1. **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Сибирский государственный университет науки и технологий**

**имени академика М.Ф. Решетнева»**

**АЭРОКОСМИЧЕСКИЙ КОЛЛЕДЖ**

**Методические указания**

для проведения лабораторных /практических занятий/

МП 02 Разработка и администрирование баз данных

**МДК 02.02 Технология разработки и защиты баз данных**

основной профессиональной образовательной программы

по специальности среднего профессионального образования

**Для специальности**

09.02.03 «Программирование в компьютерных системах»

Красноярск, 2022

Разработчики:

Аэрокосмический колледж СибГУ им. М.Ф. Решетнева,

преподаватель О.С. Гвоздиевская

Рассмотрено на заседании ЦК

«\_\_\_\_\_» **\_\_\_\_\_\_\_\_\_\_\_\_\_\_**20\_\_\_ г. Протокол № **\_\_\_\_\_\_\_\_\_\_**

Председатель цикловой комиссии

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.А. Башанова

**Инструкционно-технологическая карта**

**для выполнения практические работы №1, 2**

**Тема:** Основы работы с СУБД.

**Наименование работы:** Сбор и анализ информации и проектирование базы данных в среде СУБД

**Цель:** Собрать и проанализировать информацию и спроектировать БД..

**Формируемые компоненты:** **ПК 11.1-11.6,ОК 1-11**

**Норма времени:**8 часов

**Оборудование и программное обеспечения:** ПК, MQ Visio.MQ Word

**Основные правила ТБ при выполнении работы:**

-плавно нажимайте на клавиши, не допуская резких ударов;

-не пользуйтесь клавиатурой, если не подключено напряжение;

-работайте на клавиатуре чистыми руками;

-никогда не пытайтесь самостоятельно устранить неисправность в работе аппаратуры;

Строго запрещается:

-трогать разъемы соединительных кабелей;

-прикасаться к питающим проводам и устройствам заземления;

-прикасаться к экрану и к тыльной стороне монитора, клавиатуры;

-включать и отключать аппаратуру без указания преподавателя;

-Класть диск, книги, тетради на монитор и клавиатуру;

-Работать во влажной одежде и влажными руками

**Литература:** Кумскова, И. А. Базы данных: учебник для СПО / И. А. Кумскова.- М.: КНОРУС, 2016.-488 с.

**Контрольные вопросы при допуске:**

1. Понятия БД?
2. Что такое объект в БД?
3. Что значит анализ предметной области?

**ЗАДАНИЕ ВЫПОЛНЕНИЯ ПРАКТИЧЕСКТХ РАБОТ**

*Практическая работа №1,2 выполняется в среде проектирования VISIO и в конце занятия сдается на проверку. После проверки будет выставлена оценка.*

**Выбор задания**

Выбрать из таблицы «**Варианты заданий для лаб.работы**» вариант задания, соответствующий **номеру студента в списке учебной группы**. Для всех последующих практических работ вариант остается неизменным. Каждому студенту предоставляется свой вариант предметной области (ПО), который он будет использовать в процессе выполнения всех практических работ.

**Анализ предметной области**.

На основании выбранного варианта привести: название предприятия, цель деятельности предприятия, структура предприятия, информационные потребностей пользователей (кратко).

**Описание основных сущностей ПО**.

Здесь следует привести описание основных сущностей (объектов) ПО. Отбор сущностей производится на основе анализа информационных потребностей. Необходимо привести таблицы описания сущностей (сущностей должно быть не менее 3-х)

Таблица 1.1. Список сущностей предметной области.

|  |  |  |
| --- | --- | --- |
| N  п.п. | Наименование сущности | Краткое описание |
|  |  |  |
|  |  |  |

Здесь же приводится отбор атрибутов (не менее 5-ти) для каждого экземпляра сущности. Отбираются только те атрибуты сущностей, которые необходимы для формирования ответов на регламентированные и непредусмотренные запросы. Для каждого объекта следует привести таблицы его атрибутов.

Таблица 1.2. Список атрибутов.

|  |  |  |
| --- | --- | --- |
| N  п.п. | Наименование  атрибута | Краткое описание |
|  |  |  |
|  |  |  |

На основе анализа информационных запросов следует выявить связи между сущностями. Для выявленных связей также нужно заполнить таблицу 1.3.

Таблица 1.3. Список связей ПО.

|  |  |  |  |
| --- | --- | --- | --- |
| N  п.п. | Наименование  связи | Сущности, участвующие в  связи | Краткое описание |
|  |  |  |  |
|  |  |  |  |

**Построение инфологической модели.**

На основании ранее выбранного варианта и таблиц 1.1-1.3:

* + описать классы объектов (сущностей) и их свойства,
  + расставить существующие связи между ними,
  + на основании табл. 1.3. в письменной форме обосновать типы связей (1:1, 1:М и т.д.).

При графическом построении ИЛМ следует придерживаться единого масштаба для всей схемы. Все прямоугольники, обозначающие классы объектов, должны быть одного размера. Аналогично, все ромбы с именами связей также должны иметь одинаковый размер.

**Построение даталогической модели.**

На основании ранее выбранного варианта и таблиц 1.1-1.3, инфологической модели и нормализации БД необходимо:

* + провести соответствие ключей для каждой таблицы 1.1-1.3,
  + заполнить для каждой таблицы БД форму, согласно табл. 1.4. Таблица 1.4.

Структура таблицы для даталогической модели.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| N  п.п. | Наименование реквизита | Иденти- фикатор | Тип | Длина | Формат изобра-  жения | Ограничения и комментарий |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**СОДЕРЖАНИЕ ОТЧЕТА**

1. Название и цель работы.
2. Словесный и схематический анализ предметной области (ПО), включая схему структуры предприятия.
3. Заполненные таблицы 1.1 - 1.3. с описанием основных сущностей ПО.
4. Инфологическая модель БД, согласно варианту.
5. Обоснование типов связи в инфологической модели данных.
6. Даталогическая модель БД (табл. 1.4.).

**ПРИМЕР ОФОРМЛЕНИЯ**

**Пример. Разработать базу данных «Учеба студентов». Решение.**

**Шаг первый**. **Анализ предметной области.**

Студенты учатся на одном из факультетов, возглавляемом деканатом, в функции которого входит контроль за учебным процессом. В учебном процессе участвуют преподаватели кафедр, административно относящиеся к одному из факультетов. Каждому факультету могут принадлежать несколько кафедр. Студенты кафедр организованные в группы.

Преподаватели кафедр характеризуются фамилией именем и отчеством, должностью, научным званием, ставкой и стажом работы, адресом проживания, возрастом.

Каждая кафедра читает определенный набор закрепленных за ней дисциплин. Каждая дисциплина характеризуется своим полным названием, указанием общего количества часов и формы контроля (зачет, экзамен).

В конце каждого семестра составляется экзаменационно-зачетные ведомости, в которых указываются дисциплины и для каких групп проводится форма контроля, фамилия преподавателя и учебный год и семестр. В каждой такой ведомости составляется список студентов и выставляется оценка.

**Шаг второй. Описание основных сущностей ПО.**

В результате проведенного анализа предметной области базы данных «Учеба студентов» легко перечислить основные сущности этой БД. Так как на физическом уровне сущности соответствует таблица, то просто перечислим основные таблицы БД.

В реляционную модель проектированной БД будут входить следующие таблицы (сущности): Факультет, Кафедра, Преподаватели, Группы, Студенты, Дисциплины, Ведомости.

**Список сущностей.**

|  |  |  |
| --- | --- | --- |
| **№** | **Название** | **Назначение** |
| 1 | Факультет | Описание факультета и его деканата |
| 2 | Кафедра | Описание кафедры |
| 3 | Преподаватели | Описание состава сотрудников кафедр |
| 4 | Группы | Перечень групп, закрепленных за каждой кафедрой |
| 5 | Студенты | Перечень студентов каждой группы |
| 6 | Дисциплины | Перечень дисциплин, закрепленных за каждой кафедрой |
| 7 | Ведомости | Экзаменнационно-зачетные ведомости с перечнем  студентов и их оценками |
| 8 | Подчиненная ведомость | Это таблица внутри таблицы ведомости. Отражает связь один-ко-многим. Так как каждая ведомость выписывается  каждой конкретной группе, а студентов в ней много. |

Для каждой таблицы (сущности) приведем описание ее атрибутов. Атрибут на физическом уровне – это колонки таблицы и выражает определенное свойство объекта.

**Список атрибутов таблицы «Факультеты»**

|  |  |  |
| --- | --- | --- |
| **Ключевое поле** | **Название** | **Назначение** |
| ПК (первичный ключ) | Код факультета | Ключевое поле, предназначенное для однозначной идентификации каждой записи в таблице. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждому факультету.  Это целое число. Т.е. для идентификации каждого факультета будет применятся не названия самих факультетов, а определенный номер. Этот номер  может быть случайным целым числом или счетчик по порядку. |
|  | ФИО декана |  |
|  | Номер комнаты  деканата |  |
|  | Телефон деканата |  |

**Список атрибутов таблицы «Кафедра»**

|  |  |  |
| --- | --- | --- |
| **Ключевое поле** | **Название** | **Назначение** |
| ПК  (первичный ключ) | Код кафедры | Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждой кафедре.  Однако для идентификации каждой кафедры первичного ключа недостаточно, так как каждая кафедра принадлежит определенному факультету.  Для этого будем использовать внешний ключ. |
| ВК  (внешний ключ) | Код факультета | Внешний ключ – это атрибут отношения, который является первичным ключом другого отношения. В нашем случае это атрибут таблицы факультеты. С помощью внешнего ключа будет определено к  какому факультету принадлежит каждая кафедра. |
|  | Название кафедры |  |
|  | ФИО заведующего |  |
|  | Номер комнаты  кафедры |  |
|  | Телефон кафедры |  |

**Список атрибутов таблицы «Преподаватели»**

|  |  |  |
| --- | --- | --- |
| **Ключевое поле** | **Название** | **Назначение** |
| ПК  (первичный ключ) | Код преподавателя | Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждому преподавателю. Это например, может быть его табельный номер.  Однако для идентификации каждого преподавателя первичного ключа недостаточно, так как каждый сотрудник принадлежит определенной кафедры. Для  этого будем использовать внешний ключ. |
| ВК  (внешний ключ) | Код кафедры | С помощью данного внешнего ключа будет определено к какой кафедры принадлежит каждый преподаватель. |
|  | ФИО |  |
|  | должность | Ассистент, доцент, процессор, ст. преподаватель |
|  | научное звание | К.т.н., проф., магистр, ст.н.с., м.н.с. |
|  | ставка |  |
|  | стаж работы, |  |
|  | адрес  проживания |  |
|  | возраст |  |

**Список атрибутов таблицы «Группы»**

|  |  |  |
| --- | --- | --- |
| **Ключевое**  **поле** | **Название** | **Назначение** |
| ПК  (первичный ключ) | Код группы | Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждой группе.  Однако для идентификации каждой группы первичного ключа недостаточно, так как каждая группа принадлежит определенной кафедре. Для этого будем  использовать внешний ключ. |
| ВК  (внешний ключ) | Код кафедры | С помощью данного внешнего ключа будет определено к какой кафедре принадлежит каждая группа. |
|  | Номер группы |  |
|  | Год поступления |  |
|  | Курс обучения |  |

**Список атрибутов таблицы «Студенты»**

|  |  |  |
| --- | --- | --- |
| **Ключевое поле** | **Название** | **Назначение** |
| К  (первичный ключ) | Код студента | Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждому студенту.  Однако для идентификации каждого студента первичного ключа недостаточно, так как каждый студент принадлежит определенной группе. Для этого  будем использовать внешний ключ. |
| ВК  (внешний ключ) | Код группы | С помощью данного внешнего ключа будет определено к какой группе принадлежит каждый студент. |
|  | ФИО |  |
|  | Год рождения |  |
|  | Адрес  проживания |  |

**Список атрибутов таблицы «Дисциплины»**

|  |  |  |
| --- | --- | --- |
| **Ключевое**  **поле** | **Название** | **Назначение** |
| ПК (первичный ключ) | Код дисциплины | Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждой дисциплине.  Однако для идентификации каждой дисциплины первичного ключа недостаточно, так как каждая  дисциплина принадлежит определенной кафедре. Для этого будем использовать внешний ключ. |
| ВК (внешний ключ) | Код кафедры | С помощью данного внешнего ключа будет определено к какой кафедре принадлежит каждая  дисциплина. |
|  | Название дисциплины |  |
|  | Расчасовка |  |
|  | Форма контроля |  |

**Список атрибутов таблицы «Ведомости»**

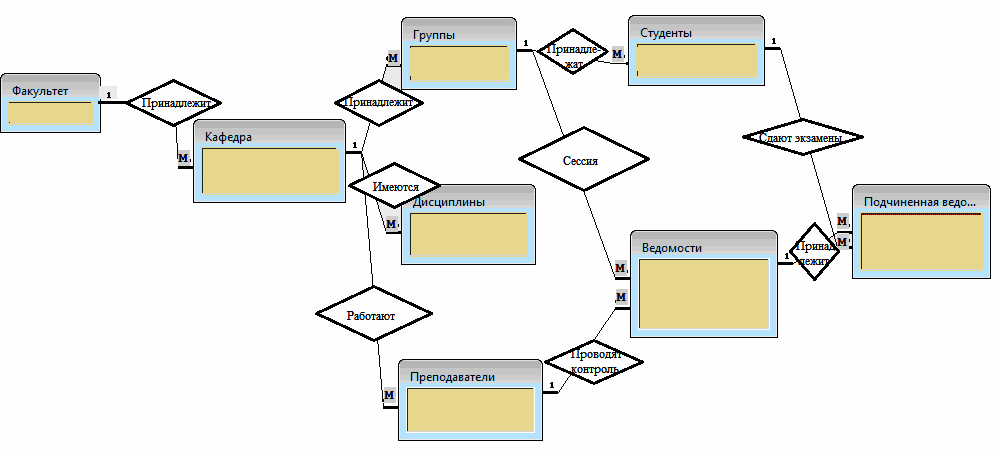
|  |  |  |
| --- | --- | --- |
| **Ключевое поле** | **Название** | **Назначение** |
| ПК  (первичный ключ) | Код ведомости | Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждой учебной ведомости.  Однако для идентификации каждой ведомости первичного ключа недостаточно, так как каждая ведомость выписывается для определенной учебной группе по определенной дисциплине и преподавателя.  Для этого будем использовать внешние ключи. |
| ВК  (внешний ключ) | Код группы | С помощью данного внешнего ключа будет определено для какой группы выписывается ведомость. |
| ВК  (внешний ключ) | Код дисциплины | С помощью данного внешнего ключа будет определено для какой дисциплины выписывается ведомость. |
| ВК  (внешний ключ) | Код преподавателя | С помощью данного внешнего ключа будет определено какому преподавателю выписывается ведомость. |
|  | Учебный год |  |
|  | Семестр |  |

**Список атрибутов таблицы «Подчиненная таблица Ведомости»**

|  |  |  |
| --- | --- | --- |
| **Ключевое поле** | **Название** | **Назначение** |
| ПК  (первичный ключ) | Код под\_ведомости | Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждой подведомости.  Однако для идентификации каждой подведомости первичного ключа недостаточно, так как каждая подведомость принадлежит определенной ведомости.  Для этого будем использовать внешний ключ. |
| ВК  (внешний ключ) | Код ведомости | С помощью данного внешнего ключа будет осуществлена связь с таблицей ведомости. |
| ВК  (внешний ключ) | Код студента | С помощью данного внешнего ключа будет определен студент |
|  | Оценка |  |

**Шаг третий.** Построение **инфологической модели**.

Инфологическую модель лучше представить графически, где будут изображены все таблицы и связи между ними. В нашем случае схема связей представлена на рисунке.



Для выявленных связей заполним таблицу

**Список связей.**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Название связи** | **Сущности, участвующие в связи** | **Назначение** |
| 1 | 1:М | Факультет-Кафедра | Одному факультету могут  принадлежать несколько кафедр |
| 2 | 1:М | Кафедра - Группа | Одной кафедре может принадлежать  несколько групп |
| 3 | 1:М | Кафедра - Дисциплины | Одной кафедре могут принадлежать  несколько читаемых дисциплин |
| 4 | 1:М | Кафедра - Преподаватели | На одной кафедре работает более  одного преподавателя |
| 5 | 1:М | Группа-Студенты | В каждой группе учится множество  студентов |
| 6 | 1:М | Группа - Ведомость | Каждой группе выписывают  несколько ведомостей |
| 7 | 1:М | Дисциплины - Ведомость | Ведомость выписывается из  множества дисциплин |
| 8 | 1:М | Преподаватели -  Ведомость | Ведомость выписывается  конкретному преподавателю |
| 9 | 1:М | Ведомость-Подчиненная  ведомость | Подчиненная ведомость принадлежит  одной конкретной ведомости |
| 10 | 1:М | Студенты-Подчиненная  ведомость | В подчиненной ведомости  перечислены все студенты группы |

Таблица 1. Варианты заданий для практической работы

|  |  |
| --- | --- |
| **№ варианта** | **Условие** |
| **Вариант №1** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – успеваемость студентов ВУЗА**. БД состоит из следующих таблиц: факультеты, кафедры, учебные группы, студенты, ведомости  успеваемости.  **Таблица факультеты** имеет следующие атрибуты: название факультета, ФИО декана, номер комнаты, номер корпуса, телефон.  **Таблица кафедра** имеет следующие атрибуты: название кафедры, факультет, ФИО заведующего, номер комнаты, номер корпуса, телефон, кол-во преподавателей.  **Таблица учебные группы** имеет следующие атрибуты: название группы, год поступления, курс обучения, кол-во студентов в группе.  **Таблица студенты** имеет следующие атрибуты: студента, фамилия, имя, отчество, группа, год рождения, пол, адрес, город, телефон.  **Таблица ведомости успеваемости** имеет следующие атрибуты: группа, студент, предмет, оценка. |
| **Вариант №2** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. **БД – информационная система супермаркета**. БД состоит из следующих таблиц: отделы, сотрудники, товары, продажа товаров, должности.   **Таблица отделы** имеет следующие атрибуты: название отдела, кол-во прилавков, кол-во продавцов, номер зала.  **Таблица сотрудники** имеет следующие атрибуты: фамилия, имя, отчество, отдел, год рождения, год поступления на работу, стаж, должность, пол, адрес, город, телефон.  **Таблица должности** имеет следующие атрибуты: название должности, сумма ставки.  **Таблица товары** имеет следующие атрибуты: название товара, отдел, страна производитель, условия хранения, сроки хранения .  **Таблица продажа товаров** имеет следующие атрибуты: сотрудника являющегося продавцом, товара дата, время, кол-во, цена, сумма. |
| **Вариант №3** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – информационная система военного округа**. БД состоит из следующих таблиц: места дислокации, вид войск, части, роты, личный состав.  **Таблица вид войск** имеет следующие атрибуты: название.  **Таблица места дислокации** имеет следующие атрибуты: страна, город, адрес, занимаемая площадь.  **Таблица части** имеет следующие атрибуты: номер части, место  дислокации, вид войск, кол-во рот.  **Таблица роты** имеет следующие атрибуты: название роты, кол-во служащих.  **Таблица личный состав** имеет следующие атрибуты: фамилия, рота, должность, год рождения, год поступления на службу, выслуга лет, награды, участие в военных мероприятиях. |
| **Вариант №4** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – информационная система библиотеки**. БД состоит из следующих таблиц: библиотеки, фонд библиотеки, тип литературы, сотрудники, пополнение фонда.  **Таблица библиотеки** имеет следующие атрибуты: название, адрес, город.  **Таблица фонд библиотеки** имеет следующие атрибуты: название фонда, библиотека, кол-во книг, кол-во журналов, кол-во газет, кол-во сборников, кол-во диссертаций, кол-во рефератов.  **Таблица тип литературы** имеет следующие атрибуты: название типа. **Таблица сотрудники** имеет следующие атрибуты: фамилия сотрудника, библиотека, должность, год рождения, год поступления на работу, образование, зарплата.  **Таблица пополнение фонда** имеет следующие атрибуты: фонд, сотрудник, дата, название источника литературы, тип литературы, издательство, дата издания, кол-во экземпляров. |
| **Вариант №5** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – информационная система туристического агентства**. БД состоит из следующих таблиц: пансионаты, туры, клиенты, путевки, вид жилья. **Таблица пансионаты** имеет следующие атрибуты: название пансионата, адрес, город, страна, телефон, описание территории, кол-во комнат, наличие бассейна, наличие медицинских услуг, наличие спа-салона, уровень пансионата, расстояние до моря.  **Таблица вид жилья** имеет следующие атрибуты: название (дом, бунгало, квартира, 1-я комната, 2-я комната и т.д.), категория жилья (люкс, полулюкс, и т.д.), пансионат, описание условий проживания, цена за номер в сутки.  **Таблица туры** имеет следующие атрибуты: название тура (Европа, средняя Азия, тибет и т.д.), вид транспорта, категория жилья на ночь(гостиница, отель, палатка и т.д.), вид питания (одноразовое, двухразовое, трехразовое, завтраки), цена тура в сутки.  **Таблица клиенты** имеет следующие атрибуты: фамилия, имя, отчество, паспортные данные, дата рождения, адрес, город, телефон.  **Таблица путевки** имеет следующие атрибуты: клиент, пансионата, вид  жилья, дата заезда, дата отъезда, наличие детей, наличие мед. страховки, кол-во человек, цена, сумма. |
| **Вариант №6** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – информационная система автопредприятия города**. БД состоит из следующих таблиц: автотранспорт, водители, маршруты, обслуживающий персонал, гаражное хозяйство.  **Таблица автотранспорт** имеет следующие атрибуты: название транспорта (автобусы, такси, маршрутные такси, прочий легковой транспорт, грузовой транспорт и т.д.), кол-во наработки, пробег, кол-во ремонтов, характеристика.  **Таблица маршруты** имеет следующие атрибуты: название маршрута, транспорт, водитель, график работы.  **Таблица водители** имеет следующие атрибуты: фамилия, имя, отчество, год рождения, год поступления на работу, стаж, должность, пол, адрес, город, телефон.  **Таблица обслуживающий персонал** имеет следующие атрибуты: должность (техники, сварщики, слесари, сборщики и др.), фамилия, имя, отчество, год рождения, год поступления на работу, стаж, пол, адрес, город, телефон.  **Таблица гаражное хозяйство** имеет следующие атрибуты: название гаража, транспорт на ремонте, вид ремонта, дата поступления, дата выдачи после ремонта, результат ремонта, персонал, производящего  ремонт. |
| **Вариант №7** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. **БД – информационная система поликлиники**. БД состоит из следующих таблиц: врачи, пациенты, история болезней, отделения, обслуживающий персонал.   **Таблица отделения** имеет следующие атрибуты: название отделения (хирургия, терапия, нервология и т.д.), этаж, номера комнат, ФИО заведующего.  **Таблица врачи** имеет следующие атрибуты: фамилия, имя, отчество,  должность, стаж работы, научное звание, адрес, номер отделения, в котором он работает.  **Таблица пациенты** имеет следующие атрибуты: фамилия, имя, отчество, адрес, город, возраст, пол.  **Таблица диагнозы** имеет следующие атрибуты: название диагноза, признаки болезни, период лечения, назначения.  **Таблица история болезни** имеет следующие атрибуты: пациент, врач, диагноз, лечение, дата заболевания, дата вылечивания, вид лечения (амбулаторное, стационарное). |
| **Вариант №8** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – информационная система больницы**. БД состоит из следующих таблиц: врачи, пациенты, история болезней, операции, лист лечения.  **Таблица врачи** имеет следующие атрибуты:, фамилия, имя, отчество, должность, стаж работы, научное звание, адрес.  **Таблица пациенты** имеет следующие атрибуты: фамилия, имя, отчество, адрес, город, возраст, пол.  **Таблица история болезни** имеет следующие атрибуты:, пациента врач, диагноз, дата заболевания, дата вылечивания, вид лечения (амбулаторное, стационарное), код операции.  **Таблица лист лечения** имеет следующие атрибуты: дата лечения, история болезни, лекарства, температура, давление, состояние больного (тяжелое, среднее, и т.д.).  **Таблица операции** имеет следующие атрибуты: описание операции  (удаление аппендицита, пластическая операция и т.д.), врач, дата операции, пациент, результат операции. |
| **Вариант №9** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – информационная система библиотек города**. БД состоит из следующих таблиц: библиотеки, читальные залы, литература, читатели, выдача лит-ры.  **Таблица библиотеки** имеет следующие атрибуты: название, адрес, город.  **Таблица читальные залы** имеет следующие атрибуты: название читального зала, библиотека, кол-во единиц лит-ры, кол-во посадочных мест, время работы, этаж, кол-во сотрудников.  **Таблица читатели** имеет следующие атрибуты: фамилия, имя, отчество,  категория читателя, место работы или обучения, возраст, дата регистрации в библиотеке.  **Таблица литература** имеет следующие атрибуты: название, категория литературы, авторы, издательство, год издательства, кол-во страниц, читальный зал.  **Таблица выдача литературы** имеет следующие атрибуты: читатель, литература, дата выдачи, срок выдачи, вид выдачи, наличие залога. |
| **Вариант**  **№10** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – информационная система автосалона**. БД состоит из следующих таблиц: автомобили, марка автомобиля, сотрудники, продажа автомобилей, покупатели.  **Таблица марка автомобиля** имеет следующие атрибуты: название марки, страна производитель, завод производитель, адрес.  **Таблица автомобиля** имеет следующие атрибуты: название автомобиля, марка, год производства, цвет, категория, цена.  **Таблица покупатели** имеет следующие атрибуты: фамилия, имя, отчество, паспортные данные, адрес, город, возраст, пол.  **Таблица сотрудника** имеет следующие атрибуты: фамилия, имя, отчество, стаж, зарплата.  **Таблица продажа автомобилей** имеет следующие атрибуты: дата, сотрудник, автомобиль, покупатель. |
| **Вариант**  **№11** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – успеваемость студентов кафедры**. БД состоит из следующих таблиц: кафедры, дисциплины, преподаватели, студенты, ведомости успеваемости.  **Таблица кафедра** имеет следующие атрибуты: название кафедры, факультет, ФИО заведующего, номер комнаты, номер корпуса, телефон, кол-во преподавателей.  **Таблица преподаватели** имеет следующие атрибуты: фамилия, имя, отчество, кафедра, год рождения, год поступления на работу, стаж, должность, пол, адрес, город, телефон.  **Таблица студенты** имеет следующие атрибуты: фамилия, имя, отчество, кафедра, год рождения, пол, адрес, город, телефон.  **Таблица дисциплины** имеет следующие атрибуты: название  дисциплины, кафедра, читаемой эту дисциплину, кол-во часов, вид итогового контроля.  **Таблица ведомости успеваемости** имеет следующие атрибуты:  преподаватель, дисциплина, студент, оценка. |
| **Вариант**  **№12** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – торговая организация**. БД состоит из следующих таблиц: торговая организация, торговая точка, продавцы, поставщики, заказы поставщикам.  **Таблица торговая организация** имеет следующие атрибуты: название торговой организации, адрес, ФИО директора, налоговый номер.  **Таблица торговая точка** имеет следующие атрибуты: название торговой точки, тип (универмаги, магазины, киоски, лотки и т.д.), торговая организация, адрес, ФИО заведующего.  **Таблица продавцы** имеет следующие атрибуты: фамилия, имя, отчество, торговая точка, должность, год рождения, пол, адрес проживания, город. **Таблица поставщики** имеет следующие атрибуты: название поставщика, тип деятельности, страна, город, адрес.  **Таблица заказы поставщикам** имеет следующие атрибуты: дата заказа, торговая точка, поставщик, название товара, кол-во, цена. |
| **Вариант**  **№13** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. **БД – проектная организация**. БД состоит из следующих таблиц: отделы, сотрудники, организации, договора, проектные работы.   **Таблица отделы** имеет следующие атрибуты: название отдела, этаж, телефон, начальник отдела.  **Таблица сотрудники** имеет следующие атрибуты: ФИО, должность (конструкторы, инженеры, техники, лаборанты, прочий обслуживающий персонал), номер отдела, в котором работает, пол, адрес, дата рождения. **Таблица организации** имеет следующие атрибуты: название организации, тип деятельности, страна, город, адрес, ФИО директора.  **Таблица договора** имеет следующие атрибуты: номер договора, дата заключения договора, организация, стоимость договора.  **Таблица проектные работы** имеет следующие атрибуты: дата начала проектной работы, дата завершения проектной работы, номер договора, отдел, осуществляющий разработку. |
| **Вариант** | На основании выбранного варианта выполнить следующее: |
| **№14** | 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – информационная система военно-морского флота**. БД состоит из следующих таблиц: базы, части, личный состав, корабли, учения.  **Базы военно-морского флота** имеет следующие атрибуты: название базы, георгафическое расположение, кол-во частей.  **Таблица части** имеет следующие атрибуты: номер части, база флота, место базирования, вид войск (морская авиация, морская пехота и т.д.). **Таблица личный состав** имеет следующие атрибуты: фамилия, часть, должность, год рождения, год поступления на службу, выслуга лет, награды,  **Таблица корабли** имеет следующие атрибуты: идентификационный номер корабля, название корабля, тип корабля, дата создания, наработка, кол-во посадочных мест, устройство двигателя (п[арусное](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D1%83%D1%81%D0%BD%D0%BE%D0%B5_%D1%81%D1%83%D0%B4%D0%BD%D0%BE), [гребное](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B5%D0%B1%D0%BD%D0%BE%D0%B5_%D1%81%D1%83%D0%B4%D0%BD%D0%BE), [пароход](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%BE%D1%85%D0%BE%D0%B4), [теплоход](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BF%D0%BB%D0%BE%D1%85%D0%BE%D0%B4), [турбоход](https://ru.wikipedia.org/wiki/%D0%A2%D1%83%D1%80%D0%B1%D0%BE%D1%85%D0%BE%D0%B4), и т.д. ), тип привода ([самоходное](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%BC%D0%BE%D1%85%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D1%81%D1%83%D0%B4%D0%BD%D0%BE), [несамоходное](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D1%81%D0%B0%D0%BC%D0%BE%D1%85%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D1%81%D1%83%D0%B4%D0%BD%D0%BE)), размещение [корпуса](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D0%BF%D1%83%D1%81_%D1%81%D1%83%D0%B4%D0%BD%D0%B0) (п[одводная лодка](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B4%D0%B2%D0%BE%D0%B4%D0%BD%D0%B0%D1%8F_%D0%BB%D0%BE%D0%B4%D0%BA%D0%B0), [ныряющее](https://ru.wikipedia.org/w/index.php?title=%D0%9D%D1%8B%D1%80%D1%8F%D1%8E%D1%89%D0%B5%D0%B5_%D1%81%D1%83%D0%B4%D0%BD%D0%BE&action=edit&redlink=1), [полупогружное](https://ru.wikipedia.org/w/index.php?title=%D0%9F%D0%BE%D0%BB%D1%83%D0%BF%D0%BE%D0%B3%D1%80%D1%83%D0%B6%D0%BD%D0%BE%D0%B5_%D1%81%D1%83%D0%B4%D0%BD%D0%BE&action=edit&redlink=1), и т.д.)  **Таблица учения**: часть, корабль, дата учения, место проведения, оценка. |
| **Вариант**  **№15** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – туристическая фирма**. БД состоит из следующих таблиц: туристы, туристичексая группа, состав групп, гостиницы, ведомости продажей.  **Таблица туристы** имеет следующие атрибуты: ФИО, паспортные данные, пол, возраст, дети.  **Таблица туры** имеет следующие атрибуты: название, страна, города, тип передвижения, тип питания, цена тура, тип проживания.  **Таблица туристическая группа** имеет следующие атрибуты: название, дата отправления, дата прибытия, тур, кол-во туристов.  **Таблица состав групп** имеет следующие атрибуты: дата продажи, турист, группа, цена билета.  **Таблица гостиницы** имеет следующие атрибуты: название гостиницы, страна, город, адрес, кол-во мест, тип гостиницы.  **Таблица ведомость продажей** имеет следующие атрибуты: дата, туристическая группа, гостиница, общая стоимость. |
| **Вариант**  **№16** | На основании выбранного варианта выполнить следующее:  1. Выполнить анализ предметной области исследуемой организации;   1. Описать основные сущности предметной области; 2. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 3. Построить инфологическую модель базы данных организации; 4. Построить даталогическую модель базы данных организации.   **БД – цирк**. БД состоит из следующих таблиц: работники цирка, представления, расписание гастролей, труппа цирка, программа цирка. **Таблица работники цирка** имеет следующие атрибуты: фамилия, имя, отчество, год рождения, год поступления на работу, стаж, должность (акробат, клоун, гимнаст, музыкант, постановщик, служащий и т.д.), пол, адрес, город, телефон.  **Таблица представления** имеет следующие атрибуты: название, режиссер-постановщик, художник-постановщик, дирижер-постановщик, автор, жанр, тип.  **Таблица расписание гастролей** имеет следующие атрибуты: представление, дата начала, дата окончания, места проведения гастроли. **Таблица труппа представления цирка** имеет следующие атрибуты: представление, актер цирка, название роли.  **Таблица программа цирка** имеет следующие атрибуты: представление, дата премьеры, период проведения, дни и время, цена билета. |
| **Вариант**  **№17** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – аптека**. БД состоит из следующих таблиц: лекартства, покупатели, продавцы, рецепты, продажа лекарств.  **Таблица лекарства** имеет следующие атрибуты: название, тип (готовое, изготовляемое), вид (таблетки, мази, настойки), цена.  **Таблица покупатели** имеет следующие атрибуты: фамилия, имя, отчество, адрес, город, телефон.  **Таблица продавцы** имеет следующие атрибуты: фамилия, имя, отчество, дата поступления, дата рождения, образование.  **Таблица рецепты** имеет следующие атрибуты: номер рецепта, дата выдачи, ФИО больного (покупатель), ФИО врача, диагноз пациента. **Таблица продажа лекарств** имеет следующие атрибуты: дата, лекарство, кол-во, рецепт, продавец. |
| **Вариант**  **№18** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области;   Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние  ключи между сущностями;   1. Построить инфологическую модель базы данных организации; 2. Построить даталогическую модель базы данных организации.   **БД – городская телефонная сеть**. БД состоит из следующих таблиц: АТС, абонент, ведомость звонков, прайс АТС, ведомость абонентской платы.  **Таблица АТС** имеет следующие атрибуты: название АТС, вид (городские, ведомственные и учрежденческие), адрес, город, кол-во абонентов.  **Таблица абоненты** имеет следующие атрибуты: фамилия, имя, отчество, вид телефона (основной, параллельный или спаренный), номер телефона, межгород (открыт/закрыт), льгота (да/нет), адрес: индекс, район, улица, дом, квартира.  **Таблица ведомость звонков** имеет следующие атрибуты: абонент, дата звонка, время начала, время окончания, межгород (да/нет).  **Таблица прайс АТС** имеет следующие атрибуты: АТС, цена на городские, цена на межгород.  **Таблица ведомость абонентской платы** имеет следующие атрибуты: абонент, месяц, год, кол-во минут на городские, кол-во минут на межгород, стоимость, сумма льготы, общая стоимость. |
| **Вариант**  **№19** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – аэропорт**. БД состоит из следующих таблиц: работники аэропорта, расписание вылетов, самолеты, бригады самолетов, ведомость продаж билетов.  **Таблица работники аэропорта** имеет следующие атрибуты: фамилия, имя, отчество, год рождения, год поступления на работу, стаж, должность (пилотов, диспетчеров, техников, кассиров, работников службы безопасности, справочной службы и других,), пол, адрес, город, телефон. **Таблица расписание вылетов** имеет следующие атрибуты: самолет, дата вылета, время вылета, место выбытия, место прибытия, маршрут (начальный и конечный пункты назначения, пункт пересадки), стоимость билета.  **Таблица самолеты** имеет следующие атрибуты: номер, год выпуска, кол-во посадочных место, грузоподъемность.  **Таблица бригады самолетов** имеет следующие атрибуты: номер бригады, самолет, работник аэропорта (пилоты, техники и обслуживающий персонал)ю  **Таблица ведомость продажи билетов** имеет следующие атрибуты: дата и время продажи, ФИО пассажира, паспортные данные, номер рейса, кол- во билетов, наличие льгот (пенсионеры, дети-сироты и т.д.), багаж (да/нет), стоимость. |
| **Вариант**  **№20** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – театр**. БД состоит из следующих таблиц: работники театра, спектакли, расписание гастролец, труппа спектакля, репертуар театра. **Таблица работники театра** имеет следующие атрибуты: фамилия, имя, отчество, год рождения, год поступления на работу, стаж, должность (актеров, музыкантов, постановщиков и служащих), пол, адрес, город, телефон.  **Таблица спектакли** имеет следующие атрибуты: название, режисер- постановщик, художник-постановщик, дирижер-постановщик, автор, жанр (музыкальная комедия, трагедия, оперетта и пр), тип (детские, молодежные и пр.).  **Таблица расписание гастролей** имеет следующие атрибуты: название, дата начала, дата окончания, места проведения гастроли, спектакль.  **Таблица труппа спектакля** имеет следующие атрибуты: спектакль, актер, название роли.  **Таблица репертуар театра** имеет следующие атрибуты: спектакль, дата премьеры, период проведения, дни и время, цена билета. |
| **Вариант**  **№21** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – железнодорожный вокзал**. БД состоит из следующих таблиц: работники ж.д.вокзала, расписание движения поездов, поезда, бригады поездов, ведомость продаж билетов.  **Таблица работники ж.д.вокзала** имеет следующие атрибуты: фамилия, имя, отчество, год рождения, год поступления на работу, стаж, должность (машинист, диспетчеров, проводник, ремонтников подвижного состава, путей, кассиров, работников службы подготовки составов, справочной службы и других,), пол, адрес, город, телефон.  **Таблица расписание движения поездов** имеет следующие атрибуты: поезд, дата отправления, время отправления, место отправления, дата прибытия, время прибытия, место прибытия, маршрут ((начальный и конечный пункты назначения, основные узловые станции), стоимость билета.  **Таблица поезда** имеет следующие атрибуты: номер, год выпуска, кол-во вагонов, тип поезда (общий, скоростной, высокоскоростной).  **Таблица бригады поездов** имеет следующие атрибуты: номер бригады, поезд, работник ж.д.вокзала (машинисты, техники, проводники и обслуживающий персонал).  **Таблица ведомость продажи билетов** имеет следующие атрибуты: дата и время продажи, ФИО пассажира, паспортные данные, номер рейса, кол- во билетов, наличие льгот (пенсионеры, дети-сироты и т.д.), стоимость. |
| **Вариант**  **№22** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – информационная система ВУЗА**. БД состоит из следующих таблиц: факультеты, кафедры, преподаватели, дисциплины, учебная нагрузка.  **Таблица факультеты** имеет следующие атрибуты: название факультета, ФИО декана, номер комнаты, номер корпуса, телефон.  **Таблица кафедра** имеет следующие атрибуты: название кафедры, ФИО заведующего, номер комнаты, номер корпуса, телефон, кол-во преподавателей.  **Таблица дисциплины** имеет следующие атрибуты: название дисциплины, кол-во часов, цикл дисциплин.  **Таблица преподаватели** имеет следующие атрибуты: фамилия, имя, отчество, кафедра, год рождения, год поступления на работу, стаж, должность, пол, город.  **Таблица учебная нагрузка** имеет следующие атрибуты: преподаватель,  дисциплина, учебный год, семестр, группы, кол-во студентов, вид итогового контроля. |
| **Вариант**  **№23** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – информационная система военного округа**. БД состоит из следующих таблиц: места дислокации, вид войск, части, роты, личный состав.  **Таблица вид войск** имеет следующие атрибуты: название вида войск. **Таблица места дислокации** имеет следующие атрибуты: страна, город, адрес, занимаемая площадь, кол-во сооружений.  **Таблица части** имеет следующие атрибуты: номер части, место дислокации, вид войск, кол-во рот, кол-во техники, кол-во вооружений. **Таблица техника** имеет следующие атрибуты: название техники, часть, характеристики.  **Таблица вооружения** имеет следующие атрибуты: название вооружения,  часть, характеристики. |
| **Вариант**  **№24** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. **БД – информационная система супермаркета**. БД состоит из следующих таблиц: отделы, клиенты, товары, продажа товаров, поставщики.   **Таблица отделы** имеет следующие атрибуты: название отдела, кол-во прилавков, кол-во продавцов, номер зала.  **Таблица клиенты** имеет следующие атрибуты: название клиента, адрес, вид оплаты.  **Таблица поставщики** имеет следующие атрибуты: название поставщика, адрес, страна, вид транспорта, вид оплаты.  **Таблица товары** имеет следующие атрибуты: название товара, отдел, поставщик, условия хранения, сроки хранения .  **Таблица продажа товаров** имеет следующие атрибуты: клиент, товар, дата, время, кол-во, цена, сумма. |
| **Вариант**  **№25** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – информационная система больницы**. БД состоит из следующих таблиц: врачи, пациенты, история болезней, отделения, лист лечения.  **Таблица отделения** имеет следующие атрибуты: название отделения (хирургия, терапия, нервология и т.д.), этаж, номера комнат, ФИО заведующего.  **Таблица врачи** имеет следующие атрибуты: фамилия, имя, отчество, должность, стаж работы, научное звание, адрес.  **Таблица пациенты** имеет следующие атрибуты: фамилия, имя, отчество, адрес, город, возраст, пол.  **Таблица история болезни** имеет следующие атрибуты: пациент, врач, диагноз, дата заболевания, дата вылечивания, вид лечения (амбулаторное, стационарное).  **Таблица лист лечения** имеет следующие атрибуты: дата лечения,  история болезни, лекарства, температура, давление, состояние больного (тяжелое, среднее, и т.д.). |

**Инструкционно-технологическая карта**

**для выполнения практической работы №3**

**Тема:** **Основы хранения и обработки данных. Проектирование БД.**

**Наименование работы: Приведение БД к нормальной форме 3НФ**

**Цель:** Приведения БД к нормальной формы ЗНФ.

**Формируемые компоненты:** **ПК 11.1-11.6,ОК 1-11**

**Норма времени:**4 часа

**Оборудование и программное обеспечения:** ПК, MQ Visio.MQ Word

**Основные правила ТБ при выполнении работы:**

-плавно нажимайте на клавиши, не допуская резких ударов;

-не пользуйтесь клавиатурой, если не подключено напряжение;

-работайте на клавиатуре чистыми руками;

-никогда не пытайтесь самостоятельно устранить неисправность в работе аппаратуры;

Строго запрещается:

-трогать разъемы соединительных кабелей;

-прикасаться к питающим проводам и устройствам заземления;

-прикасаться к экрану и к тыльной стороне монитора, клавиатуры;

-включать и отключать аппаратуру без указания преподавателя;

-Класть диск, книги, тетради на монитор и клавиатуру;

-Работать во влажной одежде и влажными руками **Литература:** Кумскова, И. А. Базы данных: учебник для СПО / И. А. Кумскова.- М.: КНОРУС, 2016.-488 с.

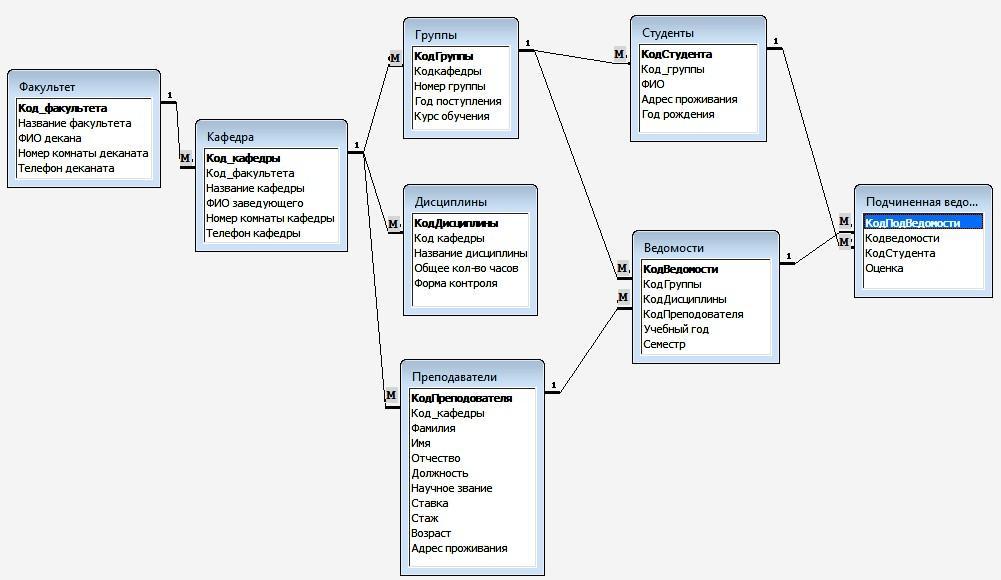
**Контрольные вопросы при допуске:**

1. Понятия БД?
2. Что такое объект в БД?
3. Что значит анализ предметной области?

**Задание:**

Даталогическая модель отражается графически в виде схемы базы данных, где указываются имена сущностей, их атрибуты и связи между сущностями.

В нашем случае схема связей представлена на рисунке.



Даталогическая модель БД представляется в виде набора таблиц специальной формы, в которых указываются наименование атрибута, идентификатор, тип, длина, формат, ограничения.

**Таблица «Факультеты»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Название** | **Идентификатор** | **Тип** | **Не пусто** | **Ограничение** |
| 1 | Код  факультета | Kod\_fakulteta | Числовой | Да | ПК (первичный  ключ) |
| 2 | Название  факультета | Name\_fakulteta | Текстовый | Нет |  |
| 3 | ФИО  декана | FIO | Текстовый | нет |  |
| 4 | Номер комнатыдеканата | N\_komnatu\_dekanata | Текстовый | Нет | Например, 123/а |
| 5 | Телефон  деканата | Telefon\_dekanata | Текстовый | Нет | Например, 41-69-99 |

**Список атрибутов таблицы «Кафедра»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Название** | **Идентификатор** | **Тип** | **Не пусто** | **Ограничение** |
| 1 | Код кафедры | Kod\_kafedru | Числовой | Да | ПК(первичный ключ) |
| 2 | Код  факультета | Kod\_fakulteta | Числовой | Да | ВК (внешний ключ) |
| 3 | Название  кафедры | Name\_kafedru | Текстовый |  |  |
| 4 | ФИО  заведующего | FIO | Текстовый | нет |  |
| 5 | Номер комнаты кафедры | N\_komnatu\_kafedru | Текстовый | Нет | Например, 123/а |
| 6 | Телефон кафедры | Telefon\_kafedru | Текстовый | Нет | Например, 41-69-99 |

**Список атрибутов таблицы «Преподаватели»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Название** | **Идентификатор** | **Тип** | **Не пусто** | **Ограничение** |
| 1 | Код  преподавателя | Kod\_prepodavately | Числовой | Да | ПК(первичный ключ) |
| 2 | Код кафедры | Kod\_kafedru | Числовой | Да | ВК (внешний ключ) |
| 3 | ФИО | FIO | Текстовый | Нет |  |
| 4 | должность | Dolgnost | Текстовый | Нет |  |
| 5 | научное  звание | Zvanie | Текстовый | Нет |  |
| 6 | ставка | Stavka | Числовой | Нет | Вещественное число Например, 0.5, 0.75,  1 |
| 7 | стаж работы, | Stag | Числовой | Нет | Вещественное число |
| 8 | адрес  проживания | Address | Текстовый | Нет |  |
| 9 | возраст | Vozrast | Числовой | нет |  |

**Список атрибутов таблицы «Группы»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Название** | **Идентификатор** | **Тип** | **Не пусто** | **Ограничение** |
| 1 | Код группы | Kod\_grupu | Числовой | Да | ПК (первичный ключ) |
| 2 | Код  кафедры | Kod\_kafedru | Числовой | Да | ВК (внешний ключ) |
| 3 | Номер  группы | N\_grupu | Текстовый | Нет | Например, МТ-461 |
| 4 | Год  поступления | God\_post | Числовой | нет |  |
| 5 | Курс обучения | Kurs | Числовой | Нет | Вычисляемое поле, как разность между текущей датой и  годом поступления |

**Список атрибутов таблицы «Студенты»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Название** | **Идентификатор** | **Тип** | **Не пусто** | **Ограничение** |
| 1 | Код  студента | Kod\_studenta | Числовой | Да | ПК (первичный ключ) |
| 2 | Код группы | Kod\_grupu | Числовой | Да | ВК (внешний ключ) |
| 3 | ФИО | FIO | Текстовый | Нет |  |
| 4 | Год  рождения | God\_rogdeniya | Числовой | нет |  |
| 5 | Адрес  проживания | Address | Текстовый | Нет |  |

**Список атрибутов таблицы «Дисциплины»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Название** | **Идентификатор** | **Тип** | **Не пусто** | **Ограничение** |
| 1 | Код  дисциплины | Kod\_disciplinu | Числовой | Да | ПК (первичный ключ) |
| 2 | Код  кафедры | Kod\_kafedru | Числовой | Да | ВК (внешний ключ) |
| 3 | Название  дисциплины | Name\_dis | Текстовый | Нет |  |
| 4 | Расчасовка | Raschasovka | Числовой | нет |  |
| 5 | Форма  контроля | Kontrol | Текстовый | Нет | Два значения –  экзамен или зачет |

**Список атрибутов таблицы «Ведомости»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Название** | **Идентификатор** | **Тип** | **Не пусто** | **Ограничение** |
| 1 | Код  ведомости | Kod\_vedomopsti | Числовой | Да | ПК (первичный  ключ) |
| 2 | Код группы | Kod\_grupu | Числовой | Да | ВК (внешний ключ) |
| 3 | Код  дисциплины | Kod\_disciplinu | Числовой | Да | ВК (внешний ключ) |
| 4 | Код  преподавателя | Kod\_prepodavately | Числовой | Да | ВК (внешний ключ) |
| 5 | Учебный год | God | Числовой | Нет |  |
| 6 | Семестр | Semester | Числовой | Нет | Диапазон от 1-10 |

**Список атрибутов таблицы «Подчиненная таблица Ведомости»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Название** | **Идентификатор** | **Тип** | **Не пусто** | **Ограничение** |
| 1 | Код  под\_ведомости | Kod\_pod\_vedomopsti | Числовой | Да | ПК (первичный  ключ) |
| 2 | Код ведомости | Kod\_ edomopsti | Числовой | Да | ВК (внешний ключ) |
| 3 | Код студента | Kod\_studenta | Числовой | Да | ВК (внешний ключ) |
| 4 | Оценка | Osenka | Числовой | Нет | Диапазон от 0-12 |

Таблица 1. Варианты заданий

|  |  |
| --- | --- |
| **№ варианта** | **Условие** |
| **Вариант №1** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – успеваемость студентов ВУЗА**. БД состоит из следующих таблиц: факультеты, кафедры, учебные группы, студенты, ведомости  успеваемости.  **Таблица факультеты** имеет следующие атрибуты: название факультета, ФИО декана, номер комнаты, номер корпуса, телефон.  **Таблица кафедра** имеет следующие атрибуты: название кафедры, факультет, ФИО заведующего, номер комнаты, номер корпуса, телефон, кол-во преподавателей.  **Таблица учебные группы** имеет следующие атрибуты: название группы, год поступления, курс обучения, кол-во студентов в группе.  **Таблица студенты** имеет следующие атрибуты: студента, фамилия, имя, отчество, группа, год рождения, пол, адрес, город, телефон.  **Таблица ведомости успеваемости** имеет следующие атрибуты: группа, студент, предмет, оценка. |
| **Вариант №2** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. **БД – информационная система супермаркета**. БД состоит из следующих таблиц: отделы, сотрудники, товары, продажа товаров, должности.   **Таблица отделы** имеет следующие атрибуты: название отдела, кол-во прилавков, кол-во продавцов, номер зала.  **Таблица сотрудники** имеет следующие атрибуты: фамилия, имя, отчество, отдел, год рождения, год поступления на работу, стаж, должность, пол, адрес, город, телефон.  **Таблица должности** имеет следующие атрибуты: название должности, сумма ставки.  **Таблица товары** имеет следующие атрибуты: название товара, отдел, страна производитель, условия хранения, сроки хранения .  **Таблица продажа товаров** имеет следующие атрибуты: сотрудника являющегося продавцом, товара дата, время, кол-во, цена, сумма. |
| **Вариант №3** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – информационная система военного округа**. БД состоит из следующих таблиц: места дислокации, вид войск, части, роты, личный состав.  **Таблица вид войск** имеет следующие атрибуты: название.  **Таблица места дислокации** имеет следующие атрибуты: страна, город, адрес, занимаемая площадь.  **Таблица части** имеет следующие атрибуты: номер части, место  дислокации, вид войск, кол-во рот.  **Таблица роты** имеет следующие атрибуты: название роты, кол-во служащих.  **Таблица личный состав** имеет следующие атрибуты: фамилия, рота, должность, год рождения, год поступления на службу, выслуга лет, награды, участие в военных мероприятиях. |
| **Вариант №4** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – информационная система библиотеки**. БД состоит из следующих таблиц: библиотеки, фонд библиотеки, тип литературы, сотрудники, пополнение фонда.  **Таблица библиотеки** имеет следующие атрибуты: название, адрес, город.  **Таблица фонд библиотеки** имеет следующие атрибуты: название фонда, библиотека, кол-во книг, кол-во журналов, кол-во газет, кол-во сборников, кол-во диссертаций, кол-во рефератов.  **Таблица тип литературы** имеет следующие атрибуты: название типа. **Таблица сотрудники** имеет следующие атрибуты: фамилия сотрудника, библиотека, должность, год рождения, год поступления на работу, образование, зарплата.  **Таблица пополнение фонда** имеет следующие атрибуты: фонд, сотрудник, дата, название источника литературы, тип литературы, издательство, дата издания, кол-во экземпляров. |
| **Вариант №5** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – информационная система туристического агентства**. БД состоит из следующих таблиц: пансионаты, туры, клиенты, путевки, вид жилья. **Таблица пансионаты** имеет следующие атрибуты: название пансионата, адрес, город, страна, телефон, описание территории, кол-во комнат, наличие бассейна, наличие медицинских услуг, наличие спа-салона, уровень пансионата, расстояние до моря.  **Таблица вид жилья** имеет следующие атрибуты: название (дом, бунгало, квартира, 1-я комната, 2-я комната и т.д.), категория жилья (люкс, полулюкс, и т.д.), пансионат, описание условий проживания, цена за номер в сутки.  **Таблица туры** имеет следующие атрибуты: название тура (Европа, средняя Азия, тибет и т.д.), вид транспорта, категория жилья на ночь(гостиница, отель, палатка и т.д.), вид питания (одноразовое, двухразовое, трехразовое, завтраки), цена тура в сутки.  **Таблица клиенты** имеет следующие атрибуты: фамилия, имя, отчество, паспортные данные, дата рождения, адрес, город, телефон.  **Таблица путевки** имеет следующие атрибуты: клиент, пансионата, вид  жилья, дата заезда, дата отъезда, наличие детей, наличие мед. страховки, кол-во человек, цена, сумма. |
| **Вариант №6** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – информационная система автопредприятия города**. БД состоит из следующих таблиц: автотранспорт, водители, маршруты, обслуживающий персонал, гаражное хозяйство.  **Таблица автотранспорт** имеет следующие атрибуты: название транспорта (автобусы, такси, маршрутные такси, прочий легковой транспорт, грузовой транспорт и т.д.), кол-во наработки, пробег, кол-во ремонтов, характеристика.  **Таблица маршруты** имеет следующие атрибуты: название маршрута, транспорт, водитель, график работы.  **Таблица водители** имеет следующие атрибуты: фамилия, имя, отчество, год рождения, год поступления на работу, стаж, должность, пол, адрес, город, телефон.  **Таблица обслуживающий персонал** имеет следующие атрибуты: должность (техники, сварщики, слесари, сборщики и др.), фамилия, имя, отчество, год рождения, год поступления на работу, стаж, пол, адрес, город, телефон.  **Таблица гаражное хозяйство** имеет следующие атрибуты: название гаража, транспорт на ремонте, вид ремонта, дата поступления, дата выдачи после ремонта, результат ремонта, персонал, производящего ремонт. |
| **Вариант №7** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. **БД – информационная система поликлиники**. БД состоит из следующих таблиц: врачи, пациенты, история болезней, отделения, обслуживающий персонал.   **Таблица отделения** имеет следующие атрибуты: название отделения (хирургия, терапия, нервология и т.д.), этаж, номера комнат, ФИО заведующего.  **Таблица врачи** имеет следующие атрибуты: фамилия, имя, отчество,  должность, стаж работы, научное звание, адрес, номер отделения, в котором он работает.  **Таблица пациенты** имеет следующие атрибуты: фамилия, имя, отчество, адрес, город, возраст, пол.  **Таблица диагнозы** имеет следующие атрибуты: название диагноза, признаки болезни, период лечения, назначения.  **Таблица история болезни** имеет следующие атрибуты: пациент, врач, диагноз, лечение, дата заболевания, дата вылечивания, вид лечения (амбулаторное, стационарное). |
| **Вариант №8** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – информационная система больницы**. БД состоит из следующих таблиц: врачи, пациенты, история болезней, операции, лист лечения.  **Таблица врачи** имеет следующие атрибуты:, фамилия, имя, отчество, должность, стаж работы, научное звание, адрес.  **Таблица пациенты** имеет следующие атрибуты: фамилия, имя, отчество, адрес, город, возраст, пол.  **Таблица история болезни** имеет следующие атрибуты:, пациента врач, диагноз, дата заболевания, дата вылечивания, вид лечения (амбулаторное, стационарное), код операции.  **Таблица лист лечения** имеет следующие атрибуты: дата лечения, история болезни, лекарства, температура, давление, состояние больного (тяжелое, среднее, и т.д.).  **Таблица операции** имеет следующие атрибуты: описание операции  (удаление аппендицита, пластическая операция и т.д.), врач, дата операции, пациент, результат операции. |
| **Вариант №9** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – информационная система библиотек города**. БД состоит из следующих таблиц: библиотеки, читальные залы, литература, читатели, выдача лит-ры.  **Таблица библиотеки** имеет следующие атрибуты: название, адрес, город.  **Таблица читальные залы** имеет следующие атрибуты: название читального зала, библиотека, кол-во единиц лит-ры, кол-во посадочных мест, время работы, этаж, кол-во сотрудников.  **Таблица читатели** имеет следующие атрибуты: фамилия, имя, отчество,  категория читателя, место работы или обучения, возраст, дата регистрации в библиотеке.  **Таблица литература** имеет следующие атрибуты: название, категория литературы, авторы, издательство, год издательства, кол-во страниц, читальный зал.  **Таблица выдача литературы** имеет следующие атрибуты: читатель, литература, дата выдачи, срок выдачи, вид выдачи, наличие залога. |
| **Вариант**  **№10** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – информационная система автосалона**. БД состоит из следующих таблиц: автомобили, марка автомобиля, сотрудники, продажа автомобилей, покупатели.  **Таблица марка автомобиля** имеет следующие атрибуты: название марки, страна производитель, завод производитель, адрес.  **Таблица автомобиля** имеет следующие атрибуты: название автомобиля, марка, год производства, цвет, категория, цена.  **Таблица покупатели** имеет следующие атрибуты: фамилия, имя, отчество, паспортные данные, адрес, город, возраст, пол.  **Таблица сотрудника** имеет следующие атрибуты: фамилия, имя, отчество, стаж, зарплата.  **Таблица продажа автомобилей** имеет следующие атрибуты: дата, сотрудник, автомобиль, покупатель. |
| **Вариант**  **№11** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – успеваемость студентов кафедры**. БД состоит из следующих таблиц: кафедры, дисциплины, преподаватели, студенты, ведомости успеваемости.  **Таблица кафедра** имеет следующие атрибуты: название кафедры, факультет, ФИО заведующего, номер комнаты, номер корпуса, телефон, кол-во преподавателей.  **Таблица преподаватели** имеет следующие атрибуты: фамилия, имя, отчество, кафедра, год рождения, год поступления на работу, стаж, должность, пол, адрес, город, телефон.  **Таблица студенты** имеет следующие атрибуты: фамилия, имя, отчество, кафедра, год рождения, пол, адрес, город, телефон.  **Таблица дисциплины** имеет следующие атрибуты: название  дисциплины, кафедра, читаемой эту дисциплину, кол-во часов, вид итогового контроля.  **Таблица ведомости успеваемости** имеет следующие атрибуты:  преподаватель, дисциплина, студент, оценка. |
| **Вариант**  **№12** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – торговая организация**. БД состоит из следующих таблиц: торговая организация, торговая точка, продавцы, поставщики, заказы поставщикам.  **Таблица торговая организация** имеет следующие атрибуты: название торговой организации, адрес, ФИО директора, налоговый номер.  **Таблица торговая точка** имеет следующие атрибуты: название торговой точки, тип (универмаги, магазины, киоски, лотки и т.д.), торговая организация, адрес, ФИО заведующего.  **Таблица продавцы** имеет следующие атрибуты: фамилия, имя, отчество, торговая точка, должность, год рождения, пол, адрес проживания, город. **Таблица поставщики** имеет следующие атрибуты: название поставщика, тип деятельности, страна, город, адрес.  **Таблица заказы поставщикам** имеет следующие атрибуты: дата заказа, торговая точка, поставщик, название товара, кол-во, цена. |
| **Вариант**  **№13** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. **БД – проектная организация**. БД состоит из следующих таблиц: отделы, сотрудники, организации, договора, проектные работы.   **Таблица отделы** имеет следующие атрибуты: название отдела, этаж, телефон, начальник отдела.  **Таблица сотрудники** имеет следующие атрибуты: ФИО, должность (конструкторы, инженеры, техники, лаборанты, прочий обслуживающий персонал), номер отдела, в котором работает, пол, адрес, дата рождения. **Таблица организации** имеет следующие атрибуты: название организации, тип деятельности, страна, город, адрес, ФИО директора.  **Таблица договора** имеет следующие атрибуты: номер договора, дата заключения договора, организация, стоимость договора.  **Таблица проектные работы** имеет следующие атрибуты: дата начала проектной работы, дата завершения проектной работы, номер договора, отдел, осуществляющий разработку. |
| **Вариант** | На основании выбранного варианта выполнить следующее: |
| **№14** | 1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – информационная система военно-морского флота**. БД состоит из следующих таблиц: базы, части, личный состав, корабли, учения.  **Базы военно-морского флота** имеет следующие атрибуты: название базы, георгафическое расположение, кол-во частей.  **Таблица части** имеет следующие атрибуты: номер части, база флота, место базирования, вид войск (морская авиация, морская пехота и т.д.). **Таблица личный состав** имеет следующие атрибуты: фамилия, часть, должность, год рождения, год поступления на службу, выслуга лет, награды,  **Таблица корабли** имеет следующие атрибуты: идентификационный номер корабля, название корабля, тип корабля, дата создания, наработка, кол-во посадочных мест, устройство двигателя (п[арусное](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D1%83%D1%81%D0%BD%D0%BE%D0%B5_%D1%81%D1%83%D0%B4%D0%BD%D0%BE), [гребное](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B5%D0%B1%D0%BD%D0%BE%D0%B5_%D1%81%D1%83%D0%B4%D0%BD%D0%BE), [пароход](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%BE%D1%85%D0%BE%D0%B4), [теплоход](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BF%D0%BB%D0%BE%D1%85%D0%BE%D0%B4), [турбоход](https://ru.wikipedia.org/wiki/%D0%A2%D1%83%D1%80%D0%B1%D0%BE%D1%85%D0%BE%D0%B4), и т.д. ), тип привода ([самоходное](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%BC%D0%BE%D1%85%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D1%81%D1%83%D0%B4%D0%BD%D0%BE), [несамоходное](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D1%81%D0%B0%D0%BC%D0%BE%D1%85%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D1%81%D1%83%D0%B4%D0%BD%D0%BE)), размещение [корпуса](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D0%BF%D1%83%D1%81_%D1%81%D1%83%D0%B4%D0%BD%D0%B0) (п[одводная лодка](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B4%D0%B2%D0%BE%D0%B4%D0%BD%D0%B0%D1%8F_%D0%BB%D0%BE%D0%B4%D0%BA%D0%B0), [ныряющее](https://ru.wikipedia.org/w/index.php?title=%D0%9D%D1%8B%D1%80%D1%8F%D1%8E%D1%89%D0%B5%D0%B5_%D1%81%D1%83%D0%B4%D0%BD%D0%BE&action=edit&redlink=1), [полупогружное](https://ru.wikipedia.org/w/index.php?title=%D0%9F%D0%BE%D0%BB%D1%83%D0%BF%D0%BE%D0%B3%D1%80%D1%83%D0%B6%D0%BD%D0%BE%D0%B5_%D1%81%D1%83%D0%B4%D0%BD%D0%BE&action=edit&redlink=1), и т.д.)  **Таблица учения**: часть, корабль, дата учения, место проведения, оценка. |
| **Вариант**  **№15** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – туристическая фирма**. БД состоит из следующих таблиц: туристы, туристичексая группа, состав групп, гостиницы, ведомости продажей.  **Таблица туристы** имеет следующие атрибуты: ФИО, паспортные данные, пол, возраст, дети.  **Таблица туры** имеет следующие атрибуты: название, страна, города, тип передвижения, тип питания, цена тура, тип проживания.  **Таблица туристическая группа** имеет следующие атрибуты: название, дата отправления, дата прибытия, тур, кол-во туристов.  **Таблица состав групп** имеет следующие атрибуты: дата продажи, турист, группа, цена билета.  **Таблица гостиницы** имеет следующие атрибуты: название гостиницы, страна, город, адрес, кол-во мест, тип гостиницы.  **Таблица ведомость продажей** имеет следующие атрибуты: дата, туристическая группа, гостиница, общая стоимость. |
| **Вариант**  **№16** | На основании выбранного варианта выполнить следующее:  1. Выполнить анализ предметной области исследуемой организации;   1. Описать основные сущности предметной области; 2. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 3. Построить инфологическую модель базы данных организации; 4. Построить даталогическую модель базы данных организации.   **БД – цирк**. БД состоит из следующих таблиц: работники цирка, представления, расписание гастролей, труппа цирка, программа цирка. **Таблица работники цирка** имеет следующие атрибуты: фамилия, имя, отчество, год рождения, год поступления на работу, стаж, должность (акробат, клоун, гимнаст, музыкант, постановщик, служащий и т.д.), пол, адрес, город, телефон.  **Таблица представления** имеет следующие атрибуты: название, режиссер-постановщик, художник-постановщик, дирижер-постановщик, автор, жанр, тип.  **Таблица расписание гастролей** имеет следующие атрибуты: представление, дата начала, дата окончания, места проведения гастроли. **Таблица труппа представления цирка** имеет следующие атрибуты: представление, актер цирка, название роли.  **Таблица программа цирка** имеет следующие атрибуты: представление, дата премьеры, период проведения, дни и время, цена билета. |
| **Вариант**  **№17** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – аптека**. БД состоит из следующих таблиц: лекартства, покупатели, продавцы, рецепты, продажа лекарств.  **Таблица лекарства** имеет следующие атрибуты: название, тип (готовое, изготовляемое), вид (таблетки, мази, настойки), цена.  **Таблица покупатели** имеет следующие атрибуты: фамилия, имя, отчество, адрес, город, телефон.  **Таблица продавцы** имеет следующие атрибуты: фамилия, имя, отчество, дата поступления, дата рождения, образование.  **Таблица рецепты** имеет следующие атрибуты: номер рецепта, дата выдачи, ФИО больного (покупатель), ФИО врача, диагноз пациента. **Таблица продажа лекарств** имеет следующие атрибуты: дата, лекарство, кол-во, рецепт, продавец. |
| **Вариант**  **№18** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области;   Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние  ключи между сущностями;   1. Построить инфологическую модель базы данных организации; 2. Построить даталогическую модель базы данных организации.   **БД – городская телефонная сеть**. БД состоит из следующих таблиц: АТС, абонент, ведомость звонков, прайс АТС, ведомость абонентской платы.  **Таблица АТС** имеет следующие атрибуты: название АТС, вид (городские, ведомственные и учрежденческие), адрес, город, кол-во абонентов.  **Таблица абоненты** имеет следующие атрибуты: фамилия, имя, отчество, вид телефона (основной, параллельный или спаренный), номер телефона, межгород (открыт/закрыт), льгота (да/нет), адрес: индекс, район, улица, дом, квартира.  **Таблица ведомость звонков** имеет следующие атрибуты: абонент, дата звонка, время начала, время окончания, межгород (да/нет).  **Таблица прайс АТС** имеет следующие атрибуты: АТС, цена на городские, цена на межгород.  **Таблица ведомость абонентской платы** имеет следующие атрибуты: абонент, месяц, год, кол-во минут на городские, кол-во минут на межгород, стоимость, сумма льготы, общая стоимость. |
| **Вариант**  **№19** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – аэропорт**. БД состоит из следующих таблиц: работники аэропорта, расписание вылетов, самолеты, бригады самолетов, ведомость продаж билетов.  **Таблица работники аэропорта** имеет следующие атрибуты: фамилия, имя, отчество, год рождения, год поступления на работу, стаж, должность (пилотов, диспетчеров, техников, кассиров, работников службы безопасности, справочной службы и других,), пол, адрес, город, телефон. **Таблица расписание вылетов** имеет следующие атрибуты: самолет, дата вылета, время вылета, место выбытия, место прибытия, маршрут (начальный и конечный пункты назначения, пункт пересадки), стоимость билета.  **Таблица самолеты** имеет следующие атрибуты: номер, год выпуска, кол-во посадочных место, грузоподъемность.  **Таблица бригады самолетов** имеет следующие атрибуты: номер бригады, самолет, работник аэропорта (пилоты, техники и обслуживающий персонал)ю  **Таблица ведомость продажи билетов** имеет следующие атрибуты: дата и время продажи, ФИО пассажира, паспортные данные, номер рейса, кол- во билетов, наличие льгот (пенсионеры, дети-сироты и т.д.), багаж (да/нет), стоимость. |
| **Вариант**  **№20** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – театр**. БД состоит из следующих таблиц: работники театра, спектакли, расписание гастролец, труппа спектакля, репертуар театра. **Таблица работники театра** имеет следующие атрибуты: фамилия, имя, отчество, год рождения, год поступления на работу, стаж, должность (актеров, музыкантов, постановщиков и служащих), пол, адрес, город, телефон.  **Таблица спектакли** имеет следующие атрибуты: название, режисер- постановщик, художник-постановщик, дирижер-постановщик, автор, жанр (музыкальная комедия, трагедия, оперетта и пр), тип (детские, молодежные и пр.).  **Таблица расписание гастролей** имеет следующие атрибуты: название, дата начала, дата окончания, места проведения гастроли, спектакль.  **Таблица труппа спектакля** имеет следующие атрибуты: спектакль, актер, название роли.  **Таблица репертуар театра** имеет следующие атрибуты: спектакль, дата премьеры, период проведения, дни и время, цена билета. |
| **Вариант**  **№21** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – железнодорожный вокзал**. БД состоит из следующих таблиц: работники ж.д.вокзала, расписание движения поездов, поезда, бригады поездов, ведомость продаж билетов.  **Таблица работники ж.д.вокзала** имеет следующие атрибуты: фамилия, имя, отчество, год рождения, год поступления на работу, стаж, должность (машинист, диспетчеров, проводник, ремонтников подвижного состава, путей, кассиров, работников службы подготовки составов, справочной службы и других,), пол, адрес, город, телефон.  **Таблица расписание движения поездов** имеет следующие атрибуты: поезд, дата отправления, время отправления, место отправления, дата прибытия, время прибытия, место прибытия, маршрут ((начальный и конечный пункты назначения, основные узловые станции), стоимость билета.  **Таблица поезда** имеет следующие атрибуты: номер, год выпуска, кол-во вагонов, тип поезда (общий, скоростной, высокоскоростной).  **Таблица бригады поездов** имеет следующие атрибуты: номер бригады, поезд, работник ж.д.вокзала (машинисты, техники, проводники и обслуживающий персонал).  **Таблица ведомость продажи билетов** имеет следующие атрибуты: дата и время продажи, ФИО пассажира, паспортные данные, номер рейса, кол- во билетов, наличие льгот (пенсионеры, дети-сироты и т.д.), стоимость. |
| **Вариант**  **№22** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – информационная система ВУЗА**. БД состоит из следующих таблиц: факультеты, кафедры, преподаватели, дисциплины, учебная нагрузка.  **Таблица факультеты** имеет следующие атрибуты: название факультета, ФИО декана, номер комнаты, номер корпуса, телефон.  **Таблица кафедра** имеет следующие атрибуты: название кафедры, ФИО заведующего, номер комнаты, номер корпуса, телефон, кол-во преподавателей.  **Таблица дисциплины** имеет следующие атрибуты: название дисциплины, кол-во часов, цикл дисциплин.  **Таблица преподаватели** имеет следующие атрибуты: фамилия, имя, отчество, кафедра, год рождения, год поступления на работу, стаж, должность, пол, город.  **Таблица учебная нагрузка** имеет следующие атрибуты: преподаватель,  дисциплина, учебный год, семестр, группы, кол-во студентов, вид итогового контроля. |
| **Вариант**  **№23** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – информационная система военного округа**. БД состоит из следующих таблиц: места дислокации, вид войск, части, роты, личный состав.  **Таблица вид войск** имеет следующие атрибуты: название вида войск. **Таблица места дислокации** имеет следующие атрибуты: страна, город, адрес, занимаемая площадь, кол-во сооружений.  **Таблица части** имеет следующие атрибуты: номер части, место дислокации, вид войск, кол-во рот, кол-во техники, кол-во вооружений. **Таблица техника** имеет следующие атрибуты: название техники, часть, характеристики.  **Таблица вооружения** имеет следующие атрибуты: название вооружения,  часть, характеристики. |
| **Вариант**  **№24** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации. **БД – информационная система супермаркета**. БД состоит из следующих таблиц: отделы, клиенты, товары, продажа товаров, поставщики.   **Таблица отделы** имеет следующие атрибуты: название отдела, кол-во прилавков, кол-во продавцов, номер зала.  **Таблица клиенты** имеет следующие атрибуты: название клиента, адрес, вид оплаты.  **Таблица поставщики** имеет следующие атрибуты: название поставщика, адрес, страна, вид транспорта, вид оплаты.  **Таблица товары** имеет следующие атрибуты: название товара, отдел, поставщик, условия хранения, сроки хранения .  **Таблица продажа товаров** имеет следующие атрибуты: клиент, товар, дата, время, кол-во, цена, сумма. |
| **Вариант**  **№25** | На основании выбранного варианта выполнить следующее:   1. Выполнить анализ предметной области исследуемой организации; 2. Описать основные сущности предметной области; 3. Расставить существующие связи между сущностями: самостоятельно добавить в каждую сущность первичные ключи и установить внешние ключи между сущностями; 4. Построить инфологическую модель базы данных организации; 5. Построить даталогическую модель базы данных организации.   **БД – информационная система больницы**. БД состоит из следующих таблиц: врачи, пациенты, история болезней, отделения, лист лечения.  **Таблица отделения** имеет следующие атрибуты: название отделения (хирургия, терапия, нервология и т.д.), этаж, номера комнат, ФИО заведующего.  **Таблица врачи** имеет следующие атрибуты: фамилия, имя, отчество, должность, стаж работы, научное звание, адрес.  **Таблица пациенты** имеет следующие атрибуты: фамилия, имя, отчество, адрес, город, возраст, пол.  **Таблица история болезни** имеет следующие атрибуты: пациент, врач, диагноз, дата заболевания, дата вылечивания, вид лечения (амбулаторное, стационарное).  **Таблица лист лечения** имеет следующие атрибуты: дата лечения,  история болезни, лекарства, температура, давление, состояние больного (тяжелое, среднее, и т.д.). |

**Инструкционно-технологическая карта**

**для выполнения практической работы №4**

**Тема:** **Основы работы с СУБД.**

**Наименование работы: Создание базы данных в среде разработки**

**Цель:** C помощью операторов языка Transact SQL научиться создавать базы данных и совокупность связанных таблиц, принадлежащих указанной базе данных

**Формируемые компоненты:** **ПК 11.1-11.6 ОК 1-11**

**Норма времени:**4 часа

**Оборудование и программное обеспечения:** ПК, MS SQL Server

**Основные правила ТБ при выполнении работы:**

-плавно нажимайте на клавиши, не допуская резких ударов;

-не пользуйтесь клавиатурой, если не подключено напряжение;

-работайте на клавиатуре чистыми руками;

-никогда не пытайтесь самостоятельно устранить неисправность в работе аппаратуры;

Строго запрещается:

-трогать разъемы соединительных кабелей;

-прикасаться к питающим проводам и устройствам заземления;

-прикасаться к экрану и к тыльной стороне монитора, клавиатуры;

-включать и отключать аппаратуру без указания преподавателя;

-Класть диск, книги, тетради на монитор и клавиатуру;

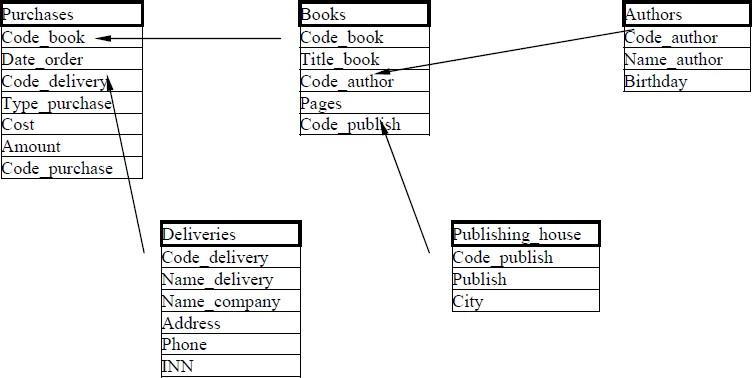
-Работать во влажной одежде и влажными руками

**Литература:** Кумскова, И. А. Базы данных: учебник для СПО / И. А. Кумскова.- М.: КНОРУС, 2016.-488 с.

**Контрольные вопросы при допуске:**

1. Что такое БД?
2. Что такое первичный ключ?
3. Какие элементы в БД есть?

***Задание*:**Создать базу данных. В качестве примера базы данных, которая будет создана программно с помощью операторов языка Transact SQL, выберем БД «Книжное дело». Схема базы данных и структура таблиц данной БД представлена в таблице.

Рисунок 1 Структура таблиц, схема базы данных

*Таблица 1.1* **Покупки (название таблицы Purchases)**

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Описание поля |
| Code\_book | Number (5) | Код закупаемой книги |
| Date order | Date | Дата заказа книги |
| Code\_delivery | Number (4) | Код поставщика, внешний ключ на таблицу  Deliveries |
| Type\_purchase | Char (1) | Тип закупки (опт/ розница) |
| Cost | Number (10,2) | Стоимость единицы товара |
| Amount | Number (4) | Количество экземпляров |
| Code\_purchase | Number (5) | Код покупки |

*Таблица 1.2***Справочник книг (название таблицы Books)**

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Описание поля |
| Code\_book | Number (5) | Код книги, |
| Title\_book | VarChar2 (100) | Название книги |
| Code\_author | Number (3) | Код автора, внешний ключ на таблицу  Authors |
| Pages | Number (4) | Количество страниц |
| Code\_publish | Number (4) | Код издательства |

*Таблица 1.3***Справочник авторов (название таблицы Authors)**

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Описание поля |
| Code\_author | Number (3) | Код автора |
| Name\_author | Char (30) | Фамилия, имя, отчество автора |
| Birthday | Date | Дата рождения |

**Справочник поставщиков (название таблицы Deliveries)***Таблица 1.4*

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Описание поля |
| Code\_delivery | Number(4) | Код поставщика |
| Name\_delivery | Char (35) | Фамилия, имя, отчество ответственного лица |
| Name\_company | Char (30) | Название компании-поставщика |
| Address | VarChar2 (100) | Юридический адрес |
| Phone | Char (11) | Телефон контактный |
| INN | Char (13) | ИНН |

*Таблица 1.5***Справочник издательств (название таблицы Publishing\_house)**

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Описание поля |
| Code\_publish | Number(4) | Код издательства |
| Publish | Char (30) | Издательство |
| City | Char (15) | Город |

**Порядок выполнения работы**

1. Создать структуру новой базы данных. Запустить **SQL Server Management Studio 2019.**

Для написания программного кода в **SQL Server Management Studio** нужно нажать кнопку «Создать запрос» («Newquery») на панели инструментов «Стандартная» («Standart»).

Создать новую базу данных с названием **DB\_Books**с помощью команды:

CREATE DATABASE DB\_BOOKS

Для выполнения команды нажать F5.

Открыть утилиту **SQL Server Management Studio**. Проверить наличие БД DB\_Books, если ее не видите в разделе Data Bases, то нажмите F5 для обновления.

1. Создать в базе данных перечисленные таблицы с помощью следующих команд (для создания новой страницы для кода в SQL Server Management Studio нажать кнопку «Создать запрос»):

use DB\_BOOKS

CREATE TABLE Authors(Code\_author INT PRIMARY KEY,name\_author CHAR(30), Birthday DATETIME)

CREATE TABLE Publishing\_house(Code\_publish INT PRIMARY KEY,Publish CHAR(30), City CHAR(20))

CREATE TABLE Books(Code\_book INT PRIMARY KEY, Title\_bookCHAR(40), Code\_author INT FOREIGN KEY REFERENCES Authors(Code\_author), Pages INT, Code\_publish INT FOREIGN KEYREFERENCES Publishing\_house(Code\_publish))

CREATE TABLE Deliveries(Code\_delivery INT PRIMARY KEY,Name\_delivery CHAR(30), Name\_company CHAR(20), AddressVARCHAR(100), Phone BIGINT, INN CHAR(13))

CREATE TABLE Purchases(Code\_purchase INT PRIMARY KEY,Code\_book INT FOREIGN KEY REFERENCES

Books(Code\_book),Date\_order SMALLDATETIME, Code\_delivery INT FOREIGN KEYREFERENCES Deliveries(Code\_delivery), Type\_purchase BIT, Cost FLOAT,Amount INT)

Запустите команду клавишей F5.

1. В утилите SQL Server Managemen Studio проверить наличие БД DB\_Books и таблиц в ней.

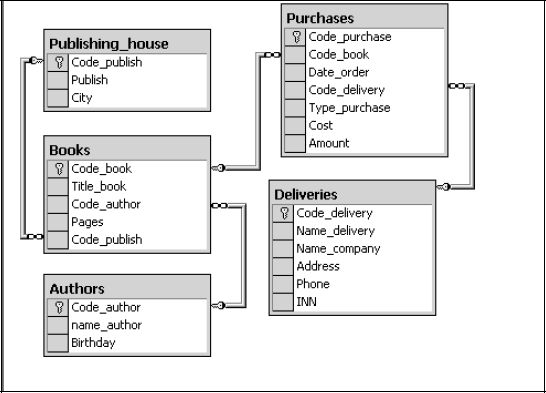
В разделе диаграмм создать новую диаграмму, в которую добавить из списка пять наших таблиц, проверить связи между таблицами (рис. 1.1).

Рис. 1.1. Результат создания диаграммы

***Использованные операторы:***

PRIMARY KEY – признак создания ключевого поля.

FOREIGN KEY**...**REFERENCES… – признак создания поля связи с другой таблицей.

CREATE TABLE – команда создания таблицы в текущей БД.

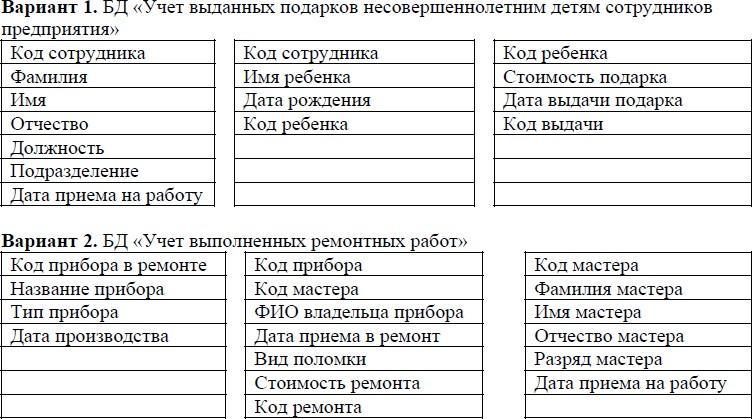
USE – сделать активной конкретную БД.

CREATE DATABASE – команда создания новой БД.

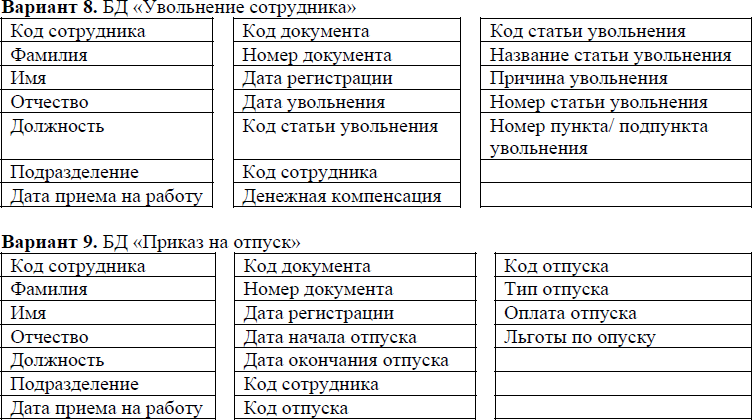
*Варианты заданий к лабораторной работе №2*

В утилите SQL Server Management Studio создать новую базу данных с помощью оператора **Create Database**, название БД определить, исходя из предметной области. Закомментировать оператор (-- – однострочный комментарий, /\* \*/ – многострочный комментарий). Программно сделать активной созданную БД с помощью оператора **Use**. Создать перечисленные таблицы c помощью операторов **Create table**, причем самостоятельно определить типы таблиц (родительская или подчиненная), типы полей и их размеры, найти поля типа Primarykey и Foreignkey. В SQL Server Management Studio в разделе диаграмм созданной БД сгенерировать новую диаграмму, проверить связи между таблицами.

Варианты заданий к лабораторной работе №2









*Требования к содержанию отчета*: итоги лабораторной работы представить в виде пояснений и копий экранов выполнения основных заданий, описанных в инструкции по выполнению лабораторной работы, также необходимо предоставить файл программы индивидуального задания по варианту, файл назвать - ФамилияСтудента\_Лаб\_1\_№ варианта**.**

**Контрольные вопросы для самоподготовки**

1. Назовите основные проводники, браузеры, редакторы, визуальные конструкторы и мастера среды SQL Server Management Studio.
2. Что такое источники и поставщики данных?
3. Что такое индекс?
4. Дайте определение типа данных.
5. В чем заключается понятие сущности, атрибута, кортежа?

**Инструкционно-технологическая карта**

**для выполнения лабораторная работа №3**

**Тема:** **Разработка и администрирование БД.**

**Наименование работы: Организация локальной сети. Настройка локальной сети**

**Цель:** знакомство с основными принципами программирования в MS SQL Server средствами встроенного языка Transact SQL.

**Формируемые компоненты:** **ПК 11.1-11.6 ОК 1-11**

**Норма времени:**4 часа

**Оборудование и программное обеспечения:** ПК, MS SQL Server

**Основные правила ТБ при выполнении работы:**

-плавно нажимайте на клавиши, не допуская резких ударов;

-не пользуйтесь клавиатурой, если не подключено напряжение;

-работайте на клавиатуре чистыми руками;

-никогда не пытайтесь самостоятельно устранить неисправность в работе аппаратуры;

Строго запрещается:

-трогать разъемы соединительных кабелей;

-прикасаться к питающим проводам и устройствам заземления;

-прикасаться к экрану и к тыльной стороне монитора, клавиатуры;

-включать и отключать аппаратуру без указания преподавателя;

-Класть диск, книги, тетради на монитор и клавиатуру;

-Работать во влажной одежде и влажными руками **Литература:** Кумскова, И. А. Базы данных: учебник для СПО / И. А. Кумскова.- М.: КНОРУС, 2016.-488 с.

**Контрольные вопросы при допуске****:**

1. Что такое первичный ключ?
2. Как ставить первичный ключ в MS SQL Server?
3. Какие элементы БД есть в MS SQL Server?

***Задание*:** освоить основные принципы программирования в MS SQL Server**.** Для этого используем пример базы данных c названием **DB\_Books**, которая была создана в лабораторной работе №1. При выполнении примеров и заданий обращайте внимание на соответствие названий БД, таблиц и других объектов проекта.

Порядок выполнения работы

Для получения более подробной информации о работе тех или иных операторов или функций можно запустить утилиту BooksOnline из состава MS SQL Server и в разделе «Указатель» набрать искомый ключевой элемент.

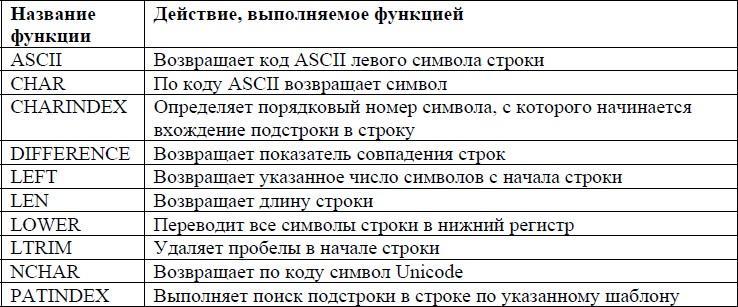
Выполнить задания, согласно варианту, представленному в табл. 2.3.

*Специальные знаки и простейшие операторы в Transact SQL*

Проверить работу описанной установки SETQUOTED\_IDENTIFIER.

*Таблица 2.2*

**Краткий обзор строковых функций**



Проверить работу описанной установки SET DATEFIRST.

*Объявление переменных*

Объявить переменную Perem1 типа денежный, а переменнуюPerem2 типа число с целой частью равной 8 и дробной частью равной

.*Таблица 2.3* **Список вариантов заданий**



Объявить переменную Perem1 типа строка длиной 100, а переменную Perem2 типа длинное целое.

Объявить переменную Perem1 типа динамическая строка с максимальной длиной 1000, а переменную Perem2 типа целое число.

Объявить переменную Perem1 типа строка длиной 30, а переменную Perem2 типа число с целой частью равной 10 и дробной частью равной3.

Объявить переменную Perem1 типа дата/ время, а переменнуюPerem2 типа число в диапазоне от 0 до 255.

*Присвоение значений переменным и вывод значений на экран*

Подсчитать среднюю цену закупленных книг (с помощью запроса SELECT) и умножить ее на значение 123,34, которое необходимо сохранить в отдельной переменной, вывести значение переменной на экран.

Подсчитать суммарную цену всех закупок книг, результат поместить в переменную, вывести значение переменной на экран.

Подсчитать количество книг в справочнике книг, результат поместить в переменную, вывести значение переменной на экран.

Определить минимальную дату рождения автора в справочнике авторов, результат поместить в переменную, вывести значение переменной на экран.

*Сочетание ключевых слов SET и SELECT*

Подсчитать количество поставщиков книг, результат поместить в переменную.

Подсчитать сумму закупок книг, результат поместить в переменную.

Подсчитать среднюю цену в таблице покупок книг, результат поместить в переменную.

Подсчитать максимальную стоимость книг в закупке, результат поместить в переменную.

*Работа с датой и временем*

Определить переменную Date1 типа дата/время. Присвоить ей значение даты 31.12.2006 в формате dd.mm.yyyy.

Определить переменную Date1 типа дата/время. Присвоить ей значение даты 31.12.2006 в формате mm.dd.yyyy.

Определить переменную Date1 типа дата/время. Присвоить ей значение даты 31.12.2006 в формате yyyy.mm.dd.

*Создание временной таблицы через переменную типа TABLE*

Создать локальную таблицу с названием TEMP и полями типа дата/время, длинное целое, строка. Добавить в нее две записи с данными и вывести результат на экран.

Создать локальную таблицу с названием TEMP и полями типа длинное целое, строка и значением по умолчанию «введите что- нибудь»,денежный. Добавить в нее две записи с данными и вывести результат на экран.

Создать локальную таблицу с названием TEMP и полями типа целое, динамическая строка, бит со значением по умолчанию «1». Добавить в нее две записи с данными и вывести результат на экран.

Создать локальную таблицу с названием TEMP и полями типа дата/время, длинное целое, строка. Добавить в нее две записи с данными и вывести результат на экран.

Создать локальную таблицу с названием TEMP и полями типа дата/время, длинное целое с автонаращиванием, динамическая строка. Добавить в нее две записи с данными и вывести результат на экран.

*Преобразование типов переменных*

Объявить переменные типа FLOAT, CHAR, TINYINT. Присвоить значения, соответствующие типам. Выполнить преобразование переменных типа FLOAT, CHAR, TINYINTвINT, DATETIME, BITсоответственно и вывести результат на экран.

Объявить переменные типа INT, DATETIME, BIT. Присвоить значения, соответствующие типам. Выполнить преобразование переменных типа INT, DATETIME, BIT в FLOAT, CHAR, TINYINT соответственно и вывести результат на экран.

Объявить переменные типа NUMERIC, VARCHAR, DATETIME. Присвоить значения, соответствующие типам. Выполнить преобразование переменных типа NUMERIC, VARCHAR, DATETIMEвFLOAT, CHAR,BIGINT соответственно и вывести результат на экран.

Объявить переменные типа BIT, NVARCHAR, DATETIME. Присвоить значения, соответствующие типам. Выполнить преобразование переменных типа BIT, NVARCHAR, DATETIME в FLOAT, INT, BIGINT соответственно и вывести результат на экран.

*Условная конструкция IF*

Подсчитать количество поставщиков в таблице Deliveries. Если их в таблице от 2 до 5, то ничего не сообщать, в противном случае вывести сообщение вида "В таблице ... поставщиков" (вместо многоточия поставить точное количество поставщиков).

Подсчитать сумму закупок книг в таблице покупок. Если полученная сумма в диапазоне от 1000 до 5000, то ничего не сообщать, в противном случае вывести сообщение вида "Сумма закупок = …" (вместо многоточия поставить точную сумму).

Подсчитать среднюю стоимость закупки книг в таблице покупок. Если полученная стоимость в диапазоне от 1000 до 5000, то ничего не сообщать, в противном случае вывести сообщение вида "Средняя стоимость закупки = …" (вместо многоточия поставить точную среднюю стоимость).

Определить минимальную стоимость закупки книг в таблице покупок. Если полученная стоимость в диапазоне от 200 до 300, то ничего не сообщать, в противном случае вывести сообщение вида "Минимальная стоимость закупки = …" (вместо многоточия поставить точную стоимость).

*Цикл WHILE*

Определить количество записей в таблице Authors. Пока записей меньше 15, делать в цикле добавление записи в таблицу с автоматическим наращиванием значения ключевого поля, а вместо имени автора ставить значение 'Автор не известен'.

Определить количество записей в таблице издательств. Пока записей меньше 20, делать в цикле добавление записи в таблицу с автоматическим наращиванием значения ключевого поля, а вместо названия издательства ставить значение 'не известно'.

Определить количество записей в таблице поставщиков. Пока записей меньше 17, делать в цикле добавление записи в таблицу с автоматическим наращиванием значения ключевого поля, а вместо названия поставщика ставить значение 'не известен'.

*Объявление курсора*

Создать статический курсор по данным таблицы Books с полями Code\_book, Title\_book.

Создать динамический курсор по данным таблицы поставщиков(таблицаDeliveries) с полями Name\_delivery, Name\_company.

Создать статический курсор по данным таблицы Books и Authors с полями Code\_book, Title\_book, Name\_author.

Создать статический курсор по данным таблицы Books и Publishing\_house с полямиCode\_book, Title\_book, Publish.

*Операторы для работы с курсором*

Создать динамический курсор для чтения по данным таблицы Deliveries с полями Code\_delivery, Name\_delivery. Вывести данные 3-й записи.

Сделать текущей БД db\_books. Поместить в курсор данные таблицы Purchases. Перебрать все записи таблицы Purchases. Просуммировать значения произведений полей Cost и Amount и результат сохранить в переменной Sum\_table, которую после суммирования вывести на экран. Закрыть и удалить из памяти курсор.

Объявить статический курсор по данным таблиц Authors и Books. Вывести данные 5-й записи.

*Использование функций для работы со строковыми переменными*

Базовый текст дан в отдельном файле по вариантам. Для выполнения этого блока заданий в начале программы, которую вы создаете, объявите переменную типа varchar и присвойте ей в качестве значения строку с любым базовым текстом, который будет анализироваться и/или исправляться в заданиях.

Удалить в тексте лишние пробелы. Лишними считаются те, которые идут непосредственно за пробелом. Подсчитать количество исправлений.

Подсчитать количество встреч каждой из следующих букв: "а", "в","и", "п" в базовом тексте.

Подсчитать доли процентов встречи следующих букв: "е", "о", если суммарный процент встречаемости всех этих букв равен 100% или процент встречаемости е% + о% равен 100%.

По правилам оформления машинописных текстов перед знаками.,!?:; пробелы не ставятся, но обязательно ставятся после этих знаков. Удалите лишние пробелы. Подсчитать количество исправлений.

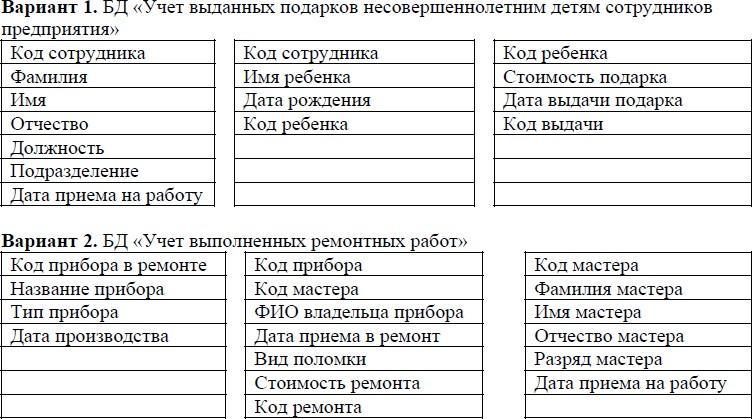
По правилам оформления машинописных текстов перед знаками.,!?:; пробелы не ставятся, но обязательно ставятся после этих знаков. Расставьте недостающие пробелы. Подсчитать количество исправлений.

Найти из исходного текста второе предложение и вернуть его в переменную Perem, а также вывести на экран весь исходный текст и найденное предложение.

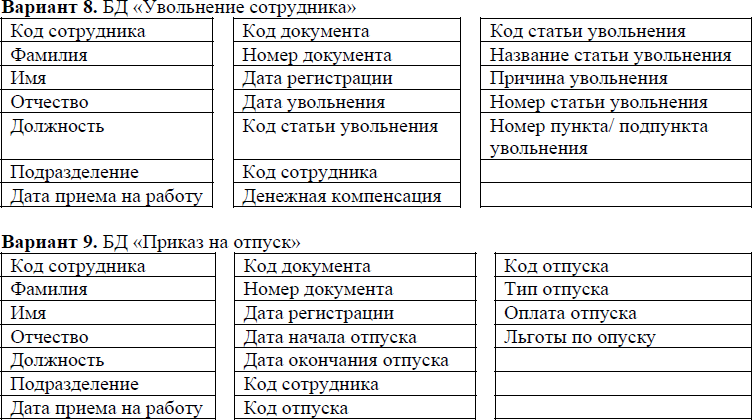
Удалить из базового текста 2, 4, 6, 8 слова.

Удалить из базового текста 3, 5, 7, 10 слова.

*Требования к содержанию отчета*: итоги лабораторной работы представить в виде пояснений и копий экранов выполнения основных заданий, описанных в инструкции по выполнению лабораторной работы.









**Контрольные вопросы для самоподготовки**

Что такое SQL?

Перечислите простейшие операторы встроенного языка Transact SQL

Опишите процедуру объявления переменных, присвоение значений переменным и вывод значений на экран на языке transact SQL?

Какие существуют логические, математические, строковые функции, функции работы с датой и временем в языке transact SQL?

Опишите синтаксис условной конструкции IF и цикла WHILE на языке transact SQL

Что такое курсоры и как они объявляются?

**Инструкционно-технологическая карта**

**для выполнения лабораторная работа №4**

**Тема:** **Разработка и администрирование БД.**

**Наименование работы: Установка и настройка SQL-сервера**

**Цель:** научиться создавать и использовать хранимые процедуры на сервере БД.

**Формируемые компоненты:** **ПК 11.1-11.6 ОК 1-11**

**Норма времени:**4 часа

**Оборудование и программное обеспечения:** ПК, MS SQL Server

**Основные правила ТБ при выполнении работы:**

-плавно нажимайте на клавиши, не допуская резких ударов;

-не пользуйтесь клавиатурой, если не подключено напряжение;

-работайте на клавиатуре чистыми руками;

-никогда не пытайтесь самостоятельно устранить неисправность в работе аппаратуры;

Строго запрещается:

-трогать разъемы соединительных кабелей;

-прикасаться к питающим проводам и устройствам заземления;

-прикасаться к экрану и к тыльной стороне монитора, клавиатуры;

-включать и отключать аппаратуру без указания преподавателя;

-Класть диск, книги, тетради на монитор и клавиатуру;

-Работать во влажной одежде и влажными руками **Литература:** Кумскова, И. А. Базы данных: учебник для СПО / И. А. Кумскова.- М.: КНОРУС, 2016.-488 с.

**Контрольные вопросы при допуске:**

1. Что такое сервер?
2. Для чего нужен сервер?

***Задание*:** освоить процедуру создания хранимых процедур

Порядок выполнения работы

Создадим процедуру без параметров:

CREATE PROCEDURE Count\_Books AS Select count(Code\_book) from Books

Go

1. Создадим процедуру c входным параметром:

CREATE PROCEDURE Count\_Books\_Pages @Count\_pages as Int

AS Select count(Code\_book) from Books WHERE Pages>=@Count\_pages

Go

1. Создадим процедуру c входными параметрами:

CREATE PROCEDURE Count\_Books\_Title @Count\_pages as Int, @Title AS Char(10) AS Select count(Code\_book) from Books WHERE Pages>=@Count\_pages

ANDTitle\_book LIKE @Title Go

1. Создадим процедуру c входными параметрами и выходным параметром:

CREATE PROCEDURE Count\_Books\_Itogo @Count\_pagesInt, @Title Char(10) , @ItogoInt OUTPUT AS Select @Itogo = count(Code\_book) from Books WHERE Pages>=@Count\_pages AND Title\_book LIKE @Title

Go

1. Создадим процедуру c входными параметрами и RETURN: CREATE PROCEDURE checkname @paramint

AS IF (SELECT Name\_author FROM authors WHERE Code\_author = @param) = 'ПушкинА.С.'

RETURN 1 ELSE RETURN 2

1. Создадим процедуру без параметров для увеличения значения ключевого поля в таблице Purchases в 2 раза:

CREATE PROC update\_proc

AS UPDATE Purchases SET Code\_purchase = Code\_purchase\*2 Процедура не возвращает никаких данных.

1. Создадим процедуру с входным параметром для получения всей информации о конкретном авторе:

CREATE PROC select\_author @k CHAR(30)

AS SELECT \* FROM Authors WHERE name\_author=@k

1. Создадим процедуру с входным параметром и значением по умолчанию для увеличения значения ключевого поля в таблице Purchases в заданное количество раза (по умолчанию в 2 раза):

CREATE PROC update\_proc @p INT = 2

AS UPDATE Purchases SET Code\_purchase = Code\_purchase \*@p Процедура не возвращает никаких данных.

1. Создадим процедуру для определения количества заказов, совершенных за указанный период. Это процедура с входным и выходным параметрами

CREATE PROC count\_purchases

@d1 SMALLDATETIME, @d2 SMALLDATETIME, @c INT OUTPUT

AS SELECT @c=count(Code\_purchase) from Purchases WHERE Date\_order BETWEEN @d1 AND @d2

SET @c = ISNULL(@c,0)

*Варианты заданий к лабораторной работе №3 Общие положения*

В утилите SQL Server Management Studio создать новую страницу для кода (кнопка «Создать запрос»). Программно сделать активной созданную БД DB\_Books с помощью оператора **Use**. Создать хранимые процедуры c помощью операторов **Create procedure**, причем самостоятельно определить имена процедур. Каждая процедура будет выполнять по одному SQL запросу, которые были выполнены во второй лабораторной работе. Причем код SQL запросов нужно изменить таким образом, чтобы в них можно было передавать значения полей, по которым осуществляется поиск.

Например, исходное задание и запрос в лабораторной работе №3:

/\*Выбрать из справочника поставщиков (таблица Deliveries) названия компаний, телефоны и ИНН (поля Name\_company, Phone и INN), у которых

название компании (поле Name\_company) 'ОАО МИР'. SELECT Name\_company, Phone, INN FROM Deliveries WHERE Name\_company = 'ОАОМИР'\*/

--В данной работе будет создана процедура:

CREATE PROC select\_name\_company @comp CHAR(30) AS

SELECT Name\_company, Phone, INN FROM Deliveries

WHERE Name\_company = @comp

--Для запуска процедуры используется команда: EXEC select\_name\_company 'ОАОМИР'

Сохранить файл программы с названием ФамилияСтудента\_ЛАб\_3. В SQL Server Management Studio в разделе хранимых процедур БД

DB\_Books проверить наличие процедур.

*Список заданий*

В утилите SQL Server Management Studio создать новую программу.

Программно сделать активной индивидуальную БД, созданную в лабораторной работе №1, с помощью оператора Use. Создать хранимые процедуры c помощью операторов Create procedure, причем самостоятельно определить имена процедур. В SQL Server Management Studio в разделе хранимых процедур индивидуальной БД проверить наличие процедур. Варианты заданий для данной работы приведены в Приложении 3.

*Требования к содержанию отчета*: итоги лабораторной работы представить в виде пояснений и копий экранов выполнения основных заданий, описанных в инструкции по выполнению лабораторной работы, предоставить файл программы индивидуального задания, с названием Фамилия Студента\_Лаб\_4\_№варианта..

***Вариант 1***

1. Вывести список сотрудников, у которых есть хотя бы один ребенок.
2. Вывести список детей, которым выдали подарки в указанный период.
3. Вывести список родителей, у которых есть несовершеннолетние дети.
4. Вывести информацию о подарках со стоимостью больше указанного числа, отсортированных по дате.

***Вариант 2***

1. Вывести список приборов с указанным типом.
2. Вывести количество отремонтированных приборов и общую стоимость ремонтов у указанного мастера.
3. Вывести список владельцев приборов и количество их обращений, отсортированный по количеству обращений по убыванию.
4. Вывести информацию о мастерах с разрядом больше указанного числа или с датой приема на работу меньше указанной даты.

***Вариант 3***

1. Вывести список цветков с указанным типом листа.
2. Вывести список кодов продаж, по которым продано цветов на сумму больше указанного числа.
3. Вывести дату продажи, сумму, продавца и цветок по указанному коду продажи.
4. Вывести список цветов и сорт для цветов с высотой больше указанного числа или цветущий.

***Вариант 4***

1. Вывести список лекарств с указанным показанием к применению.
2. Вывести список дат поставок, по которым продано больше указанного числа одноименного лекарства.
3. Вывести дату поставки, сумму, ФИО руководителя от поставщика и название лекарства по коду поступления больше указанного числа.
4. Вывести список лекарств и единицы измерения для лекарств с количеством в упаковке больше указанного числа или кодом лекарства меньше определенного значения.

***Вариант 5***

1. Вывести список сотрудников с указанной должностью.
2. Вывести список списанного оборудования по указанной причине.
3. Вывести дату поступления, название оборудования, ФИО ответственного и дату списания для оборудования, списанного в указанный период.
4. Вывести список оборудования с указанным типом или с датой поступления больше определенного значения.

***Вариант 6***

1. Вывести список блюд с весом больше указанного числа.
2. Вывести список продуктов, в названии которых встречается указанный фрагмент слова.
3. Вывести объем продукта, название блюда, название продукта с кодом блюда от указанного начального значения по определенному конечному значению.
4. Вывести порядок приготовления блюда и название блюда с количеством углеводов больше определенного значения или количеством калорий больше указанного значения.

***Вариант 7***

1. Вывести список сотрудников с указанной должностью.
2. Вывести список документов, в содержании которых встречается указанный фрагмент слова.
3. Вывести дату регистрации, тип документа, ФИО регистратора и название организации для документов, зарегистрированных в указанный период.
4. Вывести список зарегистрированных документов с определенным типом документа или с датой регистрации больше указанного значения.

***Вариант 8***

1. Вывести список сотрудников с указанной причиной увольнения.
2. Вывести список документов с датой регистрации в указанный период.
3. Вывести дату регистрации, причину увольнения, ФИО сотрудника для документов, зарегистрированных в указанный период.
4. Вывести список зарегистрированных документов с кодом документа в указанном диапазоне.

***Вариант 9***

1. Вывести список сотрудников, бравших отпуск указанного типа.
2. Вывести список документов с датой регистрации в указанный период.
3. Вывести дату регистрации, тип отпуска, ФИО сотрудника для документов, зарегистрированных в указанный период.
4. Вывести список зарегистрированных документов с кодом документа в указанном диапазоне.

***Вариант 10***

1. Вывести список сотрудников с указанной должностью.
2. Вывести список документов, в содержании которых встречается указанный фрагмент слова.
3. Вывести дату регистрации, тип документа, ФИО отправителя и название организации для документов, зарегистрированных в указанный период.

Вывести список зарегистрированных документов с указанным типом документа или с кодом документа меньше определенного значения

**Контрольные вопросы для самоподготовки**

Что такое хранимая процедура?

Для чего используется хранимая процедура?

Сколько хранимых процедур можно создать у одной таблицы?

Какие существуют виды хранимых процедур?

Какие уровни языка transact-SQL используются при создании и использовании хранимых процедур?

**Инструкционно-технологическая карта**

**для выполнения лабораторная работа №5**

**Тема:** **Разработка и администрирование БД.**

**Наименование работы: Экспорт данных базы в документы пользователя**

**Цель:** – научиться создавать клиентское приложение для работы с базой данных с применением встроенных инструментов в среде Delphi 7.

**Формируемые компоненты:** **ПК 11.1-11.6 ОК 1-11**

**Норма времени:**4 часа

**Оборудование и программное обеспечения:** ПК, MS SQL Server, Delphi 7

**Основные правила ТБ при выполнении работы:**

-плавно нажимайте на клавиши, не допуская резких ударов;

-не пользуйтесь клавиатурой, если не подключено напряжение;

-работайте на клавиатуре чистыми руками;

-никогда не пытайтесь самостоятельно устранить неисправность в работе аппаратуры;

Строго запрещается:

-трогать разъемы соединительных кабелей;

-прикасаться к питающим проводам и устройствам заземления;

-прикасаться к экрану и к тыльной стороне монитора, клавиатуры;

-включать и отключать аппаратуру без указания преподавателя;

-Класть диск, книги, тетради на монитор и клавиатуру;

-Работать во влажной одежде и влажными руками

**Литература:** Кумскова, И. А. Базы данных: учебник для СПО / И. А. Кумскова.- М.: КНОРУС, 2016.-488 с.

**Контрольные вопросы при допуске:**

1. Что такое данные?
2. Что такое экспорт?
3. Как осушествляется экспорт данных?

***Задание*:** создать клиентское приложения в Delphi 7. Для этого использовать пример базы данных c названием DB\_Books, которая была создана в лабораторной работе №2. При выполнении примеров и заданий обращайте внимание на соответствие названий БД, таблиц и других объектов проекта.

**Порядок выполнения работы**

В данной работе при создании клиентского приложения будем использовать встроенные инструменты для работы с данными (в Delphi 7 – вкладка DataControls).

В новом проекте создадим модуль данных (меню File - New - DataModule). Переименуем форму в DM (свойство Name - DM).

В основной форме (например, Form1, переименовать ее в MainForm) в коде добавить описание модуля DM, для этого после ключевого слова USES в конце списка поставить запятую и дописать название программного модуля, например UNIT2, который соответствует созданной форме DM.

На форму DM с вкладки ADO добавить компонент ADOConnection (название, например ADOConnectionl), 5 компонентов типа ADOTable (переименовать компоненты в ADOPurchases, ADOBooks, ADOAuthors, ADODeliveries, ADOPublish), с вкладки Data Access 5 компонентов типа DataSource (переименовать компоненты в DataPurchases, DataBooks, DataAuthors, DataDeliveries, DataPublish).

У компонента ADOConnectionl настроить свойства:

Connected String = нажать кнопку Build\ выбрать Поставщик данных - Microsoft OLE DB Provider for Oracle

Источник данных – ORCL Пользователь - ADMIN\_BOOKS Пароль - worlds

Получится в результате - Provider=MSDAORA.1;User ID=ADMIN\_BOOKS;Data Source=Orcl;Persist Security

Info=False LoginPrompt = True; Connected = True.

У ADOAuthors изменить следующие свойства: Connection на ADOConnectionl; TableName на Authors;

Active на True.

У DataAuthors изменить следующие свойства (это будет ссылка на таблицу):

DataSet на ADOAuthors.

У ADOPurchases изменить следующие свойства: Connection на ADOConnectionl; TableName на Purchases;

Active на True.

У DataPurchases изменить следующие свойства (это будет ссылка на таблицу):

DataSet на ADOPurchases.

У ADOBooks изменить следующие свойства: Connection на ADOConnectionl; TableName на Books;

Active на True.

У DataBooks изменить следующие свойства (это будет ссылка на таблицу):

DataSet на ADOBooks.

У ADODeliveries изменить следующие свойства: Connection на ADOConnectionl;

TableName на Deliveries; Active на True.

У DataDeliveries изменить следующие свойства (это будет ссылка на таблицу):

DataSet на ADODeliveries.

У ADOPublish изменить следующие свойства: Connection на ADOConnection1; TableName на Publishing\_house;

Active на True.

У DataPublish изменить следующие свойства (это будет ссылка на таблицу):

DataSet на ADOPublish.

На основной форме (MainForm) добавить компонент Меню с вкладки Standart. В редакторе меню сделать первый пункт «Работа с таблицами» и в подменю пункты «Авторы», «Книги», «Издательства»,

«Поставщики», «Поставки».

Создать пять форм, каждую из которых назвать (изменить свойство Name): FormAuthors, FormPurchases, FormBooks, FormDeliveries, FormPublish. В главной форме, в коде добавить описание этих форм, для этого после ключевого слова USES в конце списка через запятую дописать названия программных модулей, которые соответствуют описанным формам. В каждой из созданных форм в коде добавить описание модуля DM, для этого после ключевого слова USES в конце списка поставить запятую и дописать название программного модуля, например UNIT2, который соответствует созданной форме DM.

На основной форме в подпунктах меню в соответствующих методах Click вызвать соответствующие формы с помощью кода:

для FormAuthors:

DM.ADOAuthors.Open;

FormAuthors: =TFormAuthors.Create(Application); FormAuthors.Show;

для FormPurchases:

DM.ADOPurchases.Open; FormPurchases:=TFormPurchases. Create(Application); FormPurchases.Show;

для FormBooks:

DM.ADOBooks.Open;

FormBooks: =TFormBooks.Create(Application); FormBooks.Show;

для FormDeliveries:

DM.ADODeliveries.Open; FormDeliveries: =TFormDeliveries.

Create(Application); FormDeliveries. Show; для FormPublish:

DM.ADOPublish.Open; FormPublish:=TFormPublish.Create(Application); FormPublish.Show;

На формы FormAuthors, FormPurchases, FormBooks, FormDeliveries, FormPublish добавить с вкладки Data Controls по паре компонент типа DBGrid и DBNavigator. Настроить у DBGrid и DBNavigator свойство DataSource для связи с соответствующим источником данных.

Проверить работу приложения.

На форму FormBooks с вкладки Data Controls добавить 3 компонента типа DBEdit, 2 компонента типа DBLookupComboBox.

У 1-го компонента DBEdit изменить свойства: Name на DBE\_Code\_Book;

DataSource на DM.DataBooks; DataField на Code\_book.

У 2-го компонента DBEdit изменить свойства: Name на DBE\_Title\_book;

DataSource на DM.DataBooks; DataField на Title\_book.

У 3-го компонента DBEdit изменить свойства: Name на DBE\_Pages;

DataSource на DM.DataBooks; DataField на Pages.

У 1-го компонента DBLookupComboBox изменить свойства: Name на DBL\_Code\_author;

DataSource на DM.DataBooks; DataField на Code\_author; ListSource на DM.DataAuthors;

ListField на name\_author; KeyField на Code\_author.

У 2-го компонента DBLookupComboBox изменить свойства: Name на DBL\_Code\_Publish;

DataSource на DM.DataBooks; DataField на Code\_publish; ListSource на DM.DataPublish; ListField на Publish;

KeyField на Code\_publish.

Рядом с каждым настроенным компонентом поставить Label (надпись), у которой в свойстве Caption написать, например, "Введите название книги” и т.д.

У компонента DBGrid настроить свойство ReadOnly в режим True. Так как на форме будут использоваться компоненты

DBLookupComboBox, которые ссылаются на таблицы Authors и Publishing\_house, нужно на основной форме в подпункте меню "Книги" в методе Click добавить в начало две команды:

DM.ADOPublish.Open; DM.ADOAuthors. Open;

В противном случае при открытии формы может произойти ошибка, если по какой-то причине источники данных для компонент DBLookupComboBox будут неактивны.

Аналогично для остальных форм добавить элементы типа DBEdit (для полей данные которых будут просто набираться или редактироваться пользователем) и DBLookupComboBox (для полей связи, к которым подходит связь типа «много», они позволят выбирать данные из соответствующего справочника и пользователю не придется помнить значения кодов).

Проверить работу приложения.

Теперь необходимо создать фильтры по текущим значениям полей

таблицы. Значения для фильтра будут браться из колонок DBGrid. Например, строка formbooks.DBGrid1.Columns.Items[0].Field.Text; будет брать текущее значение из первой колонки DBGrid. Поэтому при выполнении этого задания обращайте внимание на соответствие индексов колонки порядку полей и исправьте при необходимости индексы в соответствии с вашим порядком полей в таблице.

На форму FormBooks с вкладки Standart добавить 5 компонентов типа Button.

У 1-го компонента Button изменить свойства и метод: Name на B\_Publish;

Caption на Фильтр по текущему издательству; В методе Click кнопки написать код:

dm.ADOBooks.Filter:= 'Code\_Publish = '+ form- books.DBGrid1.Columns.Items[4].Field.Text; dm.ADOBooks.Filtered:= true;

У 2-го компонента Button изменить свойства и метод: Name на B\_Title\_Book;

Caption на Фильтр по текущему названию книги.

В методе Click кнопки написать код: dm.ADOBooks.Filter:= 'Title\_book = '+ QuotedStr( form- books.DBGrid1.Columns.Items[1].Field.Text); dm.ADOBooks.Filtered:= true;

У 3-го компонента Button изменить свойства и метод: Name на B\_Author;

Caption на Фильтр по текущему автору.

В методе Click кнопки написать код: dm.ADOBooks.Filter:= 'Code\_Author = '+ form- books.DBGrid1.Columns.Items[0].Field.Text; dm.ADOBooks.Filtered:= true;

У 4-го компонента Button изменить свойства и метод: Name на B\_Pages;

Caption на Фильтр по количеству страниц в книге.

В методе Click кнопки написать код: dm.ADOBooks.Filter:= 'Pages = '+ form- books.DBGrid1.Columns.Items[3].Field.Text; dm.ADOBooks.Filtered:= true;

У 5-го компонента Button изменить свойства и метод: Name на B\_Cancel;

Caption на Снять фильтр.

В методе Click кнопки написать код:

dm.ADOBooks.Filter:= ''; dm.ADOBooks.Filtered:= false;

Аналогично для остальных форм добавить элементы типа Button, которые будут запускать фильтры по соответствующим значениям полей текущей записи в объекте Grid. Пример интерфейса формы представлен на рис. 4.1.

Проверить работу приложения.

Создать форму, назвать (изменить свойство Name) FormProcedure.

В главной форме в коде добавить описание этой формы, для этого после ключевого слова USES в конце списка через запятую дописать название программного модуля, которое соответствует созданной форме.

В созданной форме в коде добавить описание модуля DM, для этого после ключевого слова USES в конце списка поставить запятую и дописать название программного модуля, например UNIT2, который соответствует созданной форме DM.

Добавить на главной форме в меню пункт с названием Работа с процедурами. В методе Click пункта меню написать код для запуска формы FormProcedure (см. пример кода в пункте 17 текущей лабораторной работы).

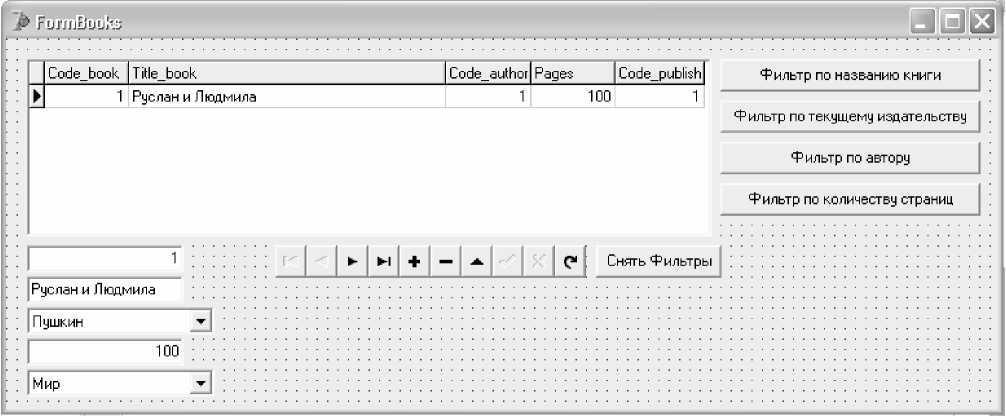


Рис. 4.1. Пример расположения компонентов на форме FormBooks

Подключить хранимую процедуру Count\_Books\_Itogo, выполненную в лабораторной работе №1, которая возвращает одно значение. На форму DM добавить компонент ADOStoredProc, переименовать компонент в ADOSP\_Count. У ADOSP\_Count изменить следующие свойства:

Connection на ADOConnection 1; ProcedureName на Count\_Books\_Itogo.

У компонента ADOSP\_Count выбрать свойство Parameters и создать 3 параметра:

для Count\_pages IN Int: Name на Count\_pages; Datatype на ftInteger; Direction на pdInput; Value-Type на

Integer;

для Title IN Char: Name на Title; Datatype на ftString; Direction на pdInput; Value-Type на

OleStr;

для Itogo OUT Int: Name на Itogo; Datatype на ftInteger;

Direction на pdOutput; Value-Type на Integer.

На форму FormProcedure добавить 3 компонента типа Edit (имена соответственно Edit1, Edit2, Edit3) и 1 компонент типа Button. Рядом с каждым компонентом Edit поставить Label и исправить их свойства Caption соответственно на «Введите количество страниц»,

«Введите название книги», «Количество найденных книг».

На кнопке поменять название на «Выполнить запрос». В методе Click кнопки написать следующий код: DM.ADOSP\_Count.Prepared;

try DM.ADOSP\_Count.Parameters.ParamByName('Count\_pages').Value:=StrToI nt(Edit1.Text); DM.ADOSP\_Count.Parameters.ParamByName('Title').Value:= Edit2.Text; DM.ADOSP\_Count.ExecProc;

Edit3.Text:= inttostr(DM.ADOSP\_Count.Parameters.ParamByName('Itogo').Value); finally DM.ADOSP\_Count.Prepared:=not(DM.ADOSP\_Count.Prepared); end;// try

Проверить работу приложения.

Для вывода данных из нескольких таблиц можно использовать

представления (View), которые создаются и хранятся на сервере.

Создать представление на сервере, которое будет хранить информацию о названии книги, авторе, издательстве. Для этого в утилите SQL \*Plus Worksheet напишите следующий код и выполните (F5): CREATE OR REPLACE VIEW admin\_books.view\_books

AS

SELECT a.name\_author AS "NAME\_AUTHOR1",

b.title\_book AS "TITLE\_BOOK1",c.publish AS "PUBLISH1" FROM (ADMIN\_BOOKS.AUTHORS a INNER JOIN ADMIN\_BOOKS.BOOKS b on

a.code\_author=b.code\_author)

INNER JOIN ADMIN\_BOOKS.PUBLISHING\_HOUSE c ON

b.code\_publish = c.code\_publish

Подключить созданное представление через ADOQuery. На форму DM добавить компонент ADOQuery, переименовать компонент в

ADOView\_Books. У ADOView\_Books изменить следующие свойства: Connection на ADOConnectionl;

SQL на

select \* from admin\_books.view\_books

Для компонента ADOView\_Books добавить на форму DM компонент DataSource и присвоить его свойствам следующие значения:

Name на Data\_Books\_All; DataSet на ADOView\_Books.

На форму FormProcedure добавить DBGrid, 1 компонент типа Button и 1 компонент типа Edit. Рядом с полем Edit поставить метку с текстом "Введите фамилию искомого автора". В свойстве Datasource компонента DBGrid выбрать Data\_Books\_All. На кнопке поменять название на «Выполнить запрос». В методе Click кнопки написать следующий код:

DM.ADOView\_Books.Open; IF trim(Edit4.Text)<>'' then begin

DM. ADOView\_Books.Filter:='name\_author1='+ QuotedStr(Edit4.Text); DM. ADOView\_Books. Filtered :=True ;

End else

DM. ADOV iew\_Books. Filtered :=False ; DB Grid 1 .Refresh;

Проверить работу приложения.

**Контрольные вопросы для самоподготовки**

Какие имеются инструменты для работы с данными в Delphi 7?

Для чего предназначена вкладка DataControls?

Для чего нужен компонент ADOConnection?

С помощью какого компонента идет подключение хранимой процедуры с клиентским приложением?

С помощью какого компонента идет подключение хранимой процедуры с клиентским приложением?

**Инструкционно-технологическая карта**

**для выполнения лабораторная работа №6**

**Тема:** **Разработка и администрирование БД.**

**Наименование работы: Импорт данных пользователя в базу данных Oracle**

**Цель:** с помощью операторов языка PL/SQL научиться создавать пользователя, который будет управлять всеми объектами создаваемой базы данных, схему, табличное пространство для хранения объектов базы данных и совокупность связанных таблиц, принадлежащих указанной базе данных в конкретной схеме

**Приобретаемые умения и знания:** Получения умений импорта данных пользователя в БД

**Формируемые компоненты:** **ПК 11.1-11.6 ОК 1-11**

**Норма времени:**4 часа

**Оборудование и программное обеспечения:** ПК, MS SQL Server, Oracle 9i

**Основные правила ТБ при выполнении работы:** ТБ при работе в компьютерном классе, соблюдать рабочую дисциплину

**Литература:** Кумскова, И. А. Базы данных: учебник для СПО / И. А. Кумскова.- М.: КНОРУС, 2016.-488 с.

**Контрольные вопросы при допуске:**

1. Что такое данные?
2. Что такое импорт?
3. Как осушествляется импорт данных?

***Задание*:** создать схему и таблицы БД с помощью операторов языка PL/SQL. В качестве примера выбрать БД «Книжное дело». Структура таблиц данной БД представлена в Приложении 1.

Порядок выполнения работы

Прежде чем создавать таблицы БД, необходимо создать схему (она будет создана автоматически при создании пользователя) и соответствующее табличное пространство, в котором будут храниться объекты БД.

Это можно сделать программно и интерактивно.

*Интерактивный способ создания* реализуется через утилиту

*Enterprise Manager Console.*

Для соединения с экземпляром БД Oracle при первом соединении выберите режим Launch standalone,

алее в появившемся окне дважды щелкните по экземпляру БД и воспользуйтесь учетной записью System с паролем по умолчанию manager и уровнем доступа SYSDBA (рис. 5.1).



Рис. 5.1 Аутентификация пользователя

Перед созданием табличного пространства необходимо создать пользователя, который будет играть роль администратора создаваемой БД. Для этого выполните действие, как показано на рис. 5.2

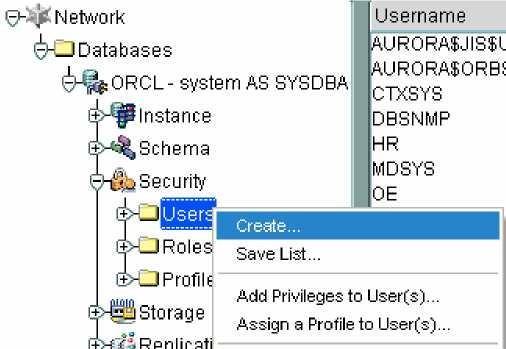


Рис. 5.2 Пример создания пользователя

Процесс создания пользователя в интерактивном режиме выглядит следующим образом (рис. 5.3):

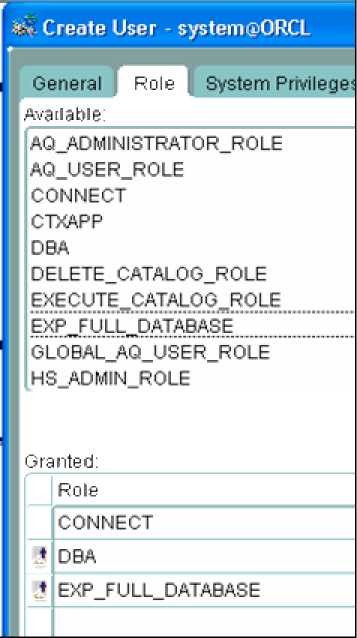


Рис. 5.3 Настройка привилегий пользователя

Теперь необходимо переподключиться к экземпляру БД под созданной учетной записью (рис. 5.4). 

Рис. 5.4 Смена соединения

Затем приступаем к созданию табличного пространства (рис. 5.5)

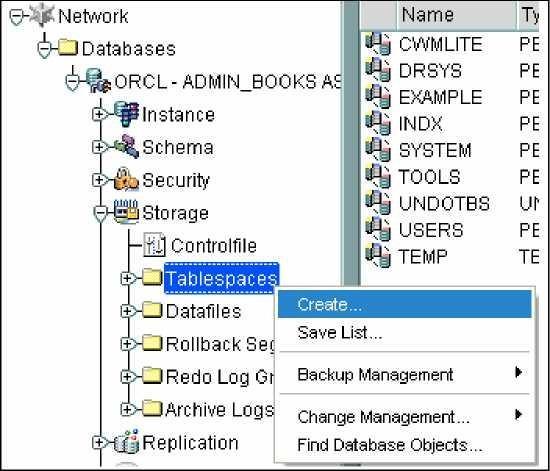


Рис. 5.5. Пример создания табличного пространства

Введем параметры, как показано на рис. 5.6

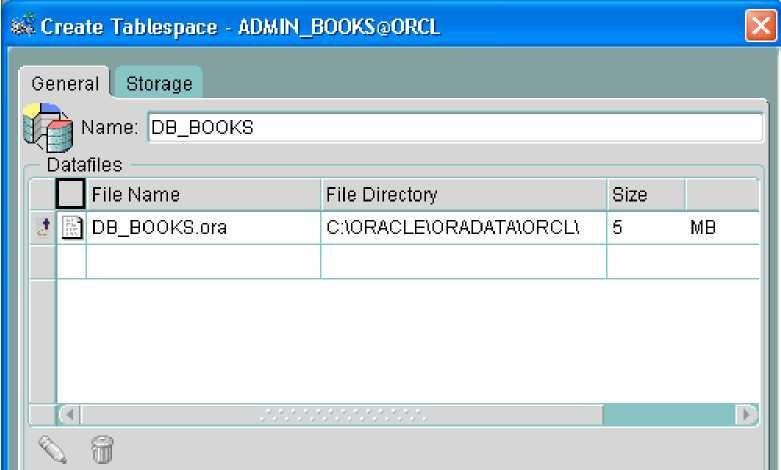


Рис. 5.6. Настройка свойств табличного пространства

* 1. Теперь переопределяем ранее созданного пользователя ADMIN\_BOOKS на работу только в этом табличном пространстве (рис. 5.7)

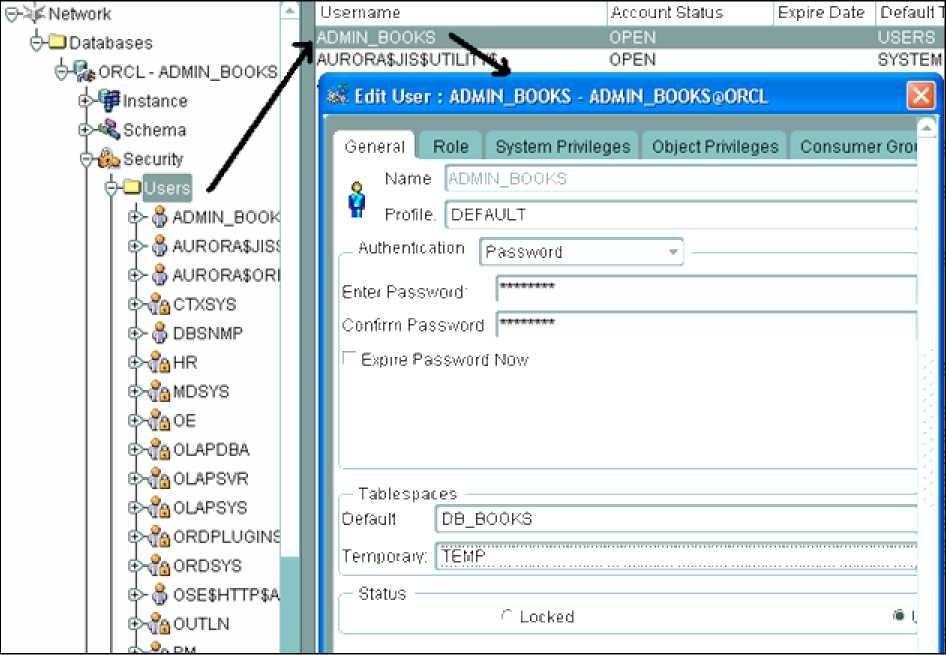


Рис. 5.7. Изменение свойств существующего пользователя

* 1. Затем можно приступать к созданию таблиц БД в табличном пространстве, как показано на рис. 5.8.
  2. Поочередно создаем таблицы, по примеру, показанному ниже на рис. 5.9

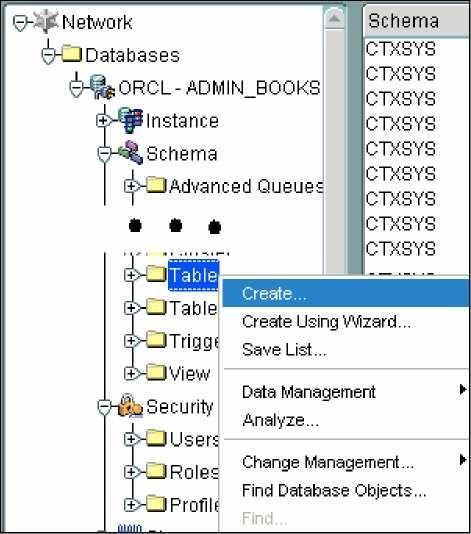


Рис. 5.8. Пример создания таблицы

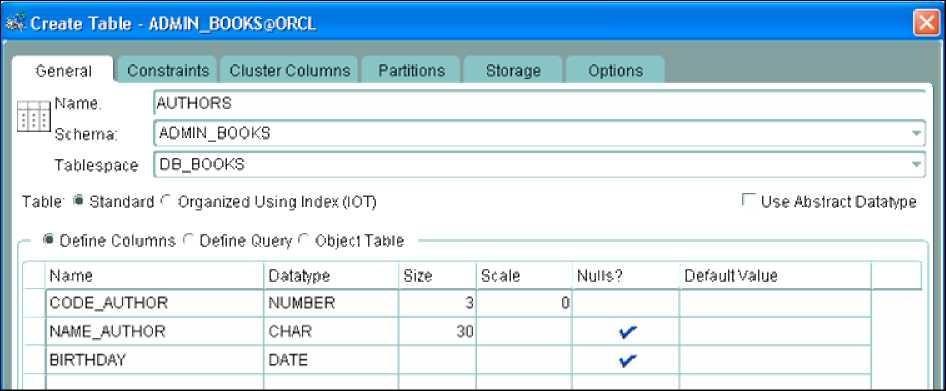


Рис. 5.9. Настройка структуры таблицы

* 1. Кроме того, при создании новой таблицы на вкладке Constraints Storage в качестве табличного пространства для хранения индекса выбираем DB\_BOOKS (рис. 5.10).

Те же самые операции можно выполнить программно, используя операторы SQL стандарта, реализованные в Oracle как операторы встроенного языка PL/SQL.

* 1. Для выполнения операторов можно запустить утилиту SQL

\*Plus Worksheet через меню Windows или через утилиту Enterprise Manager Console, выбрав на левой панели соответствующую пиктограмму, как показано на рис. 5.11.

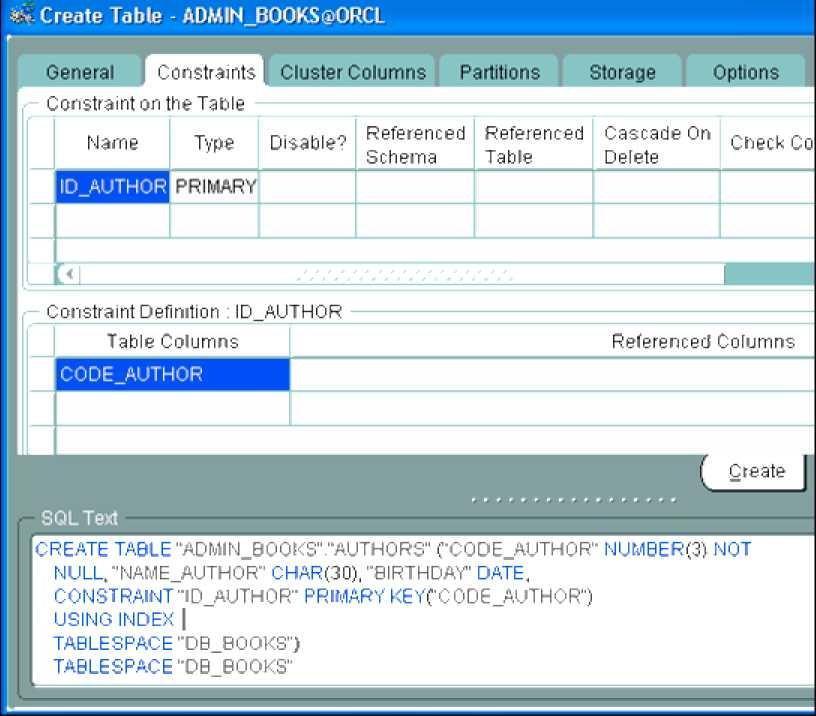


Рис. 5.10 Создание ключевого поля в таблице

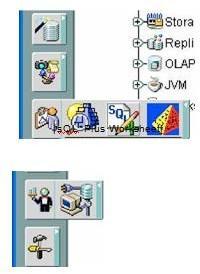


Рис. 5.11. Пример запуска утилиты SQL \* Plus Worksheet

При запуске SQL \*Plus Worksheet подключиться к серверу с помощью пользователя System с паролем по умолчанию manager и уровнем доступа SYSDBA, в качестве SERVICE ввести имя экземпляра БД, в данном при- мере это ORCL.

* 1. Вводим набор операторов для создания администратора создаваемой БД.

CREATE USER ”ADMIN\_BOOKS” PROFILE "DEFAULT"

IDENTIFIED BY "worlds" DEFAULT TABLESPACE "USERS"

TEMPORARY TABLESPACE "TEMP" ACCOUNT UNLOCK;

GRANT "CONNECT" TO "ADMIN\_BOOKS" WITH ADMIN OPTION; GRANT "DBA" TO "ADMIN\_BOOKS" WITH ADMIN OPTION; GRANT "EXP\_FULL\_DATABASE" TO "ADMIN\_BOOKS" WITH

ADMIN OPTION;

Пример ввода программного кода показан на рис. 5.12

* 1. Для запуска операторов на исполнение выделяем группу операторов мышкой и нажимаем F5. Если нет ошибок, то на каждый оператор система в нижней половине экрана выдаст сообщения:

«Пользователь создан», «Привилегии предоставлены».

* 1. Теперь необходимо переподключиться к экземпляру БД под созданной учетной записью, для этого выберите значок «Электрическая вилка» и введите в появившемся окне данные для подключения под новым пользователем.

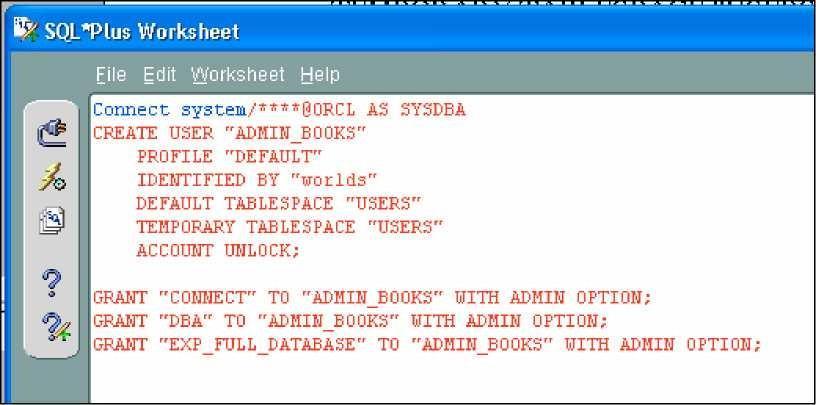


Рис. 5.12. Пример работы утилиты SQL \* Plus Worksheet

* 1. Приступаем к созданию табличного пространства программно (знак ; в конце оператора обязателен).

CREATE TABLESPACE "DB\_BOOKS" LOGGING

DATAFILE 'C:\ORACLE\ORADATA\ORCL\DB\_BOOKS.dbf SIZE 5M EXTENT MANAGEMENT LOCAL;

Для запуска оператора на исполнение выделяем его мышкой и нажимаем F5. Если нет ошибок, то на оператор система в нижней половине экрана выдаст сообщение: «Раздел создан».

* 1. Теперь переопределяем ранее созданного пользователя

ADMIN\_BOOKS на работу только в табличном пространстве DB\_BOOKS.ALTER USER "ADMIN\_BOOKS" DEFAULT TABLESPACE "DB\_BOOKS";

Для запуска оператора на исполнение выделяем его мышкой и нажимаем F5. Если нет ошибок, то на оператор система в нижней половине экрана выдаст сообщение: «Пользователь изменен».

Создаем таблицы, структура которых представлена Приложении 1.

CREATE TABLE "ADMIN\_BOOKS". "AUTHORS" ("CODE\_AUTHOR" NUMBER(3) NOT NULL, "NAME\_AUTHOR" CHAR(30), "BIRTHDAY" DATE, CONSTRAINT "ID\_AUTHOR" PRIMARY KEY("CODE\_AUTHOR") USING INDEX TABLESPACE "DB\_BOOKS") TABLESPACE "DB\_BOOKS";

CREATE TABLE "ADMIN\_BOOKS". "DELIVERIES" ("CODE\_DELIVERY" NUMBER(4) NOT NULL, "NAME\_DELIVERY" CHAR(35), "NAME\_COMPANY" CHAR(30) NOT NULL, "ADDRESS" VARCHAR2(100), "PHONE" CHAR(11)

NOT NULL, "INN" CHAR(13), CONSTRAINT "ID\_DELIVERY" PRIMARY KEY ("CODE\_DELIVERY") USING INDEX TABLESPACE "DB\_BOOKS") TABLESPACE "DB\_BOOKS";

CREATE TABLE "ADMIN\_BOOKS". "PUBLISHIN G\_HOU SE" ("CODE\_PUBLISH" NUMBER(4) NOT NULL, "PUBLISH" CHAR(30) NOT NULL, "CITY" CHAR(15), CONSTRAINT "ID\_PUBLISH" PRIMARY KEY("CODE\_PUBLISH") USING INDEX TABLESPACE "DB\_BOOKS") TABLESPACE "DB\_BOOKS";

CREATE TABLE "ADMIN\_BOOKS"."BOOKS" ("CODE\_BOOK" NUMBER(5) NOT NULL, "TITLE\_BOOK" VARCHAR2( 100) NOT NULL, "CODE\_AUTHOR" NUMBER(3) NOT NULL, "PAGES" NUMBER(4), "CODE\_PUBLISH" NUMBER(4) NOT NULL, CONSTRAINT "ID\_BOOK\_FK" PRIMARY KEY("CODE\_BOOK") USING INDEX TABLESPACE "DB\_BOOKS",

CONSTRAINT "ID\_AUTHOR\_FK" FOREIGN KEY("CODE\_AUTHOR") REFERENCES "ADMIN\_BOOKS"."AUTHORS"("CODE\_AUTHOR") ON DELETE CASCADE, CONSTRAINT "ID\_PUBLISH\_FK" FOREIGN KEY("CODE\_PUBLISH") REFERENCES

"ADMIN\_BOOKS"."PUBLISHING\_HOUSE"("CODE\_PUBLISH") ON DELETE CASCADE) TABLESPACE "DB\_BOOKS";

CREATE TABLE "ADMIN\_BOOKS". "PURCHASES" ("CODE\_BOOK" NUMBER(5) NOT NULL, "DATE\_ORDER" DATE NOT NULL, "CODE\_DELIVERY" NUMBER(4) NOT NULL, "TYPE\_PURCHASE" CHAR(1) NOT NULL, "COST" NUMBER(10,2) NOT NULL, "AMOUNT" NUMBER(4) NOT NULL, "CODE\_PURCHASE" NUMBER(5) NOT NULL, CONSTRAINT "ID\_PURCHASE" PRIMARY KEY("CODE\_PURCHASE") USING INDEX TABLESPACE "DB\_BOOKS", CONSTRAINT "ID\_BOOK\_FK2" FOREIGN KEY("CODE\_BOOK") REFERENCES "ADMIN\_BOOKS"."BOOKS"("CODE\_BOOK"), CONSTRAINT "ID\_DELIVERY\_FK" FOREIGN KEY("CODE\_DELIVERY") REFERENCES "ADMIN\_BOOKS"."DELIVERIES"("CODE\_DELIVERY")) TABLESPACE DB\_BOOKS";

Не забывайте: сначала создаются справочники, в которых есть только ключевые поля типа PRIMARY, а затем другие таблицы, в которых создаются внешние ключи типа FOREIGN KEY.

*Использованные операторы:*

PRIMARY KEY - признак создания ключевого поля.

FOREIGN KEY...REFERENCES... - признак создания поля связи с другой таблицей.

CREATE TABLE - команда создания таблицы в текущей БД.

*Общие положения*

В утилите Enterprise Manager Console создайте пример базы данных, как показано по ходу работы.

По индивидуальному варианту исходные таблицы привести к максимальной форме нормализации, для каждой таблицы определить название, типы полей, определить ключевые поля и при необходимости поля связи (внешние ключи), определить имя табличного пространства, в котором вы будете создавать объекты БД по всему комплексу лабораторных работ, а также определить имя пользователя и пароль.

В утилите Enterprise Manager Console создать вашего пользователя, табличное пространство и таблицы БД. Перед созданием очередного объекта (т.е. до нажатия клавиши Create) скопируйте SQL код создаваемого объекта, который можно увидеть, нажав клавишу Show SQL. Сохранить последовательно операторы в файле программы с названием ФамилияСтудента\_ЛАб\_5\_№варианта

*Требования к содержанию отчета*: итоги лабораторной работы представить в виде пояснений и копий экранов выполнения основных заданий, описанных в инструкции по выполнению лабораторной работы.

**Контрольные вопросы для самоподготовки**

1. Какие утилиты входят в состав Oracle 9i?
2. Охарактеризуйте основные этапы создания таблиц интерактивным способом через утилиту Enterprise Manager Console.
3. В чем разница между первичным ключом и уникальным ключом?
4. В чем заключается процедура создания пользователя и настройка привилегий пользователя?
5. Как создать и настроить свойства табличного пространства через утилиту Enterprise Manager Console?
6. Как создать и настроить структуру таблицы через утилиту Enterprise Manager Console?
7. В чем проявляются особенности создания таблиц программным способом через утилиту \* Plus Worksheet?
8. Что позволяет делать SQL \*Plus?

**Инструкционно-технологическая карта**

**для выполнения лабораторная работа №7**

**Тема:** **Разработка и администрирование БД.**

**Наименование работы: Выполнение настроек для автоматизации обслуживания базы данных**

**Цель:** изучить и освоить все способы заполнения данными таблиц Oracle

**Формируемые компоненты:** **ПК 11.1-11.6 ОК 1-11**

**Норма времени:**4 часа

**Оборудование и программное обеспечения:** ПК, MS SQL Server, Oracle 9i

**Основные правила ТБ при выполнении работы:**

-плавно нажимайте на клавиши, не допуская резких ударов;

-не пользуйтесь клавиатурой, если не подключено напряжение;

-работайте на клавиатуре чистыми руками;

-никогда не пытайтесь самостоятельно устранить неисправность в работе аппаратуры;

Строго запрещается:

-трогать разъемы соединительных кабелей;

-прикасаться к питающим проводам и устройствам заземления;

-прикасаться к экрану и к тыльной стороне монитора, клавиатуры;

-включать и отключать аппаратуру без указания преподавателя;

-Класть диск, книги, тетради на монитор и клавиатуру;

-Работать во влажной одежде и влажными руками

**Литература:** Кумскова, И. А. Базы данных: учебник для СПО / И. А. Кумскова.- М.: КНОРУС, 2016.-488 с.

**Контрольные вопросы при допуске:**

1. Что акое БД?
2. Что такое автоматизация?
3. Для чего в БД автоматизация?

**Порядок** **выполнения работы**

На примере созданного табличного пространства, таблиц и пользователя продолжим выполнение работы.

Для заполнения таблиц данными можно воспользоваться несколькими способами:

Заполнение таблиц непосредственно через утилиту Enterprise Manager Console.

Заполнение с помощью программного кода через утилиту SQL

\*Plus Worksheet.

Заполнение данными через интерфейс внешнего приложения или клиентского приложения, например, создав приложение в MS Access, Borland Delphi, PHP и т.д.

*Пошагово рассмотрим каждый пример заполнения.*

В первом случае запускаем утилиту Enterprise Manager Console.

Для соединения с экземпляром БД Oracle выберите режим Launch standalone, далее в появившемся окне дважды щелкните по экземпляру БД и воспользуйтесь учетной записью ADMIN\_BOOKS с паролем worlds и уровнем доступа SYSDBA.

В разделе Schema выбрать объекты типа Table, далее выбрать схему пользователя ADMIN\_BOOKS, далее в правой части экрана на соответствующей таблице в контекстном меню выбрать пункт Table data editor.

Далее можно заполнить таблицу необходимым количеством записей, для сохранения изменений нажать Apply.

Формат ввода даты по умолчанию для операционной среды с русскими настройками dd-Mon-yyyy, где Mon пишется первыми тремя буквами на- звания месяца по-русски.

Пример ввода представлен на рис. 6.1

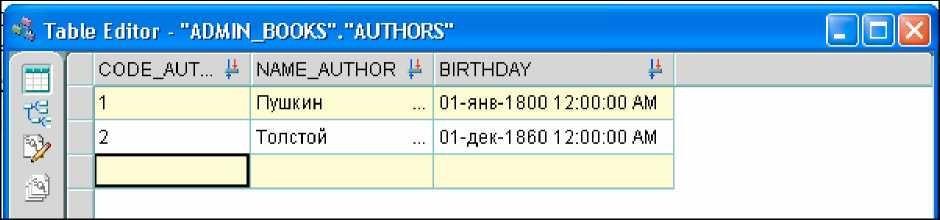


Рис. 6.1. Пример заполнения таблицы

Во втором случае программного заполнения таблиц запускаем утилиту SQL \*Plus Worksheet.

Для соединения с экземпляром БД Oracle выберите режим Launch standalone, далее в появившемся окне дважды щелкните по экземпляру БД и воспользуйтесь учетной записью ADMIN\_BOOKS с паролем worlds и уровнем доступа SYSDBA

INSERT INTO ”ADMIN\_BOOKS”.”AUTHORS” (”CODE\_AUTHOR”, ”NAME\_AUTHOR”, "BIRTHDAY”) VALUES (3, 'Лермонтов', TO\_DATE(' 12-05-1810') );

Или в случае совпадения заполняемых значений с порядком столбцов в таблице можно названия столбцов пропустить

INSERT INTO "ADMIN\_BOOKS "."AUTHORS" VALUES (4, 'Лермонтов', TO\_DATE('12-05-1810') );

Последовательно наберите в SQL \*Plus Worksheet операторы INSERT, разделяя их точкой с запятой, а также используйте в конце оператор Commit для фиксации изменений в таблице.

Пример:

INSERT INTO authors VALUES(6, 'Гончаров', TO\_DATE('12-03-

1798','dd-mm-yyyy'));

--другие операторы INSERT; COMMIT;

Для выполнения нажмите F5. Если нет ошибок, то на каждый оператор система в нижней половине экрана выдаст сообщение: «1 строка создана». На оператор Commit система выдаст сообщение:

«Фиксация обновлений завершена».

Для просмотра результата заполнения можно воспользоваться оператором SELECT, например

SELECT \* FROM ”ADMIN\_BOOKS”.”AUTHORS”;

или

SELECT \* FROM ADMIN\_B OOKS.AUTHORS;

В третьем случае заполнения таблиц через клиентское приложение создадим соединение с объектами Oracle через ODBC драйвер.

Для этого в операционной среде через ПУСК зайти в Панель управления, далее в разделе Производительность и обслуживание выбрать подраздел Администрирование, затем выбрать ссылку на утилиту Источники данных (ODBC). Далее выбрать тип «Системный источник данных» (System Name или System DSN), добавить новый источник данных, выбрав в появившемся списке драйверов Oracle. В следующем окне в качестве имени источника данных (Data Source Name) написать DB\_BooksDSN, в поле TSN Service Name указать название вашего экземпляра БД, в данном примере это ORCL, в поле User ID указать ADMIN\_BOOKS. Проверить соединение, нажав Test Connection. В случае успешной проверки сохранить результат.

Теперь можно подключаться к БД через любой интерфейс, рассмотрим пример работы через MS Access.

Создадим новый проект БД в MS Access. В разделе Таблицы в контекстном необходимо выбрать Связь с таблицами. В появившемся окне в качестве типа файлов выбрать ODBC Databases. Далее на второй вкладке Источники данных компьютера (Machine Source Name) выбрать DB\_BooksDSN. Далее в окне Oracle ODBC Driver Connect ввести пароль доступа. При успешном прохождении аутентификации появится окно Связь с таблицами, в котором необходимо выбрать пять таблиц из схемы ADMIN\_BOOKS. Если все выполнено правильно, то результат будет таким, как показано на рис. 6.2.

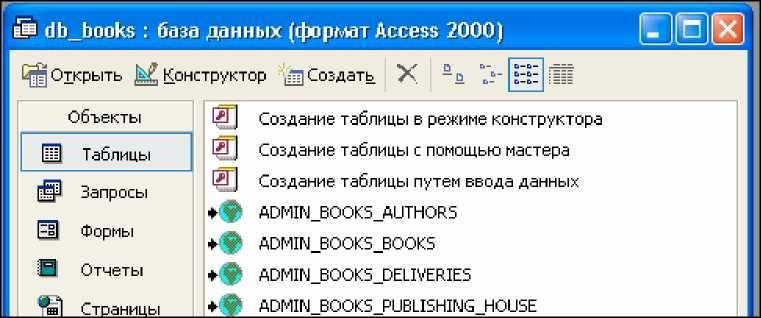


Рис.6.2. Пример связи с таблицами Oracle в Access

Теперь можно создать формы ввода данных и приступить к заполнению таблиц. Например, для таблицы Книги можно создать следующую форму (рис. 6.3)

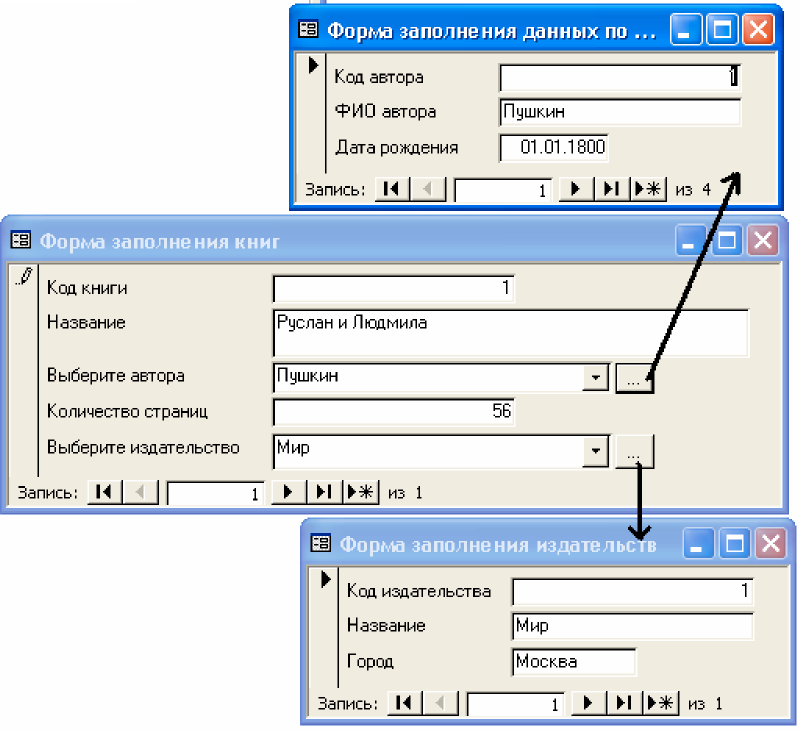


Рис. 6.3. Пример созданных форм в Access для ввода данных в таблицы Oracle

Для контроля ввода значений в отдельные столбцы таблиц можно использовать специальные объекты Domain и Sequence.

*Создание и использование последовательностей*

Последовательность (SEQUENCE) - это объект, обеспечивающий генерацию уникальных последовательностей в условиях многопользовательского асинхиронного режима доступа.

Пример создания последовательности:

CREATE SEQUENCE ’’ADMIN\_BOOKS’’.’’ID\_AUTHOR\_SEQ’’ INCREMENT BY 1 START WITH 1 MAXVALUE 99999

MINVALUE 1 NOCYCLE CACHE 20 NOORDER;

Для обращения к значениям последовательности используют следующие операторы:

sequence\_name. CURRV AL sequence\_name.NEXTV AL

Пример использования последовательности:

INSERT INTO ”ADMIN\_BOOKS”.”AUTHORS” (”CODE\_AUTHOR”, ”NAME\_AUTHOR”, "BIRTHDAY”) VALUES

(ADMIN\_BOOKS.ID\_AUTHOR\_SEQ.NextVal, 'Лермонтов', TO\_DATE('12-05-1810’) );

Варианты заданий к лабораторной работе №6

В утилите SQL \*Plus Worksheet по вашим вариантам баз данных, которые были реализованы в лабораторной работе №5, создать для полей типа Primary Key последовательности с помощью оператора CREATE SEQUENCE.

В утилите SQL \*Plus Worksheet создать файл, в котором написать последовательность операторов Insert для заполнения таблиц данными (по 10 операторов на каждую из таблиц).

Сохранить все операторы CREATE SEQUENCE и INSERT в файле программы с названием ФамилияСтудента\_ЛАб\_2\_№варианта.

*Требования к содержанию отчета*: итоги лабораторной работы представить в виде пояснений и копий экранов выполнения основных заданий, описанных в инструкции по выполнению лабораторной работы.

**Контрольные вопросы для самоподготовки**

В чем заключается процедура заполнения таблиц непосредственно через утилиту Enterprise Manager Console?

В чем заключается процедура заполнения таблиц с помощью программного кода через утилиту SQL \*Plus Worksheet?

В чем заключается процедура заполнения данными через интерфейс внешнего приложения или клиентского приложения, например, создав приложение в MS Access, Borland Delphi, PHP?

Что такое последовательность?

Как создать и использовать объект Sequence для контроля ввода значений в отдельные столбцы таблиц?

**Инструкционно-технологическая карта**

**для выполнения лабораторная работа №7**

**Тема:** **Разработка и администрирование БД.**

**Наименование работы: Мониторинг работы сервера**

**Цель:** научиться создавать профили пользователей и ролей для разграничения прав доступа к объектам создаваемой БД, а также извлекать системную информацию о пользователях и объектах для последующего использования в удаленном управлении правами доступа к объектам экземпляра БД.

**Формируемые компоненты:** **ПК 11.1-11.6 ОК 1-11**

**Норма времени:**4 часа

**Оборудование и программное обеспечения:** ПК, MS SQL Server, Oracle 9i

**Основные правила ТБ при выполнении работы:** -плавно нажимайте на клавиши, не допуская резких ударов;

-не пользуйтесь клавиатурой, если не подключено напряжение;

-работайте на клавиатуре чистыми руками;

-никогда не пытайтесь самостоятельно устранить неисправность в работе аппаратуры;

Строго запрещается:

-трогать разъемы соединительных кабелей;

-прикасаться к питающим проводам и устройствам заземления;

-прикасаться к экрану и к тыльной стороне монитора, клавиатуры;

-включать и отключать аппаратуру без указания преподавателя;

-Класть диск, книги, тетради на монитор и клавиатуру;

-Работать во влажной одежде и влажными руками

**Литература:** Кумскова, И. А. Базы данных: учебник для СПО / И. А. Кумскова.- М.: КНОРУС, 2016.-488 с.

**Контрольные вопросы при допуске:**

1. Что такое сервер?
2. Что такое мониторинг?
3. Что такое профель пользователя?

**Порядок выполнения работы**

Для просмотра информации о пользователе можно воспользоваться оператором:

SELECT username, user\_id, password, account\_status, default\_tablespace,

temporary\_tablespace, profile FROM dba\_users

WHERE username = 'SYSTEM';

Следующий пример позволяет определить список привилегий на работу с объектами:

SELECT sysauth$.\*, table\_privilege\_map.name, user$.name FROM sysauth$,table\_privilege\_map, user$

WHERE sysauth$.grantee#=user$.user# and table\_privilege\_map.privilege = sysauth$.privilege# and

user$.name='ADMIN\_BOOKS';

или, используя представление SYS.dba\_sys\_privs:

SELECT \* FROM sys.dba\_sys\_privs WHERE grantee='ADMIN\_BOOKS';

Следующий пример позволяет определить список системных привилегий введенного пользователя:

SELECT sysauth$.\*, system\_privilege\_map.name, user$.name FROM sysauth$, system\_privilege\_map, user$

GRANTEE GRANTED\_ROLE ADM

|  |  |  |
| --- | --- | --- |
| ADMIN\_BOOKS | CONNECT | YESYES |
| ADMIN\_BOOKS | DBA | YESYES |
| ADMIN\_BOOKS | EXP\_FULL\_DATABASE | YESYES |

Для определения списка таблиц, собственником которых является конкретный пользователь, необходимо выполнить следующий SQL запрос:

SELECT table\_name FROM all\_all\_tables WHERE owner = 'ADMIN\_BOOKS';

Результат:

TABLE\_NAME AUTHORS BOOKS

DELIVERIES

PUBLISHING\_HOUSE PURCHASES

Для прямого получения информации о пользователе можно обратиться к системной таблице SYS.USER$, например:

SELECT \* FROM sys.user$ WHERE name = 'ADMIN\_BOOKS'

Далее определим список всех объектов пользователя, собственником которых он является, для этого:

Определим код пользователя (колонка USER# в таблице SYS.USER$):

SELECT user#, name FROM sys.user$ WHERE name = 'ADMIN\_BOOKS';

Результат:

USER# NAME

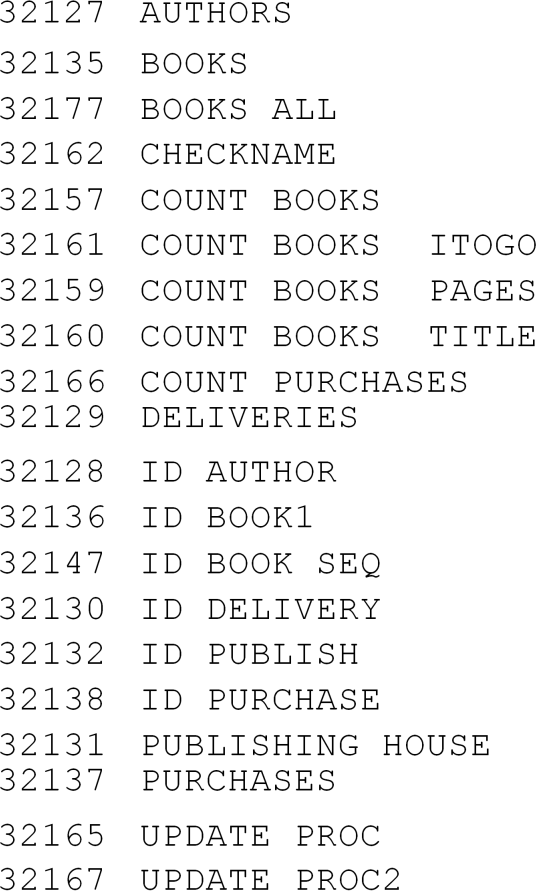
67 ADMIN\_BOOKS

Далее определим перечень объектов, собственником которых является пользователь:

SELECT OBJ#, NAME FROM SYS.OBJ$ WHERE OWNER# = 67;

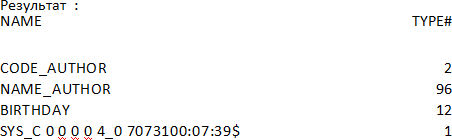
*где 67 - идентификатор пользователя ADMIN\_BOOKS, определенный по предыдущему запросу.*

Результат:

OBJ# NAME

20 строк выбрано

Можно просмотреть состав каждого объекта-таблицы, например таблицы AUTHORS, используя ее идентификационный номер, который определили в предыдущем запросе:

SELECT name, type# FROM sys.col$ WHERE obj# = '32127';

или, используя имя объекта:

SELECT c.name, c.type# FROM sys.col$ c INNER JOIN sys.obj$ o ON c.obj#= o.obj# WHERE o.name = 'AUTHORS’;

*Создание ролей*. Для упрощения управления правами доступа в системе создаются роли, которые затем можно назначать группе пользователей.

Создадим для нашего примера роли библиотекаря (LIBRAR) и читателя (READER).

*Пример создания роли библиотекаря:*

CREATE ROLE "LIBRAR" *IDENTIFIED BY "LIBRAR";*

*Пример* создания роли читателя:

CREATE ROLE "READER" IDENTIFIED BY "READER";

Библиотекарь должен обладать правами на чтение, удаление, изменение, добавление во все таблицы схемы ADMIN\_BOOKS, а также должен иметь возможность инициировать сессию и запускать на исполнение процедуры и функции схемы ADMIN\_BOOKS. Поэтому роли библиотекаря из системных привилегий назначаем CREATE SESSION, а из привилегий доступа к объектам назначаем DELETE, INSERT, UPDATE, SELECT, поз- же в лабораторной работе №5 для созданных процедур и функций добавить EXECUTE.

*Назначение системных привилегий*

Читатель должен обладать правами на чтение из таблиц ADMIN\_BOOKS.AUTHORS, ADMIN\_BOOKS.BOOKS,

ADMIN\_BOOKS.PUBLISHING\_HOUSE, а также должен иметь возможность инициировать сессию и запускать на исполнение процедуры и функции схемы ADMIN\_BOOKS. Поэтому роли читателя из системных привилегий назначаем CREATE SESSION, а из привилегий доступа к объектам назначаем SELECT.

*Синтаксис оператора представления привилегий:*

GRANT <привилегия>, ... ON < объект >, ...

TO <имя>

[WITH grant option] ;

Атрибут with grant option дает право пользователю самому раздавать права, которые он получил.

С помощью оператора grant для каждого пользователя формируется список привилегий, привилегии управляют работой сервера данных с точки зрения защиты данных. Выполнению каждой транзакции предшествует проверка привилегий пользователя, сеанс которого породил транзакцию.

Пример: grant select, update (Sales, num) ON Sales\_data TO userl with grant option;

Пользователь, предоставивший привилегию другому, называется грантор (grantor — предоставитель). Привилегия является предоставляемой, если право на нее можно предоставить другим пользователям.

PUBLIC — имя пользователя, который распространяет привилегии на все множество зарегистрированных в системе пользователей.

Роль библиотекаря названа LIBRAR с аналогичным паролем LIBRAR.

Операторы назначения прав доступа для роли библиотекаря

представлены ниже:

GRANT DELETE, INSERT, UPDATE, SELECT

ON ”ADMIN\_BOOKS’’.’’BOOKS’’ TO "LIBRAR"; GRANT DELETE, INSERT, UPDATE, SELECT

ON "ADMIN\_BOOKS"."AUTHORS" TO "LIBRAR"; GRANT DELETE, INSERT, UPDATE, SELECT

ON "ADMIN\_BOOKS"."DELIVERIES" TO "LIBRAR"; GRANT DELETE, INSERT, UPDATE, SELECT

ON "ADMIN\_BOOKS"."PUBLISHING\_HOUSE" TO "LIBRAR"; GRANT DELETE, INSERT, UPDATE, SELECT

ON "ADMIN\_BOOKS"."PURCHASES" TO "LIBRAR"; GRANT CREATE SESSION TO "LIBRAR";

Commit;

Роль читателя названа READER с аналогичным паролем READER. Операторы назначения прав доступа для роли читателя представлены ниже:

GRANT SELECT ON "ADMIN\_BOOKS"."BOOKS" TO "READER";

GRANT SELECT ON "ADMIN\_BOOKS"."AUTHORS" TO "READER";

GRANT SELECT ON "ADMIN\_BOOKS"."PUBLISHING\_HOUSE" TO "READER";

GRANT CREATE SESSION TO "READER";

Commit;

*Создание пользователей с определенной ролью*

Пример создания библиотекаря Ivanov\_Lib и присвоения роли: CREATE USER "Ivanov\_Lib"

PROFILE "DEFAULT"

IDENTIFIED BY "Ivanov\_Lib" DEFAULT TABLESPACE "DB\_BOOKS" TEMPORARY TABLESPACE "TEMP" ACCOUNT UNLOCK;

GRANT "LIBRAR" TO "Ivanov\_Lib";

Commit;

Пример создания читателя Petrov\_Read и присвоения роли: CREATE USER "Petrov\_Read"

PROFILE "DEFAULT"

IDENTIFIED BY "Petrov\_Read" DEFAULT TABLESPACE "DB BOOKS" TEMPORARY TABLESPACE "TEMP

ACCOUNT UNLOCK;

GRANT "READER" TO "Petrov\_Read"; Commit;

Управление паролями

Если система управления паролями доступна, то пользователи не могут изменить пароли командой ALTER USER. Пользователь должен изменять пароли, используя SQL\*Plus или OCIPasswordChange, в которых принимаются в качестве параметров старый и новый пароли.

Синтаксис оператора отмены привилегий REVOKE [with grant option]

< привилегии >,... ON < объект >,...

FROM <имя\_пользователя>;

Предложение with grant option сохраняет за пользователем перечисленные привилегии, но отменяет его право передавать их кому- либо другому.

Пример:

REVOKE SELECT ON "ADMIN\_BOOKS"."AUTHORS" FROM "READER";

Оператор изымания роли у пользователя:

Revoke <список ролей> from <список пользователей>. Пример:

REVOKE READER FROM "Petrov\_Read";

*Варианты заданий к лабораторной работе №7 Общие положения*

В утилите SQL \*Plus Worksheet выполнить примеры, которые даны по ходу работы.

По индивидуальному варианту базы данных, которая выполнена в лабораторной работе №5, определить 2-3 должностных лица, которые могут работать с таблицами БД. Для каждого должностного лица определить набор привилегий, которыми он может пользоваться.

В утилите SQL \*Plus Worksheet создать под каждое должностное лицо соответствующую роль, наделить эту роль определенными привилегиями.

Далее создать по одному пользователю на каждую должность и присвоить им соответствующие роли.

Сохранить последовательно операторы с указанием заданий в файле с названием ФамилияСтудента\_ЛАб\_7\_№варианта\_общее, результаты выполнения задания по варианту сохранить в файле ФамилияСтудента\_ЛАб\_7\_результаты. Операторы создания ролей, привилегий и пользователей сохранить в файле с названием ФамилияСтудента\_ЛАб\_7\_№варианта\_роли.

*Требования к содержанию отчета*: итоги лабораторной работы представить в виде пояснений и копий экранов выполнения основных заданий, описанных в инструкции по выполнению лабораторной работы.

**Контрольные вопросы для самоподготовки**

Как можно описать системные таблицы и представления в Oracle?

Какие операторы для просмотра информации о пользователе вы, знаете?

Какие операторы для определения списка привилегий на работу с объектами вам знакомы?

Что такое роль?

Можно ли изменить пароли командой ALTER USER в доступной системе управления паролями?

Какую проверку необходимо провести для обеспечения безопасности работы экземпляра БД после инсталляции программного продукта?

В какой схеме хранятся таблицы с информацией о правах доступа к системе?

Опишите синтаксис оператора представления и отмены привилегий.

**Инструкционно-технологическая карта**

**для выполнения лабораторная работа №8**

**Тема:** **Организация защиты данных в хранилищах.**

**Наименование работы: Выполнение резервного копирования**

**Цель:** научиться создавать и использовать хранимые процедуры и функции в схемах.

**Формируемые компоненты:** **ПК 11.1-11.6 ОК 1-11**

**Норма времени:**4 часа

**Оборудование и программное обеспечения:** ПК, MS SQL Server, Oracle 9i

**Основные правила ТБ при выполнении работы:**

-плавно нажимайте на клавиши, не допуская резких ударов;

-не пользуйтесь клавиатурой, если не подключено напряжение;

-работайте на клавиатуре чистыми руками;

-никогда не пытайтесь самостоятельно устранить неисправность в работе аппаратуры;

Строго запрещается:

-трогать разъемы соединительных кабелей;

-прикасаться к питающим проводам и устройствам заземления;

-прикасаться к экрану и к тыльной стороне монитора, клавиатуры;

-включать и отключать аппаратуру без указания преподавателя;

-Класть диск, книги, тетради на монитор и клавиатуру;

-Работать во влажной одежде и влажными руками **Литература:** Кумскова, И. А. Базы данных: учебник для СПО / И. А. Кумскова.- М.: КНОРУС, 2016.-488 с.

**Контрольные вопросы при допуске:**

1. Что такое БД?
2. Что такое резервноее копирования?
3. Для чего нужно резервное копирования?

***Вариант 1***

1. Вывести список сотрудников, у которых есть хотя бы один ребенок.
2. Вывести список детей, которым выдали подарки в указанный период.
3. Вывести список родителей, у которых есть несовершеннолетние дети.
4. Вывести информацию о подарках со стоимостью больше указанного числа, отсортированных по дате.

***Вариант 2***

1. Вывести список приборов с указанным типом.
2. Вывести количество отремонтированных приборов и общую стоимость ремонтов у указанного мастера.
3. Вывести список владельцев приборов и количество их обращений, отсортированный по количеству обращений по убыванию.
4. Вывести информацию о мастерах с разрядом больше указанного числа или с датой приема на работу меньше указанной даты.

***Вариант 3***

1. Вывести список цветков с указанным типом листа.
2. Вывести список кодов продаж, по которым продано цветов на сумму больше указанного числа.
3. Вывести дату продажи, сумму, продавца и цветок по указанному коду продажи.
4. Вывести список цветов и сорт для цветов с высотой больше указанного числа или цветущий.

***Вариант 4***

1. Вывести список лекарств с указанным показанием к применению.
2. Вывести список дат поставок, по которым продано больше указанного числа одноименного лекарства.
3. Вывести дату поставки, сумму, ФИО руководителя от поставщика и название лекарства по коду поступления больше указанного числа.
4. Вывести список лекарств и единицы измерения для лекарств с количеством в упаковке больше указанного числа или кодом лекарства меньше определенного значения.

***Вариант 5***

1. Вывести список сотрудников с указанной должностью.
2. Вывести список списанного оборудования по указанной причине.
3. Вывести дату поступления, название оборудования, ФИО ответственного и дату списания для оборудования, списанного в указанный период.
4. Вывести список оборудования с указанным типом или с датой поступления больше определенного значения.

***Вариант 6***

1. Вывести список блюд с весом больше указанного числа.
2. Вывести список продуктов, в названии которых встречается указанный фрагмент слова.
3. Вывести объем продукта, название блюда, название продукта с кодом блюда от указанного начального значения по определенному конечному значению.
4. Вывести порядок приготовления блюда и название блюда с количеством углеводов больше определенного значения или количеством калорий больше указанного значения.

***Вариант 7***

1. Вывести список сотрудников с указанной должностью.
2. Вывести список документов, в содержании которых встречается указанный фрагмент слова.
3. Вывести дату регистрации, тип документа, ФИО регистратора и название организации для документов, зарегистрированных в указанный период.
4. Вывести список зарегистрированных документов с определенным типом документа или с датой регистрации больше указанного значения.

***Вариант 8***

1. Вывести список сотрудников с указанной причиной увольнения.
2. Вывести список документов с датой регистрации в указанный период.
3. Вывести дату регистрации, причину увольнения, ФИО сотрудника для документов, зарегистрированных в указанный период.
4. Вывести список зарегистрированных документов с кодом документа в указанном диапазоне.

***Вариант 9***

1. Вывести список сотрудников, бравших отпуск указанного типа.
2. Вывести список документов с датой регистрации в указанный период.
3. Вывести дату регистрации, тип отпуска, ФИО сотрудника для документов, зарегистрированных в указанный период.
4. Вывести список зарегистрированных документов с кодом документа в указанном диапазоне.

***Вариант 10***

1. Вывести список сотрудников с указанной должностью.
2. Вывести список документов, в содержании которых встречается указанный фрагмент слова.
3. Вывести дату регистрации, тип документа, ФИО отправителя и название организации для документов, зарегистрированных в указанный период.
4. Вывести список зарегистрированных документов с указанным типом документа или с кодом документа меньше определенного значения

*Общие положения*

В лабораторной работе в примерах тела процедур и функций написаны так, что их можно полностью копировать, вставлять в утилиту SQL \*Plus Worksheet и запускать (F5) для их создания на сервере. Если создавать процедуру или функцию в Enterprise MANAGER Console через мастер создания, то первая строка с конструкцией CREATE OR REPLACE PROCEDURE имя\_процедуры или CREATE OR REPLACE FUNCTION имя\_функции не копируется, а само имя\_процедуры или имя\_функции заносится в поле Name.

Создать примеры хранимых процедур и функций в вашем табличном пространстве. Проверьте их работу.

По вариантам на базе таблиц, созданных в лабораторной работе

№5, в утилите SQL \*Plus Worksheet создайте процедуры и функции,

*Требования к содержанию отчета*: итоги лабораторной работы представить в виде пояснений и копий экранов выполнения основных заданий, описанных в инструкции по выполнению лабораторной работы., также каждую программу по созданным процедурам и функциям (по вариантам) сохранять на диске в отдельном файле с названием ФамилияСтудента\_ЛАб\_8\_№\_варианта\_№\_задания.

**Контрольные вопросы для самоподготовки**

Что такое хранимая процедура?

Приведите примеры создания хранимых процедур в утилите SQL \*Plus Worksheet

Приведите примеры создания хранимых процедур в утилите Enterprise MANAGER Console.

В чем отличие процедуры от функции?

Какие привилегии нужно иметь, чтобы создать процедуру или функцию?

Опишите синтаксис оператора определения процедуры и функции Oracle.

Как удалить хранимую процедуру или функцию?

**Инструкционно-технологическая карта**

**для выполнения лабораторная работа №9**

**Тема:** **Организация защиты данных в хранилищах.**

**Наименование работы: Восстановление базы данных из резервной копии**

**Цель:** научиться создавать формы отчетных документов по данным БД с применением Rave Report..

**Формируемые компоненты:** **ПК 11.1-11.6 ОК 1-11**

**Норма времени:**4 часа

**Оборудование и программное обеспечения:** ПК, MS SQL Server, Oracle 9i, Delphi

**Основные правила ТБ при выполнении работы:**

-плавно нажимайте на клавиши, не допуская резких ударов;

-не пользуйтесь клавиатурой, если не подключено напряжение;

-работайте на клавиатуре чистыми руками;

-никогда не пытайтесь самостоятельно устранить неисправность в работе аппаратуры;

Строго запрещается:

-трогать разъемы соединительных кабелей;

-прикасаться к питающим проводам и устройствам заземления;

-прикасаться к экрану и к тыльной стороне монитора, клавиатуры;

-включать и отключать аппаратуру без указания преподавателя;

-Класть диск, книги, тетради на монитор и клавиатуру;

-Работать во влажной одежде и влажными руками

**Литература:** Кумскова, И. А. Базы данных: учебник для СПО / И. А. Кумскова.- М.: КНОРУС, 2016.-488 с.

**Контрольные вопросы при допуске:**

1. Что такое БД?
2. Что такое резервная копия?
3. Для чего нужно востоновления БД из резервной копи?

**Пояснения к выполнению работы**

В данной работе используем пример базы даны c названием DB\_Books, которая была создана в лабораторной работе №5.

Рассмотрим последовательность шагов, которую требуется выполнить приложению базы данных.

В Delphi 7 есть несколько способов создания отчетов, познакомимся с компонентами Rave.

*Задание 1* Создание отчета в табличной форме по запросу, который выбирает из таблицы Books все поля, кроме кодов, из таблицы Publish\_house - название издательства и место издательства, из таблицы Authors. - имя автора.

Сначала необходимо создать приложение базы данных в Delphi. Процедура создания приложения подробно описана в лабораторной №4. Если данный проект уже был выполнен в лабораторной работе с СУБД MS SQL Server, то в нем необходимо лишь перенастроить свойство Connection String на соединение с Oracle и перекомпилировать проект

Откроем приложение базы данных Delphi, для которого требуется добавить простой отчет.

В проекте на главной форме в меню добавить пункт меню Отчеты, а также подпункты:

Отчет в табличной форме; Отчет в свободной форме;

Отчет с группировкой по двум таблицам.

В проекте на форму DM добавить компонент ADOQUERY1, у которого изменить SQL на соответствующий запрос (см. текст задания):

select \* from books, authors, publishing\_house where books.code\_author = authors.code\_author AND

books.code\_publish = publishing\_house.code\_publish,

а свойство Connection на ADOConnection1. Активизировать запрос.

В проекте на форму DM с вкладки Rave поместим компонент RvDataSetConnection. У свойства DataSet установить ссылку на ADOQUERY1.

С помощью визуального конструктора отчетов подготовим отчет и создадим файл проекта отчета. Для этого:

с помощью команды Tools\Rave Designer меню Delphi запустим изуальный конструктор Rave Reports 5.0;

выбором команды File\New Data Object откроем диалоговое окно Data Connections и выберем в нем вариант Direct Data View;

в очередном диалоговом окне в списке Active Data Connections выберем вариант RVDataSetConnection1 и нажмем кнопку Finish;

в дереве проекта в правой части окна визуального конструктора отчетов раскроем узел Data View Dictionary и в нем раскроем вновь созданный узел DataView1;

выбрав команду меню Tools\Report Wizards\Simple Table, запустим мастер создания простых таблиц в отчете, в котором выберем вариант DataView1 и нажмем кнопку Next;

на последующих шагах работы с Мастером выберем поля таблицы для отображения в отчете, при необходимости изменим очередность следования полей, установим параметры полей страницы, текст заголовков и шрифтов, используемых в отчете;

на заключительном этапе работы с Мастером нажатием кнопки Generate запустим процесс генерации отчета;

для просмотра сгенерированного отчета выберем команду File\Execute Report, в открывшемся диалоговом окне Output Options в поле Report Destination выберем переключатель Preview и нажмем ОК. Вид полученного нами отчета при просмотре приведен на рис. 9.1;

при необходимости выполним настройку параметров (шрифта

- заменить на Courier, цвета и др.) отдельных составляющих созданного отчета, все подписи сделать на русском языке (рис. 9.2);

сохраним созданный нами проект отчета в файле с произвольным именем, например Reportl, и расширением .rav с помощью команды File\Save As;

свернем или закроем диалоговое окно работы с визуальным конструктором отчета и вернемся к работе с приложением в Delphi.

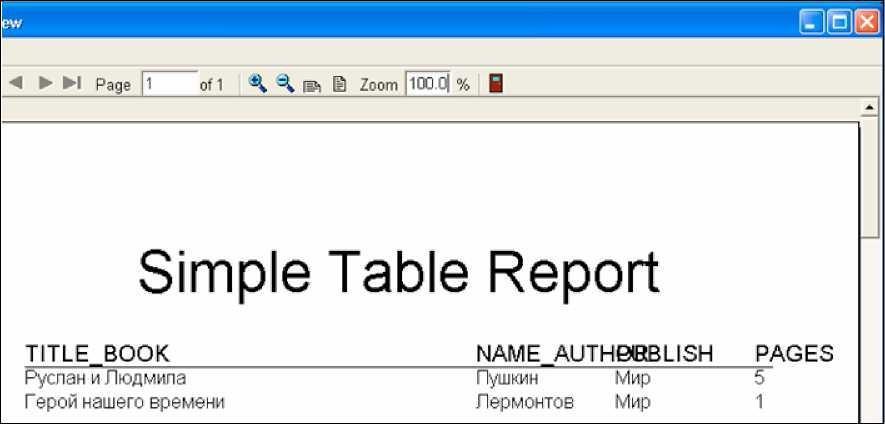


Рис. 9.1. Пример сгенерированного табличного отчета в режиме Preview

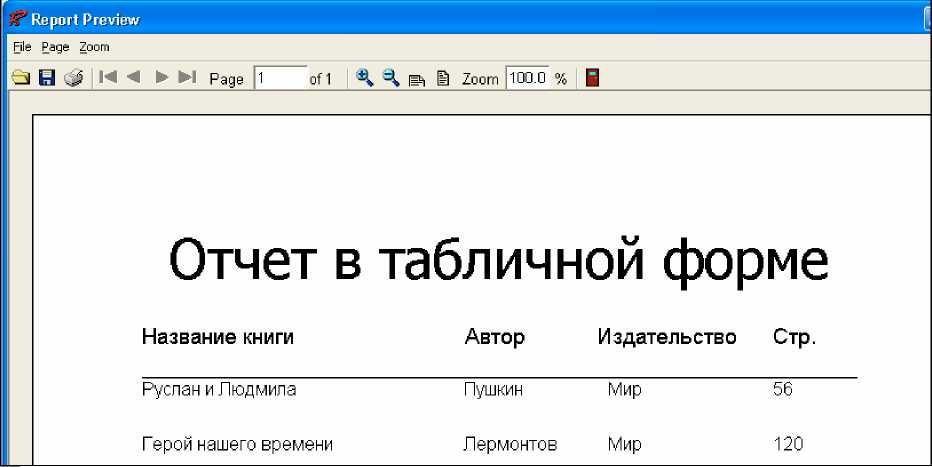


Рис. 9.2. Пример исправленного табличного отчета в режиме Preview

На главную форму MainForm клиентского приложения с вкладки Rave поместим компонент RvProject и установим значение Report1.RAV (спецификация созданного нами файла проекта) свойству ProjectFile этого компонента с помощью Инспектора объектов.

В главной форме приложения в подпункте Отчет в табличной форме в методе Click вызываем метод Execute, обеспечивающий выполнение отчета (в нашем примере Reportl) из состава проекта отчета (компонента RvProjectl):

dm.ADOQuery1.Open; RvProjectl.Open;

try

RvProjectl.Execut;

finally RvProjectl.Close; end;//try

Запустим приложение, проверим работу отчета

*Задание 2.* Создание отчета в свободной форме по запросу из первого задания. Создадим карточку книги для библиотечной картотеки.

Особенность отчета в свободной форме в том, что он создает шаблон на каждую отдельную запись таблицы, другими словами, он создается по документам, у которых нет шапки и примечаний. Примером таких документов могут служить приходный или расходный кассовые ордера, этикетка для товара или ценник в магазине, пригласительное письмо и т.д.

Продолжаем работу с визуальным конструктором отчетов Rave

Reports 5.0; Создаем новый отчет, для этого:

С помощью команды Tools\Rave Designer меню Delphi снова запустим визуальный конструктор Rave Reports 5.0.

Создаем новый отчет выбором команды File\New Report.

Устанавливаем компонент Region с вкладки Report, который является контейнером для компонентов Band (Полоса), у которого установить следующие свойства:

Column на 2;

ColumnSpacing на 0,1; Height на l0,6;

Width на 8,1.

Устанавливаем компонент DataBand component с вкладки Report в область Region, у которого установить следующие свойства:

DataView на DataView1; KeepRowTogether на True; Height на 3.

Размещаем компонент Bitmap с вкладки Standart на DataBand component и настраиваем следующие свойства:

Image на какой-либо рисунок; MatchSide на msBoth;

Height на 2,6;

Width на 3,8.

Размещаем три компонента DataText с вкладки Report на форму согласно рис. 9.3. Устанавливаем свойствам всех трех компонент следующие значения: DataView - DataView1; Font - Courier, 15, Bold; Width - 3,6; для свойства DataField - title\_book, name\_author, publish соответственно.



Рис. 9.3. Пример расположения компонентов с изображением на заднем фоне

Размещаем три компонента Text с вкладки Standart на форму согласно рис. 11.3 и устанавливаем свойству Text - “Название книги”, “Автор”, “Издательства” соответственно, свойству Font - Courier, 14.

Сохранить отчет с названием Report2.rav.

На главную форму MainForm клиентского приложения с вкладки Rave поместим компонент RvProject и установим значение Report2.RAV (спецификация созданного нами файла проекта) свойству ProjectFile этого компонента с помощью Инспектора объектов.

В главной форме приложения в подпункте Отчет в свободной форме в методе Click вызываем метод Execute, обеспечивающий выполнение отчета (в нашем примере Report2) из состава проекта отчета (компонента RvProject2):

dm.ADOQuery1 .Open; RvProject2.Open;

try

RvProject2.Execute; finally RvProject2.Close; end;//try

Запустите приложение, проверьте работу.

*Задание 3.* Создание отчета по двум таблицам. Создадим отчет с группировкой, в котором сначала будут выводиться данные автора книги из таблицы Authors, а затем список книг, которые написал этот автор.

В проекте на форму DM с вкладки Rave поместим 2 компонента RvDataSetConnection, заменить их имена на RV\_Authors и RV\_Books. У свойства DataSet установить ссылку по очереди на соответствующие компоненты ADOAuthors и ADOBooks.

Продолжаем работу с визуальным конструктором отчетов Rave Reports 5.0. Создаем новый отчет, для этого:

С помощью команды Tools\Rave Designer меню Delphi снова запустим визуальный конструктор Rave Reports 5.0.

Создаем новый отчет выбором команды File\New Report.

Выбором команды File\New Data Object откроем диалоговое окно Data Connections и выберем в нем вариант Direct Data View. В следующем окне выберем в качестве активных источников данных RV\_Authors и RV\_Books. В правом окне RaveProject будут добавлены разделы DataView2 и DataView3.

Устанавливаем компонент Region с вкладки Report, который является контейнером для компонентов Band (Полоса), у которого установить следующие свойства:

Height на 10,6;

Width на 8,1.

Устанавливаем 2 компонента Band с вкладки Report в область Region. Помещаем в каждую добавленную полосу по компоненту Text с вкладки Standart. В первой полосе у компонента Text меняем следующие свойства:

Font на Courier, 15, Bold;

Text на Отчет с группировкой.

Во второй полосе у компонента Text меняем следующие свойства:

Font на Courier, 15, Bold; Text на Ф.И.О. автора.

Устанавливаем 2 компонента DataBand с вкладки Report в область Region, у которых изменить свойство DataView на DataView2 и DataView3 соответственно. Помещаем в каждую добавленную полосу по компоненту DataText с вкладки Report. В первой полосе у компонента DataText меняем следующие свойства

DataView на DataView2; DataField на name\_author.

Ниже под компонентом DataText помещаем компонент Text, где меняем надпись на «Названия книг данного автора».

Во второй полосе у компонента DataText меняем следующие

свойства:

DataView на DataView3; DataField на Title\_book.

В отчете присутствуют два соответствующих представления данных, каждое из которых подключено к компоненту DataBand (оба компонента располагаются внутри региона).

Первый компонент DataBandl ассоциирован с основным набором данных и не обладает какими-либо специальными настройками.

Второй компонент DataBand2 определяет отношения типа

«основное/подробности», используя несколько свойств. Свойство MasterDataView ссылается на представление данных основного набора данных DataView2, а свойства MasterKey и DetailKey ссылаются на поля, ри помощи которых определяется объединение, в данном случае оба этих свойства ссылаются на поле Code\_author. Свойство ControllerBand ссылается на компонент DataBandl, который отображает данные из основного набора данных. Кроме того, для отображения данных из подчиненной таблицы, в данном случае Books, нужно настроить свойство компонента DataBand2 - BandStyle, нажав рядом со свойством кнопку

«...». В окне Band Style Editor отметить свойство Detail(D).

Если вы не хотите, чтобы отображаемые подробности переходили на следующую страницу (в этом случае основные данные и часть подробностей будут располагаться на разных страницах), присвойте свойству

KeepRowTogether значение True.

Сохранить отчет с названием Report3.rav.

На главную форму MainForm клиентского приложения с вкладки Rave поместим компонент RvProject и установим значение Report3.RAV (спецификация созданного нами файла проекта) свойству ProjectFile этого компонента с помощью Инспектора объектов.

В главной форме приложения в подпункте Отчет с группировкой в методе Click вызываем метод Execute, обеспечивающий выполнение отчета (в нашем примере Report3) из состава проекта отчета (компонента RvProject3):

dm.ADOAuthors.Open; dm.ADOBooks.Open; RvProject3.Open;

try

RvProject3.Execute; finally RvProject3.Close; end;//try

Запустите приложение, проверьте работу.

*Варианты заданий к лабораторной работе №12*

По индивидуальной базе данных, которая выдана по вариантам (из лабораторной работы №5), сделать в клиентском приложении четыре отчета, которые будут запускаться через меню главной формы:

отчет в табличной форме по одному из справочников, причем необходимо вывести итоговое количество записей в отчете;

отчет в свободной форме. Выберите одну из таблиц, по которой можно сделать или бейдж, или ценник, или пригласительный билет. При создании отчета используйте рисунок в качестве подложки;

отчет по запросу. Соедините данные всех основных таблиц, кодовые поля в запрос не помещайте. Создайте отчет в табличной форме с итоговым полем. Это может быть сумма или количество и т.п., в зависимости от содержания запросов. Каждая строка в отчете должна иметь номер по порядку. Например (см. фрагмент ниже):

Крупа 10 кг

Мука 20 кг

и т.д.;

отчет с группировкой по нескольким таблицам. Выберите одну пару связанных таблиц, определите главную и зависимую таблицы и сделайте отчет в табличной форме, в котором данные из главной таблицы расшифровываются (дополняются) данными из зависимой таблицы.

*Требования к содержанию отчета*: итоги лабораторной работы представить в виде пояснений и копий экранов выполнения основных заданий, описанных в инструкции по выполнению лабораторной работы.

Контрольные вопросы для самоподготовки

Для чего нужен RaveReports?

Какие существуют способы создания отчетов?

Какие типы соединений с источниками данных существуют в RaveReports?

1. Какой компонент необходим для настройки связи отчета с базой данных?
2. Как создать отчет с группировкой по двум таблицам?

**Инструкционно-технологическая карта**

**для выполнения лабораторная работа №10**

**Тема:** **Организация защиты данных в хранилищах.**

**Наименование работы: Реализация доступа пользователей к базе данных**

**Цель:** научиться создавать реляционную базу данных в системе управления базами данных MS SQL Express.

**Формируемые компоненты:** **ПК 11.1-11.6 ОК 1-11**

**Норма времени:**4 часа

**Оборудование и программное обеспечения:** ПК, MS SQL Server

**Основные правила ТБ при выполнении работы:**

-плавно нажимайте на клавиши, не допуская резких ударов;

-не пользуйтесь клавиатурой, если не подключено напряжение;

-работайте на клавиатуре чистыми руками;

-никогда не пытайтесь самостоятельно устранить неисправность в работе аппаратуры;

Строго запрещается:

-трогать разъемы соединительных кабелей;

-прикасаться к питающим проводам и устройствам заземления;

-прикасаться к экрану и к тыльной стороне монитора, клавиатуры;

-включать и отключать аппаратуру без указания преподавателя;

-Класть диск, книги, тетради на монитор и клавиатуру;

-Работать во влажной одежде и влажными руками **Литература:** Кумскова, И. А. Базы данных: учебник для СПО / И. А. Кумскова.- М.: КНОРУС, 2016.-488 с.

**Контрольные вопросы при допуске:**

1. Кто такой пользователь?
2. Что нужно для пользователя в БД?
3. Какой сушествует доступ для пользователя?

Практическая часть

Задание 1. Запустите утилиту SQL Server Management Studio (Пуск/Все программы/ MS SQL Server 2008R2). Настройте подключение к серверу:

компонентDatabaseEngine

локальный компьютер (localhost\SQLEXPRESS)

проверки подлинностиWindows

Перечислите имена сервера, которые предлагает. Какое имя использовали вы?

Задание 2. Перечислите команды Меню открывшегося окна SQL Server Management Studio

Задание 3. Вызовите команду Вид. Запишите назначение пиктограмм

, , , , .

Задание 4. Вызовите команду Вид/Панели инструментов. Перечислите предлагаемые панели инструментов.

Задание 5. Создайте базу данных ОборотКомпанииФамилия (вместо слова Фамилия добавьте вашу фамилию) с помощью графической утилиты SQL Server Management Studio. Для этого:

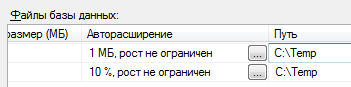
В окне «Обозреватель объектов» найдите и раскройте папку «Базы данных». Щелкните на ней правой кнопкой мыши и выберите команду «Создать базу данных…».

В открывшемся диалоговом окне «Создание базы данных» на странице «Общие» введите следующую информацию:

Имя базы данных: ОборотКомпанииФамилия

Владелец: sa

В таблице «Файлы базы данных» измените путь к файлам данных и журнала на каталог С:\Temp.



Для всех остальных параметров оставьте значения по умолчанию.

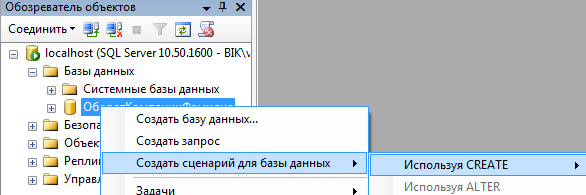
Для создания базы данных щелкните «OK». Вы должны увидеть свою новую базу данных в окне «Обозреватель объектов».

Зайдите в каталог С:\Temp, найдите два файла созданной вами базы данных:



Задание 6. Создайте базу данных ОборотКомпанииФамилия (вместо слова Фамилия добавьте вашу фамилию) с помощью SQL команды CREATE. Для этого:

Cгенерируйте сценарий базы данных на основе уже существующей базы данных. Для этого в SQL Server Management Studio в контекстном меню узла «ОборотКомпанииФамилия»



Выберите команду «Создать сценарий для базы данных – Используя CREATE– Буфер обмена». В результате в буфер обмена будет сохранен текст запроса на создание новой базы данных с параметрами, указанными при создании базы данных в Management Studio.

Для проверки работоспособности сгенерированного запроса на создание базы данных удалите созданную вами базу данных. В контекстном меню базы данных выберите команду «Удалить» и в появившемся диалоговом окне нажмите кнопку «OK». База данных со всеми файлами должна исчезнуть.

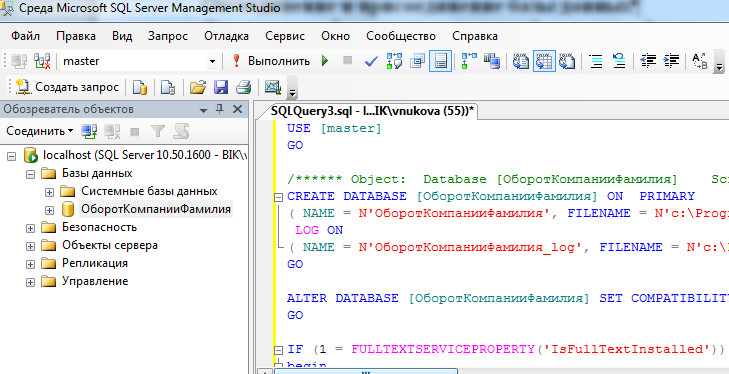
Чтобы воспользоваться сгенерированным заранее запросом на создание базы данных выполните следующие шаги:

На панели инструментов выберите команду «Создать запрос» .

В открывшемся окне редактора SQL вставьте из буфера обмена (Ctrl+V) сгенерированный запрос.

Для запуска запроса на выполнение щелкните кнопку на панели инструментов или нажмите клавишу F5.

Обновите содержимое дерева обозревателя объектов командой «Обновить» из контекстного меню узла «Базы данных». База данных ОборотКомпанииФамилия должна вновь появиться в списке доступных.

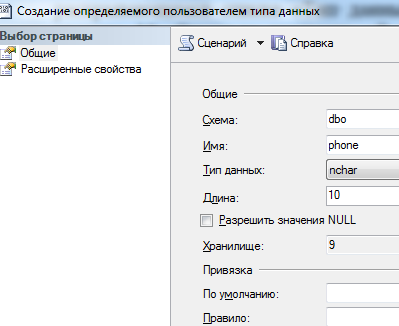


Задание 7. Cоздайте пользовательский тип данных phone, который будет использоваться в таблице Customer для хранения телефонного номера клиента. Для его создания воспользуйтесь утилитой Management Studio.

В дереве обозревателя объектов раскройте папки «Базы данных – ОборотКомпанииФамилия – Программирование - Типы». В контекстном меню узла «Определяемые пользователем типы данных» выберите команду «Создать определяемый пользователем тип данных».

В появившемся окне в текстовом поле «Имя» введите phone. В раскрывающемся списке «Тип данных» выберите nchar. В качестве длины введите 10. Отметьте параметр «Разрешить значения null», чтобы иметь возможность не указывать телефонный номер при добавлении нового клиента.

В секции «Привязки» оставьте пустые значения и щелкните на кнопке Ok. Созданный пользовательский тип данных должен появиться в дереве обозревателя объектов.



Задание 8. Создайте в базе данных пять таблицы. Первая таблица, Customer, будет хранить информацию о клиентах, вторая таблица City– справочник городов, третья, Product,   информацию о товарах, четвертая, Order, будет содержать подробную информацию о заказах и пятая, OrdItem, - о составе заказа (перечне товаров входящих в заказ). Ниже представлены все поля этих таблиц и их основные свойства.

Имя столбца

Тип данных

Разрешить null

Описание

Customer

IdCust

int, identity

нет

Уникальный идентификационный номер клиента, на который можно ссылаться в других таблицах

FName

nvarchar(20)

нет

Имя клиента

LName

nvarchar(20)

нет

Фамилия клиента

IdCity

int

нет

Ссылка на номер города

Address

nvarchar(50)

нет

Адрес клиента

Zip

nchar(5)

нет

Почтовый индекс клиента

Phone

phone

да

Телефонный номер клиента

City

IdCity

int, identity

нет

Уникальный идентификационный номер города

CityName

nvarchar(20)

нет

Название города

Product

IdProd

int, identity

нет

Уникальный идентификационный номер для каждого товара

Description

nvarchar(100)

нет

Короткое текстовое описание товара

InStock

int

нет

Количество единиц продукта на складе

Order

IdOrd

int, identity

нет

Уникальный идентификационный номер заказа

IdCust

int

нет

Ссылка на номер клиента

OrdDate

smalldatetime

нет

Дата и время размещения заказа

OrdItem

IdOrd

int

нет

Ссылка на номер заказа

IdProd

int

нет

Ссылка на номер товара

Qty

int

нет

Количество единиц товара в заказе

Price

money

нет

Цена товара

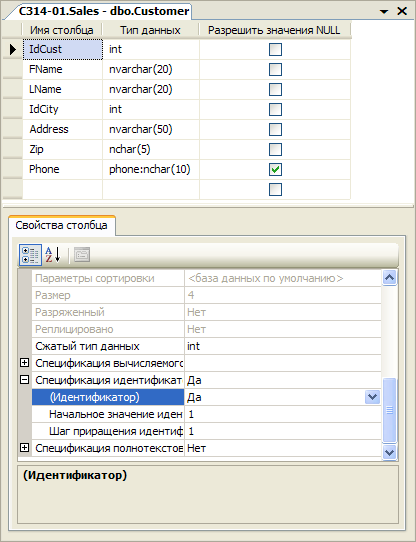
Таблицы можно создавать как в графическом интерфейсе (в утилите Management Studio), так и с помощью кодаT-SQL. Воспользуемся самым простым, графическим способом. Сначала создадим таблицу Customer:

В дереве обозревателя объектов в базе данных в контекстном меню узла «Таблицы» выберите команду «Создать таблицу…». В рабочей области должна появиться вкладка с конструктором таблиц.

В первую строку в столбце «Имя столбца» введите IdCust, в столбце «Тип данных» выберите int. Убедитесь что параметр «Разрешить значения null» отключен.

В нижней половине экрана в разделе «Свойства столбцов» введите описание поля и измените значение параметра «Спецификация идентификатора / (Идентификатор)» на «Да» для того чтобы значения номера клиента формировались автоматически. Свойство «Идентифицирующий столбец» (Identity), обычно используемое совместно с типом данных int, предназначено для автоматического приращения значения на единицу при добавлении каждой новой записи. К примеру, клиент, добавленный в таблицу первым, будет иметь значение идентификатора 1, вторым – 2, третьим – 3, и т.д.

Аналогичным образом введите описания всех остальных полей и закройте окно конструктора таблиц. Введите в качестве имени таблицы Customer. Вновь созданная таблица должна появиться в дереве обозревателя объектов в папке «Таблицы».



В соответствие с вышеприведенным описанием создайте оставшиеся четыре таблицы: City, Product, Order и OrdItem.

Задание 9. В Обозревателе объектов найдите созданную вами на прошлой лабораторной работе базу данных ОборотКомпанииФамилия.mdf, которая должна состоять из пяти таблиц: Customer, City, Product, Order, OrdItem.

Создайте сценарий для первой таблицы, используя команду Create, в буфер обмена и в новое окно редактора запросов. Вставьте созданный сценарий в отчет по лабораторной работе.

Задание 10. Создайте ограничения для поля Zip (почтовый индекс) таблицы Customer, запрещающее вводить в поле буквы.

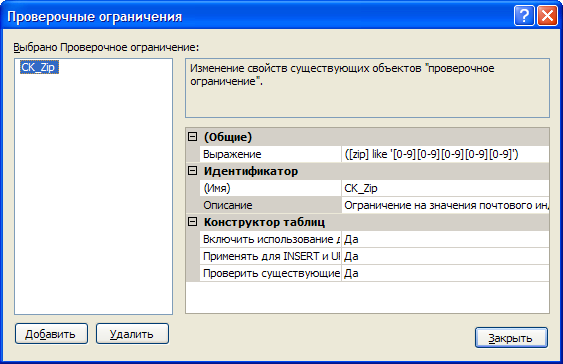
В контекстном меню папки «Ограничения» таблицы Customer выберите команду «Создать ограничение».

В открывшемся окне «Проверочные ограничения» заполните следующие поля:

Имя: CK\_Zip

Выражение: ([zip] like '[0-9][0-9][0-9][0-9][0-9]'). Данное выражение описывает ограничение, принимающее пять символов, которыми могут быть только цифры от 0 до 9.

Описание: Ограничение на значения почтового индекса



Задание 11. Создайте ограничения для полей InStock таблицы Product, Price таблицы OrdItem, запрещающие ввод в них отрицательных значений. В данном случае выражение проверки будет иметь вид (Имя поля > 0) для поля Price и (Instock>=0) для столбца InStock.

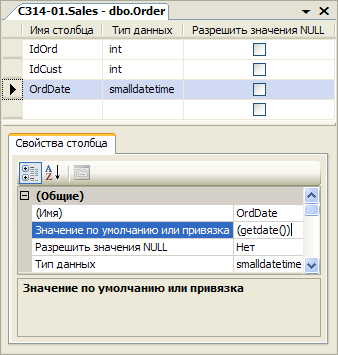
Задание 12. Создайте ограничения Использование значений по умолчанию.

Установка для полей значений по умолчанию это отличный способ избавить пользователя от излишней работы, если значения этих полей во всех записях, как правило, принимают одни и те же значения. Так в таблице заказов Order вполне логично определить по умолчанию значение поля OrdDate (дата заказа) в виде текущей даты. В этом случае при добавлении записи о новом заказе в случае пропуска этого поля оно будет автоматически заполняться значением системной даты. Для создания такого свойства выполните следующие шаги:

Раскройте папку «Столбцы» таблицы Order и в контекстном меню поля «OrdDate» выберите команду «Изменить».

В свойстве столбца «Значение или привязка по умолчанию» введите getdate(). Эта функция T-SQL возвращает текущую системную дату.

Щелкните на кнопке Сохранить и выйдите из конструктора таблиц.



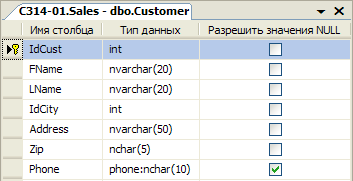
Задание 13. Установите для поля InStock (количество единиц продукта на складе) таблицы Product в качестве значения по умолчанию ноль.

Задание 14. Создайте первичный ключ для таблицы Customer   столбец IdCust. Для создания первичного ключа в таблице Customer выполните следующие шаги:

В контекстном меню таблицы Customer выберите команду «Проект».

В окне конструктора таблиц щелкните правой кнопкой мыши на поле IdCust и выберите команду «Задать первичный ключ» или нажмите кнопкуна панели инструментов. Обратите внимание на то, что слева от поля IdCustтеперь отображается значок ключа, указывающий, что поле является первичным ключом.

Закройте конструктор таблиц с сохранением изменений



Задание 15. Аналогичным образом создайте первичные ключи для остальных таблиц в соответствие с ниже приведенной таблицей.

Таблица

Первичный ключ

City

IdCity

Product

IdProd

Order

IdOrd

OrdItem

IdOrd,IdProd (для выбора нескольких столбцов при установке составного ключа воспользуйтесь клавишами Shift или Ctrl)

Задание 16. Создайте ограничения на уникальность в таблице Customer по полю Phone, которое повторяться у разных клиентов не должно.

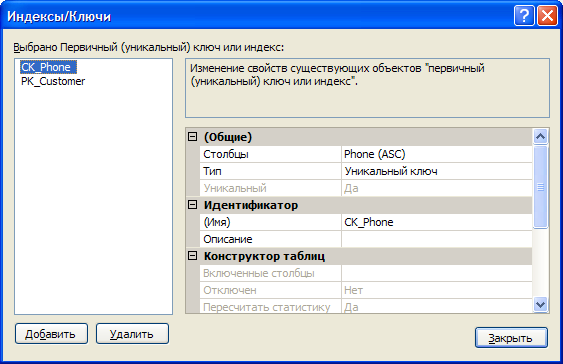
Для открытия конструктора таблиц в контекстном меню таблицы Customer выберите команду «Проект». На панели инструментов нажмите на кнопку «Управление индексами и ключами».

В открывшемся окне «Индексы и ключи» щелкните кнопку «Добавить» и введите следующие параметры для нового уникального ключа:

Столбцы: Phone

Тип: Уникальный ключ

(Имя): CK\_Phone



Закройте конструктор таблиц с сохранением изменений.

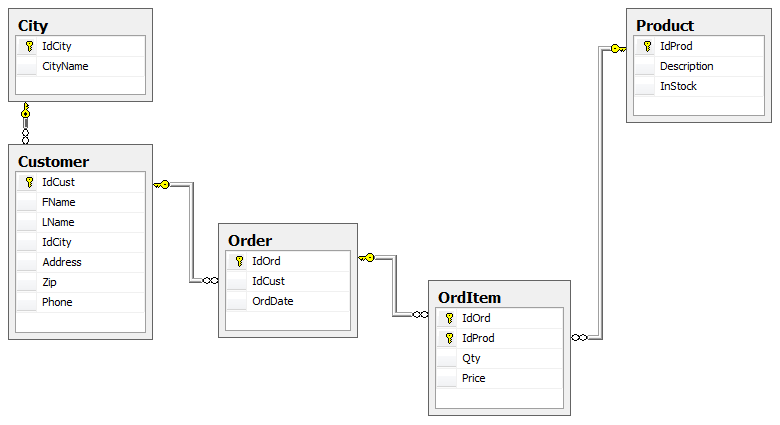
Задание 17. Аналогичным образом создайте ограничение уникальности по полю CityName таблицы City, чтобы обеспечить отсутствие в справочнике городов с одинаковыми названиями, а также по полю Description таблицы Product, чтобы иметь возможность отличить один товар от другого.

Задание 18. Создайте диаграмму базы данных:

В контекстном меню папки «Диаграммы базы данных» выберите команду «Создать диаграмму базы данных».

В диалоговом окне «Добавление таблиц» выберите все таблицы и нажмите на кнопку «Добавить».

Добавив таблицы, щелкните на кнопке «Закрыть» и вы увидите созданную диаграмму базы данных (на рисунке представлен окончательный вид диаграммы: некоторые связи у вас могут отсутствовать).



Используя диаграмму базы данных ограничения внешнего ключа можно создавать значительно быстрее: лишь перетаскивая поля из одной таблицы в другую. В качестве примера создадим внешний ключ в таблице Customer по полю IdCity для связи с таблицей City:

Выделите в таблице City поле IdCity и, не отпуская кнопку мыши, перетащите его на поле IdCity таблицы Customer.

В диалоговых окнах «Таблицы и столбцы» и «Связь по внешнему ключу» примите настройки по умолчанию.

Сохраните диаграмму базы данных под именем ILM.

Расположите таблицы в канонической форме (главные таблицы выше подчиненных) в соответствии с вышеприведенным рисунком.

Задание 19. Аналогичным образом создайте связь между таблицами Product и OrdItem по полю IdProduct. Окончательный список связей между таблицами со всеми их характеристиками представлен в следующей таблице:

Главная таблица

Подчиненная таблица

Поле связи (внешний ключ)

Правила каскадирования

City

Customer

IdCity

Без действия

Customer

Order

IdCust

Без действия

Order

OrdItem

IdOrd

Каскадное удаление

Product

OrdItem

IdProd

Без действия

Задание 20. После настройки всех ограничений можно наполнить таблицы данными. Для этого в контекстном меню таблицы выберите команду «Изменить первые 200 строк» и появившейся в рабочей области вкладке введите новые записи, заполняя все необходимые столбцы. В процессе внесения данных проверьте работоспособность всех созданных ранее ограничений:

Ограничений проверки: попробуйте ввести в поле Zip (почтовый индекс) таблицы Customer нечисловые значения, а в поля InStock таблицыProduct,Price таблицы OrdItem - отрицательные.

Значений по умолчанию: убедитесь, что при пропуске полей OrdDate и InStock таблиц Order и Product для них устанавливаются значения по умолчанию в виде текущей системной даты и нуля соответственно.

Ограничений первичного и уникального ключа: попробуйте ввести в таблицы записи с дублирующими значениями первичного или уникального ключа.

Ограничений внешнего ключа: попробуйте ввести несогласованные данные в связанные таблицы, например, заказ для несуществующего клиента или удалить запись из любой главной таблицы при наличии связанных записей в подчиненной при отсутствии правил каскадирования.

Правил каскадирования: убедитесь, что при удалении записи из таблицы Order все связанные записи из таблицы OrdItem удаляются автоматически.

**Вопросы**

1. Для чего предназначена утилита SQLServer Management Studio?

2. Перечислите предлагаемые программой SQL Server способы аутентификации.

3. Что отображается на панели Обозреватель серверов после подключения к серверу с помощью утилиты утилита SQLServer Management Studio?

4. Опишите структуру окна утилиты Management Studio?

**Инструкционно-технологическая карта**

**для выполнения лабораторная работа №11**

**Тема:** **Организация защиты данных в хранилищах.**

**Наименование работы: Мониторинг безопасности работы с базами данных**

**Цель:** научиться проводить мониторинг баз данных в системеХостТрекер

**Формируемые компоненты:** **ПК 11.1-11.6 ОК 1-11**

**Норма времени:**4 часа

**Оборудование и программное обеспечения:** ПК, MS SQL Server

**Основные правила ТБ при выполнении работы:**

-плавно нажимайте на клавиши, не допуская резких ударов;

-не пользуйтесь клавиатурой, если не подключено напряжение;

-работайте на клавиатуре чистыми руками;

-никогда не пытайтесь самостоятельно устранить неисправность в работе аппаратуры;

Строго запрещается:

-трогать разъемы соединительных кабелей;

-прикасаться к питающим проводам и устройствам заземления;

-прикасаться к экрану и к тыльной стороне монитора, клавиатуры;

-включать и отключать аппаратуру без указания преподавателя;

-Класть диск, книги, тетради на монитор и клавиатуру;

-Работать во влажной одежде и влажными руками **Литература:** Кумскова, И. А. Базы данных: учебник для СПО / И. А. Кумскова.- М.: КНОРУС, 2016.-488 с.

**Контрольные вопросы при допуске:**

1. Что такое БД?
2. Что такое мониторинг?
3. Для чего нужна мониторинг безопасности БД?

**Задание 1.**

Настройка мониторинга состояния базы данных

[Блог компании ХостТрекер](https://habr.com/ru/company/host-tracker/),

[Хостинг](https://habr.com/ru/hub/hosting/),

[Разработка веб-сайтов](https://habr.com/ru/hub/webdev/),

[Тестирование IT-систем](https://habr.com/ru/hub/it_testing/),

[Администрирование баз данных](https://habr.com/ru/hub/db_admins/)

[Сервис мониторинга ХостТрекер](http://www.host-tracker.com/) предлагает возможность автоматически следить за ключевыми показателями жизнедеятельности базы без необходимости каждый день лезть на сервера вручную. Ниже мы расскажем, как ловить популярные ошибки, защититься от внезапного переполнения базы.

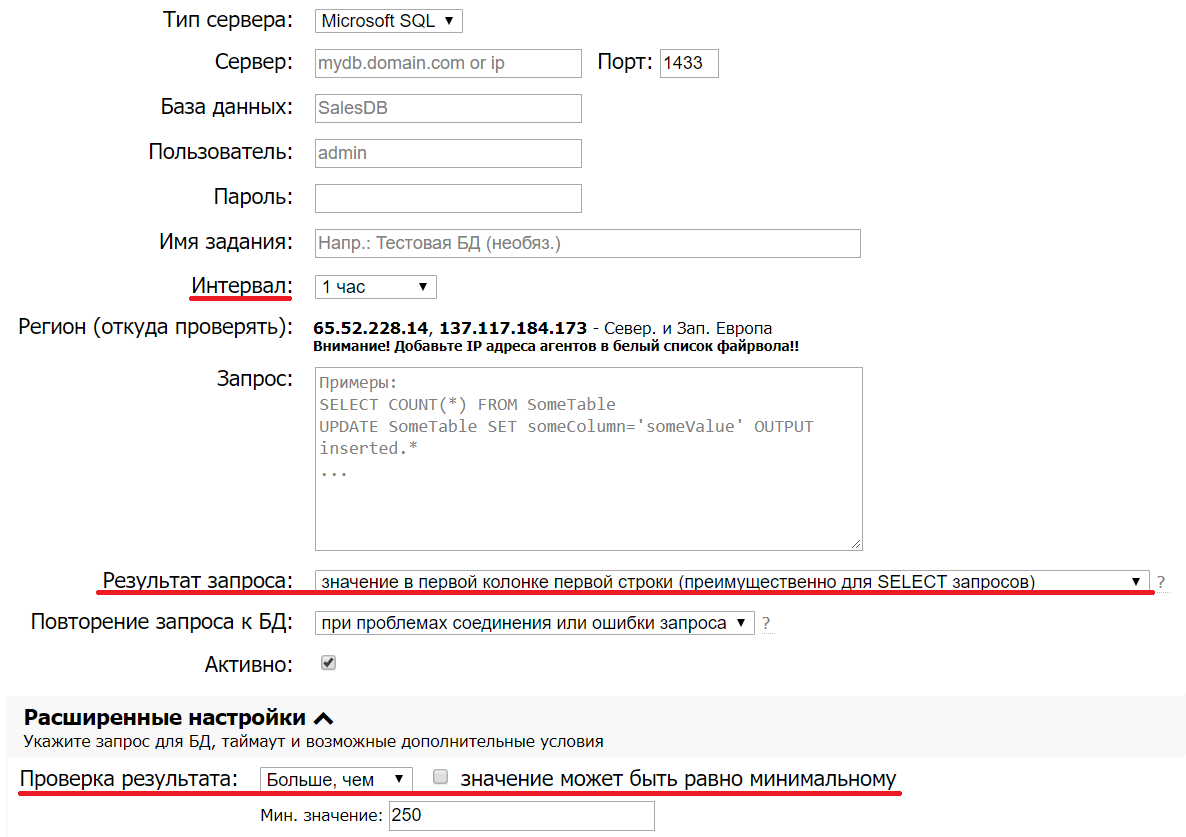
[В силу специфики сервиса](https://www.host-tracker.com/blog/about_ht), в первую очередь решаются задачи по мониторингу баз сайтов, что вместе с другими способами проверок может дать подробную диагностику проблем сайтов. Но вообще можно использовать этот инструмент для любых баз, к которым есть доступ из сети.

1. Для начала, нужно подключиться к базе. Настоятельно рекомендуем для безопасности создать отдельную учетную запись и дать ей права только на чтение и только на те таблицы, которые планируете проверять.

2.Далее, выбираете интервал проверки — от 10 минут до раза в сутки. И вводите запрос, который планируете регулярно исполнять, составив его таким образом, чтобы результат, который Вам необходимо проверять, представлял из себя одно из двух:

Первый столбец первой строки ответа сервера. Это может быть любой числовой формат, строка, массив байтов (изображение или аудио, что хранится в БД), GUID. Для чего может быть использовано — см. ниже.

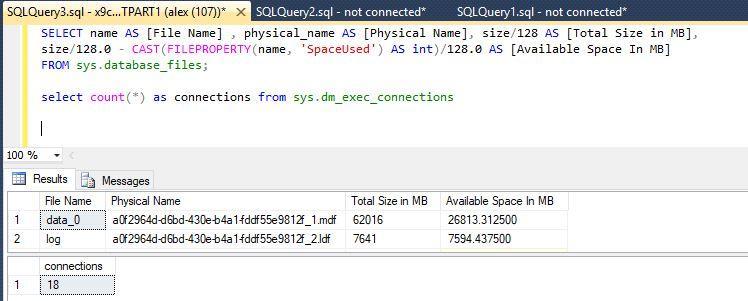
Количество затронутых запросом строк.



3. Можно настроить проверку результатов. Например, смотреть, не изменилось ли определенное значение с помощью условий «равно — не равно» (здесь как раз можно и GUID проверять, и строки). Для числовых значений также дополнительно есть условия больше/меньше или попадает/не попадает ли значение в определенный интервал.  
По любому запросу можно настроить оповещения, которые будут приходить по СМС или [другим способом](https://habr.com/company/host-tracker/blog/353316/), если искомое значение превышено.   
При необходимости, добавляем указанные адреса в файрвол.

Популярные применения

Все это настраивается один раз, возвращаться к вопросу приходится лишь в случае возникновения проблем. Самим нужно лишь написать корректный запрос. Впрочем, мы и это попытаемся исправить: ниже можно найти список запросов, которые могут быть особенно полезными при подобном мониторинге. Все — для MS SQL, но большинство запросов несложно конвертируется и для других баз.

Чтобы повысить безопасность, можно создать процедуру, в которую включить эти запросы и дать права на ее исполнение указанному логину. Либо же создать промежуточную таблицу, в которую записывать результаты этих запросов, и уже для нее предоставить права на чтение для Host-tracker.  
  
  
  
Информация о последнем успешном бэкапе:  
SELECT TOP 1 database\_name, backup\_size, backup\_start\_date, backup\_finish\_date  
FROM msdb..backupset bs  
ORDER BY backup\_set\_id DESC  
Количество подключений к базе в этот момент:  
SELECT COUNT(\*) AS connections FROM sys.dm\_exec\_connections  
Время последнего запуска сервера БД, количество ЦПУ и оперативной памяти  
SELECT sqlserver\_start\_time, cpu\_count, physical\_memory\_kb  
FROM sys.dm\_os\_sys\_info  
Размер файлов и доступное место для текущей базы данных:  
SELECT name AS [File Name] , physical\_name AS [Physical Name], size/128 AS [Total Size in MB],  
size/128.0 - CAST(FILEPROPERTY(name, 'SpaceUsed') AS int)/128.0 AS [Available Space In MB]  
FROM sys.database\_files;  
Сколько места доступно для текущей базы данных:  
SELECT (SUM(unallocated\_extent\_page\_count)\*1.0/128) AS [free space in MB]  
FROM sys.dm\_db\_file\_space\_usage  
Сколько места доступно для базы TEMPDB:  
SELECT (SUM(unallocated\_extent\_page\_count)\*1.0/128) AS [free space in MB]  
FROM tempdb.sys.dm\_db\_file\_space\_usage

Задания 2. Провести мониторинг созданной ранее базы данных.

**Вопросы**

1. Для чего предназначена утилита ХостТрекер?

2. Перечислите предлагаемые программой ХостТрекер способы мониторинга.

3. Что отображается на панели после подключения к серверу с помощью утилиты утилита ХостТрекер?

4. Опишите структуру окна утилиты ХостТрекер?

**Инструкционно-технологическая карта**

**для выполнения лабораторная работа №12,13**

**Тема:** **Организация защиты данных в хранилищах.**

**Наименование работы: Установка приоритетов и развертывание контроллеров домена**

**Цель:** научиться устанавливать приоритет в баз данных с помощью развертывания контроллеров домена.

**Формируемые компоненты:** **ПК 11.1-11.6 ОК 1-11**

**Норма времени:**4 часа

**Оборудование и программное обеспечения:** ПК, MS SQL Server

**Основные правила ТБ при выполнении работы:**

-плавно нажимайте на клавиши, не допуская резких ударов;

-не пользуйтесь клавиатурой, если не подключено напряжение;

-работайте на клавиатуре чистыми руками;

-никогда не пытайтесь самостоятельно устранить неисправность в работе аппаратуры;

Строго запрещается:

-трогать разъемы соединительных кабелей;

-прикасаться к питающим проводам и устройствам заземления;

-прикасаться к экрану и к тыльной стороне монитора, клавиатуры;

-включать и отключать аппаратуру без указания преподавателя;

-Класть диск, книги, тетради на монитор и клавиатуру;

-Работать во влажной одежде и влажными руками

**Литература:** Кумскова, И. А. Базы данных: учебник для СПО / И. А. Кумскова.- М.: КНОРУС, 2016.-488 с.

**Контрольные вопросы при допуске:**

1. Что такое БД?
2. Что такое домен?
3. Что такое приоритет?

**Задание 1.**

**Создание нового домена**

Первый домен. Домен - это набор объектов (компьютеров, пользователей и групп), определенных администратором. Для всех этих объектов используется общая база данных службы каталога (Active Directory) и общие политики безопасности. Кроме того, домен может иметь отношения безопасности с другими доменами.

Первый лес. Лес - это один или несколько доменов, совместно использующих общую схему (определения классов и атрибутов), информацию о сайтах и репликации и доступные для поиска компоненты (глобальный каталог).Домены одного леса могут быть связаны доверительными отношениями.

Первый сайт. Сайт - это фактически подсеть TCP/IP. Вы можете задавать и другие сайты, доступ к которым может происходить локально или дистанционным образом (например, через глобальную сеть [WAN]).

В ["Описание Active Directory"](https://intuit.ru/studies/professional_retraining/962/courses/274/lecture/6937)даются подробные пояснения и инструкции по работе с этими компонентами в иерархии вашей базы данных.

**Планирование развертывания контроллеров домена (DC)**

Количество контроллеров домена, которые вы создаете, зависит от размера вашей б.д. Конечно, в случае одной локальной сети (LAN) с одной подсетью ваша компания может работать с одним DC. Но это рискованно. Установка второго DC означает, что в случае отказа одного из DC аутентификация, доступ к сетевым ресурсам и другие услуги для пользователей не будут прерваны. Кроме того, если у вас очень много пользователей, то использование нескольких DC снижает нагрузку аутентификации, приходящуюся на один DC, особенно в ситуации, когда все пользователи завершают работу своих компьютеров вечером и затем загружают их и выполняют вход в сеть в 9 часов утра.

Если ваше БД охватывает несколько сайтов, то вам следует создать хотя бы один DC в каждом сайте (и еще лучше - второй DC на случай отказа). Для выполнения входа компьютерам и пользователям требуется какой-либо DC, и если этот DC находится в другом сайте, процесс входа может потребовать слишком много времени.

Поддерживайте физическую защиту своих контроллеров домена, по возможности держите их в запертых помещениях. Это не только защита от злоумышленников; это также мера защиты вашего предприятия от случайного повреждения вашими собственными пользователями.

**Установка Active Directory**

Контроллер домена отличается от других системных серверов тем, что он содержит Active Directory. Чтобы преобразовать любой компьютер Windows 2000 в контроллер домена, установите Active Directory. (На этом сервере должна действовать файловая система NTFS; если это не так, то прежде чем установить Active Directory, вы должны запустить convert.exe, чтобы заменить файловую систему FAT на NTFS.)

Все это выполняется с помощью мастеров, но подсказки и действия отличаются в зависимости от конкретной ситуации: это первый DC или один из добавляемых DC для данного домена и доступны ли службы DNS на момент установки этого DC. В следующих разделах дается описание типичных комбинаций.

**Active Directory и DNS**

Устанавливая первый DC в новом домене, вы сталкиваетесь с вечным вопросом: "Что первично - курица или яйцо?". Вы не можете установить Active Directory, если

у вас не запущена служба DNS, и вы не можете установить DNS, если у вас нет домена. Забавно, не правда ли?

Мастер установки Active Directory Installation Wizard автоматически пытается найти в своем списке сконфигурированных серверов DNS полномочный сервер DNS, который допускает обновление служебной (SRV) ресурсной записи. Если такой сервер найден, записи для этого DC будут автоматически регистрироваться этим сервером DNS после перезагрузки DC.

Но для первого сервера DC в новом домене, возможно, не найдется подходящего сервера DNS. В результате мастер автоматически добавит DNS в список ролей, устанавливаемых на этом компьютере. Предпочтительной настройкой DNS для этого сервера является конфигурирование самого себя в этой роли. Это вполне подходящее решение, так как после установки других серверов DNS вы можете удалить роль DNS с этого DC.

В качестве альтернативного решения вы можете создать сначала сервер DNS, но поскольку он не будет интегрирован с Active Directory (так как вы еще не установили Active Directory), данные для зоны DNS будут просто неструктурированным (плоским) файлом. Это не означает, что новый DC не сможет определить тот факт, что служба DNS установлена, но это повлияет на безопасность DNS. Зона DNS на основе "плоского" файла не задает никаких вопросов и не налагает никаких ограничений. Любому устройству, которое пытается зарегистрировать имя, разрешается сделать это, даже если оно заменяет такое же имя, зарегистрированное раньше. Никакие полномочия не проверяются и не налагаются на базу данных DNS, поэтому любой человек может не обращать на это никакого внимания. Но после того как вы создали свой домен и контроллеры домена, вы можете преобразовать зону на основе плоского файла в зону, интегрированную с Active Directory (информацию по установке и конфигурированию DNS см. в ["Ознакомление с DNS"](https://intuit.ru/studies/professional_retraining/962/courses/274/lecture/6923)).

**Установка первого контроллера домена (DC) в новом домене**

Чтобы установить Active Directory (AD), откройте Manage Your Server (Управление вашим сервером) из меню Start (Пуск) и щелкните на Add or Remove a Role (Добавление или удаление роли), чтобы запустить мастер Configure Your Server Wizard (Конфигурирование вашего сервера). В окне Server Role (Роль сервера) выберите Domain Controller (Active Directory), чтобы запустить мастер Active Directory Installation Wizard. Щелкните на кнопке Next, чтобы продолжить работу с мастером, используя следующие инструкции для установки вашего первого DC.

В окне Domain Controller Type (Тип контроллера домена) выберите Domain Controller For A New Domain (Контроллер домена для нового домена).

В окне Create New Domain (Создание нового домена) щелкните на Domain in a new *forest* (Домен в новом лесу).

На странице New Domain Name (Имя нового домена) введите полностью уточненное доменное имя (*FQDN*) для этого нового домена. (То есть введите companyname.com, но не *companyname*.)

В окне NetBIOS Domain Name (NetBIOS-имя домена) проверьте это NetBIOS-имя (но не *FQDN*).

В окне Database and Log Folders (Папки для базы данных и журнала) согласитесь с местоположением для папок базы данных и журнала или щелкните на кнопке Browse (Обзор), чтобы выбрать другое место, если у вас есть причина для использования другой папки.

В окне Shared System Volume (Разделяемый системный том) согласитесь с местоположением папки Sysvol или щелкните на кнопке Browse, чтобы выбрать другое место.

В окне DNS Registration Diagnostics (Диагностика регистрации DNS), проверьте, будет ли существующий сервер DNS подходящим для этого леса, или, если нет никакого сервера DNS, выберите опцию установки и конфигурирования DNS на этом сервере.

В окне Permissions (Полномочия) выберите один из следующих вариантов полномочий (в зависимости от версий Windows на клиентских компьютерах, которые будут выполнять доступ к этому DC).

Полномочия, совместимые с операционными системами до Windows 2000.

Полномочия, совместимые только с операционными системами Windows 2000 или Windows Server 2003.

Просмотрите информацию окна Summary (Сводка), и если вам нужно что-то изменить, щелкните на кнопке Back (Назад), чтобы изменить свои опции. Если все в порядке, щелкните на кнопке Next, чтобы начать установку.

После копирования всех файлов на ваш жесткий диск перезагрузите компьютер.

**Установка других контроллеров домена (DC) в новом домене**

Вы можете добавить любое число других DC в своем домене. В случае преобразования существующего рядового сервера в DC помните, что многие опции конфигурации исчезнут. Например, в этом случае будут удалены локальные пользовательские учетные записи и криптографические ключи. Если на этом рядовом сервере хранение файлов происходило с помощью EFS, то вы должны отменить шифрование. На самом деле после отмены шифрования вам следует переместить эти файлы на другой компьютер.

Чтобы создать и сконфигурировать другие DC в вашем новом домене, запустите мастер Active Directory Installation Wizard, как это описано в предыдущем разделе. Затем выполните шаги этого мастера, используя следующие указания.

В окне Domain Controller Type выберите Additional Domain Controller For An Existing Domain (Дополнительный контроллер домена для существующего домена).

В окне Network Credentials (Сетевые опознавательные данные) введите пользовательское имя, пароль и домен, указывающие пользовательскую учетную запись, которую вы хотите использовать для этой работы.

В окне Additional Domain Controller введите полное DNS-имя существующего домена, где данный сервер станет контроллером домена.

В окне Database and Log Folders согласитесь с местоположением для папок базы данных и журнала или щелкните на кнопке Browse (Обзор), чтобы выбрать место, отличное от установки по умолчанию.

В окне Shared System Volume согласитесь с местоположением папки Sysvol или щелкните на кнопке Browse, чтобы выбрать другое место.

В окне Directory Services Restore Mode Administrator Password (Пароль администратора для режима восстановления служб каталога) введите и подтвердите пароль, который вы хотите назначить учетной записи администратора этого сервера. (Эта учетная запись используется в случае загрузки компьютера в режиме Directory Services Restore Mode.)

Просмотрите информацию окна Summary, и если все в порядке, щелкните на кнопке Next, чтобы начать установку. Щелкните на кнопке Back, если вы хотите изменить какие-либо настройки.

Конфигурация системы адаптируется к требованиям для контроллера домена, после чего инициируется первая репликация. После копирования данных (это может занять определенное время, и если ваш компьютер находится в защищенном месте, то вы можете сделать для себя перерыв) щелкните на кнопке Finish в последнем окне мастера и перезагрузите компьютер. После перезагрузки появится окно мастера Configure Your Server Wizard, где объявляется, что теперь ваш компьютер стал контроллером домена. Щелкните на кнопке Finish, чтобы завершить работу мастера, или щелкните на одной из ссылок в этом окне, чтобы получить дополнительные сведения о поддержке контроллеров домена.

**Создание дополнительных контроллеров домена (DC) путем восстановления из резервной копии**

Вы можете быстро создавать дополнительные DC Windows Server 2003 в том же домене, что и какой-либо существующий DC, путем восстановления из резервной копии работающего DC Windows Server 2003. Для этого требуются только три шага (которые подробно описываются в следующих разделах).

Резервное копирование состояния системы существующего DC Windows Server 2003 (назовем его ServerOne) в том же домене.

Восстановление состояния системы в другом месте, т.е. на компьютере Windows Server 2003, который вы хотите сделать контроллером домена (назовем его ServerTwo).

Повышение статуса целевого сервера (в данном случае ServerTwo) до уровня DC с помощью команды DCPROMO /adv, введенной из командной строки.

Эту последовательность можно применять во всех сценариях: установка нового домена Windows Server 2003, модернизация домена Windows 2000 и модернизация домена Windows NT. После установки Windows Server 2003 на каком-либо компьютере, вы можете сделать этот компьютер контроллером домена (DC) с помощью данного способа.

Это особенно полезно, если ваш домен содержит несколько сайтов и ваши DC реплицируются через глобальную сеть (WAN), что намного медленнее, чем передача данных через кабель Ethernet. Когда происходит установка нового DC в удаленном сайте, первая репликация тянется очень долго. При данном способе отпадает необходимость в этой первой репликации, а в последующих репликациях копируются только изменения (что занимает намного меньше времени).

В следующих разделах приводятся инструкции по созданию DC этим способом.

Запустите Ntbackup.exe (из меню Administrative Tools или из диалогового окна Run) и выберите следующие опции в окнах мастера.

Выберите Back Up Files and Settings (Резервное копирование файлов и настроек).

Выберите Let Me Choose What To Back Up (Я выберу сам то, что нужно копировать).

Установите флажок System State (Состояние системы).

Выберите местоположение и имя для файла резервной копии. Я использовал разделяемую точку в сети и назвал файл DCmodel.bkf (файлы резервных копий имеют расширение имени .bkf ).

Щелкните на кнопке Finish.

Ntbackup выполнит резервное копирование состояния системы в указанном вами месте. Вам потребуется доступ к этой резервной копии с целевых компьютеров, поэтому проще всего использовать разделяемый ресурс сети или записать файл резервной копии на диск CD-R.

**Восстановление состояния системы на целевом компьютере**

Чтобы восстановить состояние системы на компьютере Windows Server 20010, который вы хотите сделать контроллером домена, перейдите на этот компьютер (он должен иметь доступ к файлу резервной копии, который вы создали на исходном компьютере). Запустите на этом компьютере Ntbackup.exe и выберите следующие опции в окнах мастера.

Выберите Restore Files and Settings (Восстановление файлов и настроек).

Укажите местоположение файла резервной копии.

Установите флажок System State.

Щелкните на кнопке Advanced (Дополнительно).

Выберите в раскрывающемся списке вариант Alternate Location (Другое место) и введите местоположение на локальном жестком диске (например, вы можете создать папку с именем ADRestore на диске C).

Выберите вариант Replace Existing Files (Заменить существующие файлы).

Установите флажки Restore Security Settings (Восстановить настройки безопасности) и Preserve Existing Volume Mount Points (Сохранить существующие точки монтирования томов).

Щелкните на кнопке Finish.

Ntbackup восстанавливает состояние системы в пяти подпапках в том месте, которое вы указали в мастере. Имена этих папок соответствуют следующим именам компонентов состояния системы:

Active Directory (база данных и файлы журналов)

Sysvol (политики и скрипты)

Boot Files (Файлы загрузки)

Registry (Реестр)

COM+ Class Registration Database (База данных регистрации классов COM+)

Если вы запускаете программу DCPROMO с новым ключом /adv, то она ищет эти подпапки.

В диалоговом окне Run (Выполнить) введите dcpromo /adv, чтобы запустить мастер Active Directory *Installation Wizard*. Используйте следующие инструкции, чтобы сделать свой выбор в каждом окне этого мастера.

Выберите вариант Additional Domain Controller for Exiting Domain.

Выберите вариант From These Restored *Backup Files* (Из следующих восстановленных файлов резервной копии) и укажите местоположение на локальном диске, где вы хотите восстановить резервную копию. Это должно быть место, где находятся приведенные выше пять подпапок.

Если исходный DC содержит глобальный каталог, то появится окно мастера, где спрашивается, хотите ли вы поместить глобальный каталог на этот DC. Выберите ответ Yes или No в зависимости от ваших планов конфигурирования. Процесс создания DC будет проходить несколько быстрее, если выбрать ответ Yes, но вы можете решить, что глобальный каталог нужно держать только на одном DC.

Введите опознавательные данные, позволяющие выполнить эту работу (имя и пароль администратора).

Введите имя домена, в котором будет действовать этот DC. Это должен быть домен, членом которого является исходный DC.

Введите местоположения для базы данных Active Directory и журналов (лучше всего использовать местоположения, заданные по умолчанию).

Введите местоположение для SYSVOL (и здесь лучше всего использовать местоположение по умолчанию).

Введите пароль администратора, чтобы использовать его на тот случай, если придется загружать этот компьютер в режиме Directory Services Restore Mode.

Щелкните на кнопке Finish.

Dcpromo повысит статус этого сервера до контроллера домена, используя данные, содержащиеся в восстановленных файлах, а это означает, что вам не придется ждать, пока будет выполнена репликация каждого объекта Active Directory с существующего DC на этот новый DC. Если какие-либо объекты будут изменены, добавлены или удалены после того, как вы начнете этот процесс, то при следующей репликации это будет вопросом нескольких секунд для нового DC.

По окончании этого процесса перезагрузите компьютер. После этого вы можете удалить папки, содержащие восстановленную резервную копию.

Вопросы

1. Для чего предназначена мониторин?

2. Перечислите предлагаемые способы установки контроллеров, мониторинга.

3. Что отображается на панели после установки приоритетов контроллера?

4. Опишите структуру восстановления после использования программы DCPROMO ?

**Инструкционно-технологическая карта**

**для выполнения лабораторная работа №14**

**Тема:** **Организация защиты данных в хранилищах.**

**Наименование работы: Мониторинг сетевого трафика**

**Цель:** научиться проводить мониторинг работы базы данных.

**Формируемые компоненты:** **ПК 11.1-11.6 ОК 1-11**

**Норма времени:**4 часа

**Оборудование и программное обеспечения:** ПК, MS SQL Server

**Основные правила ТБ при выполнении работы:**

-плавно нажимайте на клавиши, не допуская резких ударов;

-не пользуйтесь клавиатурой, если не подключено напряжение;

-работайте на клавиатуре чистыми руками;

-никогда не пытайтесь самостоятельно устранить неисправность в работе аппаратуры;

Строго запрещается:

-трогать разъемы соединительных кабелей;

-прикасаться к питающим проводам и устройствам заземления;

-прикасаться к экрану и к тыльной стороне монитора, клавиатуры;

-включать и отключать аппаратуру без указания преподавателя;

-Класть диск, книги, тетради на монитор и клавиатуру;

-Работать во влажной одежде и влажными руками **Литература:** Кумскова, И. А. Базы данных: учебник для СПО / И. А. Кумскова.- М.: КНОРУС, 2016.-488 с.

**Контрольные вопросы при допуске:**

1. Что такое БД?
2. Что такое мониторинг?
3. Что такое сетевой трафик?

**порядок выполнения работы**

Выполнить предложенные способы сброса настроек.

Сделайте отчет о выполнении предложенных заданий.

*Получить помощь в решении проблемы*

Чтобы устранить проблему автоматически, щелкните ссылку Устранить проблему. Затем в диалоговом окне Загрузка файла нажмите кнопку Выполнить и следуйте указаниям мастера.

Примечание. Интерфейс этого мастера может быть доступен только на английском языке, однако автоматическое исправление можно выполнять и в других языковых версиях Windows.

Примечание. Если используется не тот компьютер, на котором выявлена проблема, можно сохранить автоматическое исправление на устройстве флэш-памяти или компакт-диске, чтобы затем выполнить его на нужном компьютере.

Теперь перейдите к разделу [Проблема устранена?](https://infourok.ru/go.html?href=%23diditfix)

*Решить проблему самостоятельно*

Сброс параметров TCP/IP вручную в ОС Windows XP.

Примечание. Этот способ рассчитан на опытных пользователей. Если продвинутые способы устранения неполадок кажутся слишком сложными, обратитесь за помощью к специалисту или в службу поддержки. Сведения о том, как связаться со службой поддержки, см. на веб-сайте справки и поддержи Майкрософт по следующему адресу:

http://support.microsoft.com/contactus/?ln=ru

В программе NetShell имеется команда сброса для протокола IP. Следуйте приведенным ниже указаниям, чтобы вручную сбросить настройки TCP/IP при помощи команды reset.

Чтобы открыть командную строку, выберите в меню Пуск команду Выполнить. Скопируйте и вставьте (или введите) в поле Открыть следующую команду и нажмите клавишу ВВОД:cmd

Скопируйте и вставьте (или введите самостоятельно) в командной строке следующую команду и нажмите клавишу ВВОД:netsh int ip reset c:\resetlog.txt

Примечание. Если не хотите указывать путь для сохранения файла журнала, используйте следующую команду:netsh int ip reset resetlog.txt

Перезагрузите компьютер.

Сброс параметров TCP/IP вручную в ОС Windows Vista и Windows 7

Примечание. Этот раздел рассчитан на опытных пользователей. Если продвинутые способы устранения неполадок кажутся слишком сложными, обратитесь за помощью к специалисту или в службу поддержки. Чтобы узнать, как связаться со службой поддержки, перейдите на страницу контактных данных веб-сайта справки и поддержи Майкрософт по следующему адресу:http://support.microsoft.com/contactus/?ln=ru

В служебной программе NetShell имеется команда reset для протокола IP. Следуйте приведенным ниже указаниям, чтобы вручную сбросить параметры TCP/IP при помощи команды reset.

Чтобы открыть командную строку, нажмите кнопку Пуск, затем в поле Найти программы и файлы введите CMD.

Щелкните правой кнопкой мыши значок файла CMD.exe в списке Программы и выберите команду Запуск от имени администратора.

В окне Контроль учетных записей нажмите кнопку Да.

Скопируйте и вставьте (или введите самостоятельно) в командной строке следующую команду и нажмите клавишу ВВОД:

netsh int ip reset c:\resetlog.txt

Примечание. Если не хотите указывать путь для сохранения файла журнала, используйте следующую команду:

netsh int ip reset resetlog.txt

Перезагрузите компьютер.

При выполнении команды reset, она перезаписывает два ключа в реестре, которые используются TCP/IP. Это то же самое, что удалить и вновь установить протокол. Команда reset перезаписывает следующие два ключа в реестре:

SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\

SYSTEM\CurrentControlSet\Services\DHCP\Parameters\

Для успешного выполнения команды необходимо указать имя файла журнала, в котором будут регистрироваться действия, выполненные программой netsh. При выполнении команды вручную происходит сброс протокола TCP/IP и выполненные действия записываются в файл журнала, который в данной статье называется resetlog.txt.

Первый пример, c:\resetlog.txt, создает путь, куда файл журнала будет сохранен. Второй пример, resetlog.txt, создает файл журнала в текущем каталоге. Если журнал с таким именем уже существует, новый журнал будет добавлен в конец файла.