Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Соловьев Андрей Борисович
Должность: Директор
Дата подписания: 27.09.2023 13:12:19
Уникальный программный ключ:
c83cc511feb01f5417b9362d2700339df14aa123

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ (ФИЛИАЛ)
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО
ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
В Г. ТАГАНРОГЕ РОСТОВСКОЙ ОБЛАСТИ
ПИ (филиал) ДГТУ в г. Таганроге
ЦМК «Прикладная информатика»

Практикум

По выполнению практических работ
по дисциплине
по профессиональному модулю:
ПМ.07 Соадминистрирование и автоматизация баз данных и серверов
09.02.07 «Информационные системы и программирование»
Квалификации выпускника: Специалист по информационным системам

Таганрог 2023

Составители: Андриян И.В.

Практикум по выполнению практической работы по профессиональному содулю: ПМ.07. «Соадминистрирование и автоматизация баз данных и серверов»

ПИ (филиала) ДГТУ в г.Таганроге, 2023г.

В практикуме кратко изложены теоретические вопросы, необходимые для успешного выполнения практической работы, рабочее задание и контрольные вопросы для самопроверки.

Предназначено для обучающихся по специальности 09.02.07 «Информационные системы и программирование».

Ответственный за выпуск:

Председатель ЦМК: _____ О.В. Андриян

Введение

В учебно-методическом пособии к практикуму по курсу «***Соадминистратирование и автоматизация баз данных и серверов***» изложены сведения, необходимые для успешного выполнения практических занятий по данному курсу. Описан процесс работы с инструментарием, применяемым на практических занятиях, представлен ряд типичных задач и подходы к их решению. Практические занятия посвящены углубленному знакомству обучающихся с выявлением технических проблем, возникающие в процессе эксплуатации баз данных и серверов. Осуществлять администрирование отдельных компонент серверов.

Цель настоящего пособия – помочь обучающимся при выполнении практических работ, выполняемых для закрепления знаний по теоретическим основам и получения практических навыков работы на компьютерах.

Обучающийся должен знать: модели данных, основные операции и ограничения; технологию установки и настройки сервера баз данных; требования к безопасности сервера базы данных; государственные стандарты и требования к обслуживанию баз данных.

Обучающийся должен уметь: проектировать и создавать базы данных; выполнять запросы по обработке данных на языке SQL; осуществлять основные функции по администрированию баз данных; разрабатывать политику безопасности SQL сервера, базы данных и отдельных объектов базы данных; владеть технологиями проведения сертификации программного средства.

Данное учебно-методическое пособие предназначено для обучающихся 3 и 4 курсов.

Правила выполнения практических занятий

Практические занятия выполняются каждым обучающимся самостоятельно в полном объеме и согласно содержанию методических указаний.

Перед выполнением обучающийся должен отчитаться перед преподавателем за выполнение предыдущего занятия (сдать отчет).

Обучающийся должен на уровне понимания и воспроизведения предварительно усвоить необходимую для выполнения практических занятий теоретическую и информацию.

Обучающийся, получивший положительную оценку и сдавший отчет по предыдущему практическому занятию, допускается к выполнению следующему занятию.

Обучающийся, пропустивший практическое занятие по уважительной либо неуважительной причине, закрывает задолженность в процессе выполнения последующих практических занятий.

Практическая работа №1
Построение схемы базы данных

Цель работы: Изучить этапы проектирования реляционной БД; изучить и практически освоить процесс создания новой БД средствами СУБД MS Access, включая разработку макета таблиц в режиме Конструктора и построение схемы БД.

Форма отчета:
−выполнить задание;
−показать преподавателю;
−ответить на вопросы преподавателя.
Время выполнения: 2 ч

Краткие теоретические сведения:

Прежде чем приступить к созданию таких объектов базы данных, как таблицы, формы и отчеты, нужно разработать их проект. Главное назначение проекта—выработка четкого пути, по которому нужно следовать при его реализации. База данных–достаточно сложный объект, и время, затраченное на ее планирование, может значительно сократить сроки ее разработки. Отсутствиепродуманнойструктурыбазыданныхприводиткнеобходимостипостоянной переделкииперенастраиваниюобъектовбазыданных, таких, как формы и таблицы.

Проектированиебазыданныхцелесообразноначатьскраткогоописанияотчетов, списков и других документов, которые необходимо получить с помощью БД. Далее следует разработать эскиз объектов, требуемыхдляполученияне обходимыхрезультатовиопределитьсвязимеждуэтими объектами.

При разработке эскиза необходимо ответить на следующие вопросы:

Какими данными мы располагаем?

Какие данные будут содержать таблицы?

Какой тип и какие свойства должны иметь данные в каждом поле таблицы?

Как эти таблицы будут связаны друг с другом?

Законченный план должен содержать подробное описание всех таблиц (имена полей, типы данных и их свойства), а также связей между ними.

Проектированиепредусматриваетэтапысозданияпроектабазыданныхотконцепциидореальноговоплощения. Этапы проектирования базы данных:

1. Исследование предметной области и формулировка основных

допущений (накладываемых условий). На этом этапе составляется список всех форм и отчетов, которые могут быть затребованы пользователями вашей БД.

2. Анализ данных. Составить перечень всех элементов данных, входящих в формы и отчеты и сгруппировать их в таблицы БД.

3. Установить, какие взаимосвязи существуют между элементами данных. Определить первичные и вторичные (внешние) ключи отношений. Организовать поля данных в таблицах, причем это необходимо сделать, следуя 4-м правилам нормализации:

Правило 1: Каждое поле таблицы должно представлять уникальный тип информации. Это правило означает, что необходимо избавиться от повторяющихся полей и разделить составные поля на отдельные элементы данных.

Правило 2: Каждая таблица должна иметь уникальный идентификатор или первичный ключ, который может состоять из одного или нескольких полей.

Правило 3: В таблице не должно быть данных, не относящихся к объекту, определяемому первичным ключом.

Правило 4: Независимость полей. Это правило означает возможность изменять значения любого поля (не входящего в первичный ключ) без воздействия на данные других полей.

Результатом 3 этапа должна явиться группа таблиц, удовлетворяющих правилам нормализации. На этом же этапе необходимо установить связи между таблицами.

Порядок выполнения работы

В качестве предметной области выбрана деятельность фирмы по продаже автомобилей марки Toyota. Поставлена задача: упорядочить информацию о клиентах фирмы, ассортименте продукции и сформированных заказах.

Этапы 1-3 проектирования БД изучить теоретически, 4-5 выполнить практически.

Оформить отчет о выполнении всех этапов задания

Ответить на контрольные вопросы

1-й этап. Определение цели проектирования БД.

Проектируемая реляционная БД должна содержать структурированную информацию о клиентах, продаваемых автомобилях и заказах.

Проектируемая БД должна поддерживать выполнение, как минимум, следующих основных функций: ввод и обновление информации, просмотр и удаление.

2-й этап. Разработка информационно-логической модели предметной

области.

Вся информация о предметной области может быть логично разделена на 3 таблицы:

Клиенты, Автомобили, Заказы.

При этом выполняются основные требования к содержанию таблиц:

1. Каждая таблица содержит информацию только на одну тему.
2. Информация в таблицах не дублируется.
3. Для связи между таблицами заданы **первичные ключи**, однозначно определяющие каждую запись в таблице.

Содержание базовых таблиц приведено ниже:

Таблица Клиенты	**Таблица Автомобили**	**Таблица Заказы**
1. Код клиента (ключ)	1. Код модели (ключ)	1. Код заказа (ключ)
2. Фамилия	2. Модель	2. Код клиента
3. Имя	3. Мощность двигателя	3. Код Модели
4. Отчество	4. Цвет	4. Дата заказа
5. Адрес	5. Количество дверей	5. Скидка, %
6. Телефон	6. Заводская цена	6. Оплачено
	7. Издержки (транспортные, предпродажные)	
	8. Специальная модель	
	9. Дополнительное оснащение	

При разработке полей для каждой таблицы необходимо учитывать:

· Каждое поле должно быть связано с темой таблицы.

· Не включать в таблицу данные, которые являются результатом вычисления.

· Информацию следует разбивать на наименьшие логические единицы (Например, поля «Индекс», «Страна», «Населенный пункт», «Почтовый адрес», а не общее поле «Адрес»).

3-й этап. Определение отношений между таблицами.

Поскольку для проектируемой БД выполнены требования нормализации,

между таблицами **Клиенты-Заказы** и **Автомобили-Заказы** могут быть установлены **одно-многозначные отношения***(1 : ∞)*, которые поддерживаются реляционной СУБД.

Связь между таблицами устанавливается **с помощью ключей** Код клиента и Код модели, которые в главных таблицах Клиенты и Автомобили являются первичными, а в таблице-связке Заказы - внешними.

4-й этап. Создание таблиц БД средствами СУБД MS Access.

4.1. Загрузить СУБД MS Access. Создать **Новую базу данных.**

4.2. Создать макет таблицы Автомобили в режиме Конструктора, используя нижеприведенные данные об именах полей, их свойствах и типах данных.

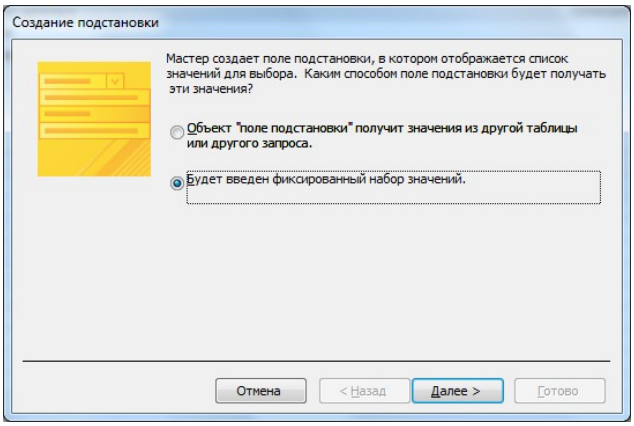
*Все поля, за исключением поля Другое оснащение, должны быть обязательными для заполнения (Свойство Обязательное поле: Да).

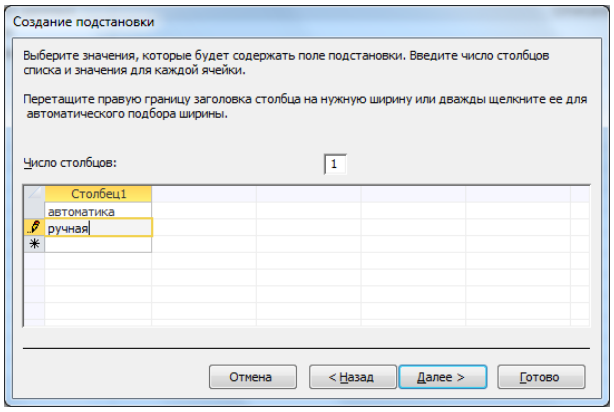
Имя поля	**Тип данных**	**Описание**	**Свойства поля**
Код модели	Числовой	Ключевое поле, код модели по заводскому каталогу	Индексированное поле: Да/Совпадения не допускаются *Ключевое поле задается в меню Правка/Ключевое поле*
Модель	Текст	Тип кузова	Размер поля: 20, Значение по умолчанию: Corolla Индексированное поле: Да/Совпадения допускаются (одна и та же модель может встречаться в БД многократно с различными вариантами оснащения)
Мощность	Текст	Мощность двигателя (кВт/л.с.)	Размер поля: 10 Индексированное поле: Нет

Имя поля	Тип данных	Описание	Свойства поля
Цвет	Текст	Цвет кузова	Размер поля: 20 Индексированное поле: Нет
Количество дверей	Числовой	2 или 4	Размер поля: Байт Формат: Основной Число десятичных знаков: 0 Значение по умолчанию: 4 Условие на значение: 2 Or 4 Сообщение об ошибке: Данное поле может содержать только 2 или 4 Индексированное поле: Нет
Коробка передач	*Мастер подстановок, фиксированный набор значений	Автоматика или ручная	Размер поля: 12 Значение по умолчанию: Ручная Условие на значение: “Ручная” Or “Автоматика” Сообщение об ошибке: Допустимы только значения “Ручная” или “Автоматика” Индексированное поле: Нет
Обивка	*Мастер подстановок, фиксированный набор значений	Велюр, кожа, ткань	Размер поля: 10 Индексированное поле: Нет
Другое оснащение	Memo	Дополнительные аксессуары	Значение по умолчанию: Радио/плейер, раздвижная крыша
Заводская цена	Денежный	Заводская продажная нетто-цена	Формат: Денежный Число десятичных знаков: Auto Индексированное поле: Нет
Транспортные издержки	Денежный	Издержки на доставку	Формат: Денежный Число десятичных знаков: Auto Индексированное поле: Нет
Предпродажные издержки	Денежный	Издержки на предпродажную подготовку	Формат: Денежный Число десятичных знаков: Auto

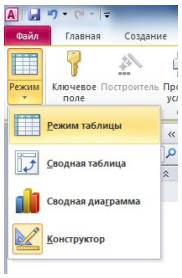
Имя поля	Тип данных	Описание	Свойства поля
			Значение по умолчанию: 105 Индексированное поле: Нет
Специальная модель	Логический	Спец. модель или стандартная	Формат: Да/Нет

*Мастер подстановокпозволяет создать для указанных полей список, из которого выбирается значение.





4.3. Перейти в режим Таблицы, сохранив созданный макет таблицы под именем Автомобили



4.4. Добавить в таблицу Автомобили 3 записи:

Код модели	12580	12653	12651
Модель	Corolla Liftback	Corolla CompactGT	Corolla CompactXL
Мощность	69/90	100/139	90/135
Цвет	Бутылочное стекло	Черный	Небесно-голубой
Количество дверей	4	2	2
Коробка передач	Автоматика	Ручная	Ручная

Обивка	Ткань	Кожа	Велюр
Другое оснащение	Радио/плейер, раздвижная крыша, лаковое покрытие “Металлик”	Радио/плейер, раздвижная крыша, алюмин. дворники	Электро-подъемник окон, раздвижная крыша
Заводская цена	39200	41100	37900
Транспортные издержки	1200	975	1050
Предпродажные издержки	105	105	105
Специальная модель	Нет	Да	Да

4.5. Создать макет таблицы Клиенты в режиме Конструктора.
*Обязательные поля Код клиента, Фамилия, Страна.

Имя поля	**Тип данных**	**Описание**	**Свойства поля** (определяют правила сохранения, отображения и обработки данных в поле)
Код клиента	Счетчик	Ключевое поле, уникальный номер клиента в БД	Индексированное поле: Да/Совпадения не допускаются *Ключевое поле задается в меню Правка/Ключевое поле*
Фамилия	Текст	Фамилия	Размер поля: 40, Индексированное поле: Да/Совпадения допускаются
Имя	Текст	Имя	Размер поля: 20, Индексированное поле: Да/Совпадения допускаются
Отчество	Текст	Отчество	Размер поля: 40, Индексированное поле: Да/Совпадения допускаются
Индекс	Числовой	Почтовый индекс	Размер поля: Длинное целое, Индексированное поле: Да/Совпадения допускаются
Страна	Текст	Название страны	Размер поля: 20, Индексированное поле: Да/Совпадения

			допускаются
Населенный пункт	Текст	Название населенного пункта	Размер поля: 40, Индексированное поле: Да/Совпадения допускаются
Почтовый адрес	Текст	Почтовый адрес	Размер поля: 50, Индексированное поле: Нет
Телефон	Текст	Контактный телефон	Размер поля: 20, Индексированное поле: Нет

4.6. Добавить в таблицу Клиенты 3 записи. (Перейти в режим Таблицы, сохранив макет таблицы под именем Клиенты)

4.7. Создать в режиме Конструктора макет таблицы Заказы.

*Все поля, за исключением поля Скидка, являются обязательными для заполнения.

Имя поля	**Тип данных**	**Описание**	**Свойства поля** (определяют правила сохранения, отображения и обработки данных в поле)
Код заказа	Счетчик	Ключевое поле, уникальный номер заказа	Индексированное поле: Да/Совпадения не допускаются *Ключевое поле задается в меню Правка/Ключевое поле*
Код модели	Числовой, *Мастер подстановок	Внешний ключ, для связи с таблицей Автомобили	Размер поля: Длинное целое Индексированное поле:Да, допускаются совпадения
Код клиента	Числовой, *Мастер подстановок	Внешний ключ, для связи с таблицей Клиенты	Размер поля: Длинное целое Индексированное поле:Да, допускаются совпадения
Дата заказа	Дата/ время	Дата формирования заказа ДД.ММ.ГГ	Формат: Краткий формат даты Индексированное поле: Да/Совпадения допускаются
Скидка	Числовой	Размер скидки в %	Размер поля: Одинарное с плавающей точкой Формат: Процентный Условие на значение: Between 0 And 1

* Используя Мастер подстановок, сформировать для полей Код клиента и Код модели список выбора из таблиц Клиенты и Автомобили(сбросить флажок Скрыть ключевое поле).

4.8. Добавить 5 записей в таблицу Заказы.

5-й этап. Создание схемы данных БД (связей между таблицами).

5.1. Выполнить опцию Схема данных из вкладки Работа с базами данных. Вдиалогом окне Добавление таблицы последовательно добавить все три таблицы. Закрыть диалоговое окно.

5.2. Установить связь между таблицами Клиенты-Заказы, Автомобили-Заказы: выделить ключевое поле в главной таблице (Клиенты или Автомобили) и перетащить его на соответствующее поле таблицы-связки Заказы. Обеспечить целостность данных.

5.3. Сохранить макет схемы данных.

Практическая работа №2
Составление словаря данных

Цель: научиться использовать представления словаря базы данных для получения информации об объектах базы данных, их структуре.

Форма отчета:

−выполнить задание;

−показать преподавателю;

−ответить на вопросы преподавателя.

Время выполнения: 2 ч

1. Для указанной таблицы, создайте сценарий, который сообщает об именах столбцов, типах данных, длине типов данных, и разрешен ли NULL. Запросите пользователя ввести имя таблицы. Дайте соответствующие псевдонимы столбцам DATA_PRECISION и DATA_SCALE. Сохраните этот сценарий в файле, script_11_01.sql.

Например, если пользователь вводит WORKS, следующие результаты:

COLUMN_NAME DATA_TYPE DATA_LENGTH PRECISION SCALE N

WORK_IDVARCHAR2 10 N

WORK_TITLE VARCHAR2 35 N

MIN_SALARY NUMBER 22 6 0 Y

MAX_SALARY NUMBER 22 6 0 Y

2. Создайте сценарий, который сообщает об имени столбца, имени ограничения, типе ограничения, условии поиска, и состоянии для указанной таблицы. Следует использовать USER_CONSTRAINTS и USER_CONS_COLUMNS, чтобы получить эту информацию. Запросите пользователя ввести имя таблицы. Сохраните сценарий в файле, script_11_02.sql.

Например, если пользователь вводит SECTIONs, следующие результаты:

COLUMN_NAME CONSTRAINT_NAME C SEARCH_CONDITION
STATUS

SECTION_NAME DEPT_NAME_NN C "SECTION_NAME" IS NOT
NULL ENABLED

SECTION_ID DEPT_ID_PK P ENABLED

MANAGER_ID DEPT_MGR_FK R ENABLED

AREA_ID DEPT_LOC_FK R ENABLED

3. Добавьте комментарий к таблице SECTIONS. Затем запросите представление USER_TAB_COMMENTS, чтобы проверить, что комментарий присутствует.

COMMENTS

Company information.

4. Найдите имена всех синонимов, которые находятся в Вашей схеме.
Ниже приведен результат выполнения.

SYNONYM_NAME TABLE_OWNER TABLE_NAME DB_LINK

EMP STAFF WORKERS

5. Вы должны определить имена и определения всех представлений в Вашей схеме. Создайте отчет, который получает информацию о представлении: имя представления и текст из представления словаря данных USER_VIEWS.

Ниже приведен результат выполнения.

```
VIEW_NAME    TEXT
------------------------------------------------------------------------------
EMP_DETAILS_VIEW
SELECT
  e.WORKER_id,
  e.WORK_id,
  e.manager_id,
  e.SECTION_id,
  d.AREA_id,
  l.LAND_id,
  e.first_name,
  e.last_name,
  e.salary,
  e.commission_pct,
  d.SECTION_name,
  j.WORK_title,
  l.city,
  l.state_province,



  c.LAND_name,
  r.PLACE_name
FROM
  WORKERs e,
  SECTIONs d,
WORKs j
  AREAs l,
  LANDS c,
  PLACEs r
WHERE e.SECTION_id = d.SECTION_id
  AND d.AREA_id = l.AREA_id
  AND l.LAND_id = c.LAND_id
  AND c.PLACE_id = r.PLACE_id
  AND j.WORK_id = e.WORK_id
WITH READ ONLY

DEPT50
SELECT WORKER_id empno, last_name WORKER,
SECTION_id deptno
FROM WORKERs
```

```
WHERE SECTION_id = 50
WITH CHECK OPTION

WORKERS_VU
SELECT WORKER_id, last_name WORKER, SECTION_id
FROM WORKERs
```

6. Найдите имена последовательностей. Напишите запрос, чтобы вывести на экран следующую информацию о последовательностях: имя последовательности, максимальное значение, приращение, и последнее число. Назовите сценарий script_11_06.sql.

Ниже приведен результат выполнения.

```
SEQUENCE_NAME    MAX_VALUE INCREMENT_BY LAST_NUMBER
------------------------------ ---------- ------------ -----------
AREAS_SEQ                 9900          100      3300
SECTIONS_SEQ              9990           10      280
WORKERS_SEQ          1.0000E+27            1        207
DEPT_ID_SEQ               1000           10       400
```

Практическая работа №3.

Разработка технических требований к серверу баз данных

Цель: Используя пример варианта расчета конфигурации серверов и рабочих мест системы «ИНФОКЛИНИКА», сформировать технические требования к собственному проекту по разработке ИС.

Форма отчета:
−выполнить задание;
−показать преподавателю;
−ответить на вопросы преподавателя.
Время выполнения: 2 ч

Вариант расчета конфигурации серверов и рабочих мест системы «ИНФОКЛИНИКА»

Вариант 1: Автономная работа клиники с одной базой данных в локальной сети клиники, толстый клиент В случае использования «толстого»

клиента требуется 1 сервер, на котором располагаются БД и исполняемые файлы, при количестве клиентов более 100 рекомендуется разделить серв ер БД и файловый сервер на разные хосты. Толстый клиент: Минимальная конфигурация: Celeron 1.6 GHz / 2 Gb RAM / HDD 80 Gb / 100 Mbit LAN / Windows 7 32 bit Рекомендуемая конфигурация: Core i3, 3.3 GHz / 4 Gb RAM / HDD 500 Gb / 100 Mbit LAN / Windows 7 64 bit Минимальные требования для сервера, обеспечивающего работу в системе до 20 «толстых» клиентов (файл сервер+ база данных на одной машине): Core i5 3GHz/ 6Gb RAM DDR-III/ 2x500Gb SATA 7200k RPM Software RAID-1/ 100Mbit FastEthernet LAN.

Таблица 2. Рекомендуемые требования для сервера при использовании толстых клиентов

Кол-во клиентов	1-30	31-60	61-100	101-200	201-500
Сервер	Intel Xeon 2.4GHz 4 ядра / 12Gb ECC RAM / 2x1TB E-SATA RAID-1 (System+Database) + 1TB SATA HDD (Backups) / 1Gbit Gigabit Ethernet LAN	Intel Xeon 2.4GHz 8 ядер/ 24Gb ECC RAM / 4x450GB E-SATA RAID-10 (System+Database) + 1TB SATA HDD (Backups) / 1Gbit Gigabit Ethernet LAN	Intel Xeon 2.4GHz 8 ядер/ 24Gb ECC RAM / 4x450GB E-SATA RAID-10 (System+Database) + 1TB SATA HDD (Backups) / 1Gbit Gigabit Ethernet LAN	Intel Xeon 2.4GHz 24 ядра/ 64Gb ECC RAM / 2x450GB SAS RAID-1 (System) + 2x250GB SAS SSD RAID-1 (Database) + 1TB SATA HDD (Backups) / 1Gbit Gigabit Ethernet LAN	Intel Xeon 2.4GHz 50 ядер / 128Gb ECC RAM / 2x450GB SAS RAID-1 (System) + 2x120GB PCI-Express SSD RAID-1 (Database) + 1TB SATA HDD (Backups) / 1Gbit Gigabit Ethernet LAN

Вариант 2. Автономная работа клиники с одной базой данных в локальной сети клиники, тонкий клиент В случае использования «тонкого»

клиента в количестве более 20 рекомендуется разделить сервер БД и сервера терминалов на разные хосты. Данные требования составлены без учета прочего сервисного и прикладного программного обеспечения – таких как антивирусы, офисные пакеты и т.п., не являющегося необходимым для функционирования МИС «ИНФОКЛИНИКА»/ «ИНФОДЕНТ», но способного повлиять на загрузку оборудования.

Таблица 3. Рекомендуемые требования для сервера при использовании тонких клиентов

Кол-во клиентов	1-20	21-50	51-100	101-200	201-500
Сервер	Intel Xeon 2.4GHz 8 ядер /16Gb ECC RAM / 4x450GB E-SATA RAID-10 (System+Database)+ 1TB SATA HDD (Backups) / 1Gbit Gigabit Ethernet LAN	Intel Xeon 2.4GHz 16 ядер /48Gb ECC RAM / 4x450GB SAS RAID-10 (System+Database) + 1TB SATA HDD (Backups) / 1Gbit Gigabit Ethernet LAN	Intel Xeon 2.4GHz 32 ядра /96Gb ECC RAM / 2x250GB E-SATA SSD RAID-1 (System) + 2x250GB SATA SSD RAID-1 (Database) + 1TB SATA HDD (Backups) / 1Gbit Gigabit Ethernet LAN	Сервер СУБД: См. Таблица 2 Сервер приложения: Intel Xeon 2.4GHz 40 ядер / 128Gb ECC Registered RAM / 2x450GB SAS SSD RAID-1 (System) / 1Gbit Gigabit Ethernet LAN	Сервер СУБД: См. Таблица 2 Сервер приложения: Intel Xeon 2.4GHz 100 ядер / 256Gb ECC RAM / 2x128GB PCI-E SSD RAID-1 (System) / 1Gbit Gigabit Ethernet LAN
Кол-во серверов	1	1	1	2	2
Пропускная способность ЛВС	1 Gbit	1 Gbit	1 Gbit	1 Gbit	1 Gbit

Вариант 3. Сеть клиник с распределенной структурой, в каждой клинике установлены локальные базы данных (БД), в центральном офисе установлена центральная база данных (ЦБД) и система репликации Требования к серверам, расположенным в каждой клинике, определяются в соответствии с Вариантом 1

Рекомендуемые требования к серверу, на котором размещена ЦБД, и работает служба репликации, обеспечивающего работу до 3 филиалов по 10 рабочих мест в каждом и до 10 клиентских мест, работающих непосредственно с ЦБД: Intel Xeon 2.4GHz 4 ядра / 8Gb ECC RAM / 2x450GB E-SATA SSD RAID-1 (System+Database)+ 1TB SATA HDD (Backups) Каналы связи должны обеспечивать передачу данных со скоростью минимум 512Kbit/c в симметричном режиме, рекомендуемые значения зависят от количества клиентских мест в филиале и определяются в соответствии с Таблицей 4.

Таблица 4

Кол-во клиентов в филиале	До 10	11-20	Более 20
Скорость передачи данных	512 Кбит/с	1 Мбит/с	2 Мбит/с и выше

Вариант 4 Сеть клиник с распределенной структурой, базы данных клиник, ЦБД и система репликации устанавливаются в Центре Обработки Данных (ЦОД), тонкий клиент Данный вариант работы предусматривает, что вся информация хранится в ЦОД, на стороне клиник базы данных не ведутся, и в случае обрыва связи продолжение работы клиник в системе невозможно.

Поэтому, для работы по этому варианту рекомендуется обеспечить наличие резервных каналов связи во всех точках сети.

Требования к клиентским рабочим местам: (толстый клиент может выступать в роли тонкого): см. Таблицу 1.

Требования к серверу БД и терминальному серверу: см. Вариант 2 Требования к серверу ЦБД и репликатора: см. Вариант 3

Требования к каналам связи: пропускная способность канала рассчитывается из числа терминальных подключений.

Подключение одного рабочего места врача требует около 100 Кбит/с, регистратора – 200 Кбит/с. около 100 Кбит/с, регистратора – 200 Кбит/с.

Практическая работа №4

Разработка требований к корпоративной сети

Цель: Используя пример разработки требований к корпоративной сети, приведенный ниже, разработать требования к корпоративной сети любого предприятия(выбирается студентом самостоятельно).

Форма отчета:
−выполнить задание;
−показать преподавателю;
−ответить на вопросы преподавателя.
Время выполнения: 2 ч

Пример.
1. Требования к локальной вычислительной сети.
Новые рабочие места ЛВС должны быть интегрированы в существующую сеть и максимально использовать имеющиеся, собственные, не арендованные ресурсы

Локальная вычислительная сеть должна включать следующие компоненты:

- информационная кабельная подсистема с пропускной способность 1000 Мб/с;

- активное оборудование (коммутаторы, маршрутизаторы);

Информационная кабельная подсистема должна строиться в соответствии с требованиями стандарта ISO/IEC 11801 Class D, категория 5E.

Общее количество автоматизированных рабочих мест –24.

Максимальная длина кабеля от информационного порта RJ45 до коммутационной панели не должна превышать 90 м.

Локальная вычислительная сеть в целом должна соответствовать категории не ниже 5E, все комплектующие (кабель, розетки, коммутационные панели, соединительные шнуры) должны соответствовать категории не ниже 5E.

Каждое автоматизированное рабочее место должно состоять из информационной розетки RJ-45 в количестве 2 штуки.

Для создания локальной вычислительной сети необходимо использовать только высококачественные компоненты, которые прошли стопроцентное тестирование в соответствии с требованиями ISO 9001 (ГОСТ 40.9001-88).

Все кабельные системы локальной вычислительной сети должны быть выполнены с учётом требований по физической защите трасс от повреждения включающих:

- прокладку кабеля за подвесным потолком, за гипсокартоновыми стенами, в металлический лотках и в кабель-каналах.

- крепление кабеля по всей трассе с помощью специальных стяжек по всей длине.

- Оборудование ЛВС и схемы его соединений должны обеспечивать двойное резервирование каналов передачи данных .

2. Общие требования к информационной кабельной подсистеме.

Информационная кабельная подсистема предназначена для передачи информации между локальными устройствами автоматизированных рабочих мест (компьютеры, активное оборудование, многофункциональными устройствами) и должна обеспечитвать подключение к узлу ТМС, устанавливаемому в рамках реализуемого Правительством Самарской области проекта «Создание телемедицинской сети Самарской области».

Количество автоматизированных рабочих мест может быть изменено Подрядчиком по согласованию с заказчиком на этапе проектирования локальной вычислительной сети.

Все порты RJ-45 расположенные на рабочих местах, а так же на коммутационной панели в коммутационном шкафу должны быть промаркированы таким способом, что бы их можно было однозначно идентифицировать. Маркировка должна быть выполнена типографским способом или при помощи лазерного принтера.

Технология прокладки кабеля должна обеспечивать сохранность эстетического вида помещений после производства монтажных работ.

3. Требования к активному оборудованию.

Оборудование должно функционировать 24 часа в сутки, 7 дней в неделю, без учета времени необходимого для проведения регламентных работ в соответствии с рекомендациями производителя.

Число портов активного оборудования должно обеспечивать функционирование 100% автоматизированных рабочих мест и иметь дополнительный запас не менее 20%.

Оборудование должно иметь возможность для установки в 19" коммутационный шкаф.

Технические требования к активному оборудованию.

Маршрутизатор – должен быть с функцией межсетевого экрана и возможности назначения листов доступа для интеграции с сетью ТМС.

Процессор	ARM, не менее, 680MHz
Память	не менее: 256MB DDR
Жесткий диск	не менее: 512MB на чипе памяти NAND, microSD слот
Ethernet порты	не менее: Пяти 10/100/1000 Mbit/s Ethernet портов с поддержкой Auto MDI/X
Производительность в режиме межсетевого экрана	не менее 1 Гбит/с
Поддержка Протоколов маршрутизации RIP, OSPF, BGP	да
Поддержка EoIP туннелей	не ограничено
Поддержка PPPoE туннеей	Не менее 500
Поддержка PPTP туннелей	Не менее 500
Поддержка L2TP туннелей	Не менее500
Поддержка OVPN туннелей	не ограничено
Поддержка VLAN интерфейсы	не ограничено
Правила брандмауэра P2P	не ограничено
NAT правила	не ограничено
Активных пользователей Хот-	500

Спот	
многоуровневые L2/L3/L4 списки контроля доступа	Для интеграции с сетью ТМС

Коммутатор

Кол-во портов Gigabit Ethernet 10/100/1000	не менее: 24 порта
Кол-во портов SFP	не менее: 4 слота
Пропускная способность	не менее: 48 Гбит/сек; 35.7 Mpps
Системная память memory	не менее: 128 Мбайт
Объем буфера пакетов	не менее: до 0.75 Мбайт
Встроенная флэш-память	не менее: 32 Мбайт
Размер базы данных адресов	не менее: 8000 MAC-адресов
Число VLAN	не менее: 1024
Число транков	не менее: 64
Число очередей	не менее: 8
Число маршрутизируемых VLAN	не менее: 32

4. Требования к кабель-каналам, информационным и электрическим розеткам.

Для реализации проекта исполнитель самостоятельно выбирает производителя кабельной системы. Тип и размер кабель канала для горизонтальной кабельной подсистемы должен быть одинаков во всех помещениях.

5. Требования к коммутационной системе.

Серверное помещение, расположенного в здании по адресу: г. Новокуйбышевск, ул. Клары Цеткин, 24, оснащается телекоммуникационным шкафом 42U. К данному шкафу подводятся кабеля вертикальных и горизонтальных кабельных систем. Так же в нем должно быть установлено активное оборудование

В шкафу необходимо придерживаться следующего расположения. Сверху вниз: органайзер, медные патч-панели в сочетании с органайзерами на 48 портов, медное активное оборудование, сервера, источники бесперебойного питания

6. Требования к электропитанию и заземлению

Система электропитания рабочих мест ЛВС предназначена для подключения компьютерной техники на рабочих местах СКС к электрической сети 220В, 50Гц. Каждое рабочее место ЛВС должно оснащаться двумя электрическими розетками 220В, 50Гц с заземляющим контактом. Компьютерные розетки должны отличаться по цвету от бытовых или иметь соответствующую маркировку.

Система электропитания рабочих мест ЛВС представляет собой выделенную распределительную электрическую сеть 380/220В, 50Гц, которая подключается к общей системе электроснабжения здания в центральном распределительном устройстве.

Система электропитания должна быть выполнена по 5-ти проводной схеме (TN-C-S) в магистральной части и по 3-проводной схеме в групповой части.

Должно быть предусмотрено равномерное распределение нагрузок по фазам.

Электроснабжение групповых этажных силовых щитов должно осуществляться от главных распределительных щитов по радиальной схеме электроснабжения. Щиты устанавливаются полностью комплектными. Конструктивное исполнение щитов должно обеспечивать выполнение требований безопасности и высокий уровень надежности. В силовых щитах обеспечить 30% резервирование по месту для возможности дополнительной установки автоматических выключателей.

Предусмотреть подключение источника бесперебойного питания, обеспечивающего электропитание сетевого и серверного (при наличии свободного места) оборудования, размещаемого в коммутационном шкафу, отдельной линией питания и от отдельного автоматического выключателя. Для удобства подключения активного и телекоммуникационного оборудования в шкафу необходимо предусмотреть электрические панели, подключаемые к ИБП, с количеством розеток, достаточным для подключения устанавливаемого в шкафу оборудования и запасом не менее 20% на развитие.

Распределительные щиты, автоматические выключатели, а также кабели должны иметь сертификаты соответствия в системе ГОСТ Р и иметь соответствующую маркировку. Электрические кабели должны иметь изоляцию

из материалов не распространяющих горение, с низким содержанием галогенов (маркировка нг LS).

Заземление элементов системы должно соответствовать требованиями главы 1.7 ПУЭ (7 издание).

Корпус коммутационного шкафа СКС должен быть заземлен отдельным проводником непосредственно с главной заземляющей шиной ВРУ.

Прокладку электрических кабелей осуществить в металлических лотках при прокладке кабельных трасс скрыто за фальшпотолком или в кабельных каналах при открытой прокладке. В рабочих кабинетах монтаж должен быть выполнен в отдельных секциях пластиковых кабельных каналов совместно с СКС.

Розетки электропитания и розетка СКС должны устанавливаться на рабочих местах ЛВС в стандартные конструктивные элементы – суппорты, рамки и т.п. и иметь единообразный дизайн.

До начала работ подрядчик должен разработать и согласовать с ответственным за электрохозяйство ЛПУ однолинейную схему выделенной распределительной системы электропитания рабочих мест ЛВС, включая электропитание коммуникационного шкафа, предоставить расчет нагрузок, поэтажные планы кабельных трасс и размещения электрооборудования, таблицу кабельных соединений. В расчетах принять, что электропотребление одним рабочим местом ЛВС составляет 350Вт. Суммарное электропотребление коммутационного шкафа принять в размере 3000Вт (с учетом дополнительно устанавливаемого серверного оборудования).

7. Надежность

Оборудование в составе локальной вычислительной сети должно обеспечивать постоянство физических характеристик канала между портом активного оборудования и абонентским оборудованием вне зависимости от трассы коммутации на панелях переключения распределительных узлов.

Постоянство физических параметров канала должно обеспечиваться при последующих перекроссировках вне зависимости от их числа (но не более определенного производителем оборудования локальной вычислительной сети).

Разрыв любого канала локальной вычислительной сети возможен только при коммутации на панелях переключения распределительных узлов.

Используемые в локальной вычислительной сети оборудование и материалы не должны допускать изменений физико-химических параметров в результате воздействия окружающей среды в течение всего гарантийного срока эксплуатации при условии соблюдения заданных производителем условий эксплуатации.

В случае выхода из строя любого из каналов должна обеспечиваться возможность перехода на использование альтернативного канала из числа резервных при помощи изменения соединений на панелях переключения распределительных узлов.

8. Безопасность

Используемое оборудование и материалы не должны допускать возможности нанесения вреда здоровью или поражения персонала электрическим током, или электромагнитными излучениями при условии соблюдения правил эксплуатации оборудования.

9. Однородность

Применить унифицированные типы кабелей и разъемов в рамках рабочих мест, горизонтальной подсистемы, подсистем внутренних магистралей, а также распределительных узлов, вне зависимости от типов подключаемого абонентского оборудования и активного оборудования различных подсистем.

10. Расширяемость

Обеспечить возможность увеличения абонентской емкости локальной вычислительной сети за счет включения дополнительных линий горизонтальной подсистемы, без необходимости прокладки новых кабельных трасс, кабельных каналов, нарушения интерьера рабочих помещений, а также без остановки работы персонала объекта.

Практическая работа №5

«Конфигурирование сети»

Цель работы: изучение принципов построения сетей по стандарту Ethernet и приобретение практических навыков оценки корректности их конфигурации.

Форма отчета:

−выполнить задание;

−показать преподавателю;

−ответить на вопросы преподавателя.

Время выполнения: 2 ч

6.1 Принципы расчета конфигурации сети

Соблюдение многочисленных ограничений, установленных для различных стандартов физического уровня сетей Ethernet, гарантирует корректную работу сети (естественно, при исправном состоянии всех элементов физического уровня). Основные характеристики и ограничения технологии Ethernet приведены в таблицах 6.1 и 6.2.

Таблица 6.1 - Общие ограничения для всех стандартов Ethernet

Характеристика	Значение
Номинальная пропускная способность	10 Мбит/с
Максимальное число станций в сети	1024
Максимальное расстояние между узлами в сети	2500 м (в 10Base-FB -2750 м)
Максимальное число коаксиальных сегментов в сети	5

Таблица 6.2 - Параметры спецификаций физического уровня для стандарта Ethernet

Параметр	10Base-5	10Base-2	10Base-T	10Base- F
Кабель	Толстый ко-аксиальный кабель RG-8 или RG-11	Тонкий ко-аксиальный кабель RG-58	Неэкраниро-ванная витая пара категорий 3,4,5	Многомод о-вый воло-конно-оптически й кабель
Максимальная длина сегмента, м	500	185	100	2000
Максимальное расстояние между узлами сети (при	2500	925	500	2500(2740 для 10Base-FB)

использовании повторителей), м				
Максимальное число станций в сегменте	100	30	1024	1024
Максимальное число повторителей между любыми станциями сети	4	4	4	4 (5 для 10Base-FB)

Наиболее часто приходится проверять ограничения, связанные с длиной отдельного сегмента кабеля, а также количеством повторителей и общей длиной сети.

Правила «5-4-3» *(допускается соединение в линию до 5 сегментов не более чем через 4 повторителя, из этих сегментов только 3 могут использоваться для подключения узлов (Trunk segments), остальные (Link segments)* используются как удлинители) для коаксиальных сетей и «4 хабов» *(число повторителей (концентраторов) между любыми двумя компьютерами в сети Ethernet не может быть больше четырех)* для сетей на основе витой пары и оптоволокна не только дают гарантии работоспособности сети, но и оставляют большой «запас прочности» сети. Например, если посчитать время двойного оборота в сети, состоящей из 4 повторителей 10Base-5 и 5 сегментов максимальный длины 500 м, то окажется, что оно составляет 537 битовых интервала. А так как время передачи кадра минимальной длины (вместе с преамбулой), составляющей 72 байт, равно 575 битовым интервалам, то видно, что разработчики стандарта Ethernet оставили 38 битовых интервала в качестве запаса для обеспечения надежности. Тем не менее в документах комитета IEEE 802.3 утверждается, что и 4 дополнительных битовых интервала создают достаточный запас надежности.

Комитет IEEE 802.3 приводит исходные данные о задержках (таблицы 6.3 и 6.4), вносимых повторителями и различными средами передачи данных, для тех специалистов, которые хотят самостоятельно рассчитывать максимальное количество повторителей и максимальную общую длину сети, не довольствуясь теми значениями, которые приведены в правилах «5-4-3» и «4 хабов».

Таблица 6.3 - Данные для расчета значения PDV*(Path Delay Value - время двойного оборота)*

Тип сегмента	База левого сегмента, bt	База промежуточного сегмента, bt	База правого сегмента, bt	Задержка среды на 1 м, bt	Максимальная длина сегмента, м
10Base-5	11,8	46,5	169,5	0,0866	500
10Base-2	11,8	46,5	169,5	0,1026	185
10Base-T	15,3	42,0	165,0	0,113	100
10Base-FB	-	24,0	-	0,1	2000
10Base-FL	12,3	33,5	156,5	0,1	2000
FOIRL	7,8	29,0	152,0	0,1	1000
AUI (>2 м)	0	0	0	0,1026	2+48

Таблица 6.4 - Уменьшение межкадрового интервала повторителями

	Передающий сегмент, bt	Промежуточный сегмент, bt
10Base-5 или 10Base-2	16	11
10Base-FB	-	2
10Base-FL	10,5	8
10Base-T	10,5	8

Особенно такие расчеты полезны для сетей, состоящих из смешанных кабельных систем, например, коаксиала и оптоволокна, на которые правила о количестве повторителей не рассчитаны. При этом максимальная длина каждого отдельного физического сегмента должна строго соответствовать стандарту, то есть 500 м для «толстого» коаксиала, 100 м для витой пары и т. д.

Чтобы сеть Ethernet, состоящая из сегментов различной физической природы, работала корректно, необходимо выполнение четырех основных условий:

- количество станций в сети - не более 1024;
- максимальная длина каждого физического сегмента - не более величины, определенной в соответствующем стандарте физического уровня;

- время двойного оборота сигнала (Path Delay Value, PDV) между двумя самыми удаленными друг от друга станциями сети - не более 575 битовых интервала;
- сокращение межкадрового интервала (Path Variability Value, PVV) при прохождении последовательности кадров через все повторители - не больше, чем 49 битовых интервала (так как при отправке кадров конечные узлы обеспечивают начальное межкадровое расстояние в 96 битовых интервала, то после прохождения повторителя оно должно быть не меньше, чем 96 - 49 = 47 битовых интервала).

Соблюдение этих требований обеспечивает корректность работы сети даже в случаях, когда нарушаются простые правила конфигурирования, определяющие максимальное количество повторителей и общую длину сети в 2500 м.

6.2 Методика расчета времени двойного оборота и уменьшения межкадрового интервала

Для упрощения расчетов обычно используются справочные данные IEEE, содержащие значения задержек распространения сигналов в повторителях, приемопередатчиках и различных физических средах (таблица 6.3). Битовый интервал обозначен как bt.

Комитет 802.3 старался максимально упростить выполнение расчетов, поэтому данные, приведенные в таблице, включают сразу несколько этапов прохождения сигнала. Например, задержки, вносимые повторителем, состоят из задержки входного трансивера, задержки блока повторения и задержки выходного трансивера. Тем не менее в таблице все эти задержки представлены одной величиной, названной базой сегмента.

Чтобы не нужно было два раза складывать задержки, вносимые кабелем, в таблице даются удвоенные величины задержек для каждого типа кабеля.

В таблице используются также такие понятия, как левый сегмент, правый сегмент и промежуточный сегмент. Поясним эти термины на примере сети, приведенной на рисунке 6.1.

Левым сегментом называется сегмент, в котором начинается путь сигнала от выхода передатчика конечного узла. На рисунке 6.1 это сегмент 1. Затем сигнал проходит через промежуточные сегменты 2-5 и доходит до приемника наиболее удаленного узла наиболее удаленного сегмента 6, который называется правым. Именно здесь в худшем случае происходит столкновение кадров и возникает коллизия.

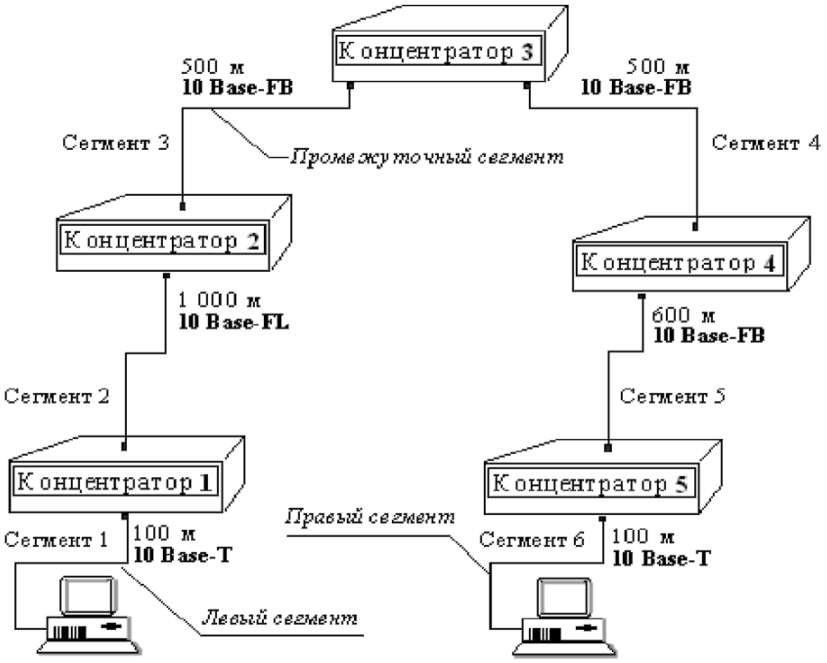


Рисунок 6.1 - Пример сети Ethernet, состоящей из сегментов различных физических стандартов

С каждым сегментом связана постоянная задержка, названная базой, которая зависит только от типа сегмента и от положения сегмента на пути сигнала (левый, промежуточный или правый). База правого сегмента, в котором возникает коллизия, намного превышает базу левого и промежуточных сегментов.

Кроме этого, с каждым сегментом связана задержка распространения сигнала вдоль кабеля сегмента, которая зависит от длины сегмента и вычисляется путем умножения времени распространения сигнала по одному метру кабеля (в битовых интервалах) на длину кабеля в метрах.

Расчет PDV заключается в вычислении задержек, вносимых каждым отрезком кабеля (приведенная в таблице задержка сигнала на 1 м кабеля умножается на длину сегмента), а затем суммировании этих задержек с базами левого, промежуточных и правого сегментов. Общее значение PDV не должно превышать 575.

Так как левый и правый сегменты имеют разные величины базовой задержки, то в случае различных типов сегментов на удаленных краях сети необходимо выполнить расчеты дважды: один раз принять в качестве левого сегмента сегмент одного типа, а во второй - сегмент другого типа. Результатом можно считать максимальное значение PDV.

Чтобы признать конфигурацию сети корректной, нужно рассчитать также уменьшение межкадрового интервала повторителями, то есть величину PVV.

Для расчета PVV также можно воспользоваться значениями максимальных величин уменьшения межкадрового интервала при прохождении повторителей различных физических сред, рекомендованными IEEE и приведенными в таблице 1.4.

6.3 Пример расчета конфигурации сети

В примере крайние сегменты сети принадлежат к одному типу - стандарту 10Base-T, поэтому двойной расчет не требуется.

Приведенная на рисунке 6.1 сеть в соответствии с правилом «4 хабов» не является корректной - в сети между узлами сегментов 1 и 6 имеются 5 хабов,

хотя не все сегменты являются сегментами 10Base-FB. Кроме того, общая длина сети равна 2800 м, что нарушает правило 2500 м. Рассчитаем значение PDV.

Левый сегмент 1:

15,3 (база) + 100-0,113 = 26,6

Промежуточный сегмент 2:

33,5 + 1000-0,1 = 133,5

Промежуточный сегмент 3:

24 + 500-0,1 = 74,0

Промежуточный сегмент 4:

24 + 500-0,1 = 74,0.

Промежуточный сегмент 5:

24 + 600-0,1 = 84,0

Правый сегмент 6:

165 + 100-0,113 = 176,3.

Сумма всех составляющих дает значение PDV, равное 568,4.

Так как значение PDV меньше максимально допустимой величины 575, то эта сеть проходит по критерию времени двойного оборота сигнала несмотря на то, что ее общая длина превышает 2500 м, а количество повторителей больше 4.

Рассчитаем значение PVV.

Левый сегмент 1 10Base-T: сокращение в 10,5 bt.

Промежуточный сегмент 2 10Base-FL: 8.

Промежуточный сегмент 3 10Base-FB: 2.

Промежуточный сегмент 4 10Base-FB: 2.

Промежуточный сегмент 5 10Base-FB: 2.

Сумма этих величин дает значение PVV, равное 24,5, что меньше предельного значения в 49 битовых интервала.

В результате сеть соответствует стандартам Ethernet по всем параметрам.

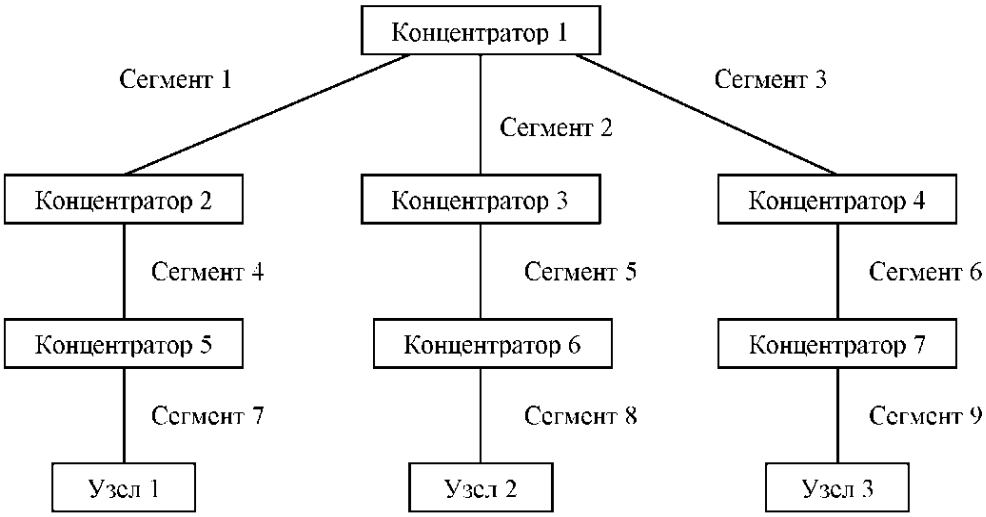
6.4 Задание на лабораторную работу

1. Ознакомиться с теоретическим материалом.
2. Произвести оценку конфигурации сети в соответствии с вариантом:

- по физическим ограничениям: на длину сегмента, на длину сети, правило «4 хаба» («5 хабов» для 10Base-FB);
- по времени двойного оборота сигнала в сети;
- по уменьшению межкадрового интервала.

3. По результатам расчетов сделать вывод о корректности конфигурации сети Ethernet.

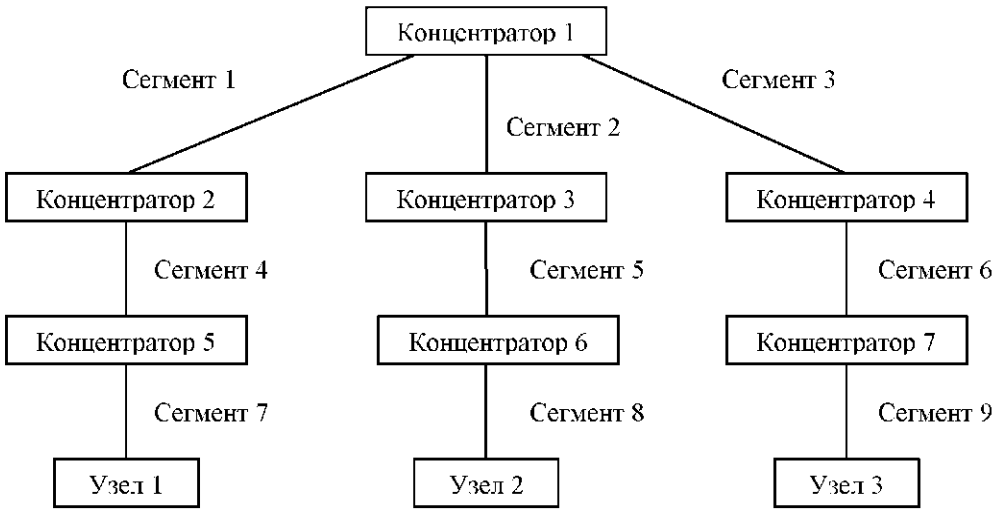
4. По результатам работы оформить отчет. Содержание отчета: исходные данные, расчеты указанных параметров, выводы.



	10 Base-FB	10 Base-FL	10 Base-T	Длина, м
Сегмент 1	+			500
Сегмент 2	+			300
Сегмент 3	+			400
Сегмент 4		+		1000
Сегмент 5		+		300
Сегмент 6		+		400
Сегмент 7			+	100
Сегмент 8			+	50
Сегмент 9			+	100

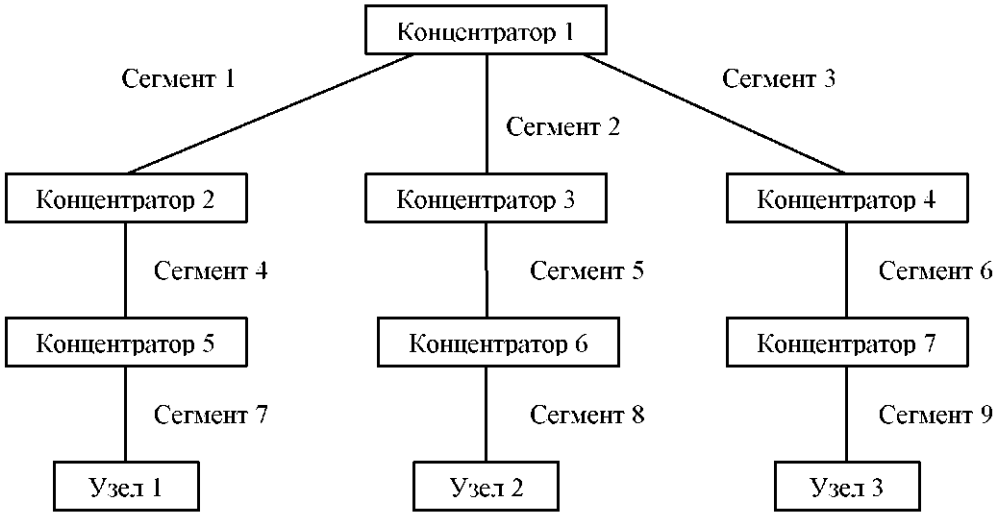
Вариант 1

Вариант 2



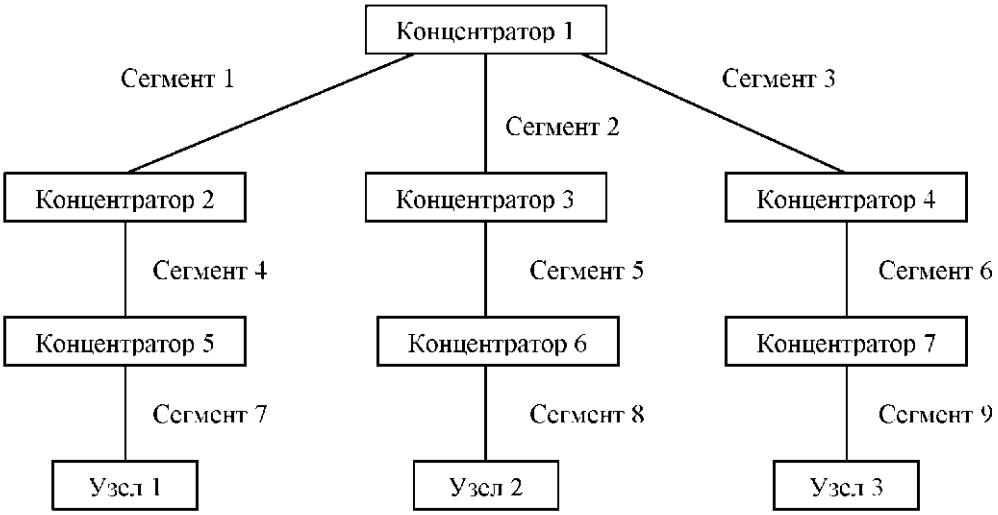
	10 Base-FB	10 Base-FL	10 Base-T	Длина, м
Сегмент 1		+		700
Сегмент 2	+			400
Сегмент 3	+			400
Сегмент 4		+		700
Сегмент 5		+		200
Сегмент 6	+			500
Сегмент 7			+	80
Сегмент 8			+	100
Сегмент 9			+	80

Вариант 3



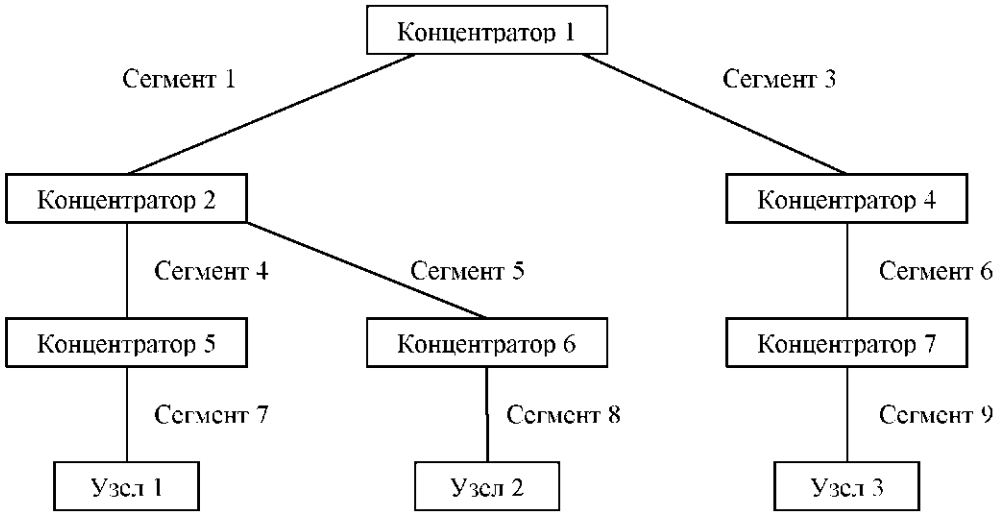
	10 Base-FB	10 Base-FL	10 Base-T	Длина, м
Сегмент 1	+			1000
Сегмент 2		+		200
Сегмент 3		+		200
Сегмент 4		+		400
Сегмент 5	+			300
Сегмент 6		+		200
Сегмент 7			+	100
Сегмент 8			+	100
Сегмент 9			+	40

Вариант 4



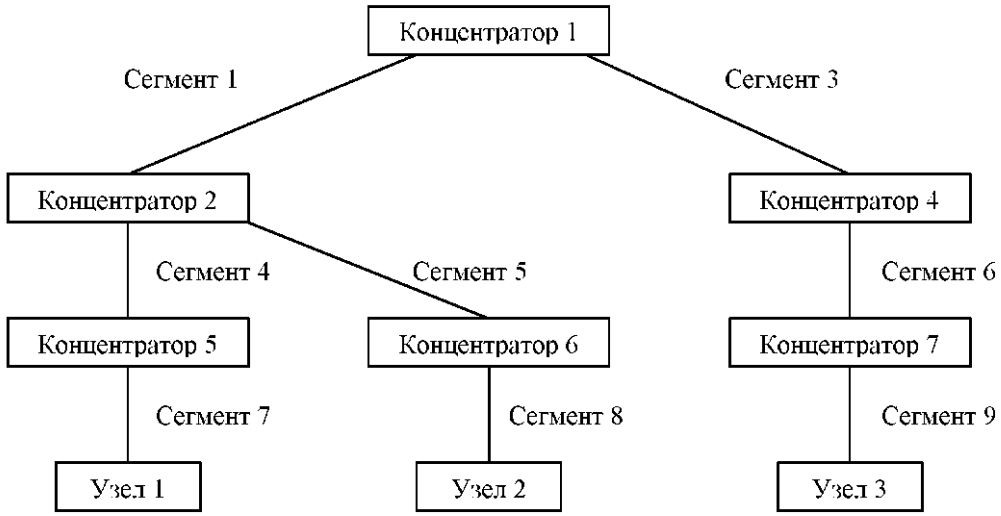
	10 Base-FB	10 Base-FL	10 Base-T	Длина, м
Сегмент 1		+		600
Сегмент 2		+		400
Сегмент 3		+		200
Сегмент 4	+			800
Сегмент 5	+			500
Сегмент 6	+			800
Сегмент 7			+	50
Сегмент 8			+	100
Сегмент 9			+	50

Вариант 5



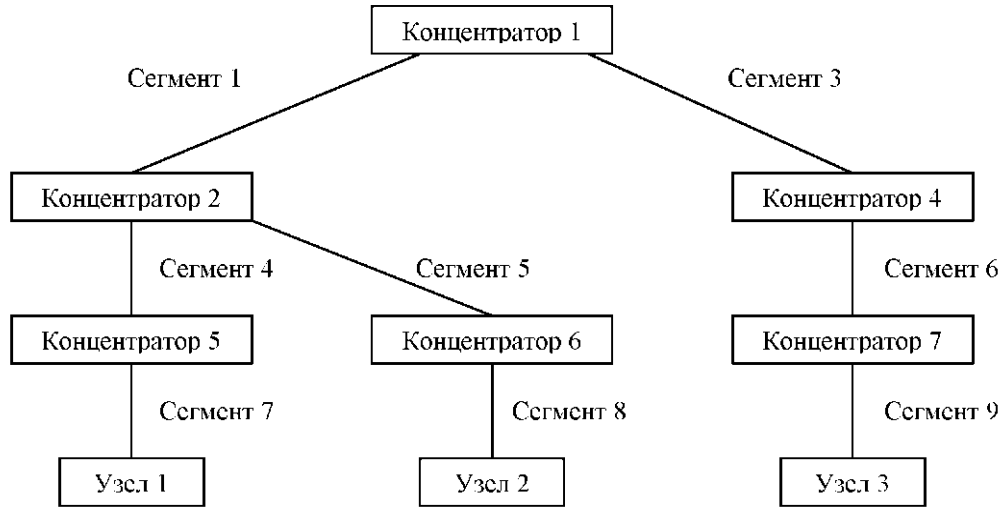
	10 Base-FB	10 Base-FL	10 Base-T	Длина, м
Сегмент 1	+			400
Сегмент 3	+			500
Сегмент 4		+		1100
Сегмент 5		+		1100
Сегмент 6		+		600
Сегмент 7			+	100
Сегмент 8			+	100
Сегмент 9			+	100

Вариант 6



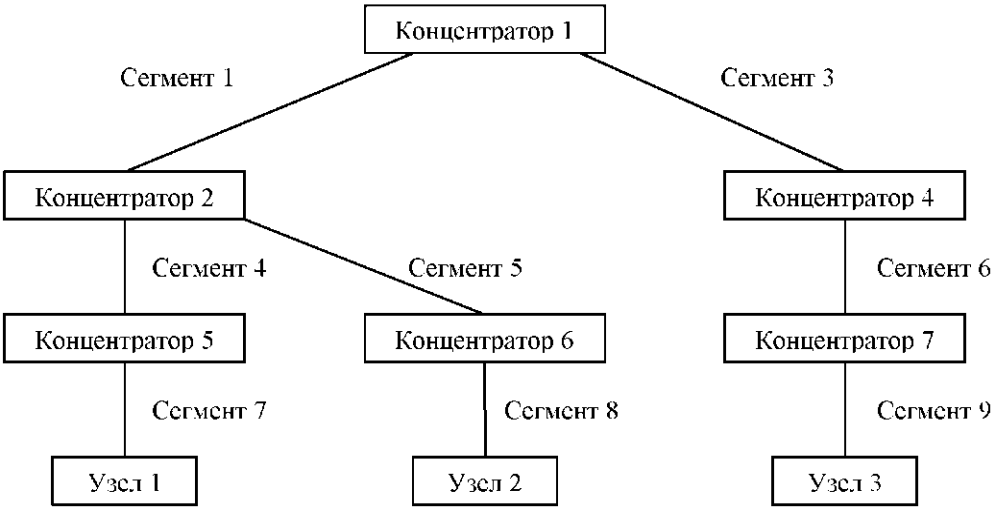
	10 Basc-FB	10 Basc-FL	10 Basc-T	Длина, м
Сегмент 1	+			500
Сегмент 3		+		500
Сегмент 4	+			1000
Сегмент 5	+			1000
Сегмент 6		+		500
Сегмент 7			+	80
Сегмент 8			+	80
Сегмент 9			+	100

Вариант 7



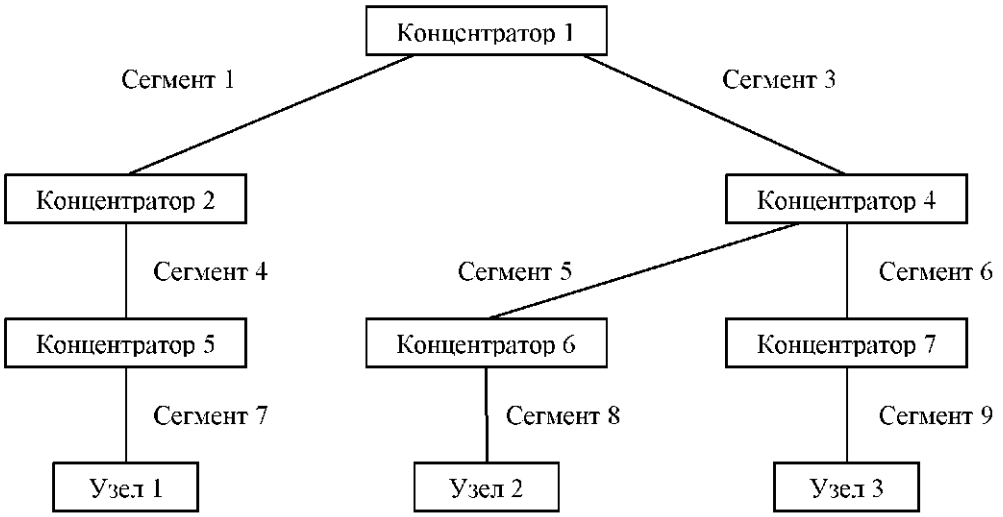
	10 Base-FB	10 Base-FL	10 Base-T	Длина, м
Сегмент 1		+		1000
Сегмент 3	+			1000
Сегмент 4		+		600
Сегмент 5		+		600
Сегмент 6	+			400
Сегмент 7			+	60
Сегмент 8			+	60
Сегмент 9			+	90

Вариант 8



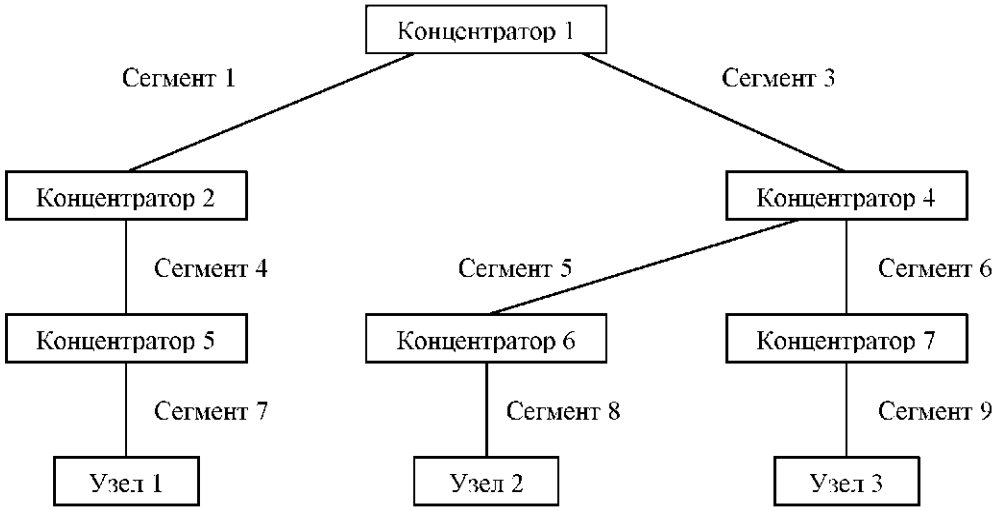
	10 Base-FB	10 Base-FL	10 Base-T	Длина, м
Сегмент 1		+		900
Сегмент 3		+		900
Сегмент 4	+			700
Сегмент 5	+			700
Сегмент 6	+			500
Сегмент 7			+	70
Сегмент 8			+	70
Сегмент 9			+	100

Вариант 9



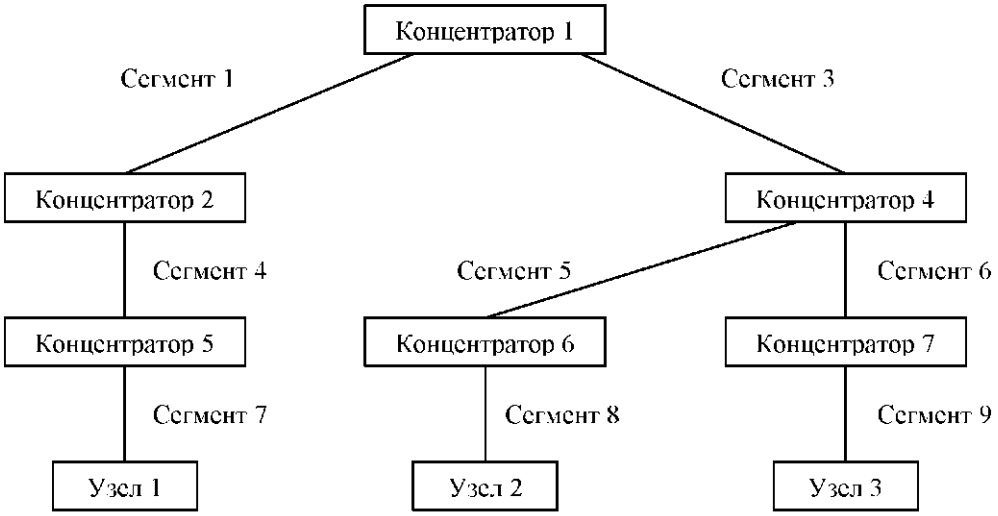
	10 Base-FB	10 Base-FL	10 Base-T	Длина, м
Сегмент 1	+			400
Сегмент 3	+			500
Сегмент 4		+		1100
Сегмент 5		+		1100
Сегмент 6		+		600
Сегмент 7			+	100
Сегмент 8			+	100
Сегмент 9			+	100

Вариант 10



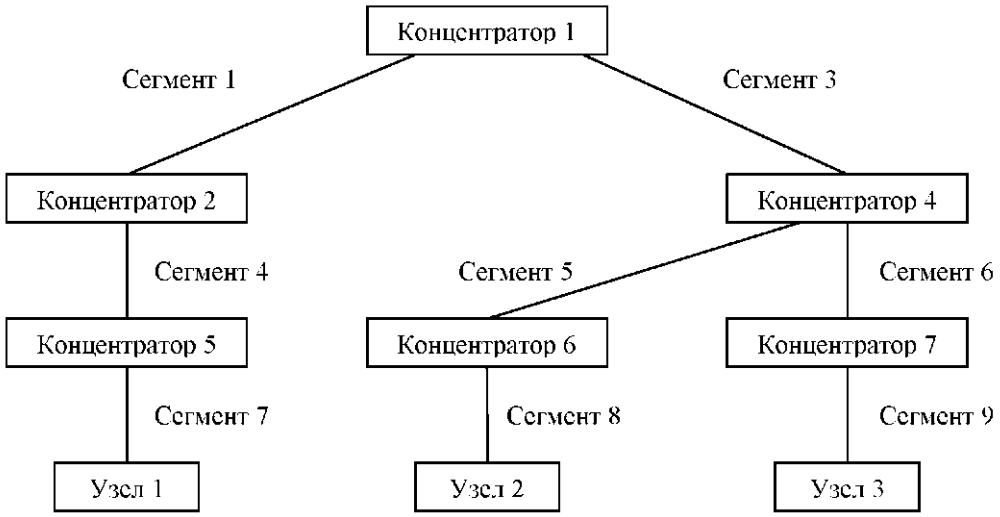
	10 Base-FB	10 Base-FL	10 Base-T	Длина, м
Сегмент 1	+			500
Сегмент 3		+		500
Сегмент 4	+			1000
Сегмент 5	+			1000
Сегмент 6		+		500
Сегмент 7			+	80
Сегмент 8			+	80
Сегмент 9			+	100

Вариант 11



	10 Base-FB	10 Base-FL	10 Base-T	Длина, м
Сегмент 1		+		1000
Сегмент 3	+			1000
Сегмент 4		+		600
Сегмент 5		+		600
Сегмент 6	+			400
Сегмент 7			+	60
Сегмент 8			+	60
Сегмент 9			+	90

Вариант 12



	10 Base-FB	10 Base-FL	10 Base-T	Длина, м
Сегмент 1		+		600
Сегмент 3		+		600
Сегмент 4	+			900
Сегмент 5	+			1000
Сегмент 6	+			500
Сегмент 7			+	70
Сегмент 8			+	80
Сегмент 9			+	90

Практическая работа №6

«Сравнение технических характеристик серверов»

Цель работы: Провести сравнительный анализ различных серверов (основных видов серверов, 5-6 видов), данные занести в таблицу.

Форма отчета:

−выполнить задание;

−показать преподавателю;

−ответить на вопросы преподавателя.

Время выполнения: 2 ч

Характеристики	Сервер 1	Сервер 2	Сервер 3	Сервер 4	Сервер 5
Базовая модель					
Процессор					
Оперативная память					

Жесткие диски					
Доп. жесткие диски					
Контроллер					
Оптический привод					
Модуль управления					
Основной адаптер					
Блоки питания					
Операционная система					

Сделать выводы по работе, оформить отчет.

Практическая работа №7

«Формирование аппаратных требований и схемы банка данных»

Цель работы: ознакомиться с принципом разработки информационного банка данных.

Форма отчета:

−выполнить задание;

−показать преподавателю;

−ответить на вопросы преподавателя.

Время выполнения: 2 ч

Требуется разработать БД, обеспечивающую автоматизацию процессов ведения и распространения информации о студентах, проживающих в общежитии (ВУЗа). Студенты, проживающие в общежитии, при поступлении заполняют необходимые документы (анкету). Потребителем информации из БД являются комендант, обеспечивающий проживание приезжих студентов. Для внесения входной информации созданы семь базы данных: база студентов, групп, происшествии, комнат, мебелей, факультетов. Входная информация поступает в БД виде:

-Информация о самих студентах;

-Информация о месте проживания(комната) ;

-Информация о родителях;

-Инвентарные номера мебелей;

-сведений о нарушении дисциплины в общежитии (сведения о студентах, проживающих в этой комнате, дата происшествия и т. д.). На выходе пользователи системы получают информацию в виде:
- сортированный список студентов общежития, курс, факультет и т.д. ;
- № комнаты с прикрепленными мебелями, с инвентарными номерами;
- информация о нарушении дисциплины с описанием, № комнаты и дата;
- информация о родителях (контактные данные), в случаях ЧС;
- количество студентов общежития по группам;

Основными требованиями к функциям БД общежития (информация о студентах) могут считаться следующие:

получить список студентов в определенной комнате; контролировать количество и дату происшествии в комнатах; ФИО студентов, ответственные за определенные мебели; список студентов по группам; получить количество студентов по группам. Каждый студент ВУЗа проходит обучение в одной из нескольких учебных групп, каждая из которых объединяет студентов одной специальности. Специальности, по которым ВУЗ обеспечивает подготовку специалистов, объединяются в факультеты, причем на разных факультетах специальности не повторяются. Каждая группа характеризуется следующими параметрами: - индекс группы; - номер курса; - специальность; - факультет У каждого студента есть родители, следовательно необходимо внести в БД следующие информации: - фамилия, имя, отчество родителей; - номера телефонов(мобильный, рабочий); - место работы; В комнатах общежития, часто происходят происшествия (драка, употребление спиртного, шум среди ночи и т.д.). Для учета таких происшествии в БД вносятся такие данные как: - Дата происшествия; - Вид происшествия; - Комната; Также в каждой комнате имеются мебели (кровать, вентилятор, стол, стулья и т.д.). Каждая мебель имеет свой индивидуальный инвентарный номер и в никакой другой комнате номер не повторяется.

Для получения списка мебелей в комнате необходимы:

- Инвентарный номер;

- Название мебели;

-Номеркомнаты;

Отчет оформить в виде скриншотов из Access.

Лабораторная работа №8
«Установка и настройка сервера MySQL»

Цель работы: ознакомиться с приложениями, включенными в состав СУБД MySQL. Получить навыки управления учетными записями пользователей и определения привилегий. Ознакомиться с утилитами, входящими в состав СУБД MySQL, получить навыки работы с ними.

Форма отчета:
−выполнить задание;
−показать преподавателю;
−ответить на вопросы преподавателя.
Время выполнения: 2 ч

Установка MySQL

Есть несколько вариантов установки, в зависимости от того, будете ли вы использовать СУБД на своём личном компьютере (что рекомендуется для полноценного освоения), или будете использовать компьютеры института.

Установка на личном компьютере

Вы можете воспользоваться тем же вариантом, что и для установки на компьютер института, но проще будет использовать специальный инсталлятор. Для установки на личном компьютере вы просто скачиваете с сайта http://dev.mysql.com/downloads/mysql/ MySQL installer - это стандартный установщик, при помощи которого вы установите MySQL так же, как любую другую прогрумму Windows. В процессе установке следует выбрать вариант "Developer Default", чтобы установить полный необходимый вам набор инстурментов. Все остальные параметры можно оставить по умолчанию. Пароль для администратора вы можете придумать сами.

Установка на компьютере института

Скачать актуальный дистрибутив СУБД MySQL можно по адресу https://dev.mysql.com/downloads/mysql/

Нам понадобится вариант Windows (x86, 64-bit), ZIP Archive. Скорее всего, вы так же можете получить его из сетевого диска, куда его предварительно скопировал преподаватель. В данном случае вы просто распаковываете скачанный архив к себе на локальный (не сетевой!) диск.

Запуск MySQL

Запуск на личном компьютере

В случае, если вы установили MySQL при помощи инсталлятора, то его запуск и остановка будут осуществляться через управление службами Windows. По умолчанию после установки сервис должен уже работать.

Запуск на компьютере института

Управление сервером будет осуществляться из командной строки. Для этого необходимо выполнить следующие команды (перейдя в папку MySQL):

1. Сначала инициализируем базу:

   ```
   bin\mysqld.exe --initialize --skip-log-syslog --
   standalone --console
   ```

2. Запомните пароль, который будет сгенерирован для пользователя root
3. Теперь запускаем базу:

   ```
   bin\mysqld.exe --skip-log-syslog --standalone --
   console
   ```

   Если вы всё проделали правильно, то в консоли должна появиться строка "mysqld.exe: ready for connections".
   После этого консоль перестаёт принимать ваш ввод и работает только на вывод информации. Не закрывайте эту консоль - вместе с ней вы закроете серверный процесс и больше не сможете общаться с СУБД! Для того, чтобы работать дальше, вам потребуется открыть ещё одну консоль.

4. И меняем пароль. Для этого запускаем клиентскую программу mysql

   ```
   Bin\mysql.exe -u root -p
   ```

   и вводим текущий пароль.

5. И, оказавшись в консоли MySQL, меняем пароль:

```
SET PASSWORD FOR 'root'@'localhost' =
PASSWORD('123');
```

Не забудьте переподключиться и проверить, что пароль поменялся! Выйти из консоли MySQL в консоль Windows можно командой

```
quit
```

Привилегии в MySQL

Для работы с базами данных в MySQL необходим пользователь, наделённый такими правами. То есть при подключении к базе данных Вы должны указывать логин пользователя и его пароль, и если доступ ему открыт, то он получит определённые права.

В **MySQL** существуют три группы привилегий: **данные**, **структура**, **администрирование**. Первая группа связана с изменением записей в таблицах, вторая группа связана с изменением структуры баз данных, а третья связана с администрированием, как бы это очевидно не звучало.

Теперь перейдём к рассмотрению каждой группы отдельно. И начнём с прав на управление данными в таблицах.

- **SELECT** - эта привилегия позволяет делать выборку (вытаскивание) записей из таблиц баз данных.
- **INSERT** - привилегия, которая необходима для добавления новых записей в таблицу.
- **UPDATE** - право, позволяющее обновлять записи в таблице.
- **DELETE** - эта привилегия позволяет удалять записи из таблицы.
- **FILE** - разрешает делать выборку записей и записывать данные в файл, а также считывать их оттуда.

Теперь перейдём к **привилегиям пользователей MySQL**, позволяющие изменять структуру таблицы и базы данных.

- **CREATE** - привилегия, позволяющая создавать новые базы данных, а также новые таблицы в базе данных.
- **ALTER** - привилегия, позволяющая переименовывать таблицы, вставлять новые поля в таблицу, удалять поля из таблицы, а также модифицировать их.
- **INDEX** - разрешает создавать индекс по определённому полю и удалять его. О том, что это такое и для чего нужно мы поговорим в одной из следующих статей.

- **DROP** - право, которое позволяет удалять либо таблицы, либо целые базы данных.
- **CREATE TEMPORARY TABLES** - возможность создавать временные таблицы, которые хранятся во время сессии, а после окончания сессии данная таблица автоматически удаляется.

И последняя группа привилегий - это привилегии, связанные с администрированием баз данных.

- **GRANT** - привилегия, которая позволяет создавать новых пользователей, а также менять права у существующих. Тут есть очень важная деталь: нельзя изменять значения привилегий, которыми сам не обладаешь. То есть если человек обладает **привилегией GRANT**, но не обладает **привилегией SELECT**, то он не может новым пользователям дать **привилегию SELECT**. Впрочем, это вполне логично.
- **SUPER** - позволяет использовать команду "**kill**", то есть убить поток. Поток - это текущее подключение другого пользователя к базе данных.
- **PROCESS** - привилегия, позволяющая выполнить команду "**processlist**", которая показывает список потоков.
- **RELOAD** - позволяет открывать и закрывать файлы журналов, а также перечитывать таблицы привилегий пользователей.
- **SHUTDOWN** - привилегия, позволяющая выполнить команду "**shutdown**", отключающая работу сервера.
- **SHOW DATABASES** - разрешает просматривать все существующие базы данных.
- **REFERENCES** - данная привилегия ещё не доступна, а только зарезервирована для использования в будущем.
- **LOCK TABLES** - позволяет блокировать таблицы от указанных потоков.
- **EXECUTE** - позволяет запускать хранимые процедуры.
- **REPLICATION CLIENT** - даёт право получать местонахождение ведущего (**master**) и ведомых (**slaves**) серверов.
- **REPLICATION SLAVE** - это привилегия, позволяющая читать ведомым журнала ведущего сервера.

И, наконец, специальные привилегии, связанные с ограничением на доступные ресурсы:

- **MAX QUERIES PER HOUR** - максимальное количество запросов в час, которое может отправить пользователь.
- **MAX UPDATES PER HOUR** - максимальное количество команд в час, которые каким-либо образом изменяют либо таблицу, либо базу данных.

- **MAX CONNECTIONS PER HOUR** - максимальное количество подключений в час, которое может сделать пользователь.

Если значение вышеназванных пределов равны "**0**", то ресурсы для пользователя не ограничены.

Создание нового пользователя в MySQL

Ранее мы вносили все изменения в настройки MySQL под root-пользователем, имея полный доступ ко всем базам данных. Однако для случаев, когда могут потребоваться более жесткие ограничения, есть способы создания пользователей с особыми наборами прав доступа.

Давайте начнем с создания нового пользователя из консоли MySQL:

```
CREATE USER 'newuser'@'localhost' IDENTIFIED BY
'password';
```

К сожалению, на данном этапе пользователь "newuser" не имеет прав делать что-либо с базами данных. На самом деле, даже если если пользователь "newuser" попробует залогиниться (с паролем "password"), он не попадет в консоль MySQL.

Таким образом, первое, что нам необходимо сделать, это предоставить пользователю доступ к информации, которая ему потребуется.

```
GRANT ALL PRIVILEGES ON * . * TO 'newuser'@'localhost';
```

Звездочки в этой команде задают базу и таблицу, соответственно, к которым у пользователя будет доступ. Конкретно эта команда позволяет пользователю читать, редактировать, выполнять любые действия над всеми базами данных и таблицами.

Поле завершения настройки прав доступа новых пользователей, убедитесь, что вы обновили все права доступа:

```
FLUSH PRIVILEGES;
```

Теперь ваши изменения вступят в силу.

Назначение привилегий

Для назначения прав конкретному пользователю можно использовать следующую схему:

```
GRANT [тип прав] ON [название базы данных].[название
таблицы] TO ‘[имя пользователя]’@'localhost’;
```

Если вы хотите дать доступ к любой базе данных или к любой таблице, поставьте звездочку (*) вместо названия базы данных или таблицы.

Каждый раз, когда вы изменяете права доступа, не забудьте использовать команду Flush Privileges.

Лишения прав доступа практически идентично их назначению:

```
REVOKE [тип прав] ON [название базы данных].[название таблицы] FROM ‘[имя пользователя]’@‘localhost’;
```

По аналогии с использованием команды DROP для удаления базы данных, вы можете использовать эту команду и для удаления пользователя.

```
DROP USER ‘demo’@‘localhost’;
```

Для тестирования учетной записи созданного пользователя, разлогиньтесь с помощью команды:

```
quit
```

и залогиньтесь снова, введя в терминате следующую команду:

```
mysql -u [имя пользователя]-p
```

Основные утилиты MySQL.

В состав дистрибутива MySQL входят следующие утилиты:

- mysqld
- mysql
- mysqladmin
- mysqlaccess
- mysqlshow
- mysqldump
- isamchk

Утилиты - это отдельные программы, которые находятся в папке bin, а не команды MySQL. Для их запуска вам нужно выйти из консоли MySQL (если вы в ней), и использовать обычную консоль Windows. Так же рекомендуется перед запуском утилит сменить текущую папку на ту, в которой у вас находится MySQL.

Утилиты mysqld и mysql были подробно рассмотрены ранее, поэтому возвращаться к ним не будем. Кратко рассмотрим остальные.

Mysqladmin

Утилита для администрирования сервера. Может использоваться администратором, а также некоторыми пользователями, которым

предоставлены определенные привилегии, например – Reload_priv, Shutdown_priv, Process_priv и File_priv. Данная команда может использоваться для создания баз данных, изменения пароля пользователя(администратор может изменить пароль любому пользователю, а рядовой пользователь – только свой собственный), перезагрузки и остановки сервера, просмотра списка процессов, запущенных на сервере. Mysqladmin поддерживает следующие команды:

- Create [database_name] Создает базу данных
- Drop [database_name] Удаляет базу данных и все таблицы в ней
- Reload Перезагружает сервер
- Shutdown Останавливает работу сервера MySQL
- Processlist Выводит список процессов на сервере
- Status Выводит сообщение о статусе сервера

Пример использования mysqladmin для изменения пароля:

```
mysqladmin -u your_name -p password new_password
```

Следует заметить, что в случае использования mysqladmin для установки пароля, не требуется использование функции password().Mysqladmin сам заботится о шифровании пароля.

Mysqlaccess

Используется для проверки привилегий пользователя для доступа к конкретной базе данных. Общий синтаксис:

```
mysqlaccess [host] [user] [db] опции
```

Полезная утилита для проверки прав доступа пользователя, если он получает сообщение Access denied, при попытке соединиться с базой данных. Опции:

- -?, --help Справка
- -u, --user=[username] Имя пользователя
- -p, --password=[password] Пароль пользователя
- -h, --host=[hostname] Имя хоста для проверки прав доступа
- -d, --db=[dbname] Имя базы данных для проверки прав доступа
- -U, --superuser=[susername] Имя суперпользователя(root)
- -P, --spassword=[spassword] Пароль администратора
- -b, --brief Выводит краткие сведения о таблице

Mysqlshow

Используется, чтобы показать, с какими базами данных работает сервер, какие таблицы содержит каждая БД и какие колонки есть в каждой таблице.Синтаксис:

```
mysqlshow [опции] [database [table [field]]]
```

Mysqlshow может использовать следующие параметры:

- -?, --help Справка
- -h, --host=[hostname] Имя сервера
- -k, --key Показать ключи для таблицы
- -p, --password=[password] Пароль пользователя
- -u, --user=[username] Имя пользователя
- -p, --port=[port] Порт для связи
- -V, --version Вывести информацию о версии

Если ввести mysqlshow без аргументов, будут показаны все базы данных, если указать имя БД, будут показаны все таблицы в ней. Не забывайте так же указывать ваше имя пользователя и пароль, если он не пустой. Команды

```
mysqlshow

mysqlshow mysql
```

Mysqldump

Программа mysqldump используется для создания дампа содержания базы данных MySQL. Она пишет инструкции SQL в стандартный вывод. Эти инструкции SQL могут быть переназначены в файл. Можно резервировать базу данных MySQL, используя mysqldump, но при этом Вы должны убедиться, что в этот момент с базой данных не выполняется никаких других действий. Программа mysqldump поддерживает следующие параметры (Вы можете использовать короткую или подробную версию):

- -#, --debug=[options] Вывести в протокол отладочную информацию. В общем виде 'd:t:o, filename`.
- -?, --help Справка.
- -c, --compleat-insert Генерируйте полные инструкции insert (не исключая значений, которые соответствуют значениям столбца по умолчанию).
- -h, --host=[hostname] Соединиться с сервером hostname.
- -d, --no-data Экспорт только схемы информации (исключая данные).
- -t, --no-create-info Экспорт только данных, исключая информацию для создания таблицы. Противоположность -d.

- -p, --password=[password] Пароль пользователя, для соединения с сервером MySQL. Обратите внимание, что не должно быть пробела между -p и паролем.
- -q, --quick Не буферизовать результаты запроса, дамп выдать непосредственно к STDOUT.
- -u, --user=[username] Имя пользователя. Если не задано, используется текущий логин.
- -v, --verbose Вывести подробную информацию относительно различных стадий выполнения mysqldump.
- -P, --port=[port] Порт для связи.
- -V, --version Информация о версии.

Вы можете направить вывод mysqldump в клиентскую программу MySQL, чтобы копировать базу данных. ПРИМЕЧАНИЕ: Вы должны убедиться, что база данных не изменяется в это время, иначе Вы получите противоречивую копию!

Для справки:

```
mysqldump -u root -p mysql user>mysql-1.sql

mysqldump -u root mysql>mysql-2.sql
```

Примечание флаг –p используется в случае, если пользователь наделен паролем.
После выполнения этой команды появился файл mysql-1.sql и mysql-2.sql. Загрузите их в текстовый редактор, чтобы поподробнее изучить.

Задание

- Запустите сервер MySQL. Зарегистрируйте своего пользователя в консольном приложении, задайте ему права.
- С помощью утилиты Mysqlshow выполните команду на просмотр структуры и состав таблиц базы Mysql.
- Приведите в отчете её схему. С помощью утилиты Mysqldump получите полный дамп базы Mysql (данные и таблицы), а также отдельные дампы таблиц и данных.

Цель: научиться устанавливать и настраивать сервер под UNIX.

Форма отчета:
−выполнить задание;
−показать преподавателю;
−ответить на вопросы преподавателя.
Время выполнения: 2 ч

При создании нового сервера Ubuntu 18.04 необходимо выполнить ряд операций конфигурирования в рамках ранней стадии базовой настройки. Это повысит уровень безопасности и удобства использования вашего сервера, а также создаст прочную основу для дальнейших действий.

Примечание: Ниже представлено руководство по выполнению в ручном режиме всех рекомендуемых нами операций для новых серверов Ubuntu 18.04. Выполнение данной процедуры в ручном режиме позволит получить и отработать базовые навыки администрирования системы для полного понимания действий, осуществляемых на сервере. В качестве альтернативы, при необходимости более быстрого запуска и работы вы можете запустить наш скрипт первоначальной настройки сервера для автоматизации данных операций.

Шаг 1 — Вход в систему под именем Root

Для входа на сервер необходимо знать **публичный IP-адрес сервера**. Также понадобится пароль или, если для аутентификации установлен SSH-ключ, закрытый ключ для учетной записи пользователя **root**. Если вы еще не вошли на сервер, вы можете использовать инструкции из нашего руководства подключение к Droplet с помощью SSH, где содержится подробное описание процесса.

Если вы еще не подключены к серверу, войдите в систему под именем **root**, используя следующую команду (замените выделенную часть команды на публичный IP-адрес вашего сервера):

```
ssh root@your_server_ip
```

Подтвердите предупреждение о подлинности хоста, если оно появится на экране. Если вы используете аутентификацию по паролю, укажите пароль учетной записи **root** для входа в систему. Если вы используете SSH-ключ, защищенный кодовой фразой, то при первом использовании ключа в каждом сеансе вам может быть предложено ввести кодовую фразу. Если

вы впервые входите на сервер с помощью пароля, вам также может быть предложено сменить пароль **root**.

О пользователе Root

Пользователь **root** является администратором в среде Linux и имеет весьма широкие права. Ввиду расширенных прав учетной записи **root** *не рекомендуется* использовать ее на постоянной основе, поскольку некоторые права, предоставляемые учетной записи **root**, дают возможность вносить деструктивные изменения, в том числе случайно.

Следующим шагом является создание альтернативной учетной записи пользователя с меньшим влиянием на повседневную работу. Мы расскажем, как получить расширенные права, когда они вам потребуются.

Шаг 2 — Создание нового пользователя

После входа в систему под именем **root** мы готовы добавить новую учетную запись пользователя, которая теперь будет использоваться для входа в систему.

В этом примере показан процесс создания нового пользователя под именем **sammy**, которое следует заменить на желаемое имя пользователя:

- adduser sammy
-

Вам будут заданы несколько вопросов, начиная с пароля учетной записи.

Введите надежный пароль и, при желании, укажите дополнительную информацию. Если это не требуется, нажмите ENTER в поле, которое вы хотите пропустить.

Шаг 3 — Предоставление прав администратора

Теперь у нас есть новая учетная запись пользователя с правами обычной учетной записи. Однако иногда нам может потребоваться выполнение задач администратора.

Чтобы избежать необходимости выхода из учетной записи обычного пользователя и входа в систему под именем **root**, мы можем настроить так называемые права “superuser” или **root** для нашей обычной учетной записи. Это позволит обычному пользователю запускать команды с правами администратора путем добавления слова sudo перед каждой командой.

Чтобы назначить данные права нашему новому пользователю, необходимо добавить нового пользователя в группу **sudo**. По умолчанию на сервере Ubuntu 18.04 пользователям группы **sudo** разрешается использовать команду sudo.

Запустите данную команду под именем **root**, чтобы добавить нового пользователя в группу **sudo** (замените выделенное слово на имя нового пользователя):

- usermod -aG sudo sammy
-

Теперь после входа в систему в качестве обычного пользователя вы можете вводить sudo перед командами для выполнения действий с правами superuser.

Шаг 4 — Установка простого брандмауэра

Серверы Ubuntu 18.04 могут использовать брандмауэр UFW, чтобы обеспечить возможность подключения только к определенным сервисам. Мы можем легко установить простой брандмауэр с помощью данного приложения.

Примечание: Если ваши серверы работают на DigitalOcean, вы можете использовать брандмауэры DigitalOcean Cloud Firewalls вместо UFW. Мы рекомендуем использовать только один брандмауэр, чтобы избежать трудноустранимых конфликтов.

Различные приложения могут регистрировать свои профили при установке UFW. Данные профили позволяют UFW управлять приложениями по имени. Сервис OpenSSH, позволяющий подключиться к нашему серверу, имеет зарегистрированный профиль в UFW.

Вы можете проверить это, набрав:

ufw app list

Output
Available applications:
OpenSSH
Необходимо убедиться, что брандмауэр разрешает SSH-соединения, чтобы можно было войти в систему в следующий раз. Мы можем разрешить эти соединения путем ввода:

ufw allow OpenSSH

Затем мы можем активировать брандмауэр путем ввода:

#ufw enable

Введите “y” и нажмите ENTER, чтобы продолжить. Можно увидеть, что SSH-соединения разрешены, путем ввода:

#ufw status

Output
Status: active

To	Action	From
OpenSSH	ALLOW	Anywhere
OpenSSH (v6)	ALLOW	Anywhere (v6)

Поскольку **брандмауэр в настоящее время блокирует все подключения, кроме SSH**, при установке и настройке дополнительных сервисов вам необходимо настроить параметры брандмауэра, чтобы разрешить соответствующий входящий трафик. Ознакомьтесь с общими операциями UFW в настоящем руководстве.

Шаг 5 — Предоставление внешнего доступа для обычного пользователя

После создания обычного пользователя для повседневной работы необходимо убедиться, что мы можем использовать SSH непосредственно в учетной записи.

Примечание: До проверки возможности входа в систему и использования sudo с новым пользователем мы рекомендуем оставаться в системе под именем **root**. В таком случае при появлении проблем их можно устранить и внести необходимые изменения под именем **root**. Если вы используете DigitalOcean Droplet и сталкиваетесь с проблемами с SSH-соединением **root**, вы можете зайти в систему Droplet, используя DigitalOcean Console.

Процесс настройки SSH-доступа для нового пользователя зависит от того, использует ли учетная запись **root** сервера пароль или SSH-ключи для аутентификации.

Если учетная запись Root использует аутентификацию по паролю

Если вы вошли в учетную запись **root**, *используя пароль*, значит, для SSH активирована аутентификация по паролю. Вы можете использовать SSH

для своей новой учетной записи пользователя, открыв новый сеанс и используя SSH с новым именем пользователя:

```
$ ssh sammy@your_server_ip
```

После ввода пароля обычного пользователя вы войдете в систему. Помните, что при необходимости запуска команды с административными правами, введите sudo перед командой:

```
$ sudo command_to_run
```

-

Вам будет предложено ввести пароль обычного пользователя при первом использовании sudo в каждом сеансе (и периодически после этого).

Для повышения безопасности вашего сервера ** мы настоятельно рекомендуем установить SSH-ключи вместо использования аутентификации по паролю**. Следуйте нашему руководству Установка SSH-ключей на Ubuntu 18.04 для настройки аутентификации с помощью ключей.

Если учетная запись Root использует аутентификацию с помощью SSH-ключей

Если вы вошли в учетную запись **root**, *используя SSH-ключи*, значит, для SSH аутентификация по паролю *деактивирована*. Для успешного входа в систему необходимо добавить копию локального открытого ключа в файл ~/.ssh/authorized_keys нового пользователя.

Поскольку ваш открытый ключ уже находится в файле ~/.ssh/authorized_keys учетной записи **root** на сервере, мы можем скопировать данную структуру файлов и каталогов в нашу новую учетную запись пользователя в рамках текущего сеанса.

Самым простым способом копирования файлов с правильными правами и полномочиями является использование команды rsync. Данная команда позволяет копировать каталог .ssh пользователя **root**, сохранять полномочия и изменять владельцев файлов. Измените выделенные части указанной ниже команды с учетом имени обычного пользователя:

Примечание: Команда rsync по-разному обрабатывает источники и приемники с завершающим слэшем и без завершающего слэша. При использовании команды rsync ниже убедитесь, что исходный каталог

(~/.ssh) **не** содержит завершающий слэш (убедитесь, что вы не используете ~/.ssh/).

Если вы случайно добавили завершающий слэш в команду, rsync скопирует *содержание* каталога ~/.ssh учетной записи **root** в корневой каталог пользователя sudo вместо копирования всей структуры каталогов ~/.ssh. Файлы будут храниться в неправильном месте, и SSH не сможет их найти и использовать.

- rsync --archive --chown=sammy:sammy ~/.ssh /home/sammy

Теперь начните новый сеанс и используйте SSH с новым именем пользователя:

```
#ssh sammy@your_server_ip
```

Вы должны войти в учетную запись нового пользователя без пароля. Помните, что при необходимости запуска команды с административными правами, введите sudo перед командой:

```
$ sudo command_to_run
```

Вам будет предложено ввести пароль обычного пользователя при первом использовании sudo в каждом сеансе (и периодически после этого).

Практическая работа №10
«Выполнение запросов к базе данных»

Цель: получить навыки формирования SQL запросов на добавление, изменение, извлечение и удаление данных на примере созданной согласно варианту базы данных. Изучить основы создания простейших триггеров.

Форма отчета:
−выполнить задание;
−показать преподавателю;
−ответить на вопросы преподавателя.
Время выполнения: 2 ч

3.1 Задание

3.2 Ход работы

1. Создать базу данных используя мастер создания БД в SQL Server Management Studio согласно схеме, представленной на рисунке 3.1.

2. Написать SQL запросы на добавление данных в таблицы. Данные представлены на рисунках 3.2 – 3.5.
3. Проделать 24 примера на извлечение данных из раздела 3.4 данного методического пособия. Протоколирую результат выполнения запросов в отчет о проделанной работе.
4. На свое усмотрения создать три триггера из примеров раздела 3.5 Протестировать их и результаты теста привести в отчете.
5. Для своей схемы БД (созданной во второй лабораторной работы) написать 25 запросов различной степени сложности (аналогично проделанным выше примерам). Результаты выполнения представить в отчете.
6. Составить отчет о проделанной работе. Структура отчета:
   - титульный лист;
   - задание;
   - описание хода выполнения работы;
   - заключение;

3.3 Подготовка к выполнению лабораторной работы

Перед тем как приступить к выполнению лабораторной работы номер 3 Вам необходимо создать базу данных используя мастер создания БД в MSSQL Server 20XX согласно ниже представленной схеме.

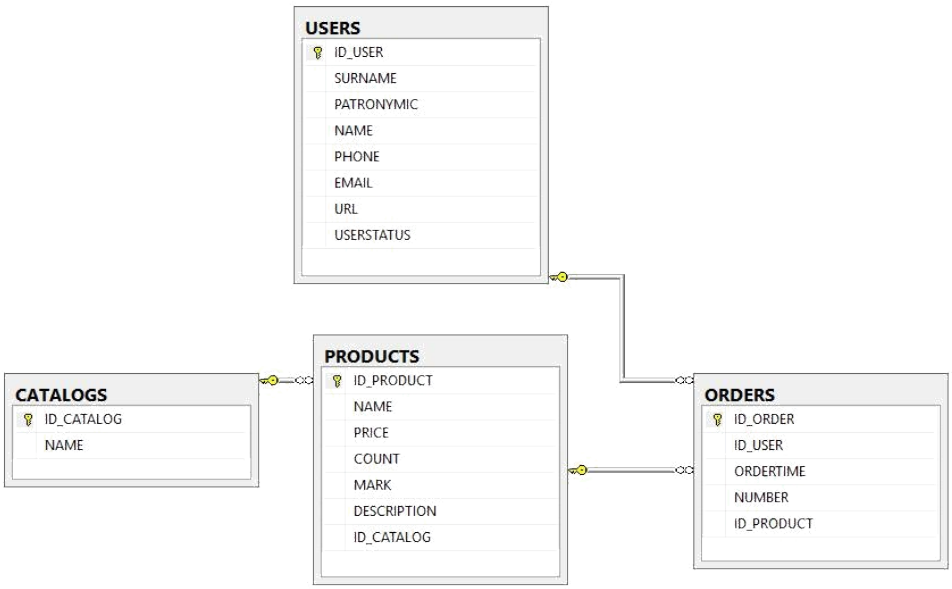


Рис. 3.1 – Схема базы данных

Следующим шагом будет заполнение БД данными. Ниже представлены данные, которыми необходимо наполнить базу данных используя SQL запросы.

	ID_CATALOG	NAME
1	1	Процессоры
2	2	Материнские платы
3	3	Видеоадаптеры
4	4	Жесткие диски
5	5	Оперативная память

Рис. 3.2 – Данные в таблице `CATALOGS`

	ID_PRODUCT	NAME	PRICE	COUNT	MARK	DESCRIPTION	ID_CATALOG
1	1	Celeron 1.8	1595.00	10	3,6	Процессор Celeron® 1.8GHz, 128kb, 478-PGA, 400Mhz, OEM 0.18	1
2	2	Celeron 2.0GHz	1969.00	2	3,7	Процессор Celeron® 2.0GHz, 128KB, 478-PGA, 400MHz, OEM	1
3	3	Celeron 2.4GHz	2109.00	4	3,9	Процессор Celeron® 2.4GHz, 128kb, 478-PGA, 400Mhz, OEM	1
4	4	Celeron D 320 2.4GHz	1962.00	1	4,1	Процессор Celeron® D 320 2.4GHz, 256kb, 478-PGA, 533Mhz, OEM	1
5	5	Celeron D 325 2.53GHz	2747.00	6	4,1	Процессор Celeron® D 325 2.53GHz, 256kb, 478-PGA, 533Mhz, OEM	1
6	6	Celeron D 315 2.26GHz	1880.00	6	4,1	Процессор Celeron® D 315 2.26GHz, 256kb, 478-PGA, 533Mhz, OEM	1
7	7	Intel Pentium 4 3.2GHz	7259.00	5	4,5	Процессор Intel® Pentium®4 3.2GHz, 1Mb, 478-PGA, 800Mhz, Hyper-Thre...	1
8	8	Intel Pentium 4 3.0GHz	6147.00	1	4,6	Процессор Intel® Pentium®4 3.0GHz, 512Kb, 478-PGA, 800Mhz, Hyper-Thr...	1
9	9	Intel Pentium 4 3.0GHz	5673.00	12	4,5	Процессор Intel® Pentium®4 3.0GHz, 1Mb, 478-PGA, 800Mhz, Hyper-Trea...	1
10	10	Gigabyte GA-8I848P-RS	1896.00	4	3,9	Материнская плата SOCKET-478 Gigabyte GA-8I848P-RS i848, (800Mhz), ...	2
11	11	Gigabyte GA-8IG1000	2420.00	2	3,8	Материнская плата SOCKET-478 Gigabyte GA-8IG1000 i865g,FSB800/53...	2
12	12	Gigabyte GA-8IPE1000G	2289.00	6	3,7	Материнская плата Socket-478 Gigabyte GA-8IPE1000G i865PE(800/533...	2
13	13	Asustek P4C800-E Delux	5395.00	4	4,1	Материнская плата Socket-478 Asustek P4C800-E Delux i875P,FSB800/5...	2
14	14	Asustek P4P800-VM\\L i865G	2518.00	6	4	Материнская плата Socket-478 Asustek P4P800-VM\\L i865G FSB800/53...	2
15	15	Epox EP-4PDA3I	2289.00	5	4	Материнская плата Socket-478 Epox EP-4PDA3I i865PE(800Mhz), 2chDD...	2
16	16	ASUSTEK A9600XT/TD	5156.00	2	4,7	Видеоадаптер ASUSTEK A9600XT/TD 128Mb DDR SDRAM, 2x400MHz ...	3
17	17	ASUSTEK V9520X	1602.00	6	4	Видеоадаптер ASUSTEK V9520X 128Mb DDR SDRAM, 400MHz DAC, AG...	3
18	18	SAPPHIRE 256MB RADEON 9550	2730.00	3	3,8	ВИДЕОКАРТА SAPPHIRE 256MB RADEON 9550, TV-out, DVI, OEM	3
19	19	GIGABYTE AGP GV-N59X128D	5886.00	6	3,6	ВИДЕОКАРТА GIGABYTE AGP GV-N59X128D FX5900XT OEM	3
20	20	Maxtor 6Y120P0	2456.00	6	4,5	Винчестер 120 GB Maxtor 6Y120P0, UDMA-133, 7200rpm, 8MB	4
21	21	Maxtor 6B200P0	3589.00	4	4	Винчестер 200 GB Maxtor 6B200P0, UDMA-133, 7200rpm, 8Mb	4
22	22	Samsung SP0812C	2093.00	5	4	Винчестер 80 GB Samsung SP0812C, SATA, 7200rpm SpinPoint P80 Serial...	4
23	23	Seagate Barracuda ST3160023A	3139.00	3	4,1	Винчестер 160 GB Seagate Barracuda ST3160023A, UDMA-100, 7200rpm,...	4
24	24	Seagate ST3120026A	2468.00	8	4,2	Винчестер 120 GB Seagate ST3120026A, UDMA-100, 7200rpm, 8MB	4
25	25	DDR-400 256MB Kingston	1085.00	20	4,8	Оперативная память DDR-400 256MB Kingston	5
26	26	DDR-400 256MB Hynix Original	1179.00	15	4,6	Оперативная память DDR-400 256MB Hynix Original	5
27	27	DDR-400 256MB PQI	899.00	10	4,2	Оперативная память DDR-400 256MB PQI	5
28	28	DDR-400 512MB Kingston	1932.00	20	4,8	Оперативная память DDR-400 512MB Kingston	5
29	29	DDR-400 512MB PQI	1690.00	12	4,2	Оперативная память DDR-400 512MB PQI	5
30	30	DDR-400 512MB Hynix	1717.00	8	4,5	Оперативная память DDR-400 512MB Hynix	5

Рис. 3.3 – Данные в таблице PRODUCTS

ID_USER	SURNAME	PATRONYMIC	NAME	PHONE	EMAIL	URL	USERSTATUS
1	Иванов	Валерьевич	Александр	58-98-78	ivanov@email.ru	NULL	active
2	Прокопчук	Иванович	Сергей	9057777777	pro@email.ru	NULL	passive
3	Семенов	Вячеславович	Игорь	9056666100	simdyanov@softtime.ru	http://www.softtime.ru	active
4	Петров	Валерьевич	Максим	NULL	kuznetsov@softtime.ru	http://www.softtime.ru	active
5	Лосев	Юрьевич	Анатолий	NULL	losev@email.ru	NULL	lock
6	Корнеев	Александрович	Александр	89-78-36	korneev@domen.ru	NULL	gold

Рис. 3.4 – Данные в таблице USERS

ID_ORDER	ID_USER	ORDERTIME	NUMBER	ID_PRODUCT
4	3	2005-04-01 10:39:38.000	1	8
5	6	2005-10-02 09:40:29.000	2	10
6	1	2005-02-18 13:41:05.000	4	20
7	3	2005-10-03 18:20:00.000	1	20
8	3	2005-03-17 19:15:36.000	1	20

Рис. 3.5 – Данные в таблице ORDERS

3.4 Примеры запросов на извлечение данных

Ниже представлен перечень (24) простых запросов к БД (схема кото-рой описана Выше. Данные примеры покрывают большой спектр кон-струкций языка SQL, начиная от простых запросов, кончая запросов с ис-пользованием функций и сортировок. Для выполнения лабораторной ра-боты, Вам необходимо проделать все 24 запроса и привести результат вы-

полнения в виде скриншота. Все конструкции языка SQL подробно описа-ны в лекционном материале.

Пример 1. Вывод данных таблицы CATALOGS

```
SELECT ID_CATALOG, NAME FROM CATALOGS

SELECT * FROM CATALOGS
```

Пример 2. Вывод данных таблицы CATALOGS с присвоением псев-донима

```
SELECT
    ID_CATALOG AS 'Идентификатор категории',
    NAME AS 'Имя категории'
FROM CATALOGS
```

Пример 3. Добавление данных с помощью SELECT в результирующую таблицу

```
SELECT NAME, ID_CATALOG, 5, 'COMMENTS' FROM
CATALOGS
```

Пример 4. Извлечение из таблицы CATALOGS записи, чей первичный ключ ID_CATALOG больше 2

```
SELECT * FROM CATALOGS
    WHERE ID_CATALOG > 2
```

Пример 5. Составное условие: извлечение из таблицы CATALOGS записи, чей первичный ключ ID_CATALOG больше 2, но меньше или ра-вен 4

```
SELECT * FROM CATALOGS
    WHERE ID_CATALOG > 2 AND ID_CATALOG <= 4
SELECT * FROM CATALOGS
    WHERE ID_CATALOG BETWEEN 3 AND 4
```

Пример 5. Противоположная конструкция, которая выводит из таблицы CATALOGS записи, чей первичный ключ ID_CATALOG меньше 3, но больше 4.

```
SELECT * FROM CATALOGS
    WHERE ID_CATALOG NOT BETWEEN 3 AND 4
```

Пример 6. Вывод записей, удовлетворяющих не диапазону, а списку

```
SELECT * FROM CATALOGS
```

```
WHERE ID_CATALOG IN (1, 2, 5)
```

Пример 7. Вывод записей, удовлетворяющих условию, заданному текстом: вывести все записи, содержащие слово процессор

```
SELECT * FROM CATALOGS
   WHERE NAME = 'процессоры'
```

Пример 8. Вывод записей, удовлетворяющих условию, заданному текстом: вывести все записи, не содержащие слово процессор

```
SELECT * FROM CATALOGS
   WHERE NOT NAME = 'процессоры'
```

Пример 9. Вывод записей, удовлетворяющих условию, заданному частью текста

```
SELECT * FROM USERS
   WHERE SURNAME LIKE 'И%'
```

Пример 10. Работа с датой: извлечение из таблицы ORDERS записи, соответствующие сделкам, осуществленным за февраль 2005 г.

```
SELECT * FROM orders
   WHERE ORDERTIME >= '2005-02-01' AND ORDERTIME
   < '2005-03-01';
```

Пример 11. Сортировка по значению одного из столбцов

```
SELECT * FROM CATALOGS
   ORDER BY ID_CATALOG
SELECT * FROM CATALOGS
   ORDER BY NAME
```

Пример 12. Извлечение из таблицы PRODUCTS записи товаров, ко-личество которых COUNT на складе от 4 до 8 с сортировкой по полю COUNT и полю MARK (для краткости выведем только столбцы COUNT и MARK)

```
SELECT COUNT, MARK FROM PRODUCTS
   WHERE COUNT BETWEEN 4 AND 8 ORDER BY COUNT,
   MARK
```

Пример 13. Изменение порядка сортировки (по умолчанию, сортировка производится в прямом порядке (ASC))

```
  SELECT ORDERTIME FROM
                 ORDERS
ORDER BY ORDERTIME DESC
```

Пример 14. Извлечение первых пяти записей с обратной сортировкой по полю COUNT

```
SELECT TOP 5 ID_PRODUCT, COUNT FROM PRODUCTS
    ORDER BY COUNT DESC
```

Пример 15. Подсчет количества проданных ТОВАРОВ

```
SELECT SUM (NUMBER) AS 'Всего продано'
    FROM ORDERS
```

Пример 16. Подсчет среднего количества товаров в одном заказе

```
SELECT AVG (NUMBER) AS 'Среднее количество'
    FROM ORDERS
```

Пример 17. Подсчет числа строк в таблице, значения столбца которых отличны от NULL

```
SELECT COUNT(ID_ORDER)
    FROM ORDERS
```

Пример 18. Подсчет числа строк в таблице, значения столбца которых отличны от NULL с присвоением псевдонима

```
SELECT COUNT(ID_ORDER) AS TOTAL
    FROM ORDERS
```

Пример 19. Извлечение максимального значения столбца ID_CATALOG

```
SELECT MAX(ID_CATALOG)
    FROM CATALOGS
SELECT TOP 1 * FROM CATALOGS
    ORDER BY ID_CATALOG
```

Пример 20. Извлечение минимального значения столбца ID_CATALOG

```
SELECT MIN(ID_CATALOG)
FROM CATALOGS
SELECT TOP 1 * FROM CATALOGS
    ORDER BY ID_CATALOG DESC
```

Пример 21. Вывод числа уникальных значений `ID_CATALOG` (сравните результат с `SELECT COUNT(ID_CATALOG) FROM PRODUCTS`)

```
SELECT COUNT(DISTINCT ID_CATALOG)
    FROM PRODUCTS
```

Пример 22. Вывод числа записей, соответствующих каждому из уникальных значений ID_CATALOG

```
SELECT ID_CATALOG, COUNT(ID_CATALOG)
```

```
FROM PRODUCTS
GROUP BY ID_CATALOG ORDER BY ID_CATALOG
```

Пример 23. Вывод числа записей, соответствующих каждому из уникальных значений ID_CATALOG больше двух

```
SELECT ID_CATALOG, COUNT(ID_CATALOG)
    FROM PRODUCTS
    WHERE ID_CATALOG > 2
    GROUP BY ID_CATALOG
    ORDER BY ID_CATALOG
```

Пример 24. Выбрать категории товаров, для которых добавлено более пяти товаров (ограничение выборки по результатам функции)

```
SELECT ID_CATALOG, COUNT(ID_CATALOG) AS TOTAL
    FROM PRODUCTS
    GROUP BY ID_CATALOG
    HAVING TOTAL > 5
    ORDER BY ID_CATALOG
```

Практическая работа №11

«Выполнение изменений в базе данных, создание триггеров»

Цель: научиться выполнять изменения в базе данных, создавать триггеры.

Форма отчета:
−выполнить задание;
−показать преподавателю;
−ответить на вопросы преподавателя.
Время выполнения: 2 ч

Триггеры являются одной из разновидностей хранимых процедур. Их исполнение происходит при выполнении для таблицы какого-либо опера-тора языка манипулирования данными (DML). Триггеры используются для проверки целостности данных, а также для отката транзакций.

Триггер – это откомпилированная SQL-процедура, исполнение кото-рой обусловлено наступлением определенных событий внутри реляцион-ной базы данных. Применение триггеров большей частью весьма удобно для пользователей базы данных. И все же их

использование часто связано с дополнительными затратами ресурсов на операции ввода/вывода. В том случае, когда тех же результатов (с гораздо меньшими непроизводитель-ными затратами ресурсов) можно добиться с помощью хранимых проце-дур или прикладных программ, применение триггеров нецелесообразно.

Триггеры – особый инструмент SQL-сервера, используемый для под-держания целостности данных в базе данных. С помощью ограничений целостности, правил и значений по умолчанию не всегда можно добиться нужного уровня функциональности. Часто требуется реализовать сложные алгоритмы проверки данных, гарантирующие их достоверность и реаль-ность. Кроме того, иногда необходимо отслеживать изменения значений таблицы, чтобы нужным образом изменить связанные данные. Триггеры можно рассматривать как своего рода фильтры, вступающие в действие после выполнения всех операций в соответствии с правилами, стандарт-ными значениями и т.д.

Триггер представляет собой специальный тип хранимых процедур, запускаемых сервером автоматически при попытке изменения данных в таблицах, с которыми триггеры связаны. Каждый триггер привязывается к конкретной таблице. Все производимые им модификации данных рас-сматриваются как одна транзакция. В случае обнаружения ошибки или нарушения целостности данных происходит откат этой транзакции. Тем самым внесение изменений запрещается. Отменяются также все измене-ния, уже сделанные триггером.

Создает триггер только владелец базы данных. Это ограничение поз-воляет избежать случайного изменения структуры таблиц, способов связи с ними других объектов и т.п.

Триггер представляет собой весьма полезное и в то же время опасное средство. Так, при неправильной логике его работы можно легко уничто-жить целую базу данных, поэтому триггеры необходимо очень тщательно отлаживать.

В отличие от обычной подпрограммы, триггер выполняется неявно в каждом случае возникновения триггерного события, к тому же он не име-ет аргументов. Приведение его в действие иногда называют запуском триггера. С помощью триггеров достигаются следующие цели:

– проверка корректности введенных данных и выполнение слож-ных ограничений целостности данных, которые трудно, если во-обще возможно, поддерживать с помощью ограничений целост-ности, установленных для таблицы;
– выдача предупреждений, напоминающих о необходимости вы-полнения некоторых действий при обновлении таблицы, реализо-ванном определенным образом;
– накопление аудиторской информации посредством фиксации све-

дений о внесенных изменениях и тех лицах, которые их выполни-ли;

– поддержка репликации.

Рассмотрим несколько простых примеров создания триггеров для существующей базе данных. В рамках лабораторной работы необходимо создать и протестировать любых три триггера (из рассмотренных приме-ров ниже, изучив их работу)

Пример 1. Триггер на добавление записи в таблицу USERS. Данный триггер в случае успешного добавления данных выводит в «Запись добав-лена»

```
CREATE TRIGGER INSERT_INDICATION --определение имени функции
    ON USERS      --для какой таблицы создается триггер
    AFTER INSERT  --когда выполнять триггер
                      --INSERT – при создании записи в таблице,
                      --DELETE – при удалении записи в таблице,
                      --UPDATE – при изменении записи в таблице,
                  --и как его выполнять
```

```
                    --AFTER – после выполнения
                    операции,
                    --INSTEAD OF – вместо выполнения
                    операции
AS
BEGIN            --тело триггера
    SET NOCOUNT ON;
    PRINT 'Запись добавлена'
END
GO
```

Тестирование работы триггера

```
INSERT INTO USERS VALUES
('Громова', 'Валерьевна', 'Анна', '55-66-89',
NULL, NULL, 'active'), ('Кремнева',
'Александровна', 'Александра', '9058956458',
'cremneva@mail.ru', NULL,'passive')
```

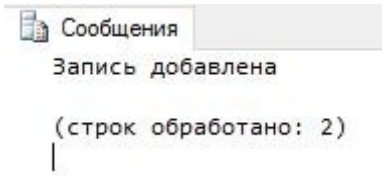


Рис. 3.6 – Результат работы триггера

Пример 2. Триггер на изменение записи в таблицу

USERS

```
CREATE TRIGGER UPDATE_INDICATION
    ON USERS
    AFTER UPDATE
AS
BEGIN
    SET NOCOUNT ON;
    PRINT 'Запись изменена'
END
```

Тестирование работы триггера

```
UPDATE USERS
    SET URL ='cremnera.tomsk.ru'
    WHERE SURNAME = 'Кремнева'
```

Пример 3. Триггер на удаление записи из таблицы USERS

```
CREATE TRIGGER DELETE_INDICATION
```

```
    ON USERS
    AFTER DELETE
AS
BEGIN
    SET NOCOUNT ON;
    PRINT 'Запись удалена'
END
GO
```

Тестирование работы триггера

```
DELETE FROM USERS
    WHERE SURNAME = 'Громова'
```

Пример 4. Триггер, демонстрирующий откат

```
CREATE TRIGGER ROLLBACK_EXAMPLE
    ON ORDERS
    AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;
    IF (SELECT NUMBER FROM inserted) < 1
    ROLLBACK
    PRINT 'Вы не можете создать заказ с количеством
    меньше 1'
END
GO
```

Тестирование работы триггера

```
INSERT INTO ORDERS VALUES (2, '2005-01-06 12:39:38',
0, 20)
```

Пример 5. Триггер на изменение количества товаров при их заказе. Количество проданного товара должно быть не меньше, чем его остаток из таблицы PRODUCTS

```
CREATE TRIGGER NUMBER_UPDATE
    ON ORDERS
    AFTER INSERT
AS
DECLARE @X INT, @Y INT
BEGIN
    SET NOCOUNT ON;
    IF NOT EXISTS(SELECT * FROM inserted
        WHERE inserted.NUMBER <= ALL (SELECT
PRODUCTS.COUNT FROM PROD-UCTS WHERE
inserted.ID_PRODUCT = PRODUCTS.ID_PRODUCT))
   BEGIN
     ROLLBACK TRAN
     PRINT 'откат! товара нет '
   END
    SELECT @Y = O.ID_PRODUCT, @X=O.NUMBER
   FROM inserted O
    UPDATE PRODUCTS
    SET PRODUCTS.COUNT = PRODUCTS.COUNT - @X
       WHERE PRODUCTS.ID_PRODUCT = @Y
END
```

```
GO
```

Тестирование работы триггера

```
INSERT INTO ORDERS VALUES ( 4, '2005-01-04 18:39:38',
12, 28)
```

Практическая работа №12
«Создание запросов и процедур на изменение структуры базы данных»

Цель: научиться создавать запросы и процедуры на изменение структуры базы данных.

Форма отчета:
−выполнить задание;
−показать преподавателю;
−ответить на вопросы преподавателя.
Время выполнения: 2 ч

Создание и изменение таблицы
Чтобы создать таблицу, используйте команду CREATE TABLE. Команда CREATE TABLE имеет следующий синтаксис:

```
CREATE TABLE table_name
 (field1 type [(size)] [NOT NULL] [index1]
 [, field2 type [(size)] [NOT NULL] [index2]
 [, ...][, CONSTRAINT constraint1 [, ...]])
```

Только необходимые элементы команды CREATE TABLE являются команды CREATE TABLE сам и имя таблицы, но обычно требуется определить некоторые поля или другими аспектами таблицы. Рассмотрим простой пример.

Предположим, что вы хотите создать таблицу для хранения имени, год и цена используемых автомобилей, которые вы собираетесь приобрести. Вы хотите разрешить до 30 символов для имени и 4 символов за год. Чтобы использовать управляющий запрос на создание таблицы, сделайте следующее:

Примечание: Во-первых, может потребоваться включить содержимое базы данных, в противном случае запрос определения данных для запуска:

Нажмите на панели сообщений кнопку Включить содержимое.

Создание таблицы
На вкладке Создание в группе макрос и код нажмите кнопку Конструктор запросов.

Закройте диалоговое окно Добавление таблицы.

На вкладке " Конструктор " в группе Тип запроса нажмите кнопку Определение данных.

Скрытые бланке и отображается вкладка объекта SQL представления.

Введите следующую инструкцию SQL:

Создание таблицы автомобилей (имя TEXT(30), год TEXT(4), валюты цена)

На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить.

Изменение таблицы
Чтобы изменить таблицу, используйте команду ALTER TABLE. Чтобы добавить, изменить или удалить столбцы (Удалить) или ограничения можно использовать команды ALTER TABLE. Команда ALTER TABLE имеет следующий синтаксис:

ALTER TABLE table_name predicate

где предикат может быть любой из следующих действий:

ADD COLUMN field type[(size)] [NOT NULL] [CONSTRAINT constraint]

ADD CONSTRAINT multifield_constraint

ALTER COLUMN field type[(size)]

DROP COLUMN field

DROP CONSTRAINT constraint

Предположим, что вы хотите добавить 10-разрядное текстовое поле для хранения сведений о состоянии каждый автомобиль. Можно сделать следующее:

На вкладке Создание в группе макрос и код нажмите кнопку Конструктор запросов.

Закройте диалоговое окно Добавление таблицы.

На вкладке " Конструктор " в группе Тип запроса нажмите кнопку Определение данных.

Скрытые бланке и отображается вкладка объекта SQL представления.

Введите следующую инструкцию SQL:

Инструкции ALTER таблицы автомобилей добавить столбец условие TEXT(10)

На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить.

К началу страницы

Создание индекса
Чтобы создать индекс в существующую таблицу, используйте команду CREATE INDEX. Команда CREATE INDEX имеет следующий синтаксис:

```
CREATE [UNIQUE] INDEX index_name
 ON table (field1 [DESC][, field2 [DESC], ...])
 [WITH {PRIMARY | DISALLOW NULL | IGNORE NULL}]
```

Только необходимые элементы являются команды CREATE INDEX имя индекс, аргумент д, имя таблицы, содержащей поля, которые вы хотите индекс и в списке полей для включения в предметном указателе.

Причины аргумент DESC индекс будет создан в порядке убывания, которая может быть полезно, если вы часто выполнение запросов, найдите максимальных значений для поля "индексировано" или, сортировка индексированные поля в порядке убывания. По умолчанию создается индекс по возрастанию.

Аргумент с ОСНОВНОЙ устанавливает индексированное поле или поля в качестве TE000126717 таблицы.

Аргумент с запретить NULL вызывает индекс, что требуется ввести значение для поля "индексировано" — то есть, не разрешены нулевые значения.

Предположим, что у вас есть таблица с именем автомобилей с полями, содержащими имя, год, цены и условия, используемые автомобилей, которые вы собираетесь приобрести. Также предположим, что таблицы стал больших

часто включают поля «месяц» в запросах. Можно создать индекс для поля «месяц», чтобы помочь запросах результатов быстрее, выполнив следующие действия:

На вкладке Создание в группе макрос и код нажмите кнопку Конструктор запросов.

Закройте диалоговое окно Добавление таблицы.

На вкладке " Конструктор " в группе Тип запроса нажмите кнопку Определение данных.

Скрытые бланке и отображается вкладка объекта SQL представления.

Введите следующую инструкцию SQL:

Создание ИНДЕКСА д YearIndex автомобилей (год)

На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить.

К началу страницы

Создание ограничения или связи
Ограничение устанавливает логического условия, которым должно удовлетворять поле или сочетание полей в при вставке значений. Например уникальности запрещает принимать значение, которое будет дублировать существующие значения для поля ограниченное поля.

Отношение — тип ограничения со ссылками на значения поля или сочетание полей в другой таблице, чтобы определить, является ли значение можно вставлять в ограниченное поле или сочетание полей. Не используйте специальный ключевое слово означает, что ограничения связи.

Чтобы создать ограничение, используйте предложение CONSTRAINT в команде CREATE TABLE или ALTER TABLE. Существует два типа предложения CONSTRAINT: один для создания ограничения на одно поле, а другой — для создания ограничения по нескольким полям.

Ограничения для одного поля
Предложение CONSTRAINT одного поля немедленно включает определение поля он ограничивает и имеет следующий синтаксис:

```
CONSTRAINT constraint_name {PRIMARY KEY | UNIQUE | NOT NULL |
 REFERENCES foreign_table [(foreign_field)]
 [ON UPDATE {CASCADE | SET NULL}]
 [ON DELETE {CASCADE | SET NULL}]}
```

Предположим, что у вас есть таблица с именем автомобилей с полями, содержащими имя, год, цены и условия, используемые автомобилей, которые вы собираетесь приобрести. Также предположим, что часто забыли значение для условия автомобилей ввода всегда, что вы хотите записать эту информацию. Вы можете создавать ограничение в поле условие, которая предотвращает оставить пустым, поля, выполнив следующие действия:

На вкладке Создание в группе макрос и код нажмите кнопку Конструктор запросов.

Закройте диалоговое окно Добавление таблицы.

На вкладке " Конструктор " в группе Тип запроса нажмите кнопку Определение данных.

Скрытые бланке и отображается вкладка объекта SQL представления.

Введите следующую инструкцию SQL:

Инструкции ALTER таблицы автомобилей ALTER СТОЛБЦА условие текст ограничение ConditionRequired NOT NULL

На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить.

Предположим, что через некоторое время вам Обратите внимание на то, что имеется много похожих значений в поле условие, которое должно быть одинаковым. Например некоторые автомобили имеют Низкое значение условия и другие пользователи имеют значение плохих.

Примечание: Если вы хотите подписаться, а также остальные процедуры, добавьте некоторые поддельных данные автомобилей таблицу, созданную на предыдущем шаге.

После очистки значений, чтобы они были более согласованное, можно создать таблицу с именем CarCondition, и одно поле с именем условие, содержащий все значения, которые вы хотите использовать для условия автомобилей:

На вкладке Создание в группе макрос и код нажмите кнопку Конструктор запросов.

Закройте диалоговое окно Добавление таблицы.

На вкладке " Конструктор " в группе Тип запроса нажмите кнопку Определение данных.

Скрытые бланке и отображается вкладка объекта SQL представления.

Введите следующую инструкцию SQL:

Создание таблицы CarCondition (условие TEXT(10))

На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить.

Создание первичного ключа в таблице с помощью инструкции ALTER TABLE:

Инструкции ALTER CarCondition ALTER СТОЛБЦА условие текст ограничение CarConditionPK ПЕРВИЧНОГО ключа таблицы

Чтобы вставить значения из поля условия в таблице автомобилей в новую таблицу CarCondition, введите следующие SQL на вкладку Вид объекта SQL:

INSERT INTO SELECT ключевое слово DISTINCT CarCondition условие из автомобилей;

Примечание: Инструкцию SQL в этом шаге является TE000126562. В отличие от управляющий запрос запрос на добавление заканчивается точкой с запятой.

На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить.

Создание связи с помощью ограничения
Если требуется, что любое новое значение, вставленные в поле условие в таблице автомобилей совпадает со значением поля условия в таблице CarCondition, нажмите можно создать связи между CarCondition и автомобилей на поле с именем условия, с помощью указанных ниже действий процедура:

На вкладке Создание в группе макрос и код нажмите кнопку Конструктор запросов.

Закройте диалоговое окно Добавление таблицы.

На вкладке " Конструктор " в группе Тип запроса нажмите кнопку Определение данных.

Скрытые бланке и отображается вкладка объекта SQL представления.

Введите следующую инструкцию SQL:

Инструкции ALTER таблицы автомобилей ALTER СТОЛБЦА условие текст

ограничение FKeyCondition ссылки CarCondition (состояние)

На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить.

Составные ограничения
Предложение CONSTRAINT нескольких полей можно использовать только из-за пределов предложение определение поля и имеет следующий синтаксис:

```
CONSTRAINT constraint_name
{PRIMARY KEY (pk_field1[, pk_field2[, ...]]) |
 UNIQUE (unique1[, unique2[, ...]]) |
 NOT NULL (notnull1[, notnull2[, ...]]) |
 FOREIGN KEY [NO INDEX] (ref_field1[, ref_field2[, ...]])
  REFERENCES foreign_table
  [(fk_field1[, fk_field2[, ...]])] |
 [ON UPDATE {CASCADE | SET NULL}]
 [ON DELETE {CASCADE | SET NULL}]}
```

Рассмотрим другой пример, в котором используется таблица автомобилей. Предположим, что вы хотите убедитесь в том, что никакие две записи в таблице автомобилей один набор значений для имени, год, условия и цены. Можно создать уникальности, к которой применяется для этих полей, выполнив следующие действия:

На вкладке Создание в группе макрос и код нажмите кнопку Конструктор запросов.

Закройте диалоговое окно Добавление таблицы.

На вкладке " Конструктор " в группе Тип запроса нажмите кнопку Определение данных.

Скрытые бланке и отображается вкладка объекта SQL представления.

Введите следующую инструкцию SQL:

ИЗМЕНИТЬ ТАБЛИЦУ автомобилей добавить ограничение NoDupes УНИКАЛЬНЫХ (название, год, условия, цена)

На вкладке Конструктор в группе Результаты нажмите кнопку Выполнить.

Практическая работа №13
«Работа с журналом аудита базы данных»

Цель работы:

- Определение необходимости использования средств аудита
- Отслеживание команд и доступа к объектам базы данных на системном

уровне

- Отслеживание режимов аудита по словарю данных.
- Просмотр и контроль результатов аудита.

Форма отчета:
−выполнить задание;
−показать преподавателю;
−ответить на вопросы преподавателя.
Время выполнения: 2 ч

2. Методические указания

Лабораторная работа направлена на использование средств аудита в целях отслеживания действий, совершённых пользователями над объектами.

Требования к результатам выполнения лабораторного практикума:

- при выполнении задания необходимо сопровождать все

проделанные действия скриншотами и описаниями к ним;

- также необходимо придерживаться строгой последовательности действий,

при выполнении заданий;

- сделать небольшие выводы по каждой части лабораторной работы;
- особо обратить внимание:
  - как использовать журнал аудита
  - как назначать привилегии как пользователям, так и объектам БД.

При составлении и оформлении отчета следует придерживаться рекомендаций, представленных на странице http://unesco.kemsu.ru/student/rule/rule.html.

3. Теоретический материал

С помощью средств аудита базы данных можно отследить и зарегистрировать действия пользователей.

Назначение аудита баз данных:

- Расследование подозрительной деятельности
- Наблюдение за функционированием базы данных
- Сбор информации о функционировании базы данных.

Операции аудита:

- Аудит команд.
- Аудит привилегий.
- Аудит объектов.

Журнал аудита:

- Вся собранная информация хранится в журнале аудита.
- Запись в журнал ведется только при включении средств аудита.

При выборе стратегии аудита базы данных придерживайтесь следующих правил:

- Общий подход:
  - Уточните цели аудита.
  - Старайтесь как можно реже изменять режимы аудита
- При расследовании подозрительной деятельности:
  - Сначала определите конкретные контролируемые действия пользователей, а затем включите аудит для этих действий.
  - Контролируйте доступ к журналу аудита.
- При сборе информации о функционировании БД:
  - Отслеживайте только интересующие вас события.
  - Архивируйте и очищайте журнал аудита.

Создание триггеров базы данных для контроля значений изменяемых данных.

Пример:

Аудит содержимого таблицы EMP.

```
CREATETRIGGER audit_employee
AFTERINSERT OR DELETE OR
UPDATEON emp
FOR EACH ROW
BEGIN
IF     auditpackage.reason IS NULL THEN
       raise_application_error(-20501,
       ’Must specify reason for update before
       performing update; use
       ”auditpackage.set_reason()’”) ;
END IF;

INSERT INTO audit_employee
```

```
VALUES (  :OLD.ssn,    :OLD.name,
          :OLD.class ,      :OLD.sal,
          :NEW.ssn,   :NEW.name,
          :NEW.class,      :NEW.sal,
          auditpackage.reason,
          USER, SYSDATE);
END;
```

```
CREATE TRIGGER audit_emp_cleanup
AFTER INSERT OR DELETE OR UPDATE
ON emp
BEGIN
auditpackage.reason := NULL;
END;
```

Хотя команды AUDIT и NOAUDIT могут быть использованы в любое время, записи в журнале аудита могут быть созданы и сохранены, только если администратор базы данных установил значение параметра инициализации AUDIT_TRAIL

Синтаксис

```
AUDIT TRAIL = значение
```

где:

- значение может быть следующим:
  - DB разрешает аудит базы данных и сохранение записей в журнале

    аудита (sys.aud$).
  - OS включает аудит базы данных и сохраняет записи в файле

    операционной системы (если это возможно в данной ОС)
  - NONE запрещает аудит базы данных (значение по умолчанию).

При изменении значения параметра инициализации AUDIT TRAIL новое значениу будет действовать только после перезагрузки экземпляра базы данных.

Если аудит не нужен, то выполните командный файл catnoaud.sql, который удалит все связанные с ним таблицы и представления. При необходимости эти таблицы всегда можно создать, выполнив командный файл calaudit.sql. Расположение командных файлов calaudit.sql и calnoaudil.sql зависит от операционной системы.

События аудита, отслеживаемые всегда

Независимо от того, включен или выключен аудит, сервер Orac1e

всегда будет записывать в журнал аудита информацию о следующих действиях в базе данных.

Запуск экземпляра

В журнал записывается следующая информация: пользователь операционной системы, запустивший экземпляр, идентификатор терминала, дата и время запуска,
разрешено или нет использование средств аудита базы данных.

Остановка экземпляра

В журнал записывается следующая информация: пользователь операционной системы, остановивший экземпляр, идентификатор терминала, дата и время остановки.

Соединение с БД с привилегиями АБД

В журнал записывается следующая информация: пользователь операционной системы, который соединился с сервером Oracle как sysoper или sysdha.

Избирательность аудита

Избирательность аудита уменьшает размер журнала. Если заданы обобщенные параметры аудита, то журнал может быстро заполняться несущественной информацией. Рекомендуется ограничивать область действия аудита:

- успешным и неудачным выполнением определенных команд SQL;
- сеансами пользователя (BY SESSION) или обращениями к конкретным объектам (BY ACCESS).

Аудит команд и привилегий

- С помощью аудита команд и привилегий можно протоколировать действия всех пользователей базы данных или только пользователей согласно заданному списку.

Аудит объектов

- Избирательное протоколирование действий, которые выполняются над конкретными объектами схемы.
- Аудит объектов всегда задается для всех пользователей базы данных. Этот режим аудита нельзя задать для списка пользователей.

  Аудит не ведется для пользователей базы данных INTERNAL и sys.
  Аудит команд

и привилегий включается в текущем сеансе с помощью команды AUDIT, но начинает реально работать для всех последующих сеансов. Для выполнения команд AUDIT и

NOAUDIT требуется привилегия AUDIT SYSTEM.

Если вы пропустили фразу WHENEVER, сервер Orac1e будет протоколировать успешные и неудачные попытки выполнения команд. При протоколировании неудачных попыток запись в журнале не будет

создана, если выполнялась неправильная по синтаксису команда.

Аудит команд

Возможен аудит команд SQL, относящихся к двум категориям.

- Аудит команд DDL задается для определенного типа объектов базы данных.
- Аудит команд SELECT и DML задается для определенного типа объектов базы данных.

Рекомендации

- Команды SQL внутри блоков PL/SQL протоколируются индивидуально по мере выполнения блока.
- Аудит команд выполняется только для локальных пользователей. Нельзя запротоколировать действия, которые пользователь выполнял в удаленной базе данных.
- Используйте триггеры для контроля изменений таблиц.

Неудачное выполнение команды будет зафиксировано только в случае, если сбой произошел из-за недостаточности привилегий или из-за обращения к несуществующему объекту. Неудачное выполнение команды не фиксируется, если выполнялась ошибочная по синтаксису команда.

Включение аудита команд

Команда AUDIT включает аудит команд.

Синтаксис

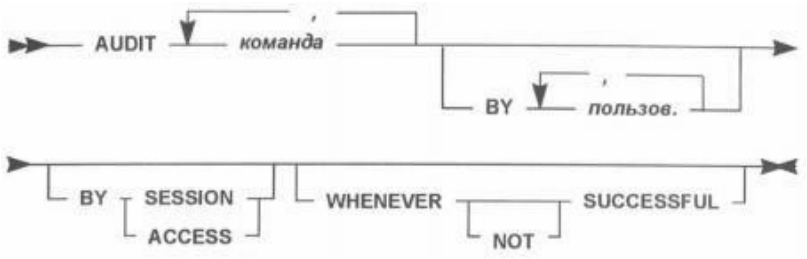


Рис.1. Синтаксис запроса на включение аудита команд.

где:

- *команда* - команды SQL, которые будут протоколироваться.
- *Пользов.* - пользователь или пользователи, для которых включается аудит команд SQL. Если пользователи не заданы, аудит включается для всех пользователей БД.
- BY SESSION - при выполнения указанных команд в журнале создается только одна протокольная запись для каждого объекта БД в каждом сеансе,

независимо от того, сколько раз выполнялись одинаковые команды SQL.

- BY ACCESS - в журнале создаются протокольные записи для каждого выполнения указанных команд. Если вы задаете аудит

команд DDL и привилегий для команд DDL, то сервер Oгас1е будет выполнять аудит в режиме BY ACCESS,
независимо от того, что вы задали. Для остальных команд SQL по умолчанию в BY SESSION

- WHENEVER SUCCESSFUL - в журнале создаются протокольные записи только успешно завершившихся команд SQL.

- NOT в журнале создаются протокольные записи только для неудачно завершившихся команд SQL

Примеры:

Чтобы отслеживать все попытки изменения атрибутов пользователей, необходимо включить аудит команд CREATE/ALTER/DROP USER, независимо от того, успешно или неудачно они завершились.

```
SQL>AUDIT user BY ACCESS;
Audit succeeded.
```

Чтобы фиксировать все соединения с базой данных, в команде включения аудита используется фраза CONNECT.

```
SQL>AUDIT connect;
Audit succeeded.
```

Выборка из представления DBA STMT AUD1T OPTS

```
SQL>COL user_name FORMAT A10
SQL>COL audit_option FORMAT A13
SQL>SELECT user_name, audit_option, success, failure
2 FROM sys.dba_stmt_audit_opts;
```

USER NAME	AUDIT OPTION	SUCCESS	FAILURE
	USER	BY ACCESS	BY ACCESS
	CREATE SESSION	BY ACCESS	BY ACCESS

Аудит привилегий

Выборочный аудит команд, которые выполняются с указанными привилегиями. Можно выполнять аудит действий конкретных или всех пользователей базы данных. Аудит привилегий функционально и синтаксически очень похож на аудит команд. Аудит должен быть очень селективным, чтобы сократить до минимума количество информации, записываемой в журнал аудита

- Возможен аудит любой системной привилегии

- Существует более 80 различных привилегий, для которых можно включить аудит.

Аудит привилегий включается командой AUDIT.

Синтаксис

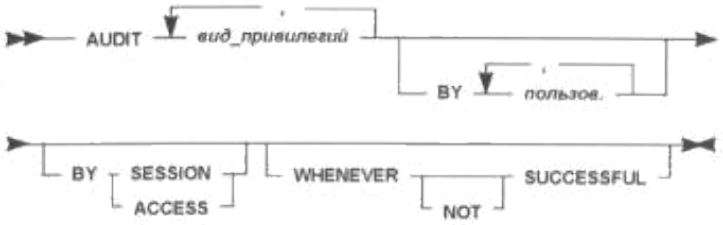


Рис.2. Синтаксис запроса на включение аудита привилегий.

где:

- вид_привилегии - задает системные привилегии (и, следовательно,

команды SQL), которые будут протоколироваться.

- пользователь - задается пользователь или пользователи, для которых включается аудит привилегий. Если пользователи не заданы, аудит будет выполняться для всех пользователей БД.
- BY SESSION - в журнале создается только одна протокольная запись для каждого пользователя и объекта БД в каждом сеансе, независимо оттого, сколько раз выполнялись одинаковые команды SQL.
- BY ACCESS - в журнале создаются протокольные записи для каждого выполнения указанных команд. Если указан аудит привилегий команд DDL, то аудит будет выполняться в режиме BY ACCESS, независимо от значения данного параметра. Для остальных команд SQL по умолчанию - BY SESSION.
- WHENEVER SUCCESSFUL - в журнале создаются протокольные записи только успешно завершившихся команд SQL.
- NOT в журнале создаются протокольные записи только для неудачно завершившихся команд SQL.

Примеры:

Чтобы следить за успешными и неудачными попытками создания таблиц или индексов пользователя scott в его схеме необходимо выполнить следующую команду:

```
SQL>AUDIT create table BY scott BY
ACCESS; Audit succeeded.
```

Чтобы следить за успешными попытками изменения таблиц, процедур, функций или пакетов пользователем scott в любой схеме, необходимо выполнить следующую комацду:

```
SQL>  AUDIT alter any table, alter any procedure
2       BY scott BY ACCESS
3       WHENEVER
SUCCESSFUL; Audit
succeeded.
```

Выборка данных из представления DBA PRIV AUDIT OPTS

```
SQL>  SELECT * FROM sys.dba_priv_audit_opts;
USER_NAME  PRIVILEGE                 SUCCESS    FAILURE
                                                BY
SCOTT      CREATE TABLE              BY ACCESS  BY ACCESS
SCOTT      ALTER ANY TABLE           BY ACCESS  NOT SET
           ALTER ANY                 BY
SCOTT      PROCEDURE                 ACCESS     NOT SET

3 rows selected.
```

Аудит объектов

Аудит объектов - это выборочное протоколирование команд, в которых есть обращение к указанным объектами базы данных. Возможен аудит команд DML и команд SQL SELECT, GRANT и REVOKE, в которых есть обращение к указанным объектами базы данных.

Контролируемые объекты

- Таблицы.
- Представления.
- Последовательности.
- Пакеты.
- Автономные хранимые процедуры.
- Автономные хранимые функции.

Нельзя вести аудит отдельных процедур пакета. Представления и процедуры могут ссылаться на другие объекты, поэтому в журнале, возможно, будут созданы несколько протокольных записей при использовании этих объектов. Аудит объектов всегда задается только для всех пользователей базы данных. Этот режим аудита нельзя задать для конкретных пользователей.

Параметры аудита объектов

Параметр	**Таблица**	**Представление**	**Последовательность**	**Момент. копия**	**Хранимые процедуры**
ALTER	X	X			

AUDIT	X	X	X	X	
COMMENT	X	X			
DELETE	X	X			
EXECUTE	X				
GRANT	X	X	X	X	
INDEX	X				

INSERT	X	X			
LOCK	X	X			
RENAME	X	X	X		
SELECT	X	X	X	X	
UPDATE	X	X			

Команда AUDIT включает аудит объектов.

Синтаксис

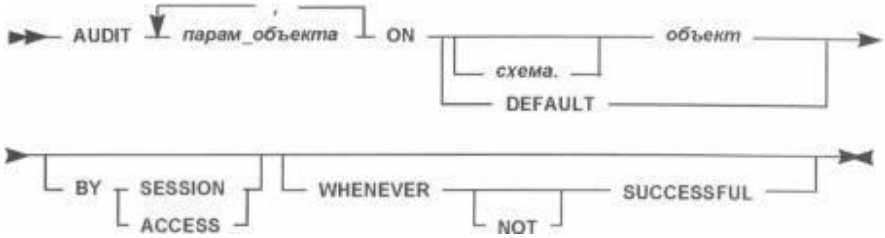


Рис.3. Синтаксис запроса на включение аудита объектов

где:

- *парам_объекта* - задает операции, для которых будет вестись аудит.
- *объект* - задает объект, для которого будет вестись аудит. Объектом могут

быть: одна или несколько таблиц, представления, пакеты хранимые процедуры и функции, моментальные копии. Если схема не задано, то сервер Oгас1е подразумевает объект в вашей схеме.

- DEFAULT - указывает, что заданные параметры аудита будут использоваться по умолчанию для всех позднее созданных объектов.
- BY SESSION - в журнале создается только одна протокольная запись для каждого сеанса, независимо от того, сколько раз выполнялись одинаковые команды SQL.
- BY ACCESS - в журнале создаются протокольные записи для каждого выполнения указанных команд. По умолчанию - BY SESSION.
- WHENEVER SUCCESSFUL - в журнале создаются протокольные записи только успешно завершившихся команд SQL.

- NOT - в журнале создаются протокольные записи только для неудачно завершившихся команд SQL.

Объект должен находиться в вашей схеме или вы должны иметь привилегию AUDIT ANY. Фразу ALL можно использовать вместо параметров аудита. ALL включает все параметры аудита, которые возможны для данного типа объектов.

Пример:

Режим создания одной протокольной записи на сеанс при выполнении процедуры CHANGEPRICE EXECUTE.

SQL> AUDIT EXECUTE ON change_price BY

SESSION; Audit succeeded.

Аудит каждого удаления информации из

таблицы EMP SQL> AUDIT DELETE ON

emp BY ACCESS; Audit succeeded.

Аудит выполнения команды GRANT на таблицу EMP. Записывать успешные попытки выполнения этой команды один раз на сеанс.

SQL> AUDIT GRANT ON emp BY SESSION WHENEVER

SUCCESSFUL; Audit succeeded.

Протоколирование удачного выполнения GRANT для

CHANGEPRICE. SQL> AUDIT GRANT ON change_price

WHENEVER SUCCESSFUL; Audit succeeded.

Протоколирование всех неудачных попыток блокировать таблицу EMP,

SQL> AUDIT LOCK ON emp BY ACCESS WHENEVER NOT

SUCCESSFUL; Audit succeeded.

Параметры аудита по умолчанию устанавливаются для объектов, которые будут созданы позднее.

Параметры аудита объектов по умолчанию

- Параметры по умолчанию устанавливаются с помощью фразы DEFAULT.
- Параметры аудита по умолчанию присваиваются всем объектам, которые будут созданы позднее.
- Изменить параметры по умолчанию можно с помощью явной установки требуемого параметра аудита.
- Любое последующее изменение параметров аудита по умолчанию не отражается на параметрах аудита для уже существующих объектов.

Для установки параметра аудита DEFAULT требуется системная привилегия AUDIT SYSTEM. Включение параметров аудита ALTER, SELECT и RENAME для всех объектов, которые будут созданы в дальнейшем.

```
SQL>  AUDIT alter, select, rename ON default;
Audit succeeded.
```

В предыдущем примере

- После создания таблицы будут протоколироваться команды ALTER,

- После создания представления будут протоколироваться команды

SELECT и RENAME, которые используют данное представление.

- После создания последовательности будут протоколироваться команды

ALTER и SELECT, которые используют данную последовательность.

- После создания моментальной копии будет протоколироваться команда

- После создания пакета или автономных процедур и функций будет протоколироваться команда RENAME, которая переименовывает данные объекты.

Выключение аудита команд и привилегий

Аудит команд и привилегий выключается командой NOAUDIT.

Синтаксис

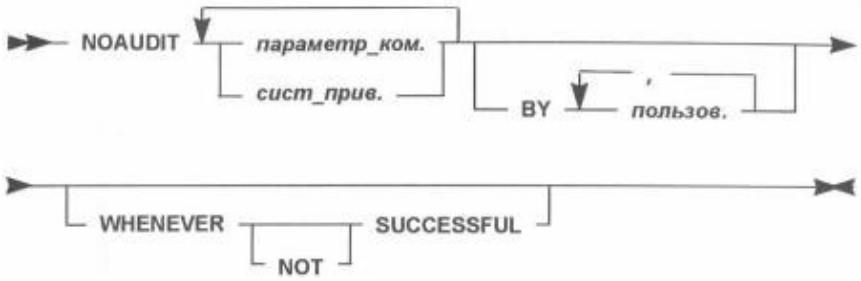


Рис.4. Синтаксис запроса на отключение аудита команд и привилегий.

где:

- *параметр_ком* - задает команды, для которых выключается аудит
- BY *пользов*. - выключает аудит команд SQL для всех последующих сеансов конкретного пользователя. Если эта фраза не указана, сервер Oracle выключает аудит для всех пользователей.
- WHENEVER SUCCESSFUL - выключает аудит удачно завершившихся команд.
- NOT выключает аудит для всех команд, завершившихся ошибкой

Для выполнения этой команды требуется привилегия AUDIT SYSTEM.

Пример:

Выключить аудит удачных и неудачных попыток создания таблиц или индексов пользователя scoff в его схеме.

SQL> NOAUDIT create table BY scott;

Noaudit succeeded.

Выключить аудит удачных попыток изменять таблицы, процедуры, функции или пакеты пользователя scott в любой схеме.

```
SQL>  NOAUDIT alter any table, alter any procedure
2       BY scott
3       WHENEVER SUCCESSFUL;
```

Noaudit succeeded.

Выключить аудит объектов можно с помощью команды NOAUDIT.

Синтаксис

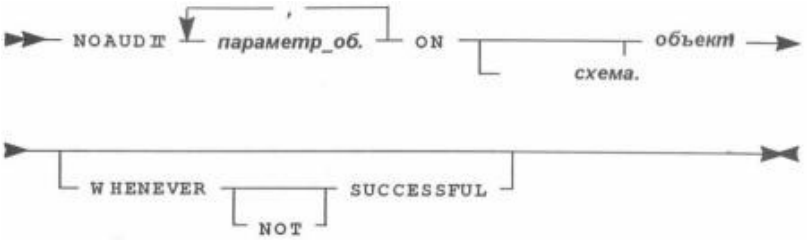


Рис.5. Синтаксис запроса на отключение аудита объектов

где:

- *параметр_об*. - задает выключаемые параметры.
- ON - ключевое слово. Указывается перед именем объекта для которого выключается аудит. Если схема пропущена, то подразумевается ваша собственная схема
- WHENEVER SUCCESSFUL выключает аудит удачно завершившихся параметр_об.
- NOT выключает аудит команд, которые завершаются с ошибкой.

Объект, для которого выключается аудит, должен быть в вашей схеме или вы должны иметь привилегию AUDIT ANY.

Примеры:

Выключить аудит выполнения процедуры

CHANGE PRICE. SQL> NOAUDIT execute

ON change_price; Noaudit succeeded.

Выключить аудит удаления информации из таблицы EMP.

SQL> NOAUDIT delete ON emp;

Noaudit succeeded.

Выключить аудит успешной выдачи привилегий на

таблицу EMP SQL> NOAUDIT grant ON emp

2 WHENEVER

SUCCESSFUL; Noaudit

succeeded.

Команда NOAUDIT удаляет ссылки на выключаемые параметры

аудита из словаря данных.

Журнал аудита

Журнал аудита содержит записи, сформированные в результате аудита команд, привилегий или объектов. Журнал аудита является таблицей словаря данных SYS.AUDS

Каждая запись журнала аудита содержит:

- Имя пользователя, выполнившего команду
- Код вида операции (число), который однозначно указывает тип выполненной команды или использованной привилегии.
- Объект, на который ссылается команда
- Дату и время выдачи команды.

Записи в журнал аудита заносятся во время фазы выполнения команды.

Рекомендации:

- Аудит не зависит от пользовательских транзакций.

Следовательно, если транзакция откатывается, протокольная запись остается.

- Аудит по обращениям (BY ACCESS) заносит записи в журнал каждый раз,

когда выполняется указанное действие.

- Аудит по сеансам (BY SESSION) (время между соединением и отсоединением заносит записи в журнал только один раз на сеанс для указанного объекта.

Контроль журнала аудита

Контроль размера и темпов роста журнала аудита сохраняет дисковое пространств и гарантирует бесперебойную работу базы данных. При заполнении журнала нельзя вставлять новые протокольные записи, поэтому команды, для которых включен аудит, будут выдавать сообщения об ошибке.

Рост журнала аудита зависит от:

- Числа включенных параметров аудита.
- Частоты выполнения команд, для которых включен аудит.

Максимальный размер журнала определяется во время создания базы данных. таблицы AUDS. Командный файл sql.bsq для создания базы данных содержит команду CREATE для таблицы AUDS

Если журнал аудита переполнен, то команды, для которых включен аудит, будут завершаться с ошибкой. Предупреждения будут получать все

пользователи, которьм выполняют такие команды. Администратор должен почистить журнал аудита перед выполнением таких команд.

Убедитесь, что размеры журнала аудита не растут слишком быстро.

Управление темпами роста журнала

- Включайте аудит только при необходимости.
- Четко выбирайте параметры аудита.
- Строго контролируйте результаты аудита объектов.

Рекомендации по ограничению аудита объектов

- Только администратор по защите имеет право включать аудит объектов.
- Администратор по защите владеет всеми объектами схемы.
- Все объекты содержаться в схемах, пользователи которых не являются реальными пользователями базы данных (т.е. не имеют системной привилегии

CREATE SESSION)

Для аудита объекта вы должны быть либо его владельцем, либо иметь привилегию AUDIT ANY.

Следовательно, для гарантии того, что только администратор по защите может включать аудит объектов, вы должны:

1. Выдать системную привилегию AUDIT ANY только администратору по защите данных.
2. Не позволять реальным пользователям базы данных создавать объекты в своих схемах.

Управление журналом аудита

Для поддержки приемлемого размера журнала аудита необходимо периодически удалять протокольные записи и удалять журнал аудита.

Очистку журнала аудита можно производить путем:

- Удаления всех протокольных записей.
- Выборочного удаления протокольных записей.
- Переноса протокольных записей в другую таблицу
- Переноса протокольных записей в файлы операционной системы.

Примеры:

Удалить все записи из журнала аудита.

```
SQL>  TRUNCATE TABLE sys.aud$;
Table truncated.
```

Удалить все протокольные записи, которые были созданы более трех месяцев

назад.

```
SQL>  DELETE FROM sys.aud$
```

2 WHERE TIMESTAMP<

SYSDATE-90; n rows deleted.

Архивирование протокольных записей в файлы операционной системы Используйте утилиту Export в режиме экспорта таблицы. Помните, что в этом

случае будет выполнен экспорт всей таблицы. Следующая команда выполнит экспорт таблицы AUDS в файл expdal.dmp.

EXP USERID=sys/password TABLES=(AUD$) FILE=expdat.dmp

Утилита Export не изменит содержимое таблицы SYS.AUDS, поэтому вам все равно придется привести в порядок журнал аудита. Если журнал аудита имеет много экстентов и большинство из них не планируется использовать, то удаление протокольных записей с помощью команды TRUNCATE может уменьшить размер выделенного журналу пространства.

Уменьшение размера журнала аудита

1. Скопируйте протокольные записи в другую таблицу базы данных или экспортируйте их в файл операционной системы.
2. Соединитесь с базой данных как пользователь sys.
3. Используя команду TRUNCATE, удалите записи из таблицы SYS.AUDS.
4. Загрузите в журнал протокольные записи, сохраненные на шаге 1.

Сохраните журнал аудита только при необходимости, в противном случае пропустите шаги 1 и 4. SYS.AUD$ - это один из немногих объектов пользователя sys, которые можно непосредственно изменять.

Протокольные записи из журнала может удалять пользователь sys или любой другой пользователь, имеющий привилегию DELETE ANY TABLE.

Если включен аудит соединений и журнал заполнен, а пользователя ждут соединения, то администратор по защите данных, должен соединиться с БД как пользователь sys (соединения пользователя sys не протоколируются) и очистить пространство, занимаемое журналом.

Если журнал аудита заполнен, то протоколируемые команды невозможно выполнять, так как невозможно создать соответствующую протокольную запись

Ограничения по параметрам хранения

- Если табличное пространство заполнено, сервер Oracle не сможет выделить следующий экстент требуемого размера. Администратор

должен расширить табличное пространство, добавив новый файл данных.

- Если размер журнала аудита достиг максимального количества экстентов, то администратору необходимо повторно создать таблицу, увеличив значения

  параметров для экстентов.

При расследовании подозрительной деятельности администратор по защите данных должен обеспечить защиту журнала аудита.

Пример:

Аудит журнала аудита.

```
SQL>  AUDIT insert, update, delete
```

2 ON sys.aud$

3 BY ACCESS;

Audit succeeded.

Для защиты журнала аудита от несанкционированного удаления только администратор по защите данных должен иметь привилегию DELETE ANY TABLE

Вывод информации аудита

Заранее определенные представления журнала аудита создаются с помощью командного файла *cataudit.sql.* Удалить все эти представления можно с помощью командного файла *catnoaud.sql.* Для просмотра протокольной информации используются представления журнала аудита.

Представления журнала аудита

Представления	**Описание**
STMT_AUDIT_OPTION_MAP	Связь номера параметра с его именем.
AUDIT_ACTIONS	Связь номера действия с его именем.
ALL_DEF_AUDIT_OPTS	Параметры аудита по умолчанию.
DBA_STMT_AUDIT_OPTS	Параметры аудита команд.
DBA_PRIV_AUDIT_OPTS	Параметры аудита системных привилегий.
DBA_OBJ_AUDIT_OPTS	Параметры аудита объектов.
USER_OBJ_AUDIT_OPTS	Параметры аудита объектов.
DBA_AUDIT_TRAIL	Все записи журнала аудита
USER_AUDIT_TRAIL	Записи журнала для данного пользователя.
DBA_AUDIT_SESSION	Записи обо всех соединениях и отсоединениях с БД.
USER_AUDIT_SESSION	Записи о соединениях и отсоединениях с БД данного пользователя.

DBA_AUDIT_STATEMENT	Все записи, относящиеся к аудиту АБД.
USER_AUDIT_STATEMENT	То же, но для данного пользователя.
DBA_AUDIT_OBJECT	Все записи, относящиеся к объектам.
USER_AUDIT_OBJECT	То же, но для текущего пользователя.
DBA_AUDIT_EXISTS	Все записи для EXISTS/NOT EXISTS

Пример

Аудит удаления данных из заданной таблицы.

SQL> CONNECT system/password

SQL> AUDIT delete ON scott.emp BY SESSION WHENEVER SUCCESSFUL;

SQL> CONNECT adams/wood

SQL> DELETE scott.emp WHERE empno = 1111;

SQL> scott/tiger

CONNECT enp WHERE empno = 1111;

SQL> DELETE system/password

SQL> scott.emp WHERE empno = 7788;

CONNECT

SQL> DELETE

4. Порядок выполнения работ

Использование средств аудита базы данных.

1. Предположим, что пользователь system должен контролировать выполняемые в базе данных операции. Для выполнения этой обязанности разрешите использование в базе данных средств аудита. Как это делается?
2. Допустим, что пользователь system должен контролировать попытки подсоединения к базе данных незарегистрированных пользователей.
    a. Установите контроль.
    b. Убедитесь, что подсоединения контролируются.
    c. Попытайтесь подсоединиться к базе данных как scott/leo. Что получилось?
    d. Подсоединитесь к базе данных как scott/tiger. Что получилось?
    e. Подсоединитесь к базе данных как system для проверки результатов аудита.
    f. В каком представлении содержится информация о попытках проникновения в базу данных незарегистрированных пользователей? Найдите в этом представлении попытку “взлома” базы данных пользователем scott.
3. Допустим, что пользователь ernie подозревается в несанкционированных изменениях квот и переназначении табличных пространств по умолчанию.
    a. Предоставьте привилегии ALTER USER пользователю ernie и установите за ним контроль.

b. Подсоединитесь к базе данных как ernie/ernie.
c. Установите пользователю scott квоту 1 Мб в табличном пространстве USER_DATA, а пользователю bert - табличное пространство по умолчанию USER_DATA.
d. Найдите изменения квот и переназначения табличных пространств по умолчанию пользователем ernie. Можно ли в журнале аудита отличить изменения квот от переназначения табличных пространств?

4. Каким образом можно контролировать успешные обновления таблицы S_EMP пользователя scott? Установите контроль.
    a. Убедитесь в том, что контроль обновлений таблицы S_EMP пользователь scott установлен?
    b. Подсоединитесь к базе данных как system и просмотрите журнал аудита, обновлял таблицу S_EMP пользователя scott?
5. Отключите аудит.
6. Каким образом можно определить для журнала аудита файл ОС (вместо базы данных)? Выполните требуемые действия и убедитесь в том, что переопределение журнала аудита прошло успешно.
7. Для архивирования содержимого таблицы SYS.AUD$ создайте в табличном пространстве USER DATA ее копию под именем AUDX.
8. Создайте специально для журнала аудита табличное пространство AUD и пересоздайте там таблицу SYS.AUD$. Там же создайте составной индекс I_AUD1 для столбцов SESSIONID и SESS$ID таблицы SYS.AUD$. Убедитесь в том, что аудит работает и запретите использование в базе данных средств аудита.

Практическая работа №14
«Мониторинг нагрузки сервера»

Цель работы: Изучить принципы работы простейших средств мониторинга сети, получить навыки решения задач, связанных с мониторингом сети.

Форма отчета:
−выполнить задание;
−показать преподавателю;
−ответить на вопросы преподавателя.

Время выполнения: 2 ч

Теоретические основы

1. Протокол ICMP

Протокол ICMP (Интернет-протокол контрольных сообщений) стека протоколов TCP/IP предназначен для передачи между сетевыми устройствами сообщений об ошибках и контрольных сообщений при помощи IP-пакетов.

В протоколе ICMP определены несколько типов сообщений, в том числе:

Destination Unreachable	Time to Live Exceeded	Parameter Problem
Source Quench	Redirect	Echo
Echo Reply	Timestamp	Timestamp Reply
Information Request	Information Reply	Address Request
Address Reply		

Например, если маршрутизатор получает пакет, который он не может доставить по указанному в нем адресу, отправителю передается ICMP-сообщение о недостижимости адреса (Destination Unreachable).

2. PING: Проверка соединения с определенным интерфейсом.

Программа ping использует протокол ICMP.

Эта команда посылает пакет эхо-запроса на другой IP-адрес и ожидает ответа. Она чаще всего используется для того, чтобы посмотреть, «жив ли» другой компьютер. Ответ на запрос содержит также данные о том, как долго пакет путешествовал до адресата. Можно использовать команду ping с различными опциями: число посланных пакетов (от 1 до 10), время жизни пакета (time to live –TTL, от 1 до 255ms), размер пакета (от 16 до 8192 байт), время ожидания (timeout, до 9999 ms) и разрешать или нет фрагментацию каждого пакета.

Формат команды в ОС Windows:

ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
[-r count] [-s count] [[-j host-list] | [-k host-list]]
[-w timeout] destination-list

Options:

-t Выполнение команды до прерывания (Ctrl+C)

-a Разрешать адреса в имена

-n count Число отправляемых пакетов.

-l size Размер буфера отправки

-f Установить флаг "Не фрагментировать".

-i TTL Установить время жизни.

-w timeout Время ожидания ответа в мс.

-v TOS Задание типа службы (поле "Type Of Service").

-r count Запись маршрута для указанного числа переходов.

-s count Штамп времени для указанного числа переходов.

-j host-list Свободный выбор маршрута по списку узлов.

-k host-list Жесткий выбор маршрута по списку узлов.

destination-list Список рассылки.

3. Программа tracert. Определение промежуточных сетевых интерфейсов между хостами. Трассировка маршрута

Программа трассировки маршрута использует протокол ICMP.

Эта утилита очень похожа на Ping, за исключением того, что она показывает все другие IP-адреса (интерфейсы), которые пакет проходит до своего места назначения. Дополнительно можно изменять различные опции, ассоциированные с Trace Route: максимальное число дозволяемых промежуточных узлов (maximum hops, от 1 до 255) и timeout (до 9999 ms).

Формат команды в ОС Windows:

tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout] target_name

Options:

-d Не разрешать адреса в имена.

-h maximum_hops Наибольшее число промежуточных узлов.

-j host-list Трассировка через определенный список хостов

-w timeout Время ожидания каждого ответа в мс.

4. Программа netstat. Сетевая статистика.

Программа netstat используется для просмотра активных соединений каждого протокола, таблиц маршрутизации, а так же детализирует статистику передачи данных.

Формат команды в ОС Windows:

netstat [-a] [-e] [-n] [-s] [-p имя] [-r] [интервал]

-a Отображение всех подключений и ожидающих портов.

(Подключения со стороны сервера обычно не отображаются).

-e Отображение статистики Ethernet. Этот ключ может применяться вместе с ключом -s.

-n Отображение адресов и номеров портов в числовом формате.

-p имя Отображение подключений для протокола "имя": tcp или udp. Используется вместе с ключом -s для отображения статистики по протоколам. Допустимые значения "имя": tcp, udp или ip.

-r Отображение содержимого таблицы маршрутов.

-s Отображение статистики по протоколам. По умолчанию выводятся данные для TCP, UDP и IP. Ключ -p позволяет указать подмножество выводящихся данных.

Повторный вывод статистических данных через указанный интервал в секундах. Для прекращения вывода данных нажмите клавиши CTRL+C. Если параметр не задан, сведения о текущей конфигурации выводятся один раз.

Задание 1. Выполните команду ping в командной строке с различными значениями параметров -t, -n, -l, -i,-w. Какие наблюдения и выводы вы сделали?

ping www.seun.ru

ping www.sgu.ru

ping www.microsoft.com

ping www.sun.com

ping212.193.38.83

Выполните ping к тем же хостам с параметром –f, увеличивая параметр -l size. При каком значении размера перестают получаться ответы?

Задание 2. Tracert

Выполните команду tracert в командной строке с различными значениями параметров. Какие наблюдения и выводы вы сделали?

Используйте, например,

tracertwww.seun.ru

tracert www.sgu.ru

tracert www.microsoft.com

tracert www.sun.com

tracert 212.193.38.83

Задание 3. Поисковые сервисы Европейского и Российского ip-регистров

Определите, кому принадлежат сети 194.85.33.0, 217.23.64.0, 212.193.38.0. Для этого используйте поисковые аппараты http://www.ripe.net/db/whois/whois.htmlиhttp://www.ripn.net:8080/nic/whois/index.html.

Пользуясь данными этих информационных систем, попробуйте определите географическое расположение сетей. Попробуйте изобразить топологическую схему соединения этих сетей.

Задание 4. Использование программы ping для исследования параметров сети.

1. Приведите сравнительные результаты выполнения команд ping по адресам 194.85.33.29, 194.85.33.30, 217.23.64.2, 212.193.38.248, 212.193.35.10 по параметрам «время отклика», TTL в форме таблицы. Объясните полученные различия.

2. Соберите средние времена прохождения 10 пакетов на указанные адреса. Сравните с р4езультатами, полученными при использовании сервиса ping в интерфейсе Looking Glass на сайте http://noc.runnet.ru. Объясните полученные различия.

3. Соберите усредненные времена прохождения 10 пакетов увеличивающегося размера по указанным адресам. Начните с 64 байт и каждый раз удваивайте размер пакета. При каком размере пакета потери превышают 50 %. Как влияет время ожидания отклика на процент

прохождения пакетов этого размера. При каком времени ожидания отклика потери для пакетов зафиксированного размера не возникают?

Представьте результаты измерений в форме таблиц, наилучшим образом проявляющим, по вашему мнению, обнаруженные зависимости.

4. Используя программу ping, оцените вклад разных сетевых участков, по которым проходит эхо-пакет между вашей рабочей станцией и интерфейсом 194.85.33.29.

Задание 5. Использование программы tracert для анализа соединений в сети.

1. Приведите сравнительны е результаты выполнения команд tracert по адресам 194.85.33.29, 194.85.33.30, 217.23.64.2, 212.193.38.248, 212.193.35.10. Объясните полученные различия.

2. Выполните трассировку к адресу 212.193.38.248 и к адресу 217.23.64.2 со стороны сайта http://noc.runnet.ru. Приведите полученные результаты.

3. Используя данные, полученные в результате выполнения трассировки и отправки эхо-пакетов между интерфейсами 212.193.38.248 и 194.85.35.100, оцените вклад разных участков сетей, соединяющих эти интерфейсы, в среднее время прохождения пакетов между ними.

4. Используя полученную в ходе выполнения всех заданий информацию, уточните схему задания 1, изобразите на ней обнаруженные вами промежуточные интерфейсы и линки сети, объединяющей подсети 194.85.33.0, 217.23.64.0, 212.193.38.0.

Практическая работа №15
«Настройка политики безопасности»

Цель: научиться настраивать политику безопасности.

Форма отчета:
−выполнить задание;
−показать преподавателю;
−ответить на вопросы преподавателя.
Время выполнения: 2 ч

Система безопасности SQL Server основана на концепции защищаемых объектов (securables), т.е. объектов, на которые можно назначать разрешения, и принципалов (principles), т.е. объектов, которым можно назначать разрешения. Принципалами могут быть логины на уровне сервера, пользователи и роли на уровне базы данных. Роли

назначаются пользователям. Разрешения на доступ к объектам могут предоставляться как непосредственно пользователям, так и через роли. Каждый объект имеет своего владельца, и права собственности также влияют на разрешения.

Общая схема системы безопасности SQL Server

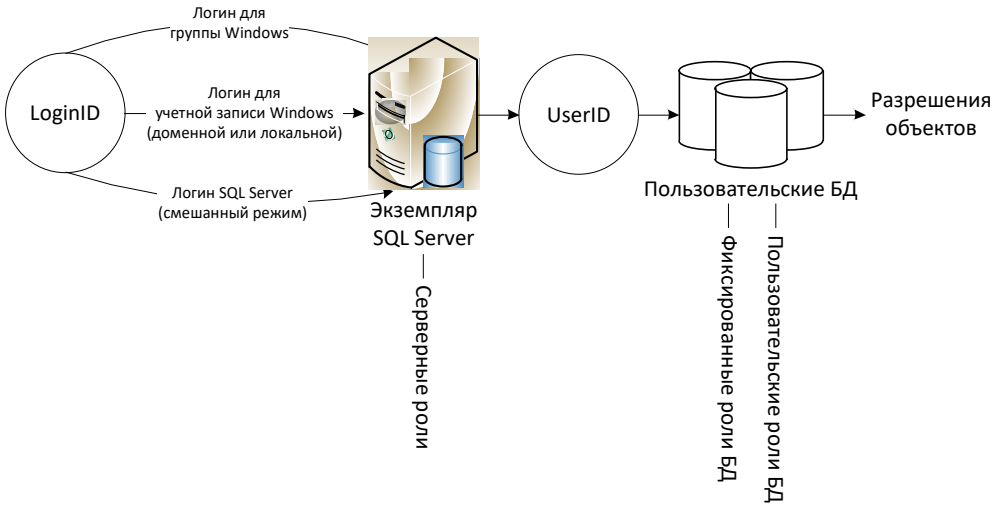


Рис. 11.1

SQL Server использует двухэтапную схему аутентификации. На уровне сервера пользователь распознается по своему идентификатору (LoginID), который может быть либо именем входа SQL Server, либо группой или учетной записью Windows. После входа на сервер пользователь получает те права, которые были назначены ему администратором на уровне сервера, в частности с помощью фиксированных серверных ролей. Если пользователь принадлежит роли sysadmin, то он имеет полный доступ ко всем функциям сервера, а также ко всем базам данных и объектам на нем.

Для получения доступа к базе данных логин пользователя должен быть сопоставлен с соответствующим ему идентификатором пользователя (UserID), который специфичен для каждой базы данных. Вполне возможна ситуация, когда пользователь был распознан в SQL Server, но у него нет

доступа ни к одной из баз данных. Также возможно и обратное: пользователю открыт доступ к базам данных, но он не был распознан сервером. Перемещение базы данных и ее разрешений на другой сервер без параллельного перемещения имен входа сервера может привести к возникновению таких "осиротевших" пользователей.

На уровне базы данных пользователю может быть предоставлен определенный набор разрешений с помощью назначения ему фиксированных ролей базы данных. Все пользователи автоматически становятся членами стандартной роли public, у которой по умолчанию нет никаких разрешений. Пользовательские роли - это дополнительные роли, служащие в качестве групп. Роли может быть разрешен доступ к объектам базы данных, а пользователю могут быть назначены роли.

Разрешения к объектам назначаются с помощью инструкций GRANT (предоставить), REVOKE (отозвать) и DENY (запретить). Запрет привилегии замещает собой ее предоставление, а предоставление привилегии замещает собой ее отзыв. Пользователю может быть предоставлено множество разрешений к объекту (индивидуальных, наследованных от роли, обеспеченных принадлежностью к роли public). Если какая-либо из этих привилегий запрещена, для пользователя блокируется доступ к объекту. В противном случае, если какая-либо из привилегий предоставляет разрешение, пользователь получает доступ к объекту.

Разрешения объекта достаточно детализированы. Существуют отдельные разрешения для каждого из возможных действий (SELECT, INSERT, UPDATE, RUN и т.д.) над объектом.

Выбор типа логина и настройка режима аутентификации

SQL Server поддерживает два типа логинов (имен входа):

логин Windows (логин для локальной учетной записи Windows, логин для доменной учетной записи Windows, логин для группы Windows);

логин SQL Server.

При использовании логинов Windows в системные таблицы базы данных master записывается информация об идентификаторе учетной записи или группы Windows (но не пароль). Аутентификация (т. е. проверка имени пользователя и пароля) производится обычными средствами Windows при входе пользователя на свой компьютер.

При использовании логина SQL Server пароль для этого логина (точнее, его хэшированное значение) хранится вместе с идентификатором логина в базе данных master. При подключении пользователя к серверу ему придется указать имя логина и пароль.

Предпочтительный вариант логина для пользователя - это логин Windows, при этом не для учетной записи, а для группы (лучше всего для локальной доменной группы). Преимуществ у такого решения множество:

пользователю достаточно помнить один пароль - для входа на свой компьютер;

повышается уровень защищенности SQL Server. Это происходит, по крайней мере, за счет того, что пароль не будет передаваться по сети открытым текстом, как это происходит по умолчанию при использовании команд CREATE LOGIN и ALTER LOGIN. Кроме того, хэши Windows более защищены, чем хэши логинов SQL Server;

проверка при входе пользователя производится быстрее.

Эти преимущества справедливы для любых логинов Windows: как для учетных записей, так и для групп. Но при использовании логинов для групп Windows появляются дополнительные преимущества:

снижается размер системных таблиц базы данных master, в результате чего аутентификация производится быстрее. На одном сервере SQL Server вполне может быть несколько тысяч логинов для пользователей Windows или, что значительно удобнее, всего пара десятков логинов для групп;

значительно упрощается предоставление разрешений для новых

учетных записей.

Использование логинов SQL Server может быть обусловлено следующей причиной: очень часто на предприятиях администрированием SQL Server и домена Windows занимаются разные люди, которым сложно согласовывать свои действия. Логины SQL Server позволяют администратору базы данных быть независимым от администратора домена. Кроме того, у логинов SQL Server есть и другие преимущества, которые принимаются во внимание разработчиками:

на предприятии вполне может не оказаться домена Windows (если, например, сеть построена на основе NetWare или UNIX);

пользователи SQL Server могут не входить в домен (например, если они подключаются к SQL Server из филиалов или через Web-интерфейс с домашнего компьютера).

Таким образом, выбор используемых типов логинов зависит от многих факторов и в каждом конкретном случае решение принимается индивидуально. Логины Windows - это удобство и защищенность, логины SQL Server- это большая гибкость и независимость от администратора сети.

При установке SQL Server одним из решений, которые следует принять, является выбор используемого режима аутентификации.

В режиме аутентификации Windows SQL Server полностью доверяет (делегирует) аутентификацию операционной системе.

В смешанном режиме аутентификация Windows и самого сервера сосуществуют независимо друг от друга.

Третьего варианта, в котором использование логинов Windows было бы запрещено, не предусмотрено: логины этого типа доступны всегда.

Установленный при инсталляции режим аутентификации можно изменить в утилите Management Studio выбрав нужный переключатель в группе «Серверная проверка подлинности» на странице «Безопасность» диалогового окна «Свойства сервера».

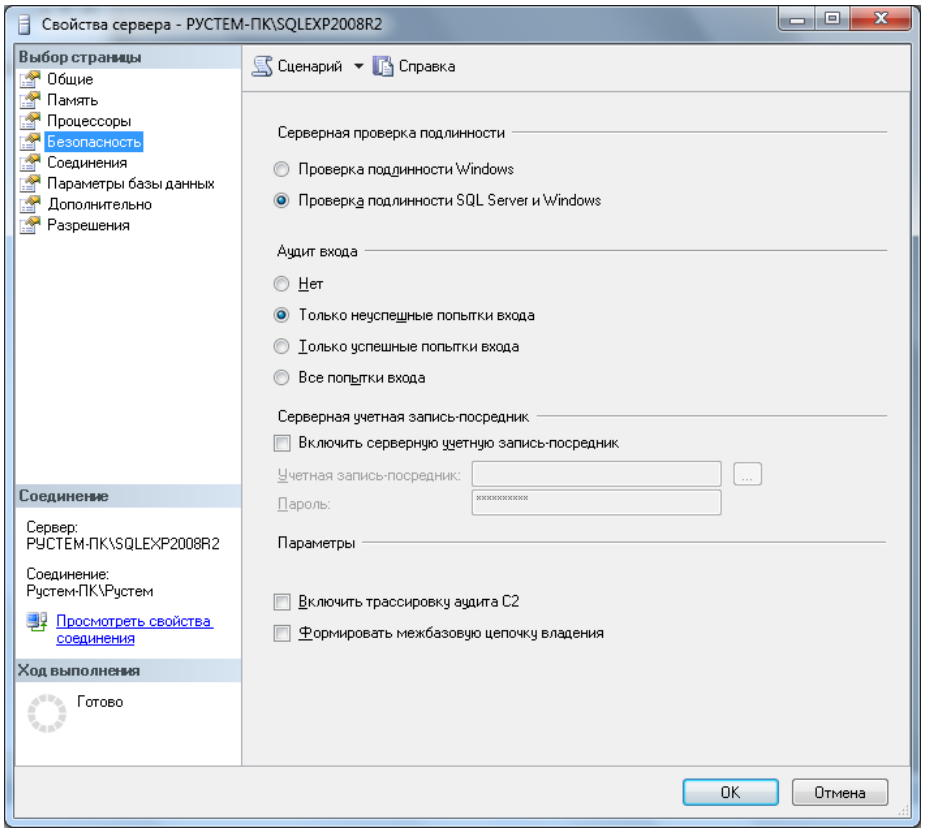


Рис. 11.2

Установите переключатель в положение «Проверка подлинности SQL Server и Windows». Таким образом, вы включите смешанный режим аутентификации, в котором и будут выполняться остальные упражнения. Для того чтобы изменение режима аутентификации вступило в силу сервер нужно перезапустить.

Создание логина и настройка его параметров

Логины любого типа создаются одинаково:

при помощи графического интерфейса - из окна «Создание имени входа». Это окно открывается с помощью команды «Создать имя входа…» контекстного меню узла «Безопасность | Имена входа» дерева обозревателя объектов в SQL Server Management Studio;

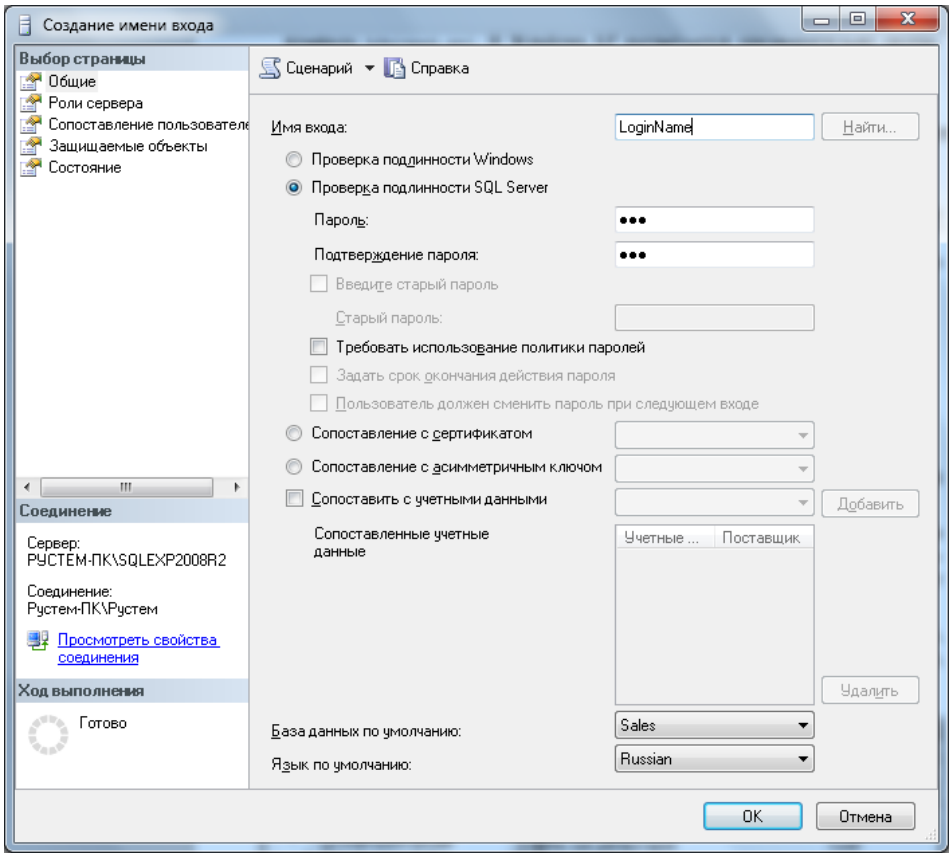


Рис. 11.3

из скрипта - при помощи команды CREATE LOGIN.

Например, команда на создание логина SQL Server с именем User1 и паролем P@sswOrd (для всех остальных параметров будут приняты значения по умолчанию) может выглядеть так:

CREATE LOGIN User1 WITH PASSWORD = 'P@sswOrd';

Если вы создаете логин Windows, вам потребуется выбрать соответствующую учетную запись или группу Windows.

Если вы создаете логин SQL Server, вам придется ввести его имя и пароль. Пароль всегда чувствителен к регистру, а логин - только тогда, когда чувствительность к регистру была определена при установке SQL Server. Конечно, кроме имени и пароля для логинов можно определить множество других параметров. Некоторые из них перечислены ниже.

База данных по умолчанию, к которой по умолчанию будет подключаться пользователь при входе на SQL Server. По умолчанию

используется база данных master. Как правило, менять этот параметр не следует: код для перехода к нужной базе данных при подключении обеспечивает клиентское приложение.

Язык по умолчанию - язык, который будет использоваться по умолчанию данным пользователем во время сеансов. В основном он влияет на формат даты и времени, которые возвращает SQL Server. В большинстве случаев для этого параметра оставляется значение по умолчанию (т. е. язык, настроенный на уровне всего сервера), если о другом значении специально не просит разработчик.

На вкладке «Состояние» свойств логина можно настроить для этого логина дополнительные параметры:

Разрешение на подключение к ядру СУБД - по умолчанию для всех логинов устанавливается значение «Предоставить», т. е. подключаться к SQL Server разрешено. Значение «Запретить», как правило, используется только в одном случае - когда вы предоставляете доступ на SQL Server логину для группы Windows, а одному или нескольким членам этой группы доступ нужно запретить. Поскольку явный запрет всегда имеет приоритет перед разрешением, то достаточно будет создать свои собственные логины Windows для этих пользователей и установить для них значение «Запретить».

Имя входа (Включено/Отключено) - конечно, все логины по умолчанию включены. Обычно отключать их приходится только в ситуации, когда какой-то пользователь увольняется или переходит на другую работу. Чтобы сэкономить время, достаточно просто отключить данный логин, а при появлении пользователя со схожими рабочими обязанностями переименовать этот логин, поменять пароль и включить. Заниматься предоставлением разрешений заново в этом случае не придется.

Имя входа заблокировано - установить этот флажок вы не можете (только снять его). Учетная запись пользователя блокируется

автоматически после нескольких попыток неверного ввода пароля для логина SQL Server, если такая блокировка настроена на уровне операционной системы, а для логина установлен флажок «Требовать использование политики паролей».

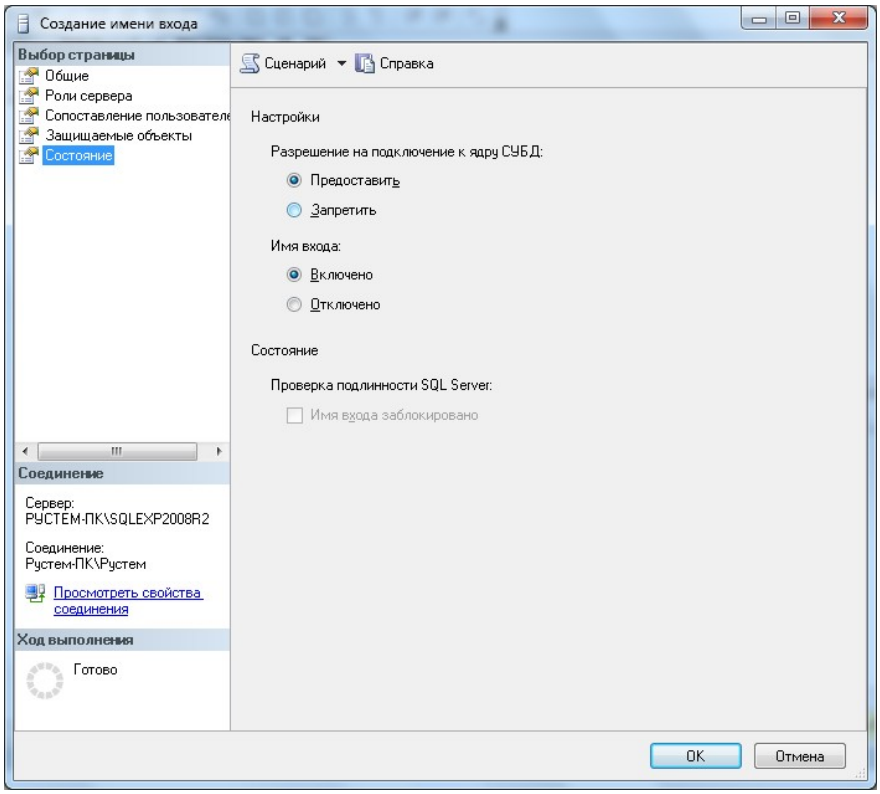


Рис. 11.4

Разрешения на уровне сервера. Фиксированные серверные роли

Создав логины, вы обеспечиваете пользователям возможность входа на SQL Server. Но сам по себе вход на сервер ничего не дает: пользователю нужны также права на выполнение определенных действий. Обычно для этой цели создаются пользователи или роли баз данных и им предоставляются разрешения (как это сделать, будет рассмотрено в следующем разделе). Однако есть и другой способ. Если вам нужно предоставить пользователю права на уровне всего сервера, а не отдельной базы данных, можно воспользоваться серверными ролями.

На графическом экране работа с ролями сервера производится или

из свойств логина (вкладка «Роли сервера»), или из свойств самой серверной роли (узел «Роли сервера» дерева обозревателя объектов Management Studio).

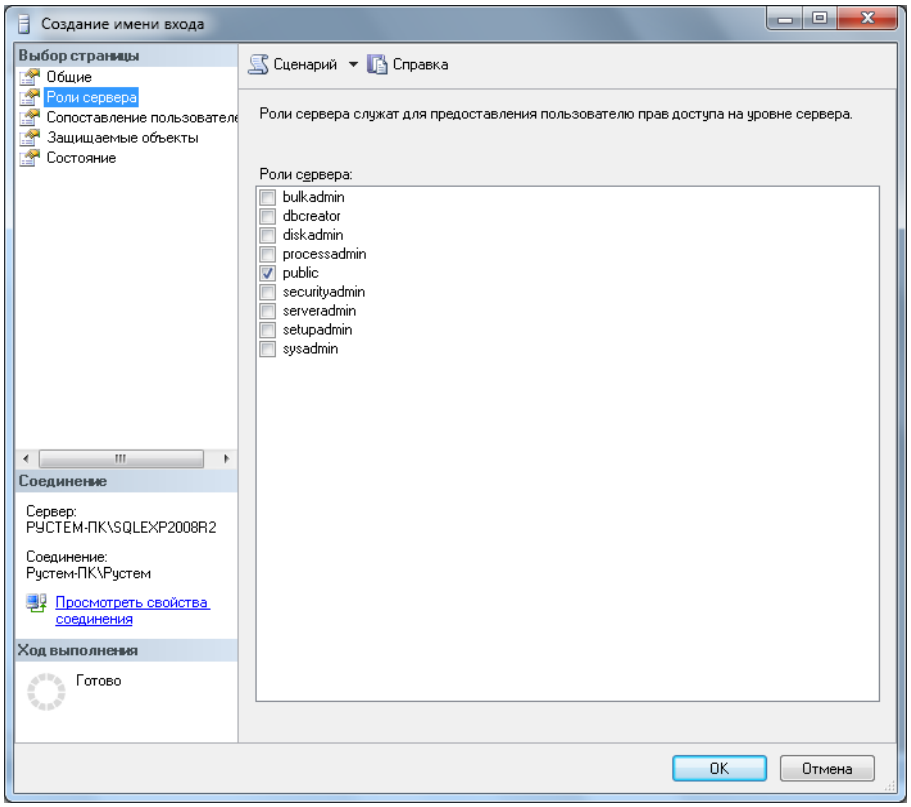


Рис. 11.5

Отметим несколько моментов, связанных с серверными ролями:

набор серверных ролей является фиксированным. Вы не можете создавать свои серверные роли (в отличие от ролей базы данных);

для предоставления прав на уровне всего сервера необязательно использовать серверные роли. Вы вполне можете предоставить эти права напрямую логину (при помощи вкладки «Разрешения» окна свойств сервера). По умолчанию каждый логин обладает на уровне всего сервера двумя правами: CONNECT SQL (т. е. подключаться к серверу, это право предоставляется логину напрямую) и VIEW ANY DATABASE (т. е. просматривать список баз данных, это право пользователь получает через серверную роль public);

серверные роли используются только в специальных случаях. Для большинства пользователей настраивать их не нужно.

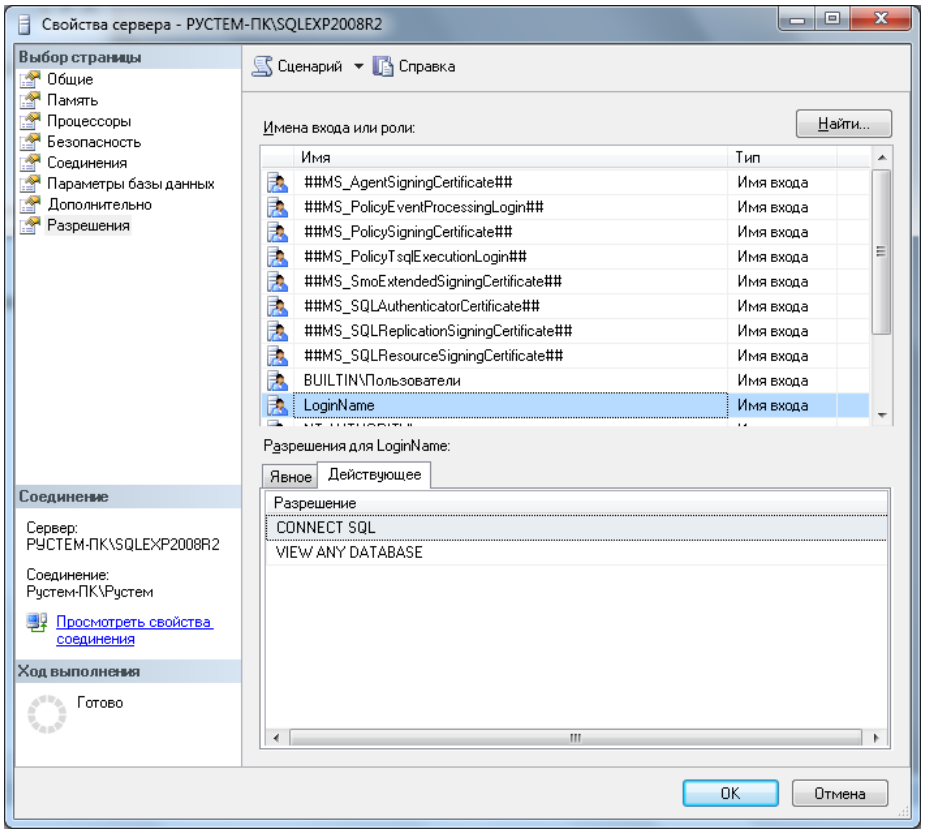


Рис. 11.6

Серверных ролей не так много, поэтому приведем их полный список с комментариями:

public - права этой роли автоматически получают все, кто подключился к SQL Server, и лишить пользователя членства в этой роли нельзя. Обычно эта роль используется для предоставления разрешений всем пользователям данного сервера;

sysadmin - логин, которому назначена эта роль, получает полные права на весь SQL Server (и возможность передавать эти права другим пользователям);

serveradmin - эта роль для оператора, который обслуживает сервер. Можно изменять любые настройки работы сервера и отключать сервер, но получать доступ к данным и изменять разрешения нельзя;

securityadmin - эта роль для того, кто выдает разрешения пользователям, не вмешиваясь в работу сервера. Он может создавать логины для обычных пользователей, изменять их пароли, предоставлять

разрешения в базах данных. Однако предоставить кому-либо административные права или изменять учетную запись администратора эта роль не позволяет;

bulkadmin - роль для сотрудников, которые выполняют массовые загрузки данных в таблицы SQL Server;

dbcreator - эта роль позволяет создавать базы данных на SQL. Server (а тот, кто создал базу данных, автоматически становится ее владельцем);

diskadmin - эта роль позволяет выполнять любые операции с файлами баз данных на диске;

processadmin - эта роль предназначена для выполнения единственной обязанности: закрытия пользовательских подключений к серверу (например, зависших);

setupadmin - права этой роли позволяют подключать внешние серверы SQL Server.

Пользователи баз данных. Схемы

После создания логинов следующая задача - спуститься на уровень базы данных и создать пользователей базы данных. Пользователи баз данных - это специальные объекты, которые создаются на уровне базы данных и используются для предоставления разрешений в базе данных (на таблицы, представления, хранимые процедуры и т.д.).

Логины и пользователи баз данных - это совершенно разные объекты. Разделение логинов и пользователей баз данных обеспечивает большую гибкость. Например, пользователь, который входит от имени одного и того же логина, сможет работать в разных базах данных от имени разных пользователей.

Создать пользователя базы данных можно:

В среде Management Studio вызвав команду «Создать пользователя…» в контекстном меню подузла «Безопасность | Пользователи» узла конкретной базы данных дерева обозревателя объектов. В открывшемся окне «Пользователь базы данных - Создать»,

снимок экрана с которым приведен ниже, необходимо указать два обязательных параметра: имя нового пользователя и выбрать соответствующий ему логин (Windows или SQL Server).

При помощи команды CREATE USER.

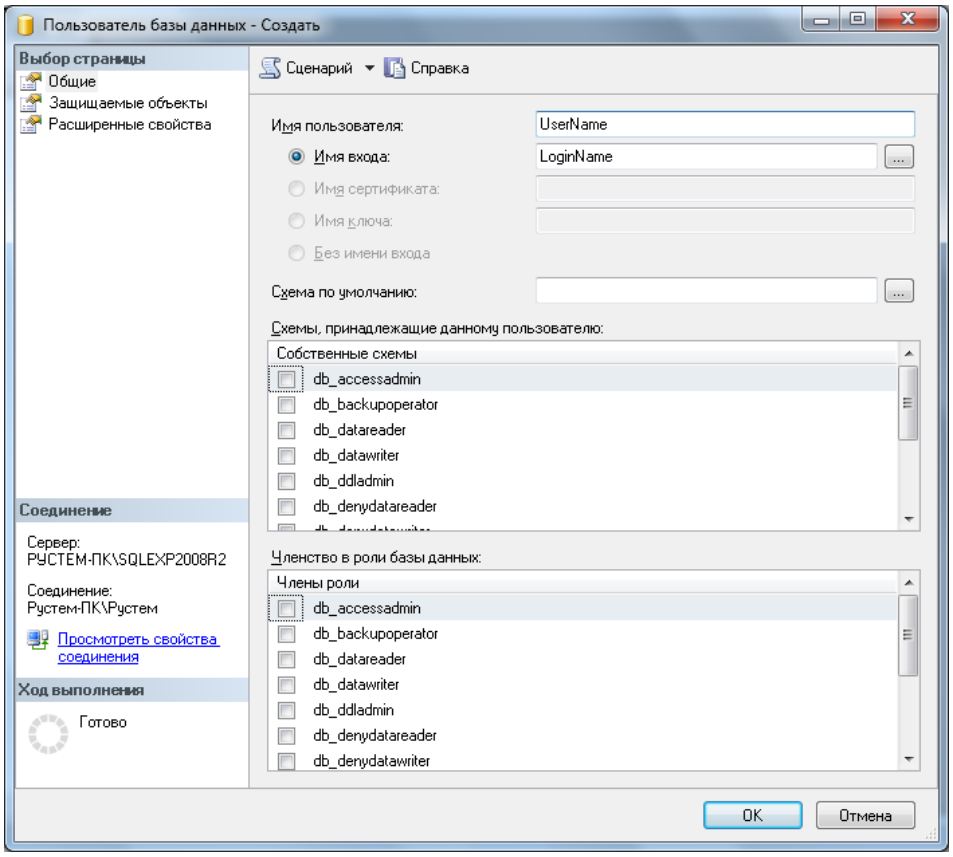


Рис. 11.7

Изменение свойств пользователя и его удаление производится из того же контейнера в Management Studio, что и создание пользователя, а также при помощи команд ALTER USER/DROP USER.

В SQL Server 2000 и в более старых версиях объект пользователя базы данных нес на себе двойную нагрузку: во-первых, он использовался для предоставления разрешений в базе данных, а во-вторых, имя пользователя базы данных использовалось для идентификации объектов. Например, обращение к таблице по полному ее имени могло выглядеть так: SELECT * FROM MyServer.MyDatabase.User1.Table1;

Если разработчик использовал в коде приложения просто имя объекта (например, SELECT * FROM Table1), то при подключении к серверу из этого приложения от имени другого пользователя могли возникнуть проблемы, поскольку вместо User1 подставлялось текущее имя пользователя (а если объект с таким полным именем не был обнаружен, то имя специального пользователя dbo).

Начиная с версии SQL Server 2005 эти две функции разделены. Для предоставления разрешений в базе данных, как и раньше, используется объект пользователя, а для именования объектов в базе данных используется специальный объект схема. Запрос с использованием полного формата имени в SQL Server теперь должен выглядеть так:

SELECT * FROM MyServer.MyDatabase.Schema1.Table1;

Схема формально определяется как набор объектов в базе данных, объединенных общим пространством имен. Проще всего представить себе схему как некий логический контейнер в базе данных, которому могут принадлежать таблицы, представления, хранимые процедуры, пользовательские функции, ограничения целостности, пользовательские типы данных и другие объекты базы данных. Этот контейнер удобно использовать как для именования объектов и их логической группировки, так и для предоставления разрешений. Например, если в базе данных есть набор таблиц со связанными данными, удобно поместить их в одну схему и предоставлять пользователям разрешения на эту схему (т. е. на этот набор таблиц).

Пользователю можно назначить схему по умолчанию. В эту схему SQL Server будет по умолчанию помещать объекты, которые создает этот пользователь. Кроме того, искать объекты, к которым обращается пользователь (например, в случае запроса вида SELECT * FROM Table1), SQL Server также будет в первую очередь в его схеме по умолчанию.

Применение схемы дает ряд дополнительных преимуществ по сравнению со старым подходом:

нескольким пользователям можно назначить одну и ту же схему по умолчанию, что может быть удобно при разработке приложений;

несколько пользователей (через группы Windows или роли баз данных) могут владеть одной и той же схемой. При этом один пользователь может являться владельцем сразу нескольких схем;

при удалении пользователя из базы данных не придется переименовывать его объекты;

упрощается предоставление разрешений для наборов объектов в базе данных.

Список схем можно увидеть в подузле «Безопасность | Схемы» узла конкретной базы данных дерева обозревателя объектов Management Studio.

При создании любой базы данных в ней автоматически создаются четыре специальных пользователя:

dbo (от database owner) - пользователь-владелец базы данных. Он автоматически создается для того логина, от имени которого была создана эта база данных. Конечно же, как владелец, он получает полные права на свою базу данных (при помощи встроенной роли базы данных db_owner);

guest (гость) - этот специальный пользователь предназначен для предоставления разрешений всем логинам, которым не соответствует ни один пользователь в базе данных. По умолчанию у этого пользователя нет права login для базы данных, и, следовательно, работать он не будет. Этот пользователь используется чаще всего для предоставления разрешений логинам на какие-то учебные/тестовые базы данных или на базы данных-справочники, доступные только на чтение;

INFORMATION_SCHEMA - этому пользователю не может соответствовать ни один логин. Его единственное значение - быть владельцем схемы INFORMATION_SCHEMA, в которой хранятся представления системной информации для базы данных;

sys - этому специальному пользователю, как и

INFORMATION_SCHEMA, не могут соответствовать логины. Он является владельцем схемы sys, которой принадлежат системные объекты базы данных.

Роли базы данных

Обычно после создания логина и пользователя базы данных следующее, что нужно сделать, - предоставить пользователю разрешения в базе данных. Один из способов сделать это - воспользоваться ролями базы данных.

Роли базы данных - это специальные объекты, которые используются для упрощения предоставления разрешений в базах данных. В отличие от серверных ролей, которые могут быть только встроенными, роли баз данных могут быть как встроенными, так и пользовательскими. Встроенные роли баз данных обладают предопределенным набором разрешений, а пользовательские роли можно использовать для группировки пользователей при предоставлении разрешений. Обычно пользовательские роли используются только для логинов SQL Server, поскольку для группировки логинов Windows обычно удобнее и проще использовать группы Windows.

Вначале перечислим встроенные роли баз данных:

public - эта специальная роль предназначена для предоставления разрешений сразу всем пользователям базы данных. Специально сделать пользователя членом этой роли или лишить его членства невозможно. Все пользователи базы данных получают права этой роли автоматически.

db_owner - этой роли автоматически предоставляются полные права на базу данных. Изначально права этой роли предоставляются только специальному пользователю dbo, а через него - логину, который создал эту базу данных;

db_accessadmin - роль для сотрудника, ответственного за пользователей базы данных. Этот сотрудник получит возможность создавать, изменять и удалять объекты пользователей баз данных, а также

создавать схемы. Других прав в базе данных у него нет;

db_securityadmin- эта роль дополняет роль db_accessadmin. Сотрудник с правами этой роли получает возможность назначать разрешения на объекты базы данных и изменять членство во встроенных и пользовательских ролях. Прав на создание и изменение объектов пользователей у этой роли нет;

db_backupoperator - эта роль дает право выполнять резервное копирование базы данных;

db_ddladmin - эта роль применяется в редких ситуациях, когда пользователю необходимо дать право создавать, изменять и удалять любые объекты в базе данных, не предоставляя прав на информацию, которая содержится в существующих объектах;

db_datareader и db_datawriter - эти встроенные роли дают право просматривать и изменять соответственно (в том числе добавлять и удалять) любую информацию в базе данных. Очень часто пользователю необходимо дать права на чтение и запись информации во всех таблицах базы данных, не предоставляя ему лишних административных разрешений (на создание и удаление объектов, изменение прав и т. п.). Самый простой вариант в этой ситуации - воспользоваться этими двумя ролями.

db_denydatareader и db_denydatawriter- эти роли противоположны ролям db_datareader и db_datawriter. Роль db_ denydatareader явно запрещает просматривать какие-либо данные, а db_denydatawriter запрещает внесение изменений. Явный запрет всегда имеет приоритет перед явно предоставленными разрешениями. Обычно эти роли используются в ситуации, когда "разрешаем всем, а потом некоторым запрещаем".

Как уже говорилось ранее, в отличие от серверных ролей, роли баз данных вы можете создавать самостоятельно. Это можно сделать из контекстного меню узла «Безопасность | Роли | Роли базы данных» обозревателя объектов в Management Studio или при помощи команды

CREATE ROLE.

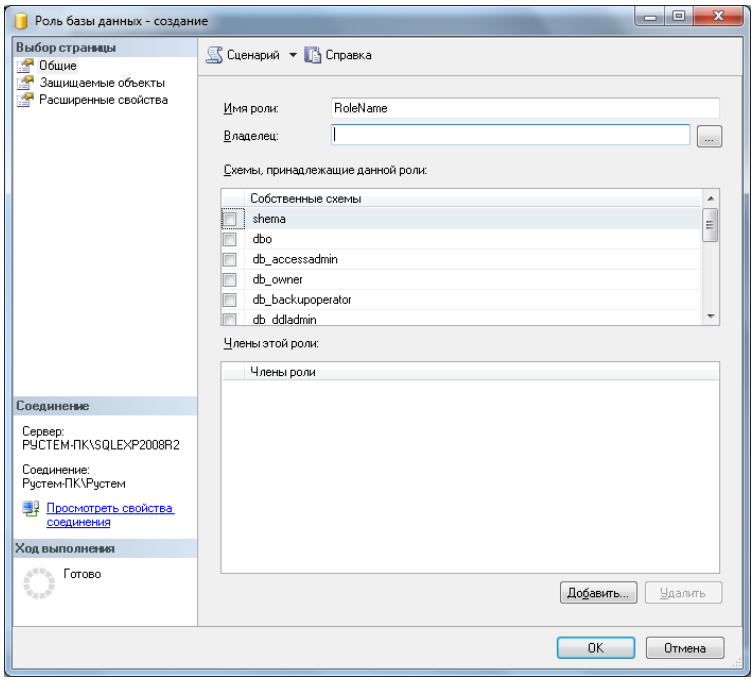


Рис. 11.8

Кроме имени роли, при создании можно также указать ее владельца (если это не указано, владельцем роли автоматически станет создавший ее пользователь базы данных). Если вы создаете роль при помощи графического интерфейса, вы можете сразу назначить роль владельцем схемы в базе данных и назначить пользователей, которые будут обладать правами этой роли.

Встроенным ролям назначены предопределенные права, изменить которые невозможно. Предоставление прав пользовательским ролям производится точно так же, как и обычным пользователям базы данных.

Предоставление прав на объекты в базе данных

Работа с разрешениями производится одинаково для всех объектов базы данных:

на вкладке «Разрешения» свойств этого объекта (эта вкладка предусмотрена не для всех объектов, для которых можно предоставить разрешения);

на странице «Защищаемые объекты» окна свойств пользователя или роли;

при помощи команд GRANT (предоставить разрешение), DENY (явно запретить что-то делать) и REVOKE (отменить явно предоставленное разрешение или запрет).

Начиная с версии 2005, в SQL Serve появилась возможность предоставлять разрешения на уровне схемы. К схеме в SQL Server могут относиться таблицы, представления, хранимые процедуры, пользовательские функции, ограничения целостности, пользовательские типы данных и другие объекты, на которые приходится предоставлять разрешения чаще всего. Если вы назначите пользователю разрешения на схему, то он получит разрешения на все объекты этой схемы.

Далее перечислены разрешения, которые можно предоставить на уровне схемы. Мы приведем только разрешения для этого объекта, поскольку разрешения схемы включают в себя разрешения, которые можно предоставить другим объектам, например, разрешения SELECT для таблиц и представлений и EXECUTE для хранимых процедур.

ALTER - возможность вносить любые изменения в свойства объекта (за исключением смены владельца).

CONTROL - тот, кому предоставлено такое разрешение, получает полные права как на сам объект, так и на информацию в нем.

DELETE - возможность удалять существующую информацию в таблицах. Применяется к таблицам, представлениям и столбцам.

EXECUTE - право запускать на выполнение. Применяется к хранимым процедурам и функциям.

INSERT - право на вставку новых данных в таблицы. Применяется к таблицам, представлениям и столбцам.

REFERENCES - разрешение, которое можно предоставить для проверки ограничений целостности. Например, пользователь может добавлять данные в таблицу с внешним ключом, а на таблицу с первичным ключом ему нельзя предоставлять права на просмотр. В этом случае на таблицу с первичным ключом ему можно предоставить право

REFERENCES - и он сможет производить вставку данных в таблицу с внешним ключом, не получая лишних прав на главную таблицу. Это разрешение можно предоставлять на таблицы, представления, столбцы и функции.

SELECT - право на чтение информации. Предоставляется для таблиц, представлений, столбцов и табличных функций.

TAKE OWNERSHIP - право на принятие на себя владения данным объектом. Владелец автоматически обладает полными правами на свой объект. Такое право можно назначить для любых объектов.

UPDATE - возможность вносить изменения в существующие записи в таблице. Предоставляется на таблицы, представления и столбцы.

VIEW DEFINITION - право на просмотр определения для данного объекта. Предусмотрено для таблиц, представлений, процедур и функций.

Для каждого разрешения мы можем установить три значения:

GRANT - разрешение предоставлено явно;

WITH GRANT - разрешение не только предоставлено данному пользователю, но он также получил право предоставлять это разрешение другим пользователям. Можно предоставлять, конечно, только вместе с GRANT;

DENY - явный запрет на выполнение действия, определенного данным разрешением. Как в любых системах безопасности, явный запрет имеет приоритет перед явно предоставленными разрешениями.

Из кода Transact-SQL разрешения можно предоставлять командами GRANT, DENY и REVOKE. Например, чтобы предоставить пользователю User1 возможность просматривать данные в таблице Table1 в схеме dbo, можно воспользоваться командой: GRANT SELECT ON dbo.Table1 TO User1;

Лишить его ранее предоставленного права можно при помощи команды: REVOKE SELECT ON dbo.Table1 TO User1;

Некоторые рекомендации, связанные с предоставлением

разрешений:

В большинстве реальных задач используются десятки и даже сотни таблиц и других объектов базы данных. Предоставлять каждому пользователю разрешения на каждый из этих объектов очень неудобно. Если есть возможность, удобнее использовать разрешения на уровне схемы или всей базы данных.

Существует общий принцип: не стоит обращаться из клиентского приложения к таблицам базы данных напрямую. Для изменения данных лучше использовать хранимые процедуры, а для запросов на чтение - хранимые процедуры или представления. Причина проста: если потребуется поменять структуру вашей базы данных (например, какую-то таблицу поделить на текущую и архивную или добавить новый столбец) не потребуется вносить изменения в клиентское приложение. Это следует помнить и при предоставлении разрешений. Отметим также, что при помощи хранимых процедур можно очень просто реализовать дополнительные проверки в добавление к обычным разрешениям;

SQL Server позволяет настраивать разрешения на уровне отдельных столбцов. На практике лучше не пользоваться такими разрешениями из-за падения производительности и усложнения системы разрешений. Если пользователю можно видеть не все столбцы в таблице (например, ему не нужны домашние телефоны сотрудников), то правильнее будет создать представление или хранимую процедуру, которые будут отфильтровывать ненужные столбцы.

Задание для самостоятельной работы:

Создайте логин SQL Server «Admin» и назначьте ей роль sysadmin;

Создайте в базе данных роль «Saler» и назначьте ей разрешения на выборку данных из всех таблиц, изменение данных в таблицах Order, OrdItem и запуск хранимой процедуры spr_getOrders;

Создайте логин SQL Server «Ivanov» и сопоставьте его с одноименным пользователем в базе данных Sales. Назначьте созданному

пользователю роль Saler.

Практическая работа №16
«Создание резервных копий базы данных»

Цель работы: ознакомиться с основными конструкциями SQL, технологиями среды MS SQL Server Management, объектами SMO (среды MS Visual Studio) для резервного копирования и восстановления БД.

Форма отчета:
−выполнить задание;
−показать преподавателю;
−ответить на вопросы преподавателя.
Время выполнения: 2 ч

Задание №1. необходимо создать резервные копии базы данных «МММ» с использованием полного резервного копирования, разностного резервного копирования и резервного копирования журнала транзакций.

Ход работы:

1. Запустите SQL Server Management Studio (SSMS), подключитесь к своему экземпляру SQL Server, используя технологию 1.
2. Создайте папку с именем c:\Student\ВашаПапка\test.
3. Откройте окно нового запроса. Измените контекст на базу данных master, используя технологию 6. Наберите и исполните следующую команду, чтобы создать полную резервную копию базы данных:

BACKUP DATABASE MMM TO DISK = 'C:\.....TEST\AW.BAK’

Ознакомьтесь с результатами запроса – какая информация обработана, сколько страниц, сколько файлов.

4. Внесите изменение в таблицу «Модель» базы данных МММ. Добавьте одну запись (придумайте сами)/
5. Откройте окно нового запроса наберите и исполните следующую команду, чтобы создать резервную копию журнала транзакций и сохранить только что внесенное изменение:

BACKUP LOG MMM TO DISK = ‘C:\.....TEST\AW1.TRN'

Ознакомьтесь с результатами запроса – какая информация обработана, сколько страниц, сколько файлов.

6. Внесите еще одно изменение в таблицу«Модель».
7. Откройте окно нового запроса наберите и исполните следующую команду, чтобы создать разностную резервную копию базы данных:

BACKUP DATABASE MMM TO DISK = ‘C:\.....\TEST\AWDIFF1.BAK' WITH DIFFERENTIAL

Ознакомьтесь с результатами запроса – какая информация обработана, сколько страниц, сколько файлов.

8. Внесите еще одно изменение в таблицу«Модель».

9. Откройте окно нового запроса наберите и исполните следующую команду, чтобы создать полную резервную копию базы данных в указанном месте на диске:

BACKUP LOG MMM TO DISK = ‘C:\....TEST\AW2.TRN'

Ознакомьтесь с результатами запроса – какая информация обработана, сколько страниц, сколько файлов.

Задание №2. необходимо организовывать со стороны клиентского приложения, созданного в Visual Studio удаленное администрирование БД (резервное копирование).

Ход работы:

В Visual Studio

1. Создайте новый проект Windows Application и сохраните его в своей папке под именем Лабы_MMM_2 семестр.
2. В главную форму добавьте меню, изображенное на рисунке 9:

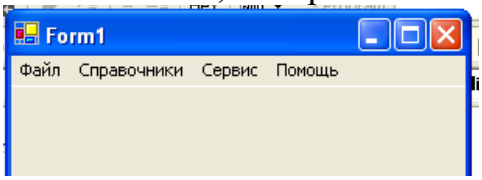


Рисунок 9 – Главное меню проекта

Файл (Открыть, Закрыть, Выход)
Справочники (Модель, Магазин, Дерево моделей)
Заказы (Работа с заказами)
Отчеты (Прайс-лист, Бланк заказов)
Администрирование БД (Резервное копирование, Безопасность)
Сервис (Калькулятор)
Помощь (Справка, О программе)

3. Добавьте новую форму в проект
4. Добавьте на только что созданную форму компоненты в соответствии с рисунком 10.

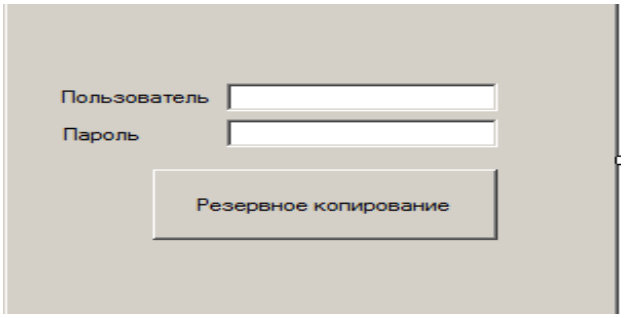


Рисунок 10 – Форма для подключения к серверу

5. Обеспечьте функциональную работу формы (напишите обработчик кнопки «Резервное копирование» с использованием объектов SMO. Описание объектов SMO, их свойств и методов см. в лекционном материале.)
6. Добавьте возможность открытия данной формы при выборе в главной форме пункта меню Администрироване БД → Резервное копирование

7. Запустите проект, проверьте работу формы.
8. Закройте проект
9. Убедитесь в появлении файла резервной копии на диске (файл, который указан в тексте программы).
10. Откройте SSMS. Добавьте в таблицу «Модель» новую строку данных (самостоятельно).
11. Средствами оболочки SSMS, выполните восстановление БД из резервной копии, созданной вашей программой
12. Убедитесь, что после восстановления добавленных строк в таблице «Модель» нет.

Задание №3. Ответьте на вопросы теста и представьте результаты преподавателю.

1. Вы выполняете разностное резервное копирование базы длимых AdveniurtWorks каждые четыре часа, начиная с 04:00. полная резервная копия создается в полночь. Какие данные будут содержаться в разностной резервной копии сделанной в полдень?
    a. A Страницы данных, измененные после полуночи.
    b. B. Экстенты, измененные после полуночи.
    c. C. Страницы данных, измененные после 08:00
    d. D. Экстенты. измененные после 08:00.
2. Вы выполняете полное резервное копирование базы данных Adventure Wortks,, которое завершается в полночь. Разностное резервное копирование выполняется по расписанию каждые четыре часа, начиная с 04:00. Резервное копирование журнала транзакций происходит по расписанию каждые пять минут. Какую информацию будет содержать резервная копия журнала транзакций, созданная в 09:15?
    a. A. Все транзакции, начатые после 09:10.
    b. B. Транзакции, завершенные после 09:10.
    c. C. Страницы, измененные после 09:10.
    d. D. Экстенты, измененные после 09:10.

Практическая работа №17
«Восстановление базы данных»

Цель: Научиться восстанавливать базу данных.

Форма отчета:
−выполнить задание;
−показать преподавателю;
−ответить на вопросы преподавателя.
Время выполнения: 2 ч

Задание №1. необходимо провести восстановление базы данных «МММ» из сделанных в задании №1 предыдущей лабораторной работы резервных копий.

Ход работы:

1. Если необходимо, запустите SSMS, подключитесь к своему экземпляру SQL Server, используя технологию 1.
2. Выполните восстановление БД из первой полной резервной копии (C:\...TEST\AW.BAK) средствами оболочки SSMS. Для этого выполните:
   - В обозревателе объектов вызовите контекстное меню на вашей БД и выберите задачу восстановления базы данных (см. рисунок 6).

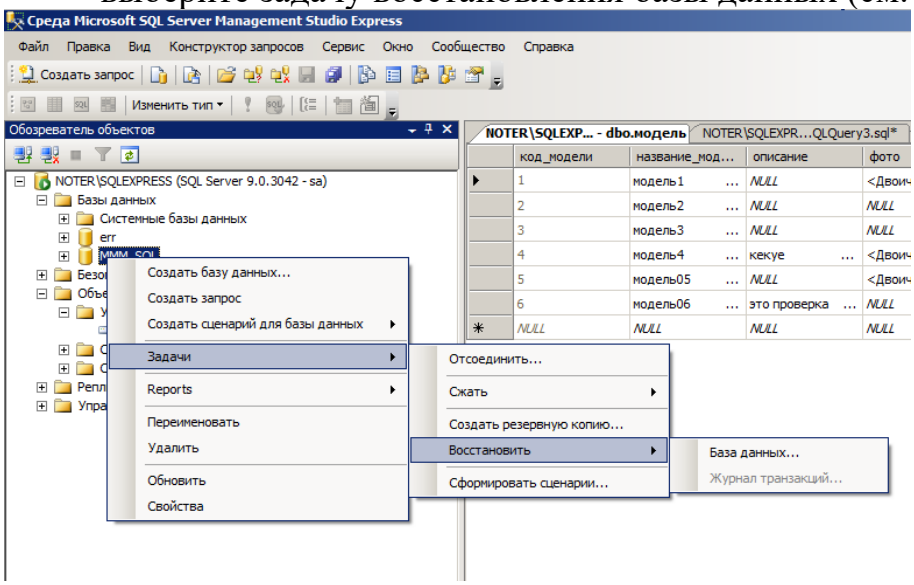


Рисунок 6 – Восстановление БД

   - В открывшемся окне необходимо задать следующие параметры восстановления

На закладке «Общие» необходимо выбрать:

a. Базу данных для восстановления (вашу МММ)
b. Выбрать источник набора данных для восстановления с устройства → файл C:\...TEST\AW.BAK
c. После определения файла-источника данных необходимо флажком выбрать базу данных для восстановления (рисунок 7).

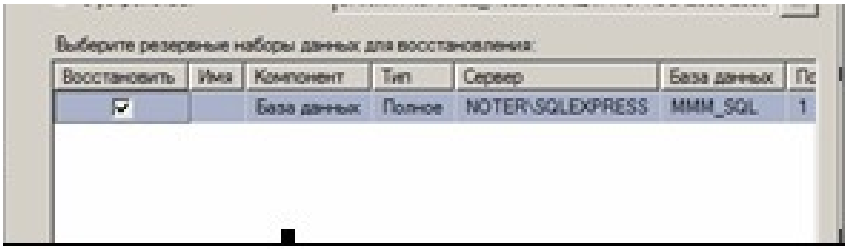


Рисунок 7- Выбор БД для восстановления

На закладке «Параметры»

a. необходимо включить опцию «Перезаписать БД» и «оставить БД готовой к использованию», (рисунок 8).

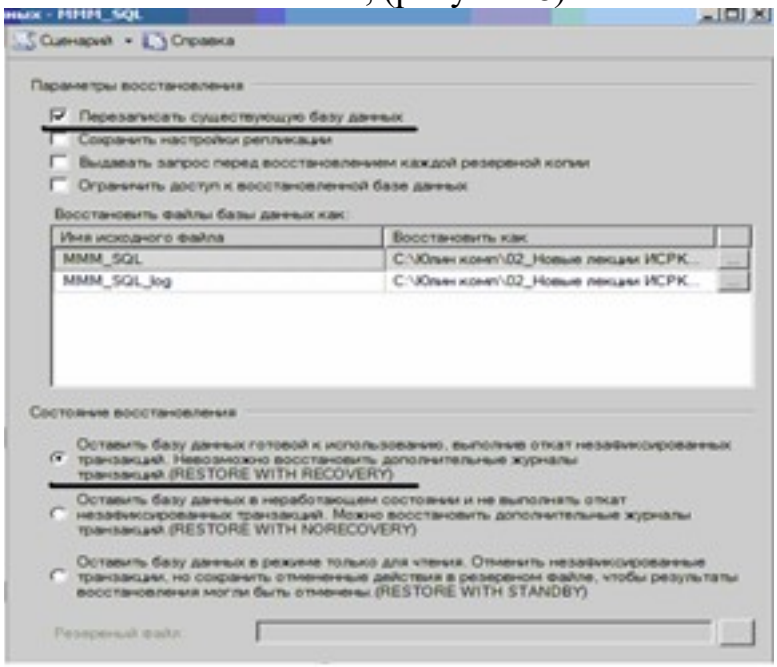


Рисунок 8 – Задание параметров восстановления

3. Нажмите ОК
4. После восстановления БД, откройте таблицу «Модель» и убедитесь, что она не содержит всех добавлений, вносимых вами в процессе выполнения упражнения, так как восстановление происходило из первой резервной копии (без изменений).

Практическая работа №18
«Восстановление носителей информации»

Цель работы*:* научиться осуществлять восстановление жесткого диска после сбоев.\

Форма отчета:

−выполнить задание;

−показать преподавателю;

−ответить на вопросы преподавателя.

Время выполнения: 2 ч

Основные положения

На сегодняшний день жёсткие диски занимают доминирующее место на рынке накопителей информации. К плюсам жёстких дисков можно отнести низкую стоимость за Гбайт памяти и практичность в использовании. Поэтому возникает необходимость в своевременном **обслуживании, тестировании** и выявлении критического состояния жесткого диска.

В состав утилит современной операционной системы, в том числе Windows 7 входят программы, позволяющие осуществлять дефрагментацию и очистку жесткого диска. Для этого необходимо выполнить команду **Пуск/Стандартные/Служебные** и из появившегося списка программ выбрать нужную.

Кроме того, современные накопители имеют систему оперативного наблюдения за своим состоянием - S.M.A.R.T. (Self-Monitoring, Analysis And Reporting Technology) – технология самодиагностики, анализа и отчета. Это набор программ, вшитых в ПЗУ диска. Эта технология позволяет в любое время оценить такие важные параметры накопителя, как количество отработанных часов, число возникших в процессе чтения/записи ошибок, температуру накопителя среднюю производительность, количество циклов запуска/останова шпинделя, время раскрутки шпинделя, количество переназначенных секторов, количество ошибок позиционирования головок и многое другое. Технология позволяет предсказать возможный выход из строя накопителя.

Исходя из огромной важности корректной работы жесткого диска, существует большое количество программ, позволяющих восстанавливать удаленные файлы с диска, файловую систему, критически важные структуры жесткого диска, такие как главная загрузочная запись, таблица разделов и т.д.

1. Partition Magic

Power Quest @ Partition Magic - это утилита, которая позволяет быстро и легко создавать, удалять, объединять или преобразовывать файловые системы и разделы на жестком диске, не уничтожая существующие данные. Новый инструмент кластерного анализа исследует FAT-дисководы и рекомендует подходящий размер кластера. Кроме того, есть возможность создавать, перемещать и изменять размер разделов типа FAT,
FAT 32, файловой системы Windows NT (Windows NT File System, NTFS), HPFS (High-Performance File System - высокопроизводительная файловая система).

Partition Magic помогает надежно устанавливать и использовать несколько операционных систем на одном жестком диске. Partition Magic включает в себя Boot Magic - мощный администратор загрузки, который помогает безопасно устанавливать новые операционные системы и позволяет выбирать через меню систему при загрузке компьютера.

Программа имеет наглядный доброжелательный интерфейс.

В версии Partition Magic 8.0 включена новая утилита - Power Quest Data Keeper. Она поможет защитить ценные данные на диске от системных сбоев, упростить процесс копиров

ания и пересылки в пределах системы, восстановить удаленный файл.

В процессе установки программы можно сделать две загрузочные дискеты - на одной будет DOS от Caldera, а на другой - Partition Magic for DOS. С помощью этих дискет можно подготовить новый
диск к работе с нуля, т.к. программа наряду с организацией разделов выполняет и их форматирование, причем эти процедуры выполняются намного быстрее, чем при использовании традиционных программ.

Прежде чем начать работу с программой Partition Magic обязательно нужно выполнить следующие рекомендации:

• Установить самые последние обновления для операционных систем Windows 95/98/Me/NT Workstation/2000/XP Professional. Удостовериться, что самые последние исправления для операционных систем Windows 95/98/Me/NT Workstation/2000/XP Professional установлены и запущены.

• Сделайте копию вашего жесткого диска. Данные на диске - самая ценная часть компьютера. Хоть это и маловероятно, чтобы Partition Magic повредил бы данные, но влияние других ошибок типа системных отказов аппаратных средств, программного обеспечения, или питания, могут привести к повреждению данных в момент выполнения программы PartitionMagic. Используя программу Power Quest's Drive Image, можно создать резервную копию раздела, который будет изменяться. Можно также использовать эту программу и для полного восстановления раздела к первоначальному состоянию.

• Создать загрузочный диск Windows. Загрузочный диск позволит загрузить Windows при возникновении проблемы.

• Запустить опцию проверки ошибок на диске. Для раздела, который будет проверяться, нажать Partition > Check for Errors. Небольшие ошибки могут быть исправлены Partition Magic, однако более серьезные ошибки прекратят выполнение программы. Проверить и исправить обычные ошибки на диске. Проверка загрузочного раздела операционной системы Windows невозможно, так как есть всегда открытые файлы. Для этого раздела, можно воспользоваться Partition > MS ScanDisk.

• Закрыть все запущенные приложения. Нельзя запускать Partition Magic вместе с другими приложениями, включая вирусные сканеры. Если осуществляется работа в сети под управлением Windows NT, перед выполнением Partition Magic, необходимо удостовериться что другие пользователи, не подключены к вашему компьютеру.

• Использовать UPS (Источник бесперебойного питания). Partition Magic не способна восстановить данные, если в процессе разделения диска происходит сбой питания. Используя источник бесперебойного питания (UPS) можно избежать проблем, вызванных сбоем питания. • Совет. Из-за несовместимости аппаратной и системной конфигурации одного компьютера с другим, не рекомендуется переносить с одного на другой компьютер, жесткий диск, разделенный с помощью программы Partition Magic, во избежание потери данных. **Проверка целостности жесткого диска.**

Программа Partition Magic проверяет целостность диска сложной системой анализа и проверки достоверности, которая скрыто начинает свою работу, каждый раз, когда запускается программа или завершается операция. Первоначальная проверка на целостность диска, сообщает о любых проблемах связанных с разделами, которые могут препятствовать нормальной работе программы Partition Magic. Проверка целостности действует как ранняя система предупреждения, которая сообщит о том, что структура диска полностью проверена и проанализирована еще до изменения.

Если физический диск проходит первоначальную проверку целостности диска, то появляется таблица разделов, и вы можете начинать работу с программой. В случае появления сообщения об ошибке вместо таблицы разделов, указывается проблема с жестким диском, а не с программой Partition Magic (так как никакие изменения с диском еще не проводились). Необходимо исправить проблему с жестким диском и перезапустить Partition Magic . Для получения **дополнительной информации** можно воспользоваться кнопкой помощи на панели инструментов.

В дополнение проверки целостности при запуске программы, Partition Magic выполняет еще две проверки в течение любой операции. До операции разделения диска проверяется файловая система (наподобие CHKDSK или MS ScanDisk) , после проверяется целостность данных. Partition Magic анализирует диск и немедленно сообщает о найденных ошибках.

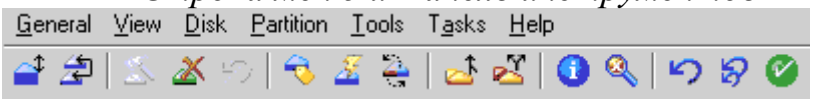
Интерфейс программы.

Интерфейс программы Partition Magic состоит из панели действия, строки меню, инструментальной панели, карты жестких дисков, списка разделов, кнопок мастера и строки легенда. Можете показать или скрыть, а также установить размеры для различных частей интерфейса. Выполнить настройку главного окна программы любым удобным способом для

различных частей интерфейса. Если выбранный жесткий диск содержит логические разделы, то они показываются внутри расширенного раздела.

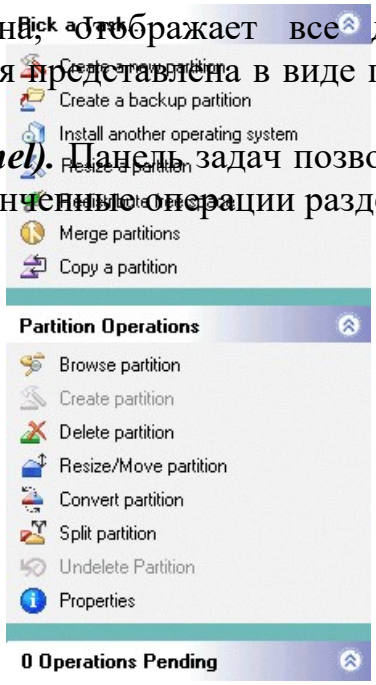
Строка меню и Панель инструментов (Menu Bar and Toolbar). В главном окне программы Partition Magic, строка меню и панель инструментов находятся наверху окна. Строка меню дает возможность доступа к любой из настроек Partition Magic, в то время как панель инструментов обеспечивает доступ к обычно используемым вариантам. Можно скрыть панель инструментов, что увеличит видимую область главного окна. Опция "Disks " на строке меню будет видна, только если у установлен второй жесткий диск.

Строка меню и Панель инструментов



Информация о разделах жесткого диска (Partition Information). Информационная область окна, отображает все данные для выбранного жесткого диска. Информация представлена в виде панели задач, карты диска и списка разделов.

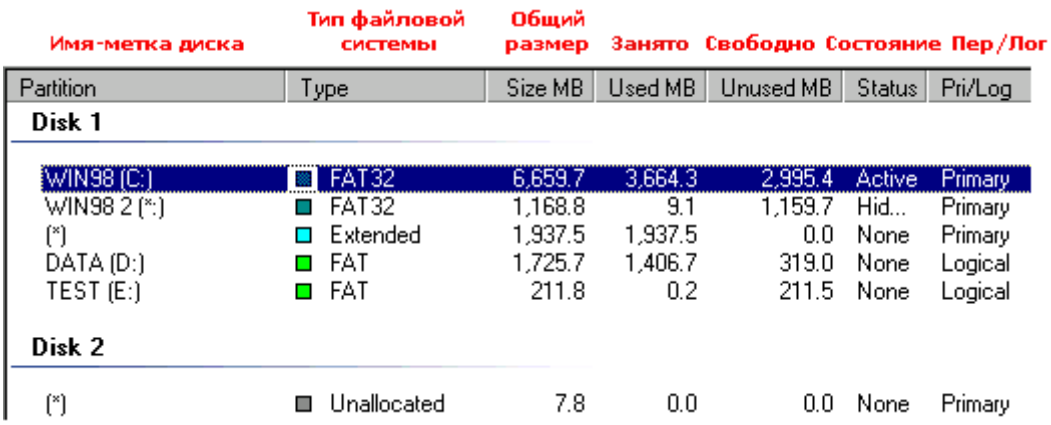
Панель задач (Action Panel). Панель задач позволяет выбрать задачу, а также увидеть текущие незаконченные операции разделения диска.



Карта диска (Disk Map). На карте очень наглядно изображены разделы диска, с возможностью масштабировать. (Для масштабирования нажимать **View > Scale Disk Map**). Каждый раздел на карте обозначается цветом (согласно легенде), которая приведена внизу окна. Освобожденное место на карте диска обозначается блоком темно – серого цвета.

Если у вас имеется второй жесткий диск то, возможно вы должны передвинуть карту что бы увидеть всю доступную информацию. Вы можете переместить карту дисков вверх или вниз, для более удобного просмотра.

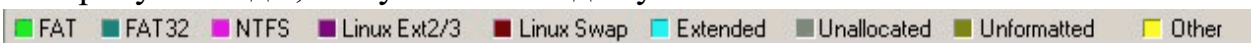
Список разделов (Partition List). Список разделов выводит информацию о каждом разделе на вашем жестком диске, конкретно это: имя диска, метки, тип файловой системы, размер в мегабайтах, количество используемого и неиспользуемого пространства в мегабайтах, состоянии, и является раздел первичным или логическим.



Разделы диска обозначается: названием тома, буквой с двоеточием. Звездочка (*) заменяет букву в том случае, если раздел является:

- Скрытым разделом.
- Расширенным разделом.
- Разделом с файловой системой, которая не поддерживается активной операционной системой.
- Высвобожденным пространством. Состояние раздела, может быть:
- ***Активным (Active)***: Раздел диска, с которого загружается компьютер.
- ***Скрытым (Hidden)***: К разделу, который не имеет букву диска, нельзя обратиться из текущей операционной системы. Разделы диска могут быть скрыты операционной системой (возможно, скрыть все первичные разделы кроме активного) или вы можете использовать Partition Magic чтобы самостоятельно скрыть нужный вам раздел. В среде Windows 2000/XP, скрытые разделы могут иметь имя.
- ***Никакой (None)***: Разделы, которые ни активны, ни скрыты.

Легенда (Legend). Легенда – это цветовые обозначения различных файловых систем, которые должны помочь пользователям понять цвета, которые используются в панели задач, карте диска, списка разделов. Можно скрыть строку легенды, что увеличит видимую область главного окна.



Четыре шага, для выполнения задачи.

Можно выполнить задачу двумя различными способами. Первый способ - использовать мастер программы Partition Magic из опускающего меню панели задач. Второй способ - это сделать вручную.

Чтобы выполнить задачу вручную надо:

1. Выбрать жесткий диск или раздел.
2. Выбрать задачу (operation).
3. Применить выбранные задачи к вашей системе

Выбор жесткого диска и раздела. Можете выделить раздел сразу, не выделяя первый жесткий диск. Для этого необходимо нажать на выбранном разделе на карте диска или выбрать его из списка в главном окне. Есть две задачи, которые всегда могут быть выполнены: удалить все разделы и вывести подробную информацию о жестком диске. Когда

выделяется жесткий диск, его разделы отображаются в списке разделов главного окна.

Выбор задачи (Selecting an operation). После того как были выбраны диск и раздел, используя строку меню или панель задач, выбрать операции. Есть несколько вариантов выполнения выбранной операции, для этого надо:

- В строке меню нажать **Partition**, затем нужную операцию. Справка советует этот метод как предпочтительный.
- На панели инструментов выбрать нужную операцию и нажмите <Enter>.
- На карте диска или в списке, выбрать раздел и щелкнуть на нем правой клавишей, затем выбрать нужную операцию.

Если операция недоступна, значить она не может быть применена к данному разделу. Partition Magic начинает выполнять немедленно операции по сбору информации, проверки на

ошибки, MS ScanDisk. Остальные операции помещаются в очередь в диалоговом окне ***Текущие действия (Operations Pending)*** и ожидают нажатия кнопки ***Применить (Apply).***

Отмена последней операции. Можно в любой момент отменить последнюю операцию, которые помещаются в очередь в диалоговом окне *Текущие действия* (Operations Pending). Есть несколько вариантов выполнения выбранного действия, для этого надо:

- Нажать **General** > **Undo Last Change**.
- На панели задач нажмите кнопку **Undo** (отмена) которая находиться внизу панели задач.
- Нажмите **Click View** > **Operations Pending** > **Undo Last**. • На панели инструментов нажмите кнопку **Undo** (отмена). • Нажмите клавиши (Ctrl+Z).

Отмена всех операций. Чтобы отменить все операции сразу, которые помещены в очередь в диалоговом окне *Текущие действия* (Operations Pending) надо:

- Нажать **General** > **Discard All Changes**.
- Нажать **View** > **Operations Pending** > **Discard All**. • Нажать клавиши <Ctrl+D>.

Применение изменений к системе. Сделанные изменения отображаются на карте диска, в списке разделов. Однако реальные изменения будут произведены только после нажатия кнопки Apply (применить). Если кнопка на панели задач активна, а текущие операции находятся в ожидании, значит, изменения еще не были произведены.

Для применения выбранных операций надо: • Нажать **General** > **Apply Changes**.

- Нажать кнопку **Apply** на панели задач главного окна программы PartitionMagic. • Нажать кнопку **Apply** на панели инструментов.
- Нажать клавиши <Ctrl+A>.

Индикатор движения процесса может не двигаться в течение нескольких минут. ***Изменение настроек (preferences) Partition Magic.***

1. Нажать General > Preferences.
2. Поставить галочку напротив надписи "Allow 64K FAT Clusters for Windows NT/2000/XP". Установка этой опции позволит создать файловую систему FAT с размером кластера равного 64К, а также позволит программе Partition Magic создавать FAT разделы размером до 4Гб. Но операционные системы Dos, Windows 3x/95/98/Me не поддерживают размеры кластеров больше 32К. Поэтому нельзя получить доступ к разделу с размером кластера 64К, используя эти операционные системы.
3. Установка галочки в маленьком окне с надписью "Skip bad sector checks" позволит пропустить проверку жесткого диска на сбойные секторы. Однако если жесткий диск имеет сбойные секторы,

то возможна потеря всех данных, поэтому не рекомендуется ее включать (по умолчанию отключена).

4. Установка галочки в маленьком окне с надписью " Set as Read-Only for Partition Magic" не позволит программе произвести какие-либо изменения с жестким диском.

Советы.

- Если установить галочку в как шаге 2, то опция "размер кластера 64К" становится доступной в задачах *Изменение/Перемещение* раздела (Resize/Move Partition), и в диалоговых окнах *Изменения размера кластера* (Resize Clusters).
- Если использовать разные операционные системы, то не рекомендуется использовать кластеры размером 64К.
- В процессе разделения программа Partition Magic выполняет проверку на сбойные секторы. Дисковые интерфейсы IDE и SCSI устроены так, что часто обрабатывают сбойные секторы внутри, делая излишним дополнительную проверку. Partition Magic позволяет отключать проверку на сбойные секторы. Если проверка отключена, то все операции выполняются гораздо быстрее.
- Если установлено два диска, то можно отключить изменение одного из них, как в шаге 4. В шаге 4 есть, исключения даже если выбрана эта опция, то некоторые загрузочные файлы Windows NT все равно могут быть изменены.

Ниже представлены особенности других программ по обслуживанию жестких дисков в процессе их эксплуатации.

2. PARAGON PARTITION MANAGER

Функции программы во многом совпадают с возможностями предыдущей пр ограммы - любые разделы можно создавать, удалять, форматировать, перемещать, конвертировать между файловыми системами, объединять и изменять их атрибуты, уменьшать или увеличивать размер разделов - и все это без потери данных. Кроме того, программа от отечественных производителей. Программа способна работать практически с любыми накопителями -

3. ACRONIS DISK DIRECTOR SUITE

Еще одна отечественная разработка. Возможности утилиты по редактированию разделов дублируют функциональность предыдущих. Дополнительно в комплект входит утилита Acronis Disk Editor, благодаря которой можно вручную редактировать огромное количество параметров жесткого диска и содержащихся на нем разделов. В частности, можно править таблицу разделов, загрузочные секторы FAT и

4. ACRONIS RECOVERY EXPERT

Нередко проблемы потери данных выходят за рамки гибели пары файлов, порой случается и так, что бесследно исчезают и целые разделы. Список причин, в результате которы х может случиться подобная неприятность, довольно обширен - простая невнимательность или неосторожность пользователя, сбой в работе жесткого диска, проказы вируса, ошибка в исполняемой программе, скачок напряжения в сети и многое другое. Помочь может эта программа. Сначала она сканирует неразмеченную область диска на предмет нахождения пропавших разделов.

5. PARTITION TABLE DOCTOR

Одна из самых распространенных неприятностей - это частичное повреждение главной загрузочной записи (Master Boot Record), таблицы разделов (Partition Table) или заг рузочных секторов (Boot Sectors), в результате чего система может вообще отказаться запускаться. Справиться с этими проблемами, и поможет данная

6. PARAGON MOUNT EVERYTHING

В последнее время все большую популярность набирают файловые системы NTFS, Ext2, Ext3. Но далеко не у всех установлены ОС, поддерживающие эти системы. Поэтому возникают проблемы совместимости при появлении в системе нового накопителя с другой файловой системой. Данная программа позволяет решить эти проблемы: моментально подключает разделы NTFS, Ext2, Ext3 в любой версии Windows, после чего работа с ними никак не будет отличаться от использования стандартных разделов FAT. Подключенным разделам присваивается буква, на них можно

7. DISK DIRECTOR SUITE

Эта программа предназначена для профессиональной работы с жестким диском. Это комплексный программный пакет, который включает в себя менеджер разделов, позволяющий осуществлять копирование, перемещение и изменение любых разделов Windows и Linux без риска потери данных, инструмент для восстановления разделов на жестком диске, а также менеджер загрузки, позволяющий установить несколько ОС на один ПК и

разделы даже в ситуациях, когда загрузка компьютера невозможна. Программа оснащена паролем на вход и файлом помощи.

8. EASY RECOVERY PRO

Эта программа предназначена для восстановления утраченных или недоступных (в результате их повреждения) данных. Утилита позволяет без особого труда восстановить данные на жестком диске при утере их вследствие случайного удаления, атаки вирусов, повреждения из-за отключения или резких колебаний напряжения в электросети, ошибок в программе, проблем при создании разделов, неправильного включения ПК, повреждения структуры файловой системы. При помощи команды Drive Test можно проверить диск на наличие физических проблем.

9. FILE RECOVERY

Утилита предназначена для восстановления удаленных или стертых в результате форматирования жесткого диска, данных. Работает с файловыми системами FAT 12/16/32 и NTFS, а также умеет восстанавливать зашифрованные и сжатые файлы. Имеется возможность восстановления информации не только на жестком диске, но и на съемных носителях - дискетах, картах SmartMedia, CompactFlash, Memory Stick и т.д.

10. RESTORER2000 Data RECOVERY

Это мощная программа, которая поможет быстро и просто восстановить нужные файлы, утерянные в результате случайного

11. HDD Temperature Pro

Это очень маленькая утилита, предназначена для отслеживания состояния жестких дисков. Используя технологию SMART, встроенную во все современные жесткие диски, она анализирует и показывает текущую температуру диска. Здесь возможна установка максимальной температуры накопителя, при превышении которой программа выдаст сообщение. Можно сделать так, чтобы эта утилита самостоятельно

12. TREESIZE

Эта утилита предназначена для мониторинга пространства на жестком диске и его освобождении. Она умет искать старые и неиспользуемые, а также временные файлы и удаляет их. С помощью этой утилиты можно найти папки, которые занимают больше всего места на диске, сравнить их объем в

Восстановление удаленных файлов. Общие сведения о программе Easy Recovery Pro

Easy Recovery Pro на сегодняшний день - это одна из лучших программ своего класса. Облегченный вариант - Easy Recovery Lite - входит в состав пакета комплексного обслуживания системы Fix-It Utilities.

Easy Recovery умеет работать почти со всеми более-менее распространенными файловыми системами: FAT12, FAT16, FAT32, NTFS, Novell, стандартами ZIP и JAZ-приводов, поддерживаются также и SCSI-жесткие диски. Одно из важнейших достоинств программы заключается в том, что у нее не только удобный и понятный Windows-интерфейс, доступный неопытным пользователям, но и есть возможность создать комплект загрузочных дискет с полноценной DOS-версией Easy Recovery. Сделано это для того, чтобы в случае серьезных неполадок, когда нет возможности

загрузить Windows (а, соответственно, и "виндовскую" версию Easy Recovery), вас всегда был бы доступ к жесткому диску, и вы могли бы восстанавливать файлы непосредственно из MS-DOS. Такой режим наиболее предпочтителен при крупных сбоях - на сбойный диск ничего не записывается, Easy Recovery

работает для него в режиме Read only («Только чтение»), поэтому и файлы на нем будут в большей сохранности.

Первое, что бросается в глаза сразу после запуска программы - очень долгий процесс сканирования диска. Однако это не является недостатком, а совсем наоборот - свидетельствует о ее неслабых возможностях. Дело в том, что как уже отмечалось, быстрые, простые программы получают информацию об удаленных файлах и шансах на их восстановление из структуры директорий таблицы размещения файлов. Времени это, конечно, занимает очень мало, но ведь файл может еще быть на диске даже в том случае, если больше никаких его следов не осталось, да и сама таблица размещения файлов и корневая директория могут быть разрушены. Вот тут-то и спасет вас Easy Recovery - она просканирует целиком весь жесткий диск, кластер за кластером, пытаясь собрать все кусочки каждого файла воедино.

При этом допускается полная потеря обоих копий таблицы FAT, повреждение Root Folder и загрузочного сектора диска. Разумеется, если что-то из этого все-таки сохранилось, то будет в полной мере использовано. Кстати, если вы регулярно дефрагментируете диск, то шансы на успех еще больше увеличиваются - файл, у которого используемые кластеры идут друг за другом, восстановить проще.

Таким образом, Easy Recovery - это одна из немногих программ, которая справляется не только с восстановлением ошибочно удаленных файлов, но и восстанавливает информацию на диске после повреждения его вирусами, форматирования, переразбиения на разделы, порчи при скачках напряжения питания, сбоях аппаратного оборудования или программ.

Из "виндовского" интерфейса вы, разумеется, тоже получите все эти возможностей, но только в том случае, если диск с операционной системой невредим. Поэтому целесообразно сделать заранее загрузочные дискеты Easy Recovery - с ними ваши данные будут иметь как бы дополнительный "спасательный круг". Правда, поскольку Easy Recovery с поврежденным диском работает только на чтение, то придется запастись вторым винчестером или другим носителем, прежде чем приступать к восстановлению больших объемов данных. Причем доступ к диску вы, скорее всего, получите, даже если ваша ОС его не обнаруживает.

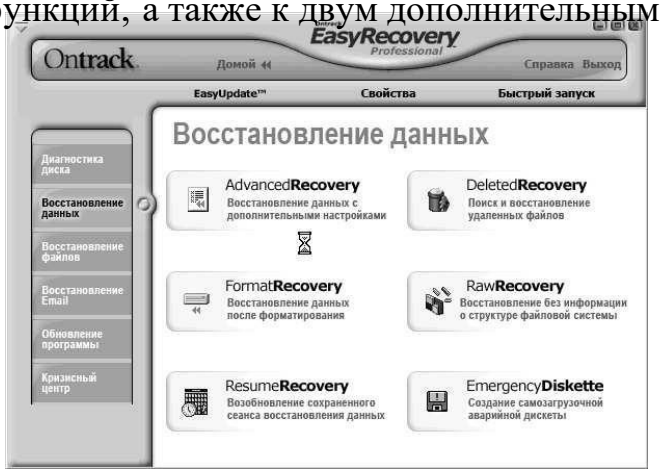
Конечно, с DOS-вариантом программы работать сложнее, поэтому желательно предварительно изучить инструкцию, чтобы разобраться во всех многочисленных опциях Easy Recovery.

Приятных и полезных дополнительных функций у Easy Recovery немало: так, например, "виндовская" версия умеет проводить диагностический тест диска, аналогичный тому, что используется стандартным ScanDisk. При восстановлении файлов сохраняются длинные имена. В соответствии с последними стандартами, программа способна обновляться через Интернет.

Восстановление файлов с помощью EasyRecovery

Запустите EasyRecovery (Пуск – Тема – Осмотр носителя – 4 Восстановление данных – EasyRecovery Professional).

После загрузки программы на экране появляется окно, в левой части которого размещено меню в виде кнопок, обеспечивающих доступ к четырем категориям функций, а также к двум дополнительным сервисам:



• Диагностика диска – утилиты для проверки физических параметров диска и целостности файловой системы;

• Восстановление данных – утилиты для поиска и восстановления удаленных и поврежденных данных;

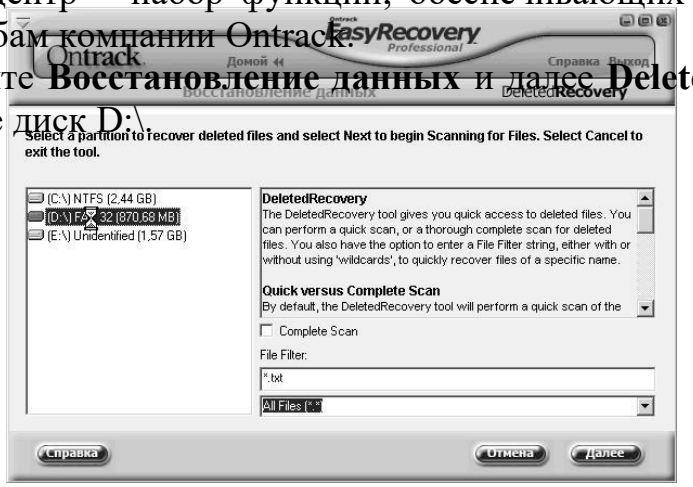
• Восстановление файлов – специализированные утилиты для восстановления файлов, созданных приложениями из семейства MS Office (кроме Outlook), а также ZIP-архивов;

• Восстановление Email – специализированная утилита для восстановления файлов Outlook;

• Обновление программы – сервисные функции, позволяющие получать информацию и выполнять обновление лицензионной версии EasyRecovery через Интернет;

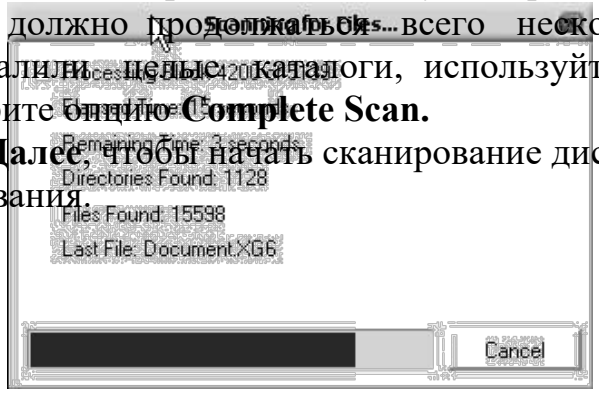
• Кризисный центр – набор функций, обеспечивающих доступ к сервисным веб-службам компании Ontrack.

В меню выберите **Восстановление данных** и далее **DeletedRecovery**. В левой части выберите диск D:\.



Примечание. Если вы удалили один или несколько файлов, быстрое сканирование должно найти эти файлы. Поиск будет производиться только в файловой системе (это должно продолжаться всего несколько секунд). В случае, когда вы удалили целые каталоги, используйте опцию полного поиска. Для этого выберите опцию **Complete Scan.**

Нажмите кнопку **Далее**, чтобы начать сканирование диска. Вы увидите окно прогресса сканирования.

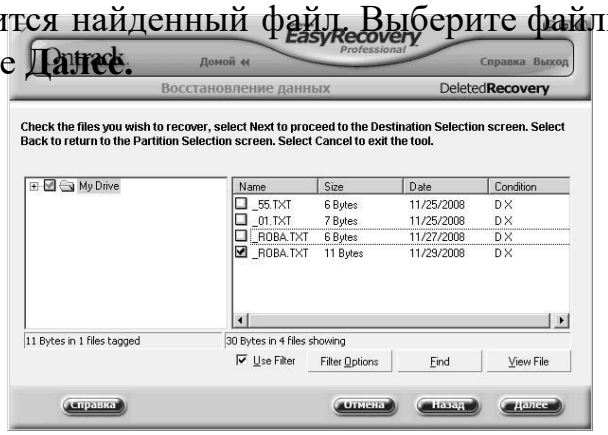


• **Processing block** показан сканированный блок диска и число всех блоков

до момента сканирования

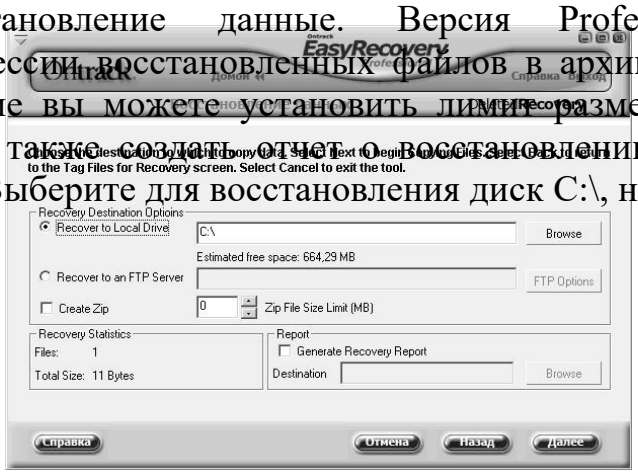
- **Elapsed time** время, которое прошло от момента начала сканирования
- **Remaining time** предполагаемое время, которое осталось до окончания операции • **Directories found** количество найденных на диске каталогов
- **Files found** количество найденных файлов
- **Last file** название последнего найденного файла

После окончания сканирования вы увидите список найденных файлов. Однако надо помнить, что не каждый найденный с помощью EasyRecovery файл возможно восстановить. Поле Condition в списке файлов показывает в каком состоянии находится найденный файл. Выберите файлы, которые хотите восстановить и щелкните **Далее.**

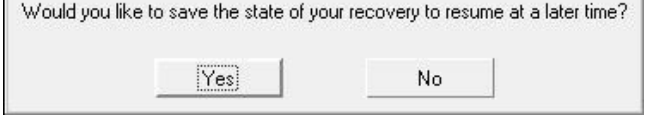


Примечание.Первый символ имени удаленного файла заменен символом подчеркивания.

В следующем окне в поле **Recovery Statistics** находится короткая статистика о восстановленных файлах, включающая количество файлов, которые вы выбрали для восстановления, а также их полный размер. Выберите директорию, в которую их надо записать (**Recover to Local Drive**). Вы также можете отправить восстановленные файлы непосредственно на сервер FTP (**Recover to an FTP Server**). Помните, что EasyRecovery не позволит записать файлы в раздел, с которого происходит восстановление данные. Версия Professional предлагает возможность компрессии восстановленных файлов в архив ZIP (**Create ZIP**). На ваше усмотрение вы можете установить лимит размера файла ZIP (**ZIP File Size Limit**), а также создать отчет о восстановлении файлов (**Generate Recovery Report**). Выберите для восстановления диск C:\, нажмите **Далее**.



В следующем окне нажмите Готово.



EasyRecovery может записать установки восстановления, чтобы потом вы смогли продолжить операцию восстановления других файлов. Нажмите кнопку **No**.

Вы восстановили данные.

Просмотрите восстановленный файл.

Практическая работа №19
«Восстановление удаленных файлов»

Цель: Получение теоритических и практических навыков программного восстановления данных.

Форма отчета:

−выполнить задание;

−показать преподавателю;

−ответить на вопросы преподавателя.

Время выполнения: 2 ч

Восстановление данных TestDisk

TestDisk — свободная программа для восстановления данных, предназначенная прежде всего для восстановления потерянных разделов на носителях информации, а также для восстановления загрузочного сектора, после программных или человеческих ошибок (например, потеря MBR).

- Установка **<sudo apt-get install testdisk>**.
- Запускаем TestDisk **<sudo testdisk>**.
- Появляется окошко приветствия TestDisk, нам предлагается вести лог работы (для выполнения данной работы лог не требуется).
- Выбираем нужный диск и нажимаем **Enter**.
- Предлагается выбрать тип таблицы разделов, обычно TestDisk определяет все правильно, так что нажимаем **Enter**.
- Выбираем **Analise**.
- Выбираем **QuickSearch**.
- Нам выводят таблицу разделов. Выбираем раздел и нажимаем **P**, чтобы вывести список файлов.
- Выбираем файлы для восстановления и нажимаем **C**.
- Выбираем папку, куда будут сохранены файлы и нажимаем **C**.

Восстановление данных PhotoRec

PhotoRec - это утилита, входящая в состав пакета TestDisk. Предназначена для восстановления испорченных файлов с карт памяти цифровых фотоаппаратов (CompactFlash, Secure Digital, SmartMedia, Memory Stick, Microdrive, MMC), USB flash-дисков, жестких дисков и CD/DVD. Восстанавливает файлы большинства распространенных графических форматов, включая JPEG, аудио-файлы, включая MP3, файлы документов в форматах Microsoft Office, PDF и HTML, а также архивы, включая ZIP. Может работать с файловыми системами ext2, ext3, ext4 FAT, NTFS и HFS+, причем способна восстановить графические файлы даже в том случае, когда файловая система повреждена или отформатирована.

- Установка **<sudo apt-get install testdisk>**.
- Запускаем PhotoRec **<sudo photorec>**.
- Выбираем нужный диск и нажимаем **Enter**.
- В нижнем меню можно выбрать **File Opt**, чтобы выбрать типы файлов для восстановления (по умолчанию выбраны все).
- Чтобы начать восстановление нажмите **Enter**, выбрав **Search**.
- У нас выбрана система ext4, поэтому выбираем первый вариант [ext2/ext3].
- Если выбрать пункт **FREE**, то поиск будет произведен в пустом пространстве и в этом случае будут восстановлены только удаленные файлы, а если выбрать **WHOLE**, то поиск будет произведен на всем диске.
- Теперь нужно указать директорию, куда будем сохранять нужные нам файлы. Выбираем нужную папку и нажимаем **C**.
- Выбираем файлы для восстановления и нажимаем **C**.

Восстановление данных Extundelete

Extundelete – утилита, позволяющая восстанавливать файлы, которые были удалены с разделов ext3/ext4.

- Установка: **<sudo apt-get install extundelete>**.
- Как только вы поняли, что удалили нужные файлы, необходимо отмонтировать раздел: **<umount /dev/<partition> >**
- Зайдите в каталог, в который будут восстанавливаться удаленные данные. Он должен быть расположен на разделе отличном от того, на котором хранились восстанавливаемые данные: **cd /<путь_к_каталогу_куда_восстанавливать_данные>**
- Запустите **extundelete**, указав раздел, с которого будет происходить восстановление и файл, который необходимо восстановить: **sudo extundelete /dev/<partition> –restore-file /<путь к файлу>/<имя_файла>**
- Можно так же восстанавливать содержимое каталогов: **sudo extundelete /dev/<partition> –restore-directory /<путь_к_директории>**

Восстановление данных Foremost.

Foremost - консольная программа, позволяющая искать файлы на дисках или их образах по hex-данным, характерным заголовкам и окончаниям. Программа проверяет файлы на предмет совпадения заранее определённых hex-кодов (сигнатур), соответствующих наиболее распространённым форматам файлов. После чего экстрагирует их из диска/образа и складывает в каталог, вместе с подробным отчётом о том, чего, сколько и откуда было восстановлено. Типы файлов, которые foremost может сразу восстановить: jpg, gif, png, bmp, avi, exe, mpg, wav, riff, wmv, mov, pdf, ole, doc, zip, rar, htm, cpp. Есть возможность добавлять свои форматы (в конфигурационном файле /etc/foremost.conf), о которых программа не знает.

- Установка: **<sudo apt-get install foremost>**
- Пример использования для восстановления изображений с диска **/dev/sdb** в каталог **~/out_dir**: **<sudo foremost -t jpg,gif,png,bmp -i /dev/sdb -o ~/out_dir>**

Задания к лабораторной работе

- Добавьте в виртуальную машину виртуальный жесткий диск.
- Запустите виртуальную машину с Linux.
- Запустите fdisk (gdisk или parted) и создайте таблицу разделов MBR с разделами.
- Отформатируйте созданные разделы в файловую систему ext4.
- Установите TestDisk.
- Удалите MBR (или таблицу разделов) с помощью команды DD.
- Восстановите MBR (или таблицу разделов) с помощью TestDisk.
- Смонтируйте восстановленные разделы и создайте там произвольные файлы.
- Удалите созданные файлы.
- С помощью TestDisk восстановите данные.
- Создайте произвольный каталог и запишите туда данные каталога /var/log/ .
- Удалите данные с созданного каталога.
- С помощью PhotoRec восстановите данные.
- Создайте произвольный каталог и запишите туда данные каталога /etc/ .
- С помощью Extundelete или Foremost восстановите данные.

Практическая работа №20
«Мониторинг активности портов»

Цель работы: формирование умений и навыков блокировки и

разблокировки портов подключения устройств

Форма отчета:
−выполнить задание;
−показать преподавателю;
−ответить на вопросы преподавателя.
Время выполнения: 2 ч

Основные положения

Понятие порта в компьютере многозначно. Самое общее определение: ***порт*** - это соединение (физическое или логическое), через которое принимаются и отправляются данные. Обмен данными между любыми устройствами возможен только при наличии утвержденного стандарта на интерфейс.

В состав аппаратного обеспечения порта входит специализированный разъём, предназначенный для подключения оборудования определённого типа. Часто этот специализированный разъем и называют портом, например USB-порт, но есть разъемы, которые портами называть не принято, например, RJ11. Как правило, каждый порт имеет обозначение, которое размещается рядом с разъемом.

Основные порты, используемые в компьютерах,
ноутбуках: · USB-порт;
· IEEE 1394 (FireWire) ;
· Порт eSATA и комбинированный порт
USB/eSATA; · Сетевой порт Ethernet;
· Порт SCSI;
· Последовательный порт RS-232;
·Порты для подключения внешних мониторов VGA, DVI, S-Video, HDMI,
DisplayPort; · Порт для док-станции и порт репликатор;
· Порты для модулей расширения PCMCIA, ExpressCard.

USB - Universal Serial Bus - универсальная последовательная шина. USB-порты являются своего рода стандартом для подключения внешних устройств, к которому стремятся все производители этих устройств. К портам USB подключаются: мыши, клавиатуры, принтеры, сканеры, модемы, кардридеры, флэш-накопители, фотоаппараты, сотовые телефоны, плееры, жёсткие диски, оптические дисководы и др.

IEEE 1394 - высокоскоростной последовательный порт для цифровых видеоустройств. Компания Apple продвигает стандарт IEEE 1394 под маркой FireWire, компания Sony – под маркой i.LINK. IEEE 1394 применяется для подключения видеокамер, цифровых фотоаппаратов и других мультимедийных устройств, а также принтеров, сканеров, внешних жестких дисков.

Основные преимущества по сравнению с USB 2.0 - более высокая скорость передачи, большая стабильность, большая длина кабеля до оконечного устройства.

eSATA - External Serial ATA (Advanced Technology Attachment - присоединение по передовой технологии) – последовательный интерфейс для подключения внешних устройств, поддерживающий режим «горячей замены». Стандарт eSATA предусматривает подключение внешних жестких дисков, оптических дисков, RAID-массивов. Скорость передачи данных гораздо выше, чем у USB 2.0 или IEEE 1394.

Недостатки eSATA:

• максимальная длина кабеля не превышает 2 метров;

• жёсткие диски, подключаемые через eSATA, потребуют дополнительного источника питания - это могут быть как разъёмы USB или 1394, так и розетка.

Порт Ethernet предназначен для подключения ноутбука к компьютерной сети с помощью сетевого кабеля через разъем RJ45 (RJ-45). Технология Ethernet описывается стандартами IEEE группы 802.3. Существует несколько стандартов технологии Ethernet. Стандарты различаются скоростью передачи данных и передающей средой. В ноутбуках обычно устанавливают порт Ethernet 10/100/1000, который поддерживает стандарты

10BASE-T, 100BASE-TX и 1000BASE-T для расстояний до 100 м. Стандарт 10BASE-T позволяет передавать данные со скоростью 10 Мбит/с. Для передачи используется 4 провода кабеля витой пары категории 3 или категории 5. По стандарту 100BASE-TX скорость передачи данных составляет 100 Мбит/с. Стандарт применяется для построения сетей топологии «звезда». Задействована витая пара категории 5, поддерживается дуплексная передача данных. Стандарт 1000BASE-T - гигабитный (Gigabit, Geth) Ethernet позволяет передавать данные со скоростью до 1 Гбит/с. Стандарт предусматривает использование витой пары категорий 5е.

RS-232 (англ. Recommended Standard) - стандарт последовательной асинхронной передачи двоичных данных между двумя устройствами на расстоянии до 15 метров. Порт RS-232 в последнее время не часто встречается в бизнес-ноутбуках, но может быть полезен в промышленных ноутбуках.

Он используется для реализации систем сбора данных в реальном времени, подключения научного ряда контактов. Карты Type III поддерживают 16- или 32-разрядный интерфейс. Они имеют толщину 10,5 мм, что позволяет устанавливать на карту стандартные разъёмы внешних интерфейсов и избавиться, таким образом, от дополнительных кабелей. Разъем имеет четыре ряда контактов. Разъем PCMCIA представляет собой щель шириной 54 мм, которая закрыта либо откидной шторкой, либо пластиковой заглушкой.

Разъем (слот) PCMCIA (вверху) и заглушка, внизу – кардридер.

Большинство ноутбуков оснащается лишь одним разъемом PCMCIA типа II. А современные ноутбуки уже обходятся и вовсе без этих разъемов.

Порт ExpressCard. Стандарт ExpressCard для карт расширения был разработан ассоциацией PCMCIA на смену стандарту PC Card. Новый стандарт был создан на базе новой скоростной последовательной шины PCI Express. Стандарт ExpressCard не только более производительный, чем PC Card, но и более универсальный. Через ExpressCard можно подключаться к шине USB. Карты ExpressCard бывают двух типов, отличающихся по ширине: 34 мм и 54 мм. Соответственно и разъемы бывают двух типов ExpressCard/34 и ExpressCard/54. При этом карты 34 мм можно устанавливать как в разъем ExpressCard/34, так и в разъем ExpressCard/54. Через разъемы ExpressCard подключают ТВ-тюнеры, звуковые карты, карты Wi-Fi, флеш-накопители (они часто подключаются через USB-составляющую интерфейса ExpressCard), модемы для работы в сотовых сетях и др.

Разъем RJ11(RJ-11 Registered jack) – разъем модема ноутбука. Используется для подключения к Интернету через модем по телефонной линии.

Сравнение средств мониторинга действий пользователей

Одной из важных особенностей современных корпоративных сетей является их размер, который зачастую исчисляется тысячами, а и иногда и десятками тысяч компьютеров. При этом деятельность пользователей может быть распределена среди различных компьютеров, а одна и та же проблема часто решается группами пользователей. Важной

задачей является контроль работы, как отдельных пользователей, так и групп пользователей.

Основными целями контроля являются: обеспечение информационной безопасности, выявление случаев некорректного, непрофессионального или нецелевого использования ресурсов, оценка характеристик функционирования корпоративной сети и параметров использования ресурсов.

Основной задачей обеспечения информационной безопасности является «раннее обнаружение» внутренних вторжений, т.е. выявление действий пользователей, которые могут предшествовать внутренним вторжениям. Чем крупнее организация, тем актуальней является для нее проблема предотвращения внутренних вторжений, в частности кражи информации, так как именно кража является конечной целью большинства внутренних вторжений. Связано это с тем, что в больших организациях затрудняется контроль над обращением информации и существенно возрастает цена ее утечки. Указанные обстоятельства определяют высокий уровень озабоченности данной проблемой со стороны крупного бизнеса и правительственных организаций. Решение данной проблемы

заключается в применении "жесткой" политики информационной безопасности в организации и использовании средств мониторинга действий пользователей.

Spector 360

Spector 360 включает в себя средства для автоматического развертывания и удаленного управления, осуществляет запись разнообразных действий, включая: Email, чаты, мгновенные сообщения, посещаемые веб-сайты, онлайновые поисковые запросы, нажимаемые клавиши и используемые программы. Spector 360 также включает в себя средство для записи образов экрана в режиме видеокамеры.

Все эти инструменты ведут запись одновременно, скрытно, под защитой тройного уровня безопасности. Приложение Recorder хорошо конфигурируется и может быть настроено для записи только интересующих Вас событий.

В дополнение к мониторингу и ведению записи Spector 360 обладает развитой системой определения и обнаружения ключевых слов, которая будет немедленно извещать о каждом случае, когда пользователь контролируемого ПК отклонится от допустимого использования ПК или Интернет.

Регистратор Spector 360 можно перевести в скрытый режим, который обеспечивает невозможность обнаружения программы неуполномоченными пользователями. В скрытом режиме Spector 360 не будет виден пользователю в системном меню задач, диспетчере задач или в меню установки/удаления программ панели управления.

При помощи Spector 360 вы можете сгенерировать высококачественные отчеты для руководства, которые могут регулярно распечатываться или рассылаться по почте.

Spector 360 разработан для коммерческих, образовательных и правительственных организаций, использующих сети на платформе Windows.

Security Curator

Security Curator – это система обеспечения информационной безопасности нового поколения, объединяющая в себе возможность наблюдения за деятельностью сотрудников, контроля их действий и блокировки потенциально опасных путей утечки информации.

Security Curator ведёт мониторинг в реальном времени практически всех действий сотрудников при работе за компьютером. Информация о действиях пользователей обновляется в реальном режиме времени. При этом постоянно производится сохранение снимков экрана при совершении любых действий, также существует возможность наблюдения за рабочим столом пользователя в режиме он-лайн. В случае работы пользователем с USB-устройствами производится резервное копирование файлов.

Внедрение **Security Curator** позволяет ограничить доступ к нежелательным сайтам, программам и приложениям на **определенный промежуток времени** либо постоянно. Например, работодатель может разрешить сотрудникам посещать сайты ВКонтакте и Одноклассники

только во время обеденного перерыва, а доступ к бухгалтерской программе 1С запретить после окончания рабочего дня и на выходных.

Activity Monitor

Этот мощный инструмент позволяет отслеживать любые действия в сети и предоставляет вам детальную информацию о том, что, как и когда делали ваши сотрудники. Будь то сеть библиотеки, университета или коммерческой организации, Activity Monitor поможет вам установить эффективный контроль над ней.

Приложение состоит из серверной и клиентской частей. Сервер Activity Monitor может быть установлен на любом компьютере в сети. Модуль-шпион(агент) устанавливается на всех компьютерах, действия на которых вы хотите отслеживать. Он может быть установлен даже удалённо с системы, на которой установлена серверная часть Activity Monitor.

Действия на сетевых компьютерах отслеживаются удалённо. Вы можете настроить программу таким образом, что она будет отслеживать и регистрировать действия на всех компьютерах в сети одновременно. Данные мониторинга могут быть использованы для более глубокого анализа и создания детальных отчётов.

Activity Monitor является эффективным средством повышения общей производительности труда в компаниях, использующих данную программу для мониторинга

локальных сетей. Проще говоря, этот мощный инструмент от Softactivity экономит ваши деньги.

NetVizor

NetVizor — Программа для мониторинга сети. NetVizor позволяет наблюдать за всей локальной сетью из одного рабочего места. Программа может следить за рабочими станциями и индивидуальными пользователями, которые используют различные компьютеры, находящимся в сети.

Программа позволяет следить за сетевыми компьютерами, осуществлять фильтрацию контента и управлять сетевыми компьютерами дистанционно.

Существует возможность ведения журналов адресов посещенных сайтов, соединений с интернетом, открываемых файлов, чатов, пересылаемых

Мониторинг	**Spector 360**	**Security Curator**	**Activity Monitor**	**NetVisor**
Экран	+	+	+	+
Снимки экрана	+	+	+	+
Запущенные процессы	+	+	+	+
Время запуска и выключения программ	+	+	+	+
Бесплатные сервисы электронной почты	+	-	+	+
Нажатие клавиш	+	+	+	+
E-mail	+	+	+	+
Посещенные сайты	+	+	+	+
Переписка в IM агентах	+	+	+	+
Социальные сети	+	+	+	+
Поисковые запросы	+	+	+	+
USB устройства	+	+	+	-
Обнаружение ключевых слов	+	-	-	+
Установка, удаление программ	+	+	+	+
Контроль рабочего времени	+	+	+	+

Загружаемые файлы	+	+	+	+
Доступ к файлам, папкам	+	+	+	+
Активность пользователя	+	+	+	+
FTP	+	+	+	+
Сетевые соединения	+	+	+	+

Выборочный мониторинг	+	+	+	
Запись по расписанию	+	+	+	
Контроль	**Spector 360**	**Security Curator**	**Activity Monitor**	**Net Visor**
Блокировка событий (запуск приложений, сайты, запрет файловых операций)	-	+	+	+
Блокировка запуска любых процессов	-	+	+	+
Блокировка подключения/отключения всех типов USB накопителей и устройств	-	+	-	-
Блокировка сетевых соединений (по порту, ip адресу)	+	+	+	+
Блокировка сайтов по домену	+	+	+	
Блокировка чатов и Интернет пейджеров	+	+	+	
Блокировка доступа в Интернет по протоколу или порту	+	+	+	
Запрет действий с файлами/папками	-	+	+	
Отчетность	**Spector 360**	**Security Curator**	**Activity Monitor**	**Net Visor**
Генерация отчетов с привязкой к отдельному пользователю	+	+	+	+

Поиск по ключевым словам	+	+	+	
Генерация графических отчетов	+	+	+	
Конвертация отчетов в PDF	+	+	-	
Конвертация отчетов в HTML	+	+	+	
Конвертация отчетов в CSV	+	+	+	
Конвертация отчетов в Excel	+	-	+	-
Конвертация отчетов в Rich Text	+	-	-	-
Экспорт отчетов	+	-	-	-
Отправка отчетов по электронной почте	+	+	-	
Отправка отчетов по FTP	+	+	-	-
Печать отчетов	+	+	+	
Генерация отчетов по расписанию	+	+	-	-

Управление	**Spector 360**	**Security Curator**	**Activity Monitor Visor**	**Net**
Централизованное управление клиентами	+	+	+	+
Централизованное управление лицензиями	+ +	+	+	
Централизованное конфигурирование безопасности	+	+	+	+
Централизованное конфигурирование сети	+	-	-	-
Централизованное конфигурирование WEB-фильтра	+ +	+	+	+
Резервирование и восстановление базы данных	+	-	-	-
Управление резервными копиями	+	-	-	-
Многопользовательский дискреционный контроль доступа к данным	+	-	-	+
Разделение доступа к функциям администрирования	+	-	-	+
Возможность группировки компьютеров	+	+	+	
Возможность группировки пользователей	+	-	+	-

Безопасность	**Spector 360**	**Security Curator**	**Activity Monitor Visor**	**Net**
Контроль компьютеров в сети	+ +	+	+	
Удаленная установка	+ +	+	+	
Невидимый режим работы	+ +	+	-	

Авторизация при запуске административного модуля	+	+	+	+

Стоимость	**Spector 360**	**Security Curator**	**Activity Monitor**	**Net Visor**
Цена за 1 лицензию (от 5 до 99 хостов)	~ 5200 руб.	~ 1800 руб.	~ 1400 руб.	~ 1600 руб.
Цена за 1 лицензию (от 100 до 249 хостов)	~ 4000 руб.	~ 1600 руб.	~ 700 руб.	~ 1100 руб.
Цена за 1 лицензию (от 250 до 1000 хостов)	~ ~	~	~	

3500 руб. 1500 руб. 600 руб.
200 руб.

Определенно, Spector 360 незаменим в крупных организациях, где решаются задачи оперативного мониторинга огромного количества рабочих станции.

Если делать акцент на возможность контроля и блокировки действий пользователей, тут подойдет Security Curator, NetVisor и Activity Monitor.

Рассмотрим два способа улучшения безопасности работы сети.

Шаг 1. Меняем учетную запись администратора (Пользователь Администратор с пустым паролем — это уязвимость) **(убираем уязвимость 1)**

При установке Windows XP в автоматическом режиме с настройками по умолчанию мы имеем пользователя **Администратор** с пустым паролем и любой **User** может войти в такой ПК с правами администратора. Чтобы решить проблему выполним команду **Мой компьютер-Панель управления-Администрирование-Управление компьютером-Локальные пользователи-Пользователи** (рис. 1).

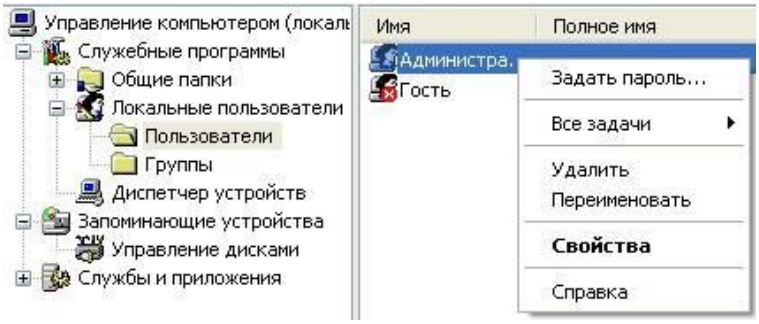


Рис. 1 - Окно Управление компьютером

Здесь по щелчку правой кнопкой мыши на **Администраторы** зададим администратору пароль, например, 12345. Теперь в окне **Администрирование** зайдем в **Локальную политику безопасности**. Далее идем по веткам дерева: **Локальные политики-Параметры безопасности-Учетные записи: Переименование учетной записи Администратор** (рис..2).

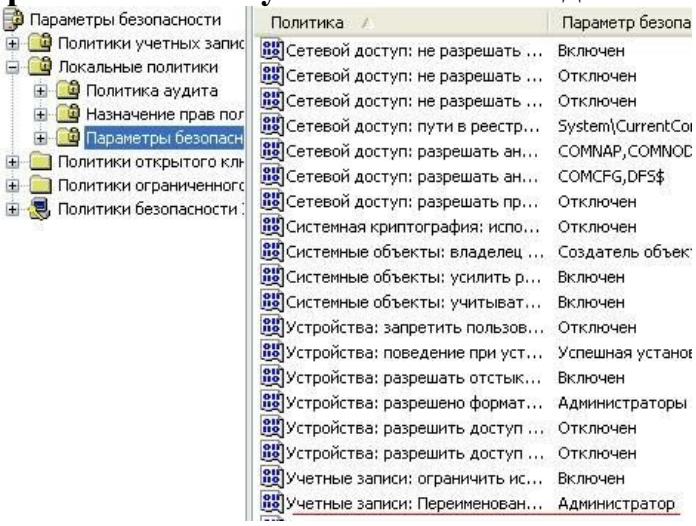


Рис. 2 - Находим в системном реестре запись Переименование учетной записи Администратор

Здесь пользователя **Администратор** заменим на **Admin** (рис. 3).

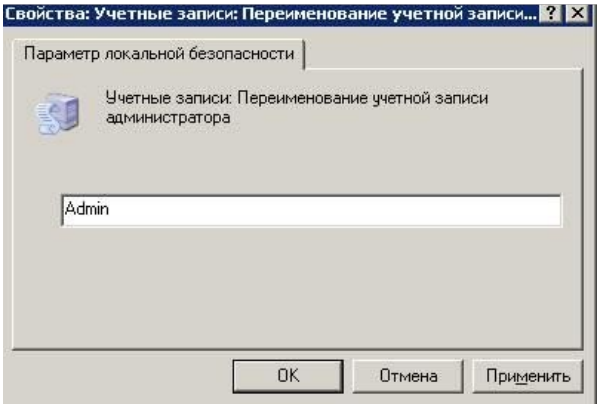


Рис. 3 - Пользователю Администратор присваиваем новое имя

Перезагружаем ОС. После наших действий получилась учетная запись Admin с паролем 12345 и правами администратора (рис. 4).

Рис. 4 - Окно входа в ОС Windows XP

Все, теперь мы имеем пользователя **Администратор** с паролем, одна из уязвимостей системы устранена.

Операцию по изменению имени пользователя и заданию пароля мы также могли бы выполнить без использования системного реестра, использовав окно **Учетные записи пользователей**, что гораздо проще (рис. 5).

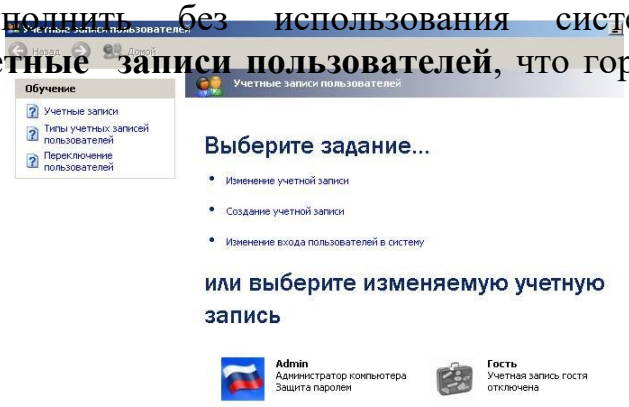


Рис. 5 - Окно Учетные записи пользователей **Примечание**

Учетная запись Гость позволяет входить в ПК и работать на нем (например, в Интернет) без использования специально созданной учетной записи. Запись Гость не требует ввода пароля и по умолчанию блокирована.

Гость не может устанавливать или удалять программы. Эту учетную запись можно отключить, но нельзя удалить.

Шаг 2. Делаем окно приветствия пустым (убираем уязвимость 2)

У нас окно входа в систему содержит подсказку Admin, давайте ее уберем, сделав окно пустым. Для начала в окне **Учетные записи пользователей** жмем на

кнопку **Изменение входа пользователей в систему** и уберем флажок **Использовать страницу приветствия** (рис. 6 и рис. 7).

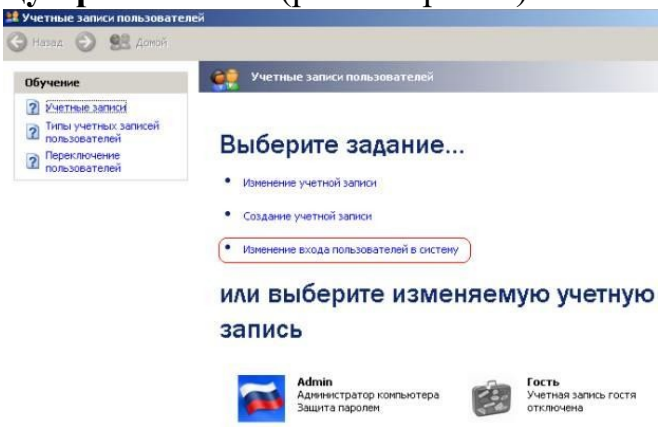


Рис. 6 - Окно Учетные записи пользователей

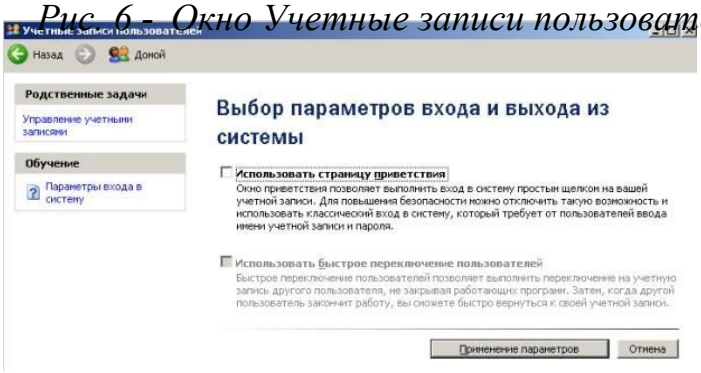


Рис. 7 - Убираем флажок Использовать страницу приветствия

Теперь повысим безопасность сети еще на одну условную ступень, сделав оба поля окна приветствия пустыми (рис. 8).

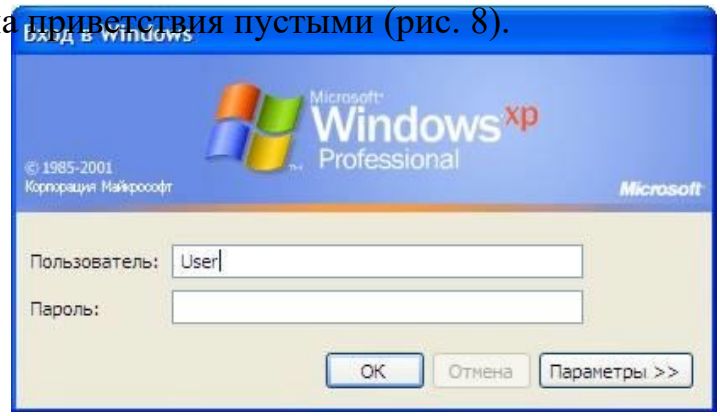
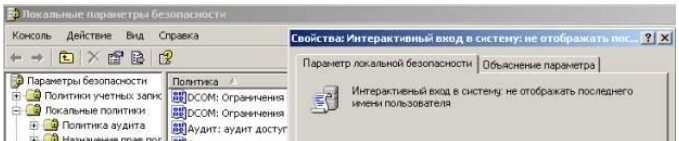


Рис. 8 - Обе строки данного окна сделаем пустыми

Выполним команду **Панель управления - Администрирование – Локальные политики безопасности - Локальные политики - Параметры безопасности -Интерактивный вход: не отображать последнего имени пользователя**. Эту запись необходимо включить (рис. 9).



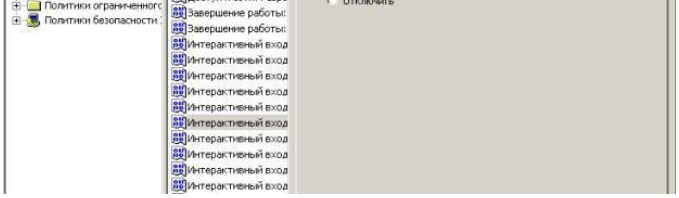


Рис. 9 - Активируем переключатель Включить

Теперь после завершения сеанса пользователь должен угадать не только пароль, но и имя пользователя (рис. 10).

Рис. 10 - Обе строки окна приветствия пусты

Выявление сетевых уязвимостей сканированием портов ПК

Злоумышленники используют сканирование портов ПК для того, чтобы воспользоваться ресурсами чужого ПК в Сети. При этом необходимо указать **IP** адрес ПК и открытый **port,** к примеру, **195.34.34.30:23**. После этого происходит соединение с удаленным ПК с некоторой вероятностью входа в этот ПК.

· TCP/IP port — это адрес определенного сервиса (программы), запущенного на данном компьютере в Internet. Каждый открытый порт — потенциальная лазейка для взломщиков сетей и ПК. Например, SMTP (отправка почты) — 25 порт, WWW — 80 порт, FTP —21 порт.

· Хакеры сканируют порты для того, чтобы найти дырку (баг) в операционной системе. Пример ошибки, если администратор или пользователь ПК открыл полный доступ к сетевым ресурсам для всех или оставил пустой пароль на вход к компьютер.

Одна из функций администратора сети - выявить недостатки в функционировании сети и устранить их. Для этого нужно просканировать сеть и закрыть (блокировать) все необязательные (открытые без необходимости) сетевые порты. Ниже, для примера, представлены службы TCP/IP, которые можно отключить:

· finger — получение информации о пользователях

· talk— возможность обмена данными по сети между пользователями · bootp — предоставление клиентам информации о сети

· systat — получение информации о системе

· netstat — получение информации о сети, такой как текущие соединения

· rusersd — получение информации о пользователях, зарегистрированных в данный момент

Просмотр активных подключений утилитой Netstat

Команда **netstat** обладает набором ключей для отображения портов, находящихся в активном и/или пассивном состоянии. С ее помощью можно получить список серверных приложений, работающих на данном компьютере. Большинство серверов находится в режиме **LISTEN**— ожидание запроса на соединение. Состояние **CLOSE_WAIT** означает, что соединение разорвано. **TIME_WAIT** — соединение ожидает разрыва. Если

соединение находится в состоянии **SYN_SENT**, то это означает наличие процесса, который пытается, установить соединение с сервером. **ESTABLISHED** — соединения установлены, т. е. сетевые службы работают (используются).

Итак, команда netstat показывает содержимое различных структур данных, связанных с сетью, в различных форматах в зависимости от указанных опций. Для сокетов (программных интерфейсов) TCP допустимы следующие значения состояния

· CLOSED — Закрыт. Сокет не используется. · LISTEN — Ожидает входящих соединений.

· SYN_SENT — Активно пытается установить соединение.

· SYN_RECEIVED — Идет начальная синхронизация соединения. · ESTABLISHED — Соединение установлено.

· CLOSE_WAIT — Удаленная сторона отключилась; ожидание закрытия сокета. · FIN_WAIT_1 — Сокет закрыт; отключение соединения.

· CLOSING — Сокет закрыт, затем удаленная сторона отключилась; ожидание подтверждения.

· LAST_ACK — Удаленная сторона отключилась, затем сокет закрыт; ожидание подтверждения.

· FIN_WAIT_2 — Сокет закрыт; ожидание отключения удаленной стороны.

· TIME_WAIT — Сокет закрыт, но ожидает пакеты, ещё находящиеся в сети для обработки

Примечание

Что такое «сокет» поясняет рис. 11. Пример сокета – 194.86.6..54:21

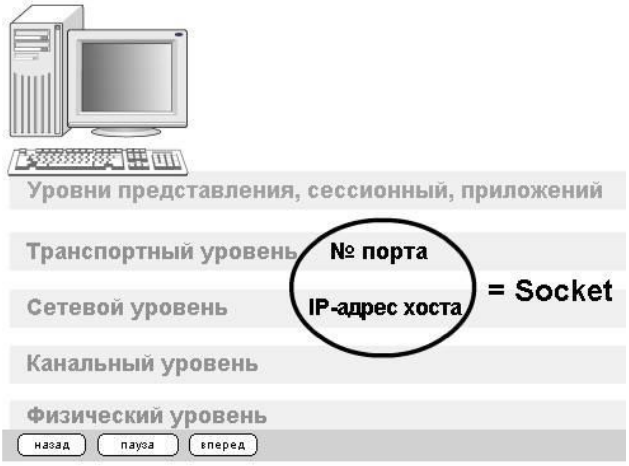
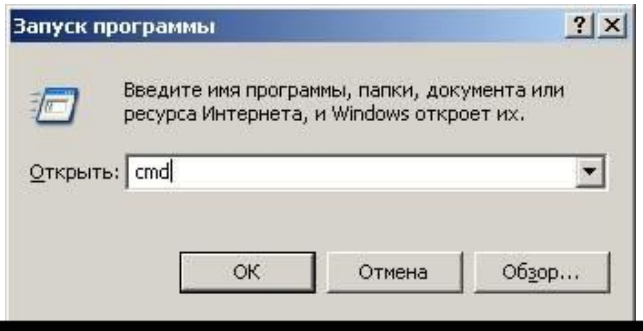
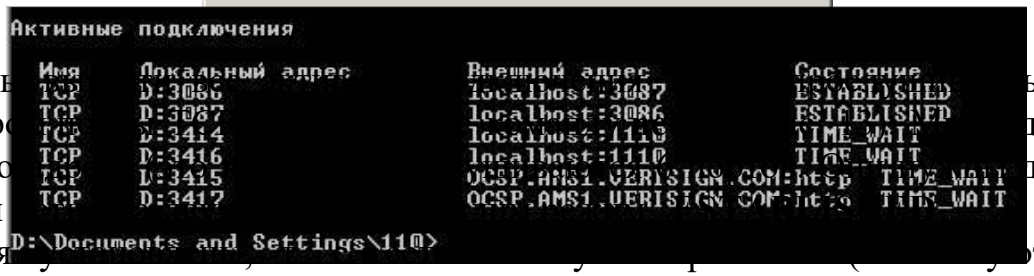


Рис. 11 - Сокет это № порта + IP адрес хоста

Практический пример. Обнаружение открытых на ПК портов утилитой Netstat Для выполнения практического задания на компьютере необходимо выполнить команду **Пуск-Выполнить**. Откроется окно **Запуск программы**, в нем введите команду **cmd** (рис. 12).





Чтобы ые компьютеры дим Локального для поддержки соединения ются). Четыре порта используются в режиме **TIME_WAIT** — соединение ожидает

разрыва.

Рис. 13 - Список активных подключений на тестируемом ПК

Запустите на вашем ПК Интернет и зайдите, например на **www. yandex. ru**. Снова выполните команду **netstat** (рис. 14). Как видим, добавилос личными состояния

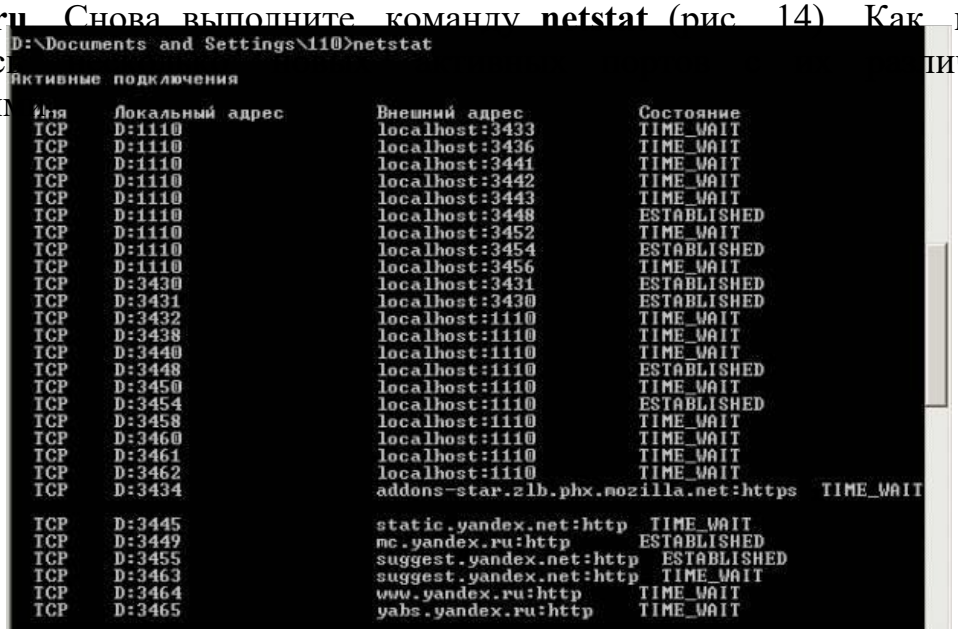


Рис. 14 - Активные подключения при работе ПК в Интернет Команда **netstat** имеет следующие опции – табл. 1.

Таблица 1 - Ключи для команды netstat

Опци я	**Назначение**
-a	Показывать состояние всех сокетов; обычно сокеты, используемые серверными процессами, не показываются
-A	Показывать адреса любых управляющих блоков протокола, связанных с сокетами; используется для отладки
-i	Показывать состояние автоматически сконфигурированных (auto-configured) интерфейсов. Интерфейсы, статически
-n	Показывать сетевые адреса как числа. netstat обычно показывает адреса как символы. Эту опцию можно использовать с любым
-r	Показать таблицы маршрутизации. При использовании с опцией -s показывает статистику маршрутизации
-s	Показать статистическую информацию по протоколам. При использовании с опцией -r показывает статистику маршрутизации
-f семейст	Ограничить показ статистики или адресов управляющих блоков только указанным семейством_адресов, в качестве которого
-I интерфе йс	Выделить информацию об указанном интерфейсе в отдельный столбец; по умолчанию (для третьей формы команды) используется интерфейс с наибольшим объёмом переданной информации с момента последней перезагрузки системы. В качестве интерфейса
-p	Отобразить идентификатор/название процесса создавшего сокет (-p, --programs display PID/Program name for sockets)

Практическая работа №21
«Блокирование портов»

Цель: научиться блокировать порты.

Форма отчета:

−выполнить задание;

−показать преподавателю;

−ответить на вопросы преподавателя.

Время выполнения: 2 ч

Программа NetStat Agent

Представьте ситуацию: ваше Интернет-соединение стало работать медленно, компьютер постоянно что-то качает из Сети. Вам поможет программа NetStat Agent. С ее помощью вы сможете найти причину проблемы и заблокировать ее. Иначе говоря, **NetStat Agent** — полезный набор инструментов для мониторинга Интернет соединений и диагностики сети. Программа позволяет отслеживать TCP и UDP соединения на ПК, закрывать нежелательные соединения, завершать процессы, обновлять и освобождать DHCP настройки адаптера, просматривать сетевую статистику для адаптеров и TCP/IP протоколов, а также строить графики для команд **Ping** и**TraceRoute** (рис. 15).

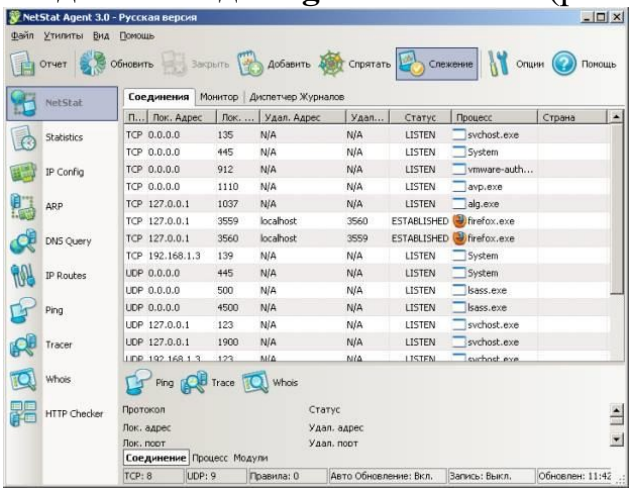


Рис. 15 - Главное окно программы

NetStat Agent В состав программы NetStat Agent вошли следующие утилиты:

· **NetStat** — отслеживает TCP и UDP соединения ПК (при этом отображается географическое местоположение удаленного сервера и имя хоста).

· **IPConfig** — отображает свойства сетевых адаптеров и

конфигурацию сети. · **Ping** — позволяет проверить доступность хоста в сети.

· **TraceRoute** — определяет маршрут между вашим компьютером и конечным хостом, сообщая все IP-адреса маршрутизаторов.

· **DNS Query** — подключается к DNS серверу и находит всю информацию о домене (IP адрес сервера, MX-записи (Mail Exchange) и др.).

· **Route** — отображает и позволяет изменять IP маршруты на ПК. · **ARP** — отслеживает ARP изменения в локальной таблице.

· **Whois** — позволяет получить всю доступную информацию об IP-адресе или домене. · **HTTP Checker** — помогает проверить, доступны ли Ваши веб-сайты.

· **Statistics** — показывает статистику сетевых интерфейсов и TCP/IP протоколов.

Сканер портов Nmap (Zenmap)

Nmap — популярный сканер портов, который обследует сеть и проводит аудит защиты. Использовался в фильме «Матрица: Перезагрузка» при взломе компьютера. Наша задача не взломать, а защитить ПК, поскольку одно и то же оружие можно использовать как для защиты, так и для нападения. Иначе говоря, сканером портов **nmap** можно определить открытые порты компьютера, а для безопасности сети пользователям рекомендуется закрыть доступ к этим портам с помощью брандмауэра (рис. 16).

Рис. 16 - Интерфейс программы Nmap

Обычно для того, чтобы просканировать все порты какого-либо компьютера в сети вводится команда **nmap –p1-65535 IP-адрес_**компьютера или **nmap –sV IP-адрес компьютера,** а для сканирования сайта — команда**nmap –sS –sV –O - P0 адрес сайта.**

Монитор портов TCPView

TCPView — показывает все процессы, использующие Интернет-соединения. Запустив **TCPView**, можно узнать, какой порт открыт и какое приложение его использует, а при необходимости и немедленно разорвать соединение – рис. 17.

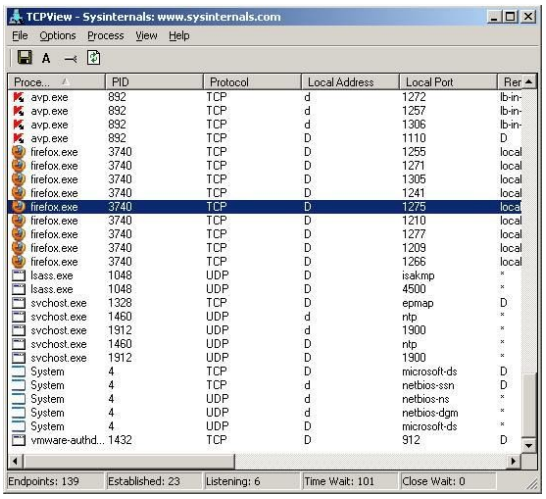


Рис. 17 - Главное окно программы TCPView

Просмотрите активные сетевые подключения локального ПК с помощью монитора портов tpiview. Определите потенциально возможные угрозы (какие порты открыты, и какие приложения их используют). При необходимости можно закрыть установленное приложением TCP-соединение или процесс правой кнопкой мыши.

Практическая работа №22
«Проверка наличия и сроков действия сертификатов»

Цель работы: научиться анализировать сертификаты соответствия.

Форма отчета:

−выполнить задание;

−показать преподавателю;

−ответить на вопросы преподавателя.

Время выполнения: 2 ч

Порядок выполнения работы:

1. Изучить теоретическую часть методических указаний;
2. Рассмотреть сертификат соответствия и провести его анализ, опираясь на приведенные вопросы;
3. Ответить на контрольные вопросы письменно;
4. Сделать выводы по проделанной работе.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Сертификат соответствия – это специальный документ, который подтверждает, что продукция качественная и соответствует российским стандартам (ГОСТ, ТУ).

Орган по сертификации после анализа протоколов испытаний, оценки производства, анализа других документов о соответствии продукции, осуществляет оценку соответствия продукции установленным требованиям. Результаты этой оценки отражают в заключении эксперта. На основании данного заключения орган по сертификации принимает решение о выдаче сертификата, оформляет сертификат и регистрирует его. Сертификат действителен только при наличии регистрационного номера. В сертификате указывают все документы, служащие основанием для выдачи сертификата, в соответствии со схемой сертификации. Срок действия сертификата устанавливает орган по сертификации с учетом срока действия нормативных документов на продукцию, но не более чем на три года.

Продукция, на которую выдан сертификат, маркируется знаком соответствия, принятым в системе. Маркирование продукции знаком соответствия осуществляет изготовитель (продавец) на основании сертификата соответствия.

Критериями для определения периодичности и объема инспекционного контроля являются степень потенциальной опасности продукции, стабильность производства, объем выпуска, наличие системы качества, стоимость проведения инспекционного контроля и т.д.

Сертификат соответствия ГОСТ Р имеет следующие пункты:

1. № сертификата соответствия:

В данной строке указывается уникальный номер СС. **Пример: РОСС RU.АЮ40.С12345**

Расшифровка номера:

RU — сокращенное обозначение страны производителя товара. В данном случае Россия.

АЮ40 — сокращенное обозначение органа по сертификации выдавшего данный сертификат. Каждый орган по сертификации имеет как полное словесно название, так и сокращенное обозначение, состоящее из двух букв и двух цифр.

Буква **С** в последней части номера обозначает код типа объекта сертификации:

А — партия (единичное изделие), сертифицированная на соответствие обязательным требованиям;

В — серийно выпускаемая продукция, сертифицированная на соответствие обязательным требованиям;

С — партия (единичное изделие), сертифицированная на соответствие требованиям нормативных документов;

Н — серийно выпускаемая продукция, сертифицированная на соответствие требованиям нормативных документов;

Е — транспортное средство, на которое выдается одобрение типа транспортного средства.

Оставшиеся цифры являются просто внутренним (для органа по сертификации) порядковым номером сертификата, в порядке включения в Государственный реестр.

2. Срок действия сертификата соответствия:

В данном пункте указывается срок действия СС. Если окончание срока действия сертификата не указано или указан прочерк, это обозначает, что сертификат бессрочный.

3. Орган по сертификации:

В данном пункте указывается полное словесное название органа по сертификации, выдавшего сертификат, а также его адрес и телефон.

4. Сертифицируемая продукция:

В этом пункте указывается полное название продукции, а также возможно упоминание о номере контракта поставки, инвойса, размера партии или указание слов «серийный выпуск».

5. Соответствует требованиям нормативных документов:

Данный пункт заполняется органом по сертификации и сообщает, требования каких документов соответствует данная продукция.

6. Изготовитель:

В данном пункте указывается полное название фирмы производителя, и его юридический адрес. В данном пункте возможно указание только одной фирмы.

7. Сертификат выдан:

В данном пункте указывается полное название фирмы держателя сертификата, его юридический адрес, ИНН (для российских фирм) и возможен телефон. Фирма- производитель продукции и фирма держатель сертификата могут быть как различными, так и одним и тем же лицом. В данном пункте возможно указание только одной фирмы.

8. На основании:

В данном пункте указываются документы, на основании которых орган по сертификации выдал данный сертификат. Ими могут быть: протоколы сертификационных испытаний продукции, декларации соответствия, зарубежные сертификаты (например, сертификаты систем качества: ISO , TUFF), или акты осмотра помещений, акты отбора образцов.

9. Дополнительная информация:

В данном пункте указываются дополнительные сведения.

10. Код ОК 005 (ОКП) (расположен справа):

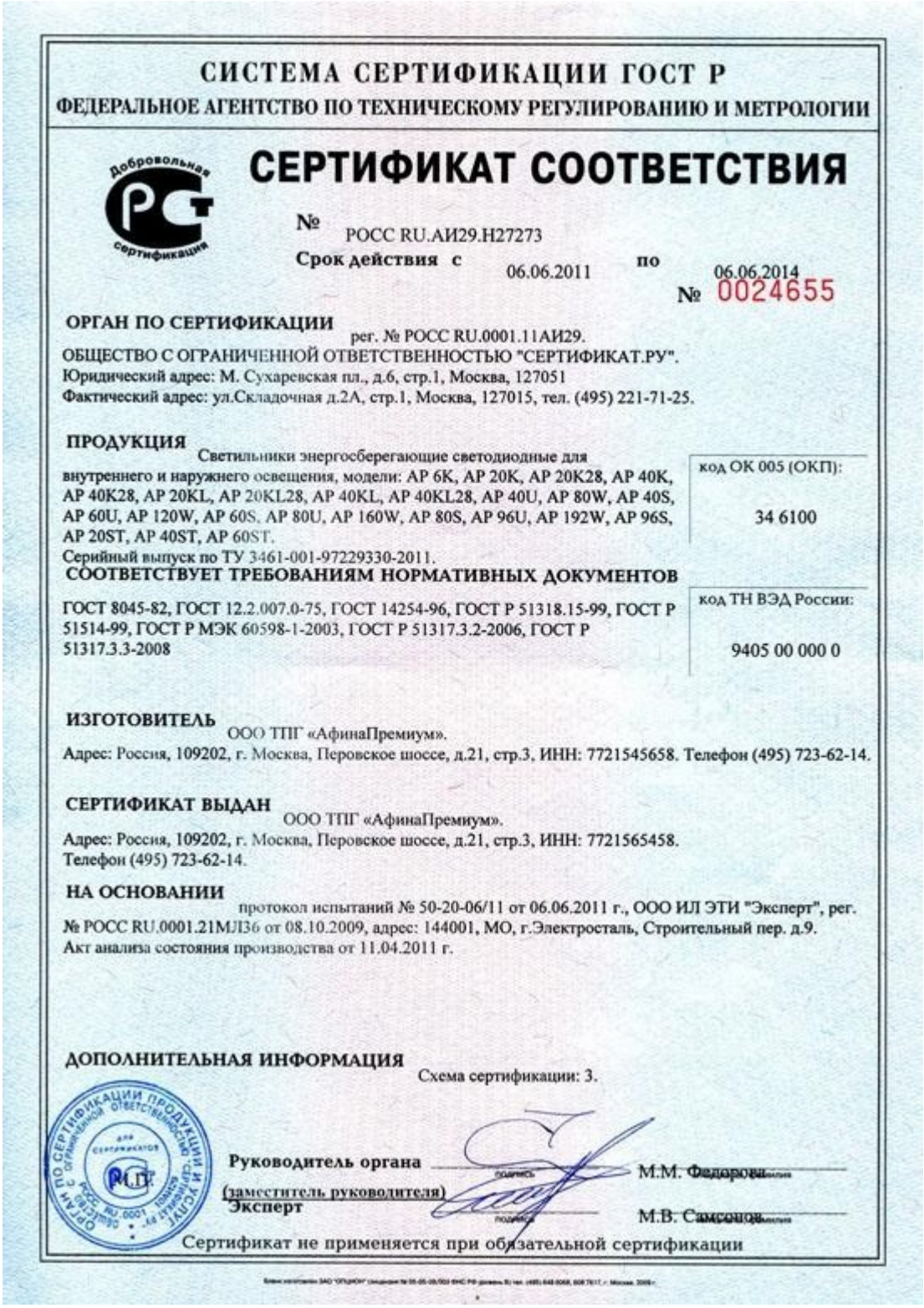
В данном пункте указывается код ОКП (Общероссийский классификатор продукции). В коде ОКП 6 цифр.

11. Код ТН ВЭД (расположен справа):

В данном пункте указывается код ТН ВЭД (Товарная номенклатура внешнеэкономической деятельности). В сертификатах наличие кода ТН ВЭД не обязательно. В коде ТН ВЭД 10 цифр.

2. ПРАКТИЧЕСКАЯ ЧАСТЬ

1. Рассмотреть приведенный ниже сертификат соответствия и провести его анализ, письменно ответив на вопросы.



СИСТЕМА СЕРТИФИКАЦИИ ГОСТ Р

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

Добровольная сертификация

СЕРТИФИКАТ СООТВЕТСТВИЯ

№ РОСС RU.АИ29.Н27273

Срок действия с 06.06.2011 по 06.06.2014

№ 0024655

ОРГАН ПО СЕРТИФИКАЦИИ рег. № РОСС RU.0001.11АИ29.
ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ "СЕРТИФИКАТ.РУ".
Юридический адрес: М. Сухаревская пл., д.6, стр.1, Москва, 127051
Фактический адрес: ул.Складочная д.2А, стр.1, Москва, 127015, тел. (495) 221-71-25.

ПРОДУКЦИЯ
Светильники энергосберегающие светодиодные для внутреннего и наружнего освещения, модели: АР 6К, АР 20К, АР 20К28, АР 40К, АР 40К28, АР 20KL, АР 20KL28, АР 40KL, АР 40KL28, АР 40U, АР 80W, АР 40S, АР 60U, АР 120W, АР 60S, АР 80U, АР 160W, АР 80S, АР 96U, АР 192W, АР 96S, АР 20ST, АР 40ST, АР 60ST.
Серийный выпуск по ТУ 3461-001-97229330-2011.

код ОК 005 (ОКП): 34 6100

СООТВЕТСТВУЕТ ТРЕБОВАНИЯМ НОРМАТИВНЫХ ДОКУМЕНТОВ
ГОСТ 8045-82, ГОСТ 12.2.007.0-75, ГОСТ 14254-96, ГОСТ Р 51318.15-99, ГОСТ Р 51514-99, ГОСТ Р МЭК 60598-1-2003, ГОСТ Р 51317.3.2-2006, ГОСТ Р 51317.3.3-2008

код ТН ВЭД России: 9405 00 000 0

ИЗГОТОВИТЕЛЬ ООО ТПГ «АфинаПремиум».
Адрес: Россия, 109202, г. Москва, Перовское шоссе, д.21, стр.3, ИНН: 7721545658. Телефон (495) 723-62-14.

СЕРТИФИКАТ ВЫДАН ООО ТПГ «АфинаПремиум».
Адрес: Россия, 109202, г. Москва, Перовское шоссе, д.21, стр.3, ИНН: 7721565458.
Телефон (495) 723-62-14.

НА ОСНОВАНИИ протокол испытаний № 50-20-06/11 от 06.06.2011 г., ООО ИЛ ЭТИ "Эксперт", рег. № РОСС RU.0001.21МЛ36 от 08.10.2009, адрес: 144001, МО, г.Электросталь, Строительный пер. д.9.
Акт анализа состояния производства от 11.04.2011 г.

ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ
Схема сертификации: 3.

Руководитель органа (заместитель руководителя) подпись М.М. Федоров
Эксперт подпись М.В. Самсонов

Сертификат не применяется при обязательной сертификации

1. Какой орган по сертификации выдал сертификат соответствия?
2. На какую продукцию выдан сертификат?
3. Какой срок действия сертификата?
4. Требованиям каких нормативных документов соответствует сертификат?
5. Кто является изготовителем продукции?
6. На основании чего выдан сертификат?

2. Ответить на контрольные вопросы письменно:

1. Что такое сертификат соответствия?
2. На основании какого документа орган по сертификации принимает решение о выдаче сертификата?
3. При каком условии сертификат действителен?
4. Чем маркируется продукция, на которую выдан сертификат?
5. Кто осуществляет маркирование продукции знаком соответствия?
6. Что является критериями для определения периодичности и объема инспекционного контроля?

Практическая работа №23
«Разработка политики безопасности корпоративной сети»

Цель: разработать политику безопасности предприятия.

Форма отчета:
−выполнить задание;
−показать преподавателю;
−ответить на вопросы преподавателя.
Время выполнения: 2 ч

Теоретические сведения

Политика безопасности трактуется как набор норм, правил и практических приемов, которые регулируют управление, защиту и распределение ценной информации. На практике политика безопасности трактуется несколько шире – как совокупность документированных административных решений, направленных на обеспечение безопасности информационного ресурса. Результатом политики является высокоуровневый документ, представляющий систематизированное изложение целей, задач, принципов и способов достижения информационной безопасности. Данный документ представляет методологическую основу практических мер (процедур) по реализации ОБИ и содержит следующие группы сведений.
1. Основные положения информационной безопасности.
2. Область применения.
3. Цели и задачи обеспечения информационной безопасности.
4. Распределение ролей и ответственности.
5. Общие обязанности.
Основные положения определяют важность ОБИ, общие проблемы безопасности, направления их решения, роль сотрудников, нормативно-правовые основы.
Областью применения политики безопасности являются основные активы и подсистемы АС, подлежащие защите. Типовыми активами являются программно-аппаратное и информационное обеспечение АС, персонал, в отдельных случаях – информационная инфраструктура предприятия.

Цели, задачи, критерии ОБИ вытекают из функционального назначения предприятия. Например, для режимных организаций на первое место ставится соблюдение конфиденциальности. Для сервисных информационных служб реального времени важным является обеспечение доступности (оперативной готовности) подсистем. Для информационных хранилищ актуальным может быть обеспечение целостности данных и т. д. Здесь указываются законы и правила организации, которые следует учитывать при проведении работ по ОБИ.
Типовыми целями могут быть следующие:
- обеспечение уровня безопасности, соответствующего нормативным документам предприятия;
- следование экономической целесообразности в выборе защитных мер;
- обеспечение соответствующего уровня безопасности в конкретных функциональных областях АС;
- обеспечение подотчетности всех действий пользователей с информационными ресурсами и анализа регистрационной информации;
- выработка планов восстановления после критических ситуаций и обеспечения непрерывности работы АС и др.
Если предприятие не является изолированным, цели и задачи рассматриваются в более широком контексте: должны быть оговорены вопросы безопасного взаимного влияния локальных и удаленных подсистем.
В рассматриваемом документе могут быть конкретизированы некоторые стратегические принципы безопасности (вытекающие из целей и задач ОБИ). Таковыми являются стратегии действий в случае нарушения политики безопасности предприятия и сторонних организаций, взаимодействия с внешними организациями, правоохранительными органами, прессой и др. В качестве примера можно привести две стратегии ответных действий на нарушение безопасности:
«выследить и осудить», когда злоумышленнику позволяют продолжить действия с целью его компрометации и наказания (данную стратегию одобряют правоохранительные органы!);
«защититься и продолжить», когда организация опасается за уязвимость информационных ресурсов и оказывает максимальное противодействие нарушению.

Практическое задание:

Вы являетесь работодателем. Вам необходимо разработать политику безопасности предприятия (согласно Вашему варианту).

Номер варианта	Предприятие
1	Поликлиника
2	Туристическое агентство

3	Колледж
4	Офис благотворительного фонда
5	Офис страховой компании
6	Издательство
7	Рекрутинговое агентство
8	Интернет-магазин
9	Консалтинговая фирма
10	Рекламное агентство

Порядок выполнения работы:
Ознакомиться с пунктами практической работы.
Оформите свой отчет.
Содержание отчета:
Название, цель работы, задание данной практической работы.
Номер варианта, условие задания своего варианта решение заданий.
Перечень контрольных вопросов.
Вывод о проделанной работе.

Выполните задание в соответствии со своим вариантом.
Сделайте вывод о проделанной работе.
Контрольные вопросы:
Дайте определение понятию объект политики безопасности.
Дайте определение понятию субъект политики безопасности.
Какое основное назначение политики информационной безопасности?

Практическая работа №24
«Получение сертификата»

Цель: Изучение нормативной базы регламентирующих документов оценки соответствия, изучение правил проведения сертификации, приобретение навыков заполнения бланков по сертификации.

Форма отчета:

−выполнить задание;
−показать преподавателю;
−ответить на вопросы преподавателя.

Время выполнения: 2 ч

Выполнить задания по следующим документам:

- ФЗ от 27.12.2002 №184 «О техническом регулировании».
- Постановление от 13.07.1997 №1013 «Об утверждении перечня товаров, подлежащих обязательной сертификации, и перечня работ и услуг, подлежащих обязательной сертификации».
- Постановление от 28.04.1999 «О правилах проведения сертификации пищевых продуктов и продовольственного сырья» (измен. Ред. 18.06.02 №43).
- Порядок сертификации услуг общественного питания.
- ФЗ-№88 от 12.06.2008 «Технический регламент на молоко и молочную продукцию».
- Постановление от 02.02.2001 №11 «Об утверждении и введении в действие правил по проведению сертификации парфюмерно-косметической продукции» (измен. Ред. 18.06.2002 №40).

Задание 1. Разработать стандарт организации (определить круг лиц, разрабатывающих стандарт, утверждающих, определить объекты стандартизации и требования, предъявляемые к этим объектам).

Задание 2. Решить ситуационную типовую задачу. Задачи прилагаются.

Задание 3. Разработать анкету опроса покупателей для выявления требований к качеству реального и прогнозируемого ассортимента (анкетирование проводится среди других участников команды). Проанализировать анкетные данные. После опроса разрабатывается реклама на новый товар.

Задание №4. Заполнить заявку на проведение оценки соответствия и акт отбора образцов (на парфюмерно-косметическую и молочную продукцию). Определить код по ОКП парфюмерно-косметической продукции.

Задание №5. Определить порядок проведения оценки соответствия предъявленной продукции. Составить схему порядка проведения оценки соответствия по разделу 3. (дать оценку ГМС по разработанной схеме).

Задание №6. Определить критерии инспекционного контроля по парфюмерно-косметическим товарам (п.5.3). Дать оценку ГМС по выполненному заданию.

Задание №7. Определить виды испытаний, предъявляемой на оценку соответствия продукции, по нормативным документам. Дать оценку ГМС по выполненному заданию.

Отчет оформить в виде ответов на задания в текстовом виде.

Список литературы

1. *Илюшечкин, В. М.* Основы использования и проектирования баз данных : учебник для среднего профессионального образования / В. М. Илюшечкин. — испр. и доп. — Москва : Издательство Юрайт, 2019. — 213 с. — (Профессиональное образование). — ISBN 978-5-534-01283-5. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://biblio-online.ru/bcode/437670

2. *Стасышин, В. М.* Базы данных: технологии доступа : учебное пособие для среднего профессионального образования / В. М. Стасышин, Т. Л. Стасышина. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 164 с. — (Профессиональное образование). — ISBN 978-5-534-09888-4. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://biblio-online.ru/bcode/442342

3. *Советов, Б. Я.* Базы данных : учебник для среднего профессионального образования / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2019. — 420 с. — (Профессиональное образование). — ISBN 978-5-534-09324-7. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://biblio-online.ru/bcode/438438

4. *Гордеев, С. И.* Организация баз данных в 2 ч. Часть 1 : учебник для среднего профессионального образования / С. И. Гордеев, В. Н. Волошина. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 310 с. — (Профессиональное образование). — ISBN 978-5-534-11626-7. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://biblio-online.ru/bcode/445767

5. *Нестеров, С. А.* Базы данных : учебник и практикум для среднего профессионального образования / С. А. Нестеров. — Москва : Издательство Юрайт, 2019. — 230 с. — (Профессиональное образование). — ISBN 978-5-534-11629-8. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://biblio-online.ru/bcode/445770