

Документ подписан простой электронной подписью

Информация о владельце: **Министерство науки и высшего образования РФ**

ФИО: Баламирзоев Назим Лиодинович

Должность: Ректор

Дата подписания: 07.06.2024 08:35:10

Уникальный программный ключ:

5cf0d6f89e80f49a334f6a4ba58e91f3326b9926

**ФГБОУ ВО «ДГТУ»**

**Методические указания  
к прохождению учебной практики по  
ПМ.04 «Разработка, администрирование и защита баз данных»  
для студентов специальности  
09.02.07 «Информационные системы и программирование»,  
квалификация «Программист»**

**Махачкала, 2022**

## **Введение**

Методические указания по выполнению практических работ по учебной практике по ПМ.04 «Разработка, администрирование и защита баз данных» разработаны в соответствии с рабочей программой учебной практики и предназначены для приобретения необходимых практических навыков и закрепления теоретических знаний, полученных обучающимися при изучении учебных дисциплин, обобщения и систематизации знаний перед экзаменом.

Методические указания предназначены для обучающихся специальности 09.02.07 Информационные системы и программирование.

### **Рекомендации по оформлению практической работы**

Задания выполняются обучающимися по шагам. Необходимо строго придерживаться порядка действий, описанного в практической работе.

Результаты выполнения практических заданий необходимо сохранять в своей папке на компьютере.

В случае пропуска занятий обучающийся осваивает материал самостоятельно в свободное от занятий время и сдает практическую работу с пояснениями о выполнении.

### **Критерии оценки практической работы**

- наличие цели выполняемой работы, выполнение более половины основных заданий (удовлетворительно);
- наличие цели выполняемой работы, выполнение всех основных и более половины дополнительных заданий (хорошо);
- наличие цели выполняемой работы, выполнение всех основных и индивидуальных заданий (отлично).

## КРАТКАЯ ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

**База данных – интеграция данных**, предназначенных для решения нескольких задач разных пользователей. Должны учитываться требования к данным каждого пользователя (его представление о данных и связях между ними). Эти требования должны обобщаться в **единое представление**, которое и будет служить основой для построения единой БД.

**Проектирование БД** – процесс создания схемы базы данных и определения необходимых ограничений целостности.

Основные задачи:

- хранение в БД всей необходимой информации.
- получение данных по всем необходимым запросам.
- сокращение избыточности и дублирования данных.
- обеспечение целостности данных (исключение противоречий, предотвращение потерь и т.д.).

### Этапы проектирования

---

- **Сбор данных**
- **Концептуальное проектирование**
- **Логическое проектирование**
- **Физическое проектирование**

Полнота и достаточность базы данных в какой-либо предметной области в значительной степени зависит от **сбора информации** предметной области базы данных. **Предметной областью** называется часть реальной системы, подлежащая исследованию с целью ее описания. Предметная область делится на полную и фрагментарную, при этом каждый фрагмент может представлять собой свою предметную область. Например, для университета можно выделить следующие фрагменты: факультет, кафедра, деканат, бухгалтерия, отдел кадров и др.

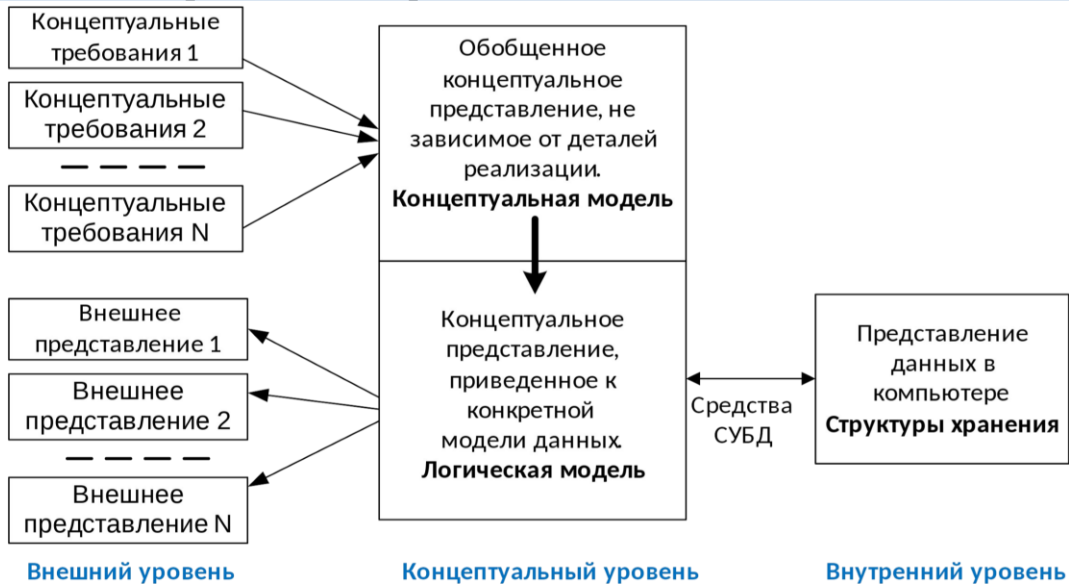
**Концептуальное (инфологическое) проектирование** — построение семантической (смысловой) модели предметной области. Такая модель создаётся без ориентации на какую-либо конкретную СУБД и модель данных. Термины «семантическая модель», «модель базы данных», «модель предметной области», «концептуальная модель», «концептуальная модель базы данных», «концептуальная модель предметной области» и «инфологическая модель» являются синонимами. *Концептуальное проектирование* - создание концептуальной модели данных, исходя из представлений пользователей о предметной области. Конкретный вид и содержание концептуальной модели базы данных определяется выбранным для этого формальным аппаратом. Обычно используются графические нотации, подобные ER (entity-relation)-диаграммам.

**Логическое проектирование** - преобразование концептуальной модели на основе выбранной модели данных в логическую модель, не зависящую от особенностей СУБД

**Физическое проектирование** - описание конкретной реализации базы данных

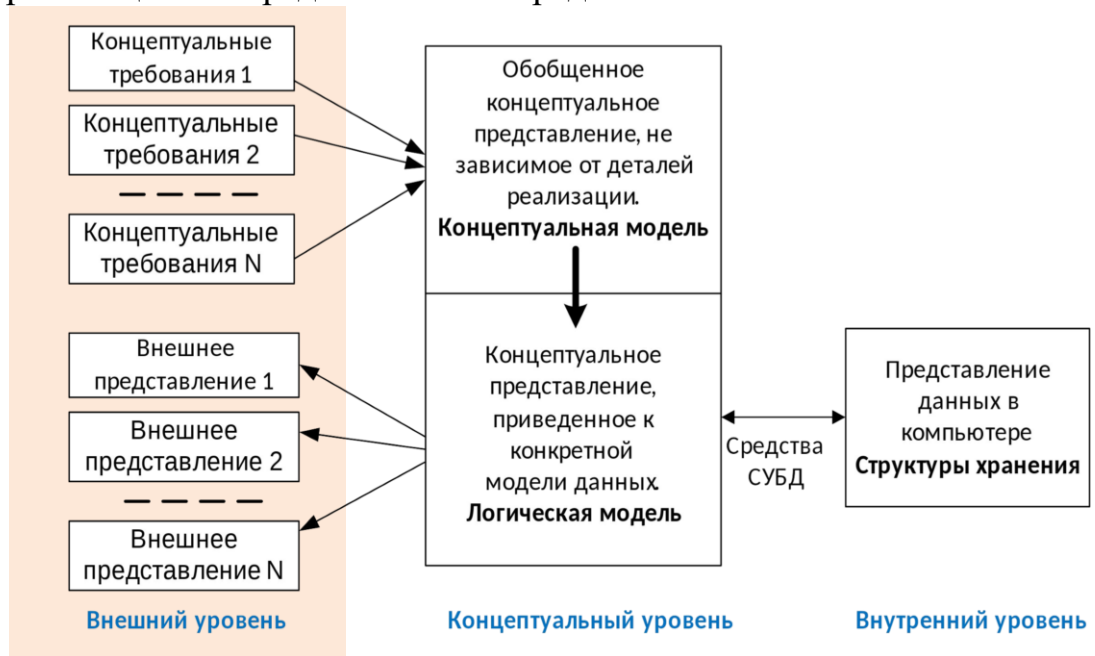
## Три уровня данных

**Основная цель** - отделение пользовательского представления о данных от их физического представления.



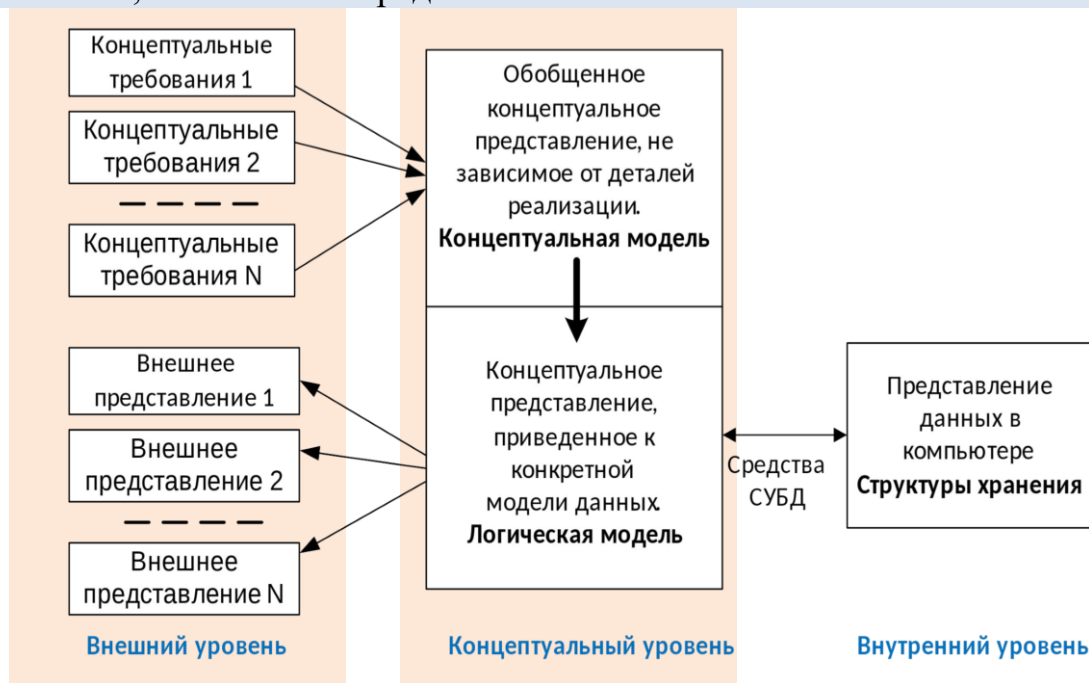
Прикладные программы работают с логической моделью.

Пользователю представляется подмножество логической модели, отражающее его представление о предметной области.



**Внешнее представление** – для специалиста предметной области (пользователя). Работают только с интерфейсом, с элементами

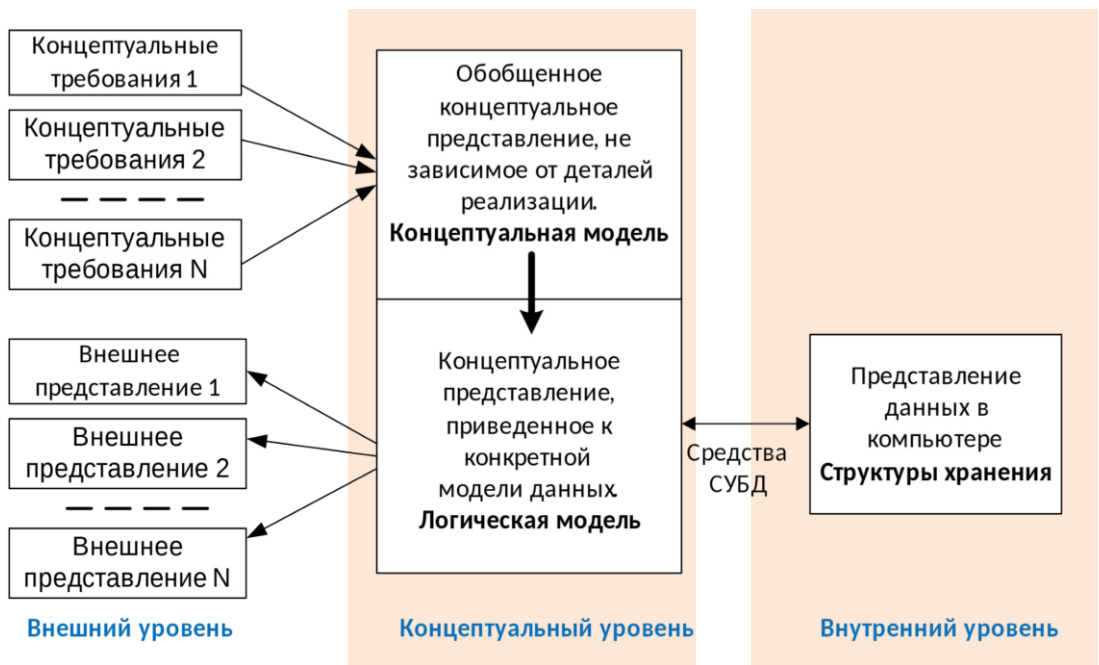
управления. Пользовательский интерфейс должен быть удобным, понятным, связанным с предметной областью.



**Внешнее представление и логическая модель** – для прикладного программиста.

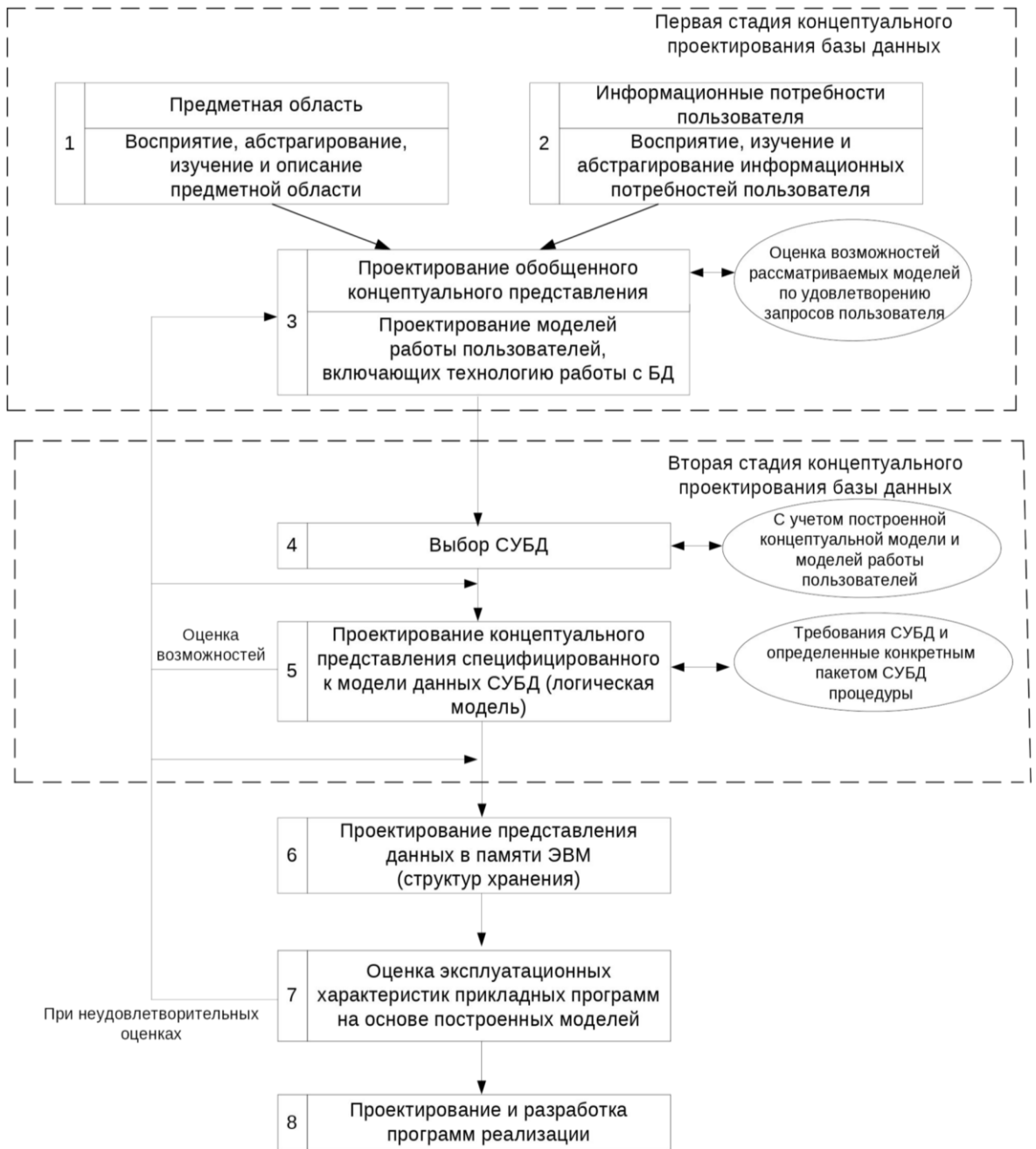
Программисты разрабатывают:

- пользовательский интерфейс и функционал приложения (это связано с предметной областью),
- взаимодействие с логическими структурами данных (абстракция, не связанная с предметной областью).



**Логическая модель и внутреннее представление – для администратора базы данных.**

Администратор БД может быть полностью абстрагирован от предметной области.





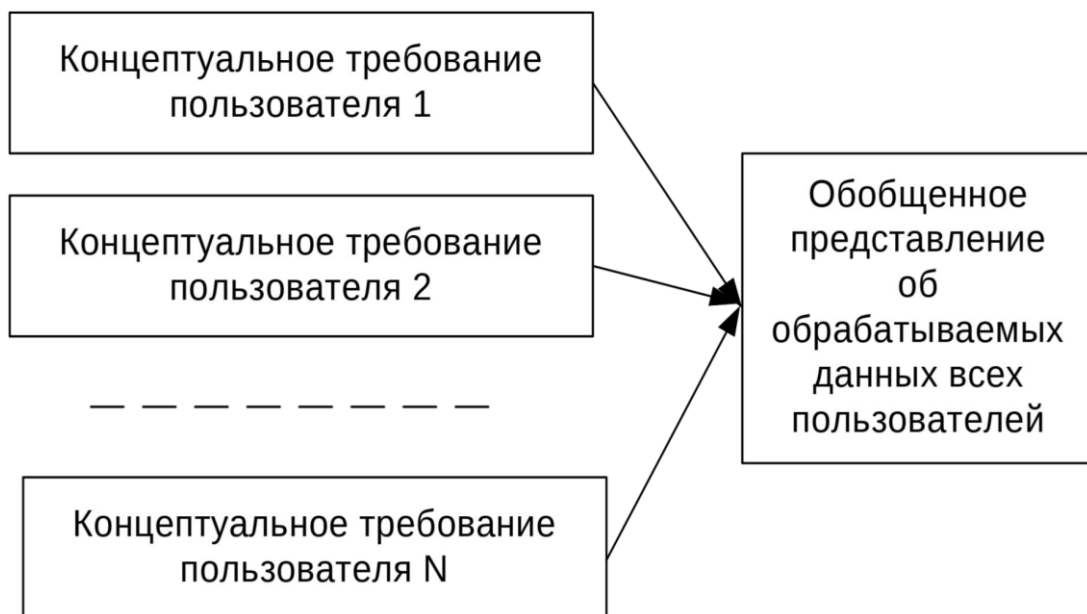
## 1 ЭТАП - КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ

---

- Определение сущностей
- Определение связей между сущностями
- Создание ER-модели предметной области
- Определение атрибутов (имя, тип, значение по умолчанию, NULL)
- Определение наборов допустимых значений атрибутов
- Определение первичных ключей для сущностей

### Концептуальная модель базы данных

Информационное описание предметной области с учетом логических взаимосвязей.



## Терминология

**Сущность** (Entity) или объект – то, о чем будет накапливаться информация в информационной системе.

**Правильная (сильная) сущность** не зависит от существования других объектов.

**Слабая сущность** не может существовать при отсутствии некоторой другой сущности.

**Атрибут** – поименованное свойство (характеристика) сущности. Совокупность атрибутов = запись об объекте.

Конкретная сущность = **экземпляр сущностей**. Экземпляры сущностей должны однозначно идентифицироваться по одному или нескольким атрибутам.

**Связь (отношение)** описывает имеющуюся в предметной области логическую связь между сущностями.

**Сущность: СТУДЕНТ**

**Атрибуты:**

- номер зачетной книжки,
- фамилия,
- дата рождения,
- место рождения

**Экземпляр сущностей:**

{123456, Магомедов, 01.01.2000, Махачкала }

## Связи в ER-диаграмме

• **Связь 1:1.** Одиночный экземпляр сущности одного класса связан с одиночным экземпляром сущности другого класса.

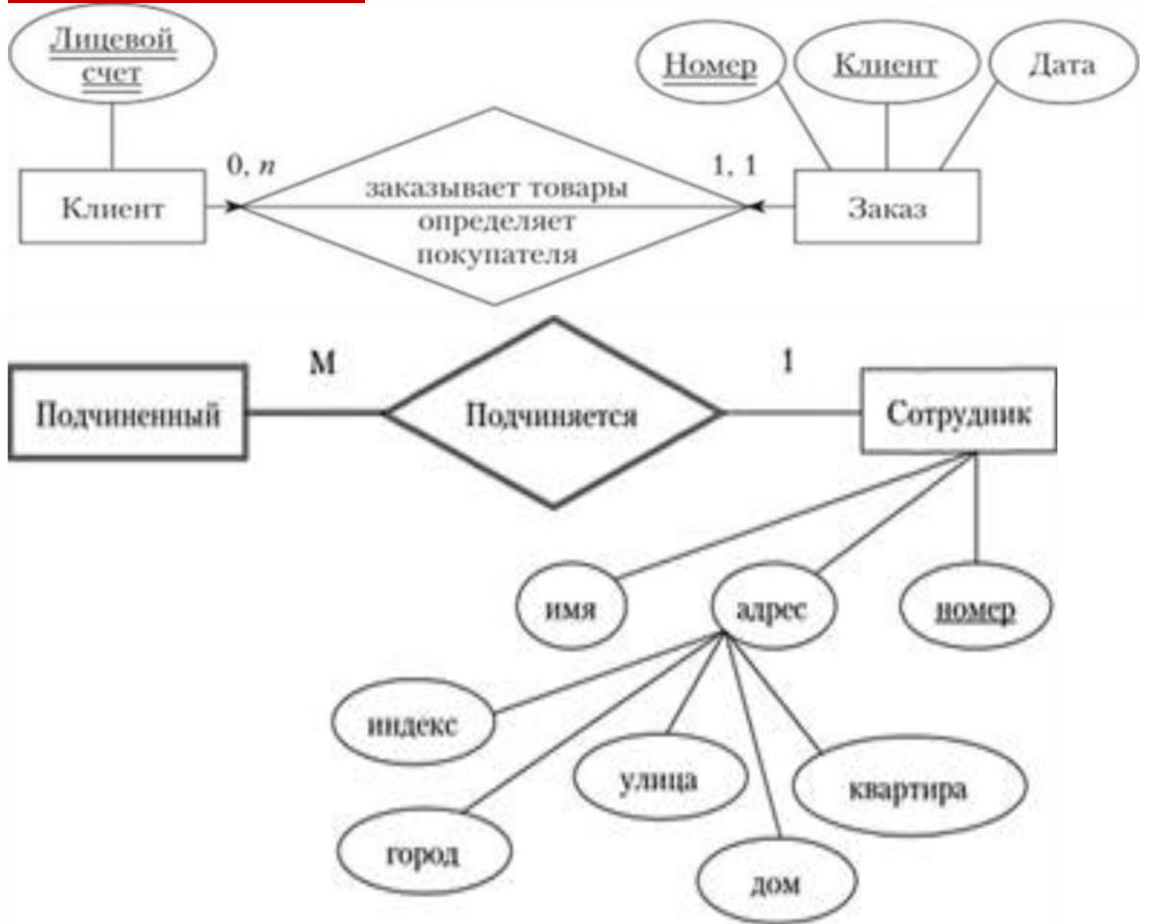
• **Связь 1:M.** Единичный экземпляр сущности одного класса связан со многими экземплярами сущности другого класса.

• **Связь M:N.** Несколько экземпляров сущности одного класса связаны с несколькими экземплярами сущности другого класса.

## Диаграмма сущностей–связей (Entity–Relationship)



## Нотация Чена



## Нотация Мартина (Crow's Foot)



One



Many



One (and only one)



Zero or one



One or many



Zero or many

## 2 ЭТАП - ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ БД

- Выбор модели данных
- Определение наборов структурированных данных.  
Нормализация таблиц (для реляционной модели)
- Проверка модели данных на предмет выполнения всех транзакций бизнес-процессов
- Определение требований поддержки целостности данных.

### Логическое проектирование БД

- **Сущности** концептуальной модели описываются как записи, состоящие из полей.
- Каждый **атрибут** описывается как поле с типом и характеристиками, возможными в выбранной модели данных.
- **Связи** концептуальной модели описываются в понятиях, соответствующих выбранной модели данных.
- Определяется порядок реализации запросов пользователей к базе данных с помощью типовых операций СУБД, поддерживающих выбранную модель данных, и т.д.

### Нотация IDEF1X



### **3 ЭТАП - ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ БД**

---

- Проектирование таблиц средствами выбранной СУБД
- Реализация бизнес-правил в среде выбранной СУБД
- Проектирование физической организации СУБД
- Разработка стратегии защиты БД
- Организация мониторинга функционирования БД и ее настройка

## ПРАКТИЧЕСКАЯ ЧАСТЬ

### ПРАКТИЧЕСКАЯ РАБОТА № 1

#### «Преобразование реляционной БД в сущности и связи»

**Цель работы:** выработать практические навыки моделирования предметной области и построения ER-модели данных, закрепить технологию проектирования БД, закрепить основные понятия теории реляционных баз данных, освоить технологию построения ER-диаграмм, научиться получать реляционные БД из ER-диаграмм.

#### Теоретические сведения

##### Сущность-связь

Работа с базой данных начинается с построения модели. Наиболее распространенной является ER-модель (entity-relationship model) - модель "Сущность-связь". Для "ручного" построения ER-модели на практике будем использовать простую систему обозначений, предложенную Питером Ченом (обозначения, встречающиеся в разных источниках, могут отличаться от нижеприведенных):





Сущность (объект)	 Сотрудник
Атрибут сущности (свойство, характеризующее объект)	 ФИО
Ключевой атрибут (атрибут, входящий в первичный ключ)	 Номер сотр.
Связь	 Работает

Рисунок 1.1. Сущность - связь

Первичный ключ - атрибут или группа атрибутов, однозначно идентифицирующих объект. Первичный ключ может состоять из нескольких атрибутов, тогда подчеркивается каждый из них.



### **Связи между объектами могут быть 3-х типов:**

*Один - к одному.* Этот тип связи означает, что каждому объекту первого вида соответствует не более одного объекта второго вида, и наоборот. Например: сотрудник может руководить только одним отделом, и у каждого отдела есть только один руководитель.

*Один - ко многим.* Этот тип связи означает, что каждому объекту первого вида может соответствовать более одного объекта второго вида, но каждому объекту второго вида соответствует не более одного объекта первого вида. Например: в каждом отделе может быть множество сотрудников, но каждый сотрудник работает только в одном отделе.

*Многие - ко многим.* Этот тип связи означает, что каждому объекту первого вида может соответствовать более одного объекта второго вида, и наоборот. Например: каждый счет может включать множество товаров, и каждый товар может входить в разные счета.

### **Реляционная структура данных**

В конце 60-х годов появились работы, в которых обсуждались возможности применения различных табличных даталогических моделей данных. Наиболее значительной из них была статья сотрудника фирмы IBM д-ра Эдварда Кодда (Codd E.F., A Relational Model of Data for Large Shared Data Banks. SACM 13: 6, June 1970), где впервые был применен термин «реляционная модель данных».

Будучи математиком по образованию, Э. Кодд предложил использовать для обработки данных аппарат теории множеств (объединение, пересечение, разность, декартово произведение). Он показал, что любое представление данных сводится к совокупности двумерных таблиц особого вида, известного в математике как *отношение* – relation.

*Наименьшая единица данных реляционной модели – это отдельное атомарное (неразложимое) для данной модели значение данных.*

Так, в одной предметной области фамилия, имя и отчество могут рассматриваться как единое значение, а в другой – как три различных значения.

Доменом называется множество атомарных значений одного и того же типа. Так, в таблице 1.1 домен пунктов отправления (назначения) – множество названий населенных пунктов, а домен номеров рейса – множество целых положительных чисел.

Таблица 1.1. Рейс

Номер рейса	Дни недели	Пункт отправления	Время вылета	Пункт назначения	Время прибытия	Тип самолета	Стоимость билета
138	2_4_7	Баку	21:12	Москва	0:52	ИЛ-86	115.00
57	3_6	Ереван	7:20	Киев	9:25	ТУ-154	92.00
1234	2_6	Казань	22:40	Баку	23:50	ТУ-134	73.50
242	1 по 7	Киев	14:10	Москва	16:15	ТУ-154	57.00
86	2_3_5	Минск	10:50	Сочи	13:06	ИЛ-86	78.50
137	1_3_6	Москва	15:17	Баку	18:44	ИЛ-86	115.00
241	1 по 7	Москва	9:05	Киев	11:05	ТУ-154	57.00
577	1_3_5	Рига	21:53	Таллин	22:57	АН-24	21.50
78	3_6	Сочи	18:25	Баку	20:12	ТУ-134	44.00
578	2_4_6	Таллин	6:30	Рига	7:37	АН-24	21.50

Смысл доменов состоит в следующем. Если значения двух атрибутов берутся из одного и того же домена, то, вероятно, имеют смысл сравнения, использующие эти два атрибута (например, для организации транзитного рейса можно дать запрос «Выдать рейсы, в которых время вылета из Москвы в Сочи больше времени прибытия из Архангельска в Москву»). Если же значения двух атрибутов берутся из различных доменов, то их сравнение, вероятно, лишено смысла: стоит ли сравнивать номер рейса со стоимостью билета?

Отношение на доменах  $D_1, D_2, \dots, D_n$  состоит из заголовка и тела. На рис. 1.2 приведен пример отношения для расписания движения самолетов (таблица 1).  $A_i$  - атрибуты,  $V_i$  - значения атрибутов.

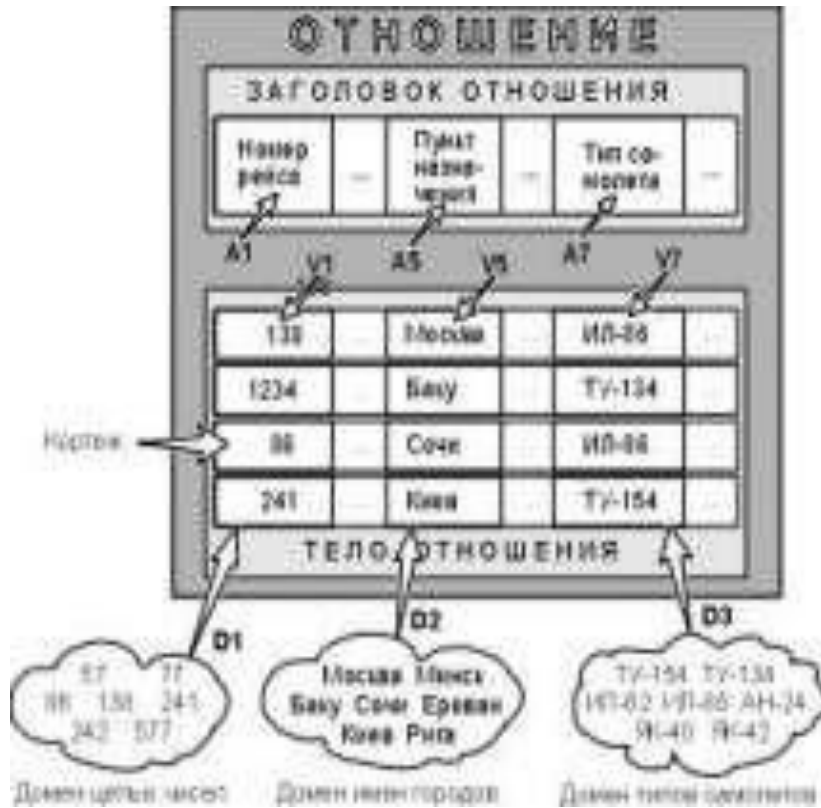


Рисунок 1.2. Отношение с математической точки зрения

Заголовок отношения состоит из такого фиксированного множества атрибутов  $A_1, A_2, \dots, A_n$ , что существует взаимно однозначное соответствие между этими атрибутами  $A_i$  и определяющими их доменами  $D_i$  ( $i=1,2,\dots,n$ ).

Тело отношения состоит из меняющегося во времени множества кортежей, где каждый кортеж состоит в свою очередь из множества пар атрибут-значение  $(A_i:V_i)$ , ( $i=1,2,\dots,n$ ), по одной такой паре для каждого атрибута  $A_i$  в заголовке.

Для любой заданной пары атрибут-значение  $(A_i:V_i)$   $V_i$  является значением из единственного домена  $D_i$ , который связан с атрибутом  $A_i$ .

Степень отношения – это число его атрибутов.

Отношение степени один называют унарным, степени два – бинарным, степени три – тернарным, ..., а степени  $n$  –  $n$ -арным. Степень отношения «Рейс» равна 8 (таблица 1.1).

*Кардинальное число или мощность отношения – это число его кортежей.*

Мощность отношения «Рейс» равна 10. Кардинальное число отношения изменяется во времени в отличие от его степени.

Поскольку отношение – это множество, а множества по определению не содержат совпадающих элементов, то никакие два кортежа отношения не могут быть дубликатами друг друга в любой произвольно заданный момент времени.

*Пусть  $R$  – отношение с атрибутами  $A_1, A_2, \dots, A_n$ . Говорят, что множество атрибутов  $K=(A_i, A_j, \dots, A_k)$  отношения  $R$  является возможным ключом  $R$  тогда и только тогда, когда удовлетворяются два независимых от времени условия:*

1. Уникальность: в произвольный заданный момент времени никакие два различных кортежа  $R$  не имеют одного и того же значения для  $A_i, A_j, \dots, A_k$ .
2. Минимальность: ни один из атрибутов  $A_i, A_j, \dots, A_k$  не может быть исключен из  $K$  без нарушения уникальности.

Каждое отношение обладает хотя бы одним возможным ключом, поскольку по меньшей мере комбинация всех его атрибутов удовлетворяет условию уникальности. Один из возможных ключей (выбранный произвольным образом) принимается за его первичный ключ. Остальные возможные ключи, если они есть, называются альтернативными ключами.

Вышеупомянутые и некоторые другие математические понятия явились теоретической базой для создания реляционных СУБД, разработки соответствующих языковых средств и программных систем, обеспечивающих их высокую производительность, и создания основ теории проектирования баз данных.

Также на практике широко используются неформальные эквиваленты этих понятий: Отношение – Таблица, Кортеж – Строка таблицы или Запись, Атрибут – Столбец Таблицы или Поле.

При этом принимается, что «запись» означает «экземпляр записи», а «поле» означает «имя и тип поля».

### **Задание для практической работы**

#### **Задание 1. Проектирование реляционных Баз Данных.**

##### **Вариант 1.**

Построить реляционную таблицу Базы Данных об окончивших общеобразовательных учреждениях студентами группы, содержащую данные о названии школы, годе окончания, виде аттестата.

##### **Вариант 2.**

Построить реляционную таблицу Базы Данных домов, где живут студенты группы, содержащую данные о районе расположения дома, количестве этажей в нем и номере этажа, где живет студент.

##### **Вариант 3.**

Построить реляционную таблицу Базы Данных дней рождения студентов группы, содержащую данные о дате рождения, месте рождения и национальности.

##### **Вариант 4.**

Построить реляционную таблицу Базы Данных сотовых телефонов студентов группы, содержащую данные о модели телефона, типе корпуса, операторе.

##### **Вариант 5.**

Построить реляционную таблицу Базы Данных студентов факультета, содержащую данные о курсе, группе, специальности, квалификации.

##### **Вариант 6.**

Построить реляционную таблицу Базы Данных учебных дисциплин, содержащую данные о названии дисциплины, количестве часов лекций, практических занятий.

Вариант 7.

Построить реляционную таблицу Базы Данных преподавателей кафедры, содержащую данные о фамилии, должности, возрасте преподавателя.

Вариант 8.

Построить реляционную таблицу Базы Данных учебных аудиторий факультета, содержащую данные о номере кабинета, площади кабинета, количестве посадочных мест.

Вариант 9.

Построить реляционную таблицу Базы Данных библиотечного фонда факультета, содержащую данные о наименовании литературного источника, ФИО авторов, годе издания, количестве экземпляров.

Вариант 10.

Построить реляционную таблицу Базы Данных корпусов университета, содержащую данные о номере корпуса, месте расположения, количестве аудиторий.

Вариант 11.

Построить реляционную таблицу Базы Данных практик студентов, содержащую данные о типе практике, продолжительности, руководителе практики.

Вариант 12.

Построить реляционную таблицу Базы Данных результатов экзаменационных сессий студентов, содержащую данные о названии учебной дисциплины, оценки, дате экзамена.

## **Технология выполнения работы и оформление отчета**

- 1.1. Придумайте заголовок отношения и запишите его в отчет.
- 1.2. Определите атрибуты отношения. Дополните список атрибутов по варианту двумя дополнительными атрибутами.
- 1.3. Начертите сетку таблицы в отчет и занесите в нее атрибуты.
- 1.4. Выполните сбор данных по фрагменту предметной области, занесите полученные данные в таблицу.
- 1.5. На чертеже таблицы укажите чему соответствуют понятия: Заголовок отношения, тело отношения, атрибут отношения, кортеж отношения.
- 1.6. Определите и запишите в отчет степень отношения и мощность отношения.
- 1.7. Дайте определение первичного ключа. Укажите Первичный ключ получившегося отношения
- 1.8. Докажите, что у вас получилась реляционная таблица, для этого укажите типы данных всех атрибутов.

### **Задание 2. Проектирование Баз Данных. ER-диаграммы.**

Формулировка задания. По описанию предметной области построить логическую модель БД методом ER-диаграмм, на основании которой построить набор таблиц БД.

Вариант 1.

Описание предметной области (Ресторан).

Посетители ресторана обслуживаются за столиками. За одним столом может располагаться не более 4 посетителей, каждый из которых может сделать заказ тех или иных блюд. Столики обслуживают официанты. У одного официанта в обслуживании несколько столов.

Задачи для БД:

- Есть ли свободные столы?
- Сколько посетителей обслужил официант за смену?

- Сколько каких блюд было реализовано?

Вариант 2.

Описание предметной области (Колледж).

Студенты колледжа объединены в группы. Набор дисциплин, изучаемых студентом, зависит от номера группы в которой он учится. Преподаватели читают дисциплины и выставляют зачеты студентам. Один преподаватель может читать несколько дисциплин, но каждую дисциплину ведет один преподаватель.

Задачи для БД:

- Какие дисциплины изучает студент?
- Какая оценка у студента по данной дисциплине?
- Кто выставил эту оценку?

Вариант 3.

Описание предметной области (Театральная касса).

В театральной кассе продаются билеты на спектакли. Стоимость билета зависит от ряда, театра и спектакля. Каждый день в театре может идти не более одного спектакля. Спектакль характеризуется названием и автором. Каждый покупатель может купить сколько угодно билетов на любые спектакли.

Задачи для БД:

- Какие спектакли идут в определенный день?
- Есть ли билеты на конкретный спектакль?
- Сколько стоит конкретный билет?

Вариант 4.

Описание предметной области (Грузоперевозки).

АТП имеет грузовые автомобили с гос. номерами и организует перевозки для своих заказчиков. Стоимость перевозки зависит от расстояния и грузоподъемности автомобиля, который ее выполняет. Каждый заказчик может сделать заказ нескольких перевозок. Одну перевозку выполняет один грузовик.



Задачи для БД:

- Какие грузовики свободны?
- Какой заказчик сделал самый дорогой заказ?
- Какой грузовик выполнил наибольшее количество заказов?

## **Технология выполнения работы**

### **1. Построение ER-диаграммы.**

1.1. Выберите из описания предметной области все существительные. Продумайте, какие из них будут соответствовать сущностям, а какие атрибутам сущностей. Зарисуйте в отчет все сущности с их атрибутами согласно обозначениям, принятым в ER-диаграммах.

1.2. На рисунке подчеркиванием атрибутов обозначьте для каждой сущности уникальный идентификатор (Ключ). При необходимости добавьте сущностям атрибуты, которые помогут однозначно отличить каждый экземпляр сущности.

1.3. Определите и включите в схему связи сущностей. Подпишите названия связей и пронумеруйте связи. Для первой связи укажите тип и модальность. Для всех связей запишите их прочтение слева направо и справа налево.

1.4. Если в схеме присутствуют связи типа «много-со-многими» уберите их путем ввода дополнительной сущности. Измененную схему зарисуйте в отчет.

### **2. Получение реляционной схемы из ER-диаграммы.**

2.1. Каждая сущность превращается в таблицу. Имя сущности – имя таблицы. Набор всех таблиц – БД. Вспомните, что такое схема БД. Запишите схему вашей БД в отчет.

2.2. Зарисуйте все полученные таблицы с их заголовками и названиями столбцов. Выделите потенциальные и внешние ключи (если есть) для каждой таблицы. Укажите столбцы, допускающие неопределенные значения.

Докажите, что полученные отношения находятся в Первой нормальной форме.

### **Пояснение. Первая нормальная форма**

Отношение находится в 1НФ, если все его атрибуты являются простыми, все используемые домены должны содержать только скалярные значения. Не должно быть повторений строк в таблице.

Например, есть таблица «Автомобили»:

Фирма	Модели
BMW	M5, X5M, M1
Nissan	GT-R

Нарушение нормализации 1НФ происходит в моделях BMW, т.к. в одной ячейке содержится список из 3 элементов: M5, X5M, M1, т.е. он не является атомарным. Преобразуем таблицу к 1НФ:

Фирма	Модели
BMW	M5
BMW	X5M
BMW	M1
Nissan	GT-R

## ПРАКТИЧЕСКАЯ РАБОТА 2

### *Тема: Нормализация реляционной модели данных методом декомпозиции отношений*

**Цель работы:** освоение основных понятий и определений нормализации отношений, получение практических навыков по нормализации реляционной модели данных методом декомпозиции.

#### **Теоретические сведения**

**Нормализация** – это разбиение таблицы на две или более, обладающих лучшими свойствами при включении, изменении и удалении данных. Окончательная цель нормализации сводится к получению такого проекта базы данных, в котором каждый факт появляется лишь в одном месте, т.е. исключена избыточность информации. Это делается не столько с целью экономии памяти, сколько для исключения возможной противоречивости хранимых данных.

В ненормализованной форме таблица может хранить информацию о двух и более сущностях. Также она может содержать повторяющиеся столбцы. Также столбцы могут хранить повторяющиеся значения. В нормализованной же форме каждая таблица хранит информацию только об одной сущности.

Процесс нормализации – формальный метод для оптимизации столбцов отношений и устранения аномалий.

#### **Избыточность данных и аномалии обновления**

Основная цель проектирования реляционной БД – группирование атрибутов в отношениях таким образом, чтобы минимизировать избыточность данных (сокращение объема вторичной памяти для хранения БД) и повышение надежности при работе с данными.

Обычно процесс проектирования отношений реляционной БД ведется на основе разработанной ER-диаграммы или на основе просто здравого смысла разработчика. В общем случае при таком подходе расположение атрибутов в отношениях *неоптимальное*. При работе с отношениями, содержащими избыточные данные, могут возникать проблемы – *аномалии обновления*.

Аномалии обновления делят на три вида:

- *аномалии вставки* – возникают при добавлении новых несогласованных данных (нарушающих целостность данных в отношении);
- *аномалии изменения* – возникают при изменении части ранее введенных данных; частичное обновление сведений приведет к нарушению целостности данных отношения;
- *аномалии удаления* – возникают при удалении строк из отношений.

Обычно для решения проблем избыточности и аномалий выполняется деление отношения на такие отношения, в которых избыточности не будет. Для выполнения такого процесса необходимо выявить все зависимости между атрибутами отношения (потеря одной такой зависимости меняет модель внешнего мира).

### Функциональные зависимости

Выявление смысловой зависимости между данными – один из способов формализации смысловой информации о данных.

*Функциональная зависимость* описывает связь типа «многие-к-одному» между атрибутами отношения, где «много» – детерминант функциональной зависимости. Функциональная зависимость является семантическим свойством атрибутов отношения.

Если в отношении  $R$ , содержащем атрибуты  $A$  и  $B$ , атрибут  $B$  функционально зависит от атрибута  $A$  ( $A$  является детерминантом атрибута  $B$ )  $A \twoheadrightarrow B$ , то в каждом кортеже этого отношения каждое конкретное значение атрибута  $A$  всегда связано только с одним значением атрибута  $B$ .

Особенности функциональных зависимостей, лежащие в основе процесса нормализации:

- функциональная зависимость является специализированным правилом целостности – она накладывает ограничения на допустимые значения атрибутов отношений; эту особенность можно использовать при обновлении БД, т.к. зная, какие функциональные зависимости есть в отношении, можно понять, нарушат ли новые данные целостность данных отношения;
- функциональная зависимость является обобщением понятия

потенциального ключа; функциональные зависимости позволяют определить все потенциальные ключи отношения (и соответственно – первичный ключ): все атрибуты отношения, которые не являются частью первичного (или потенциального) ключа, должны функционально зависеть от этого ключа; если не все остальные атрибуты отношения зависят от некоторого детерминанта, то этот детерминант не является потенциальным ключом этого отношения.

### **Нормальные формы и нормализация методом декомпозиции**

*Нормализация* – это формальный метод анализа отношений на основе их первичного ключа и существующих функциональных зависимостей.

Суть процесса нормализации:

- в нормализованных отношениях не разрешаются никакие функциональные зависимости, кроме функциональных зависимостей вида  $K \twoheadrightarrow A$ , где  $K$  – потенциальный ключ отношения  $R$ , а  $A$  – неключевой атрибут;
- если же отношение  $R$  имеет функциональные зависимости  $B \twoheadrightarrow A$ , где  $B$  не является потенциальным ключом, то в отношении  $R$  будет наблюдаться избыточность данных.

В процессе нормализации реляционных отношений применяются концепции *нормальных форм*. Говорят, что отношение находится в определенной нормальной форме, если оно удовлетворяет правилам этой нормальной формы. В настоящее время используется семь нормальных форм, которые зависят друг от друга путем усложнения (вложенности) набора правил:

$1NF \twoheadrightarrow 2NF \twoheadrightarrow 3NF \twoheadrightarrow НФБК \twoheadrightarrow 4NF \twoheadrightarrow 5NF \twoheadrightarrow 6NF$ .

Каждая нормальная форма, таким образом, **удовлетворяет всем предыдущим нормальным формам**. Более высокая нормальная форма приводит к более строгому формату отношения (меньшее число аномалий обновления).

**Примечание.** БД можно построить и на отношениях, находящихся в первой нормальной форме, но такая БД будет сильно подвержена аномалиям и избыточности данных.

**На практике желательно использовать, как минимум, *3NF***, чтобы устранить большинство аномалий обновления.

1) *1НФ*. Отношение находится в *1НФ* тогда и только тогда, когда в любом допустимом значении этого отношения каждый кортеж содержит только одно значение для каждого из атрибутов, т.е. это значение не имеет внутренней структуры (множество, таблица и т.п.). Отношения в *1НФ* имеют большое количество аномалий обновления.

2) *2НФ*. Отношение находится в *2НФ* тогда и только тогда, когда оно находится в *1НФ*, и каждый атрибут отношения, не входящий в состав первичного ключа, характеризуется полной функциональной зависимостью от этого первичного ключа.

**Полной функциональной зависимостью** называется такая зависимость  $A \square B$ , когда  $B$  функционально зависит от  $A$  и не зависит ни от какого подмножества  $A$  (т.е. удаление какого-либо атрибута из  $A$  приведет к утрате этой функциональной зависимости). *2НФ* устраняет в отношении частичные функциональные зависимости неключевых атрибутов от первичного ключа, которые выносятся в отдельное отношение вместе с копиями своих детерминантов (частей первичного ключа, от которого они зависят).

3) *3НФ*. Отношение находится в *3НФ* тогда и только тогда, когда оно находится в *2НФ* и не имеет не входящих в первичный ключ атрибутов, которые находились бы в транзитивной функциональной зависимости от этого первичного ключа.

Транзитивной функциональной зависимостью называется зависимость  $A \square C$ , если существуют зависимости  $A \square B$  и  $B \square C$  (говорят, что атрибут  $C$  транзитивно зависит от  $A$  через атрибут  $B$ ), при условии, что атрибут  $A$  функционально не зависит ни от атрибута  $B$ , ни от атрибута  $C$ .

*3НФ* устраняет в отношении транзитивные функциональные зависимости неключевых атрибутов от первичного ключа, которые выносятся в отдельное отношение вместе с копиями своих детерминантов. В *3НФ* устранено большинство аномалий от первичного ключа, но отношение в этой форме имеет аномалии в случае наличия более чем одного потенциального ключа.

*Декомпозиция* – формирование отношений БД путем разделения их на более мелкие, если эти отношения не выполняют правила необходимой нормальной формы.

## Рекомендации по выполнению работы

*Этап 1.* Выделить функциональные зависимости для каждого отношения исходной реляционной схемы. Проверить практический смысл выделенных функциональных зависимостей.

*Этап 2.* Для каждого отношения (включая и вновь создаваемые) последовательно применить правила нормальных форм. При несоблюдении текущего правила в отношении выполнить его декомпозицию (удалить проблемный атрибут из отношения с образованием нового отношения, первичным ключом которого будет детерминант рассматриваемой функциональной зависимости (этот атрибут только копируется в новое отношение)). Нормализованное отношение должно удовлетворять как минимум *3НФ*.

*Этап 3.* Для полученной нормализованной реляционной схемы проверить смысл ссылок.

*Этап 4.* Реализовать полученные реляционные отношения в виде таблиц в среде целевой СУБД.

*Этап 5.* Оформить отчет по работе.

### Пример приведения отношения к 3НФ

Рассмотрим отношение «Экзаменационная ведомость»

<u>Код студента</u>	Фамилия	<u>Код экзамена</u>	Предмет и дата	Оценка
1	Иванов	1	Математика, 05.06.2019	4
2	Петров	1	Математика, 05.06.2019	5
1	Иванов	2	Физика, 10.06.2019	5
2	Петров	2	Физика, 10.06.2019	5

Первичный ключ таблицы состоит из атрибутов: Код студента, Код экзамена

**Отношение находится в первой нормальной форме (1НФ), если все атрибуты отношения принимают простые значения (атомарные или неделимые), не являющиеся множеством или кортежем из более**

### элементарных составляющих.

Наше отношение не находится в 1НФ.

Приведем отношение к 1НФ:

<u>Код студента</u>	Фамилия	<u>Код экзамена</u>	Предмет	Дата	Оценка
1	Иванов	1	Математика	05.06.2019	4
2	Петров	1	Математика	05.06.2019	5
1	Иванов	2	Физика	10.06.2019	5
2	Петров	2	Физика	10.06.2019	5

Для исследования наличия 2НФ следует проанализировать функциональные зависимости между атрибутами отношения.

Единственный способ определить функциональные зависимости – внимательно проанализировать семантику (смысл) атрибутов.

Примеры функциональных зависимостей для отношения ЭКЗАМЕНАЦИОННАЯ ВЕДОМОСТЬ:

Код студента → Фамилия

Код студента, Код экзамена → Оценка

Код экзамена → Дата

Код экзамена → Предмет

**Отношение находится в 2НФ, если оно находится в 1НФ и каждый неключевой атрибут зависит от всего первичного ключа (не зависит от части ключа).**

**Отношение находится в 3НФ, если оно находится в 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа. Отношение находится в 3НФ в том и только том случае, если все**



**неключевые атрибуты отношения взаимно независимы и полностью зависят от первичного ключа.**

Продолжим рассмотрение примера с отношением ЭКЗАМЕНАЦИОННАЯ ВЕДОМОСТЬ. Для более краткой записи процесса нормализации введем следующие обозначения: КС – код студента, КЭ – код экзамена, Ф – фамилия, П – предмет, Д – дата, О – оценка.

Наше отношение примет вид:  $R=(КС, КЭ, Ф, П, Д, О)$

Выпишем функциональные зависимости:

$КС, КЭ \rightarrow Ф, П, Д, О$  (КС, КЭ - первичный ключ отношения, все неключевые атрибуты зависят от первичного ключа)

При этом некоторые атрибуты зависят не от всего ключа в целом:

$КЭ \rightarrow П$

$КЭ \rightarrow Д$  (предмет и дата зависят только от кода экзамена)

$КС \rightarrow Ф$  (фамилия студента зависит только от кода студента)

В соответствии с определением, **отношение находится во второй нормальной форме (2НФ), если оно находится в 1НФ и каждый неключевой атрибут зависит от первичного ключа и не зависит от части ключа.** Здесь атрибуты П, Д, Ф зависят от части ключа. Чтобы избавиться от этих зависимостей необходимо произвести декомпозицию отношения.

Выделим неполные зависимости в отдельные отношения. Если какие-то атрибуты зависят от одной части ключа, объединяем их в одну таблицу.

Получим отношение  $R1(КС, Ф)$  - это отношение находится в 2 НФ, так как ключ отношения простой и частичной зависимости быть не может. Так как в этом отношении нет транзитивных зависимостей, отношение  $R1(КС, Ф)$  находится в 3НФ.

Второе отношение  $R2(КЭ, П, Д)$  - зависимости неключевых атрибутов от части ключа нет, следовательно отношение находится в 2НФ. Транзитивных зависимостей в этом отношении также нет, следовательно отношение находится в 3НФ.

Исходное отношение приведено к виду:  $R(КС, КЭ, О)$ . Из него выведены неключевые атрибуты, зависящие от части ключа. Неключевой

атрибут O зависит от ключа КС, КЭ в целом, а не от его части. Значит, это отношение находится в 2НФ. Транзитивные зависимости отсутствуют, то есть отношение находится в 3НФ.

Таким образом все полученные отношения находятся в 3НФ.

Между таблицами установлены связи, как показано на рис. 1.

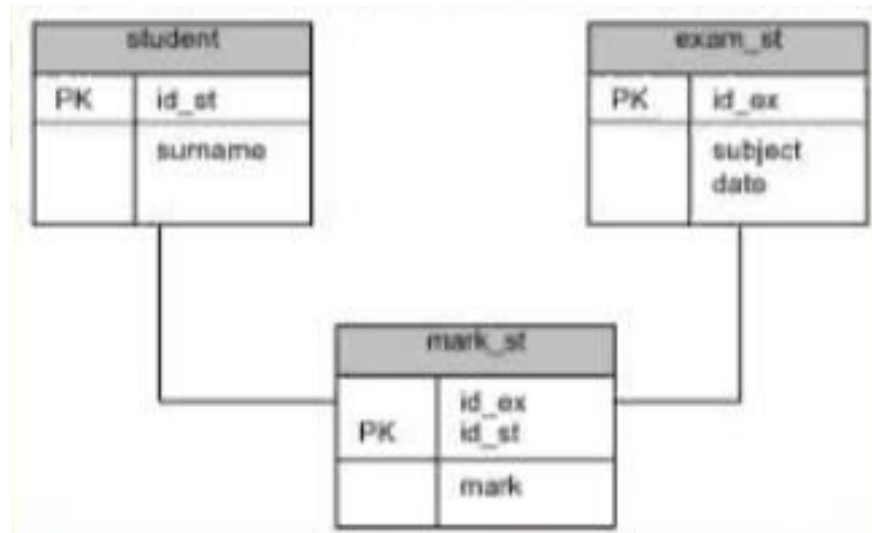


Рис. 2.1. Полученная реляционная модель (нормализованная)

Использованы следующие обозначения:

id\_st – код студента;

surname – фамилия;

id\_ex – код экзамена;

subject – предмет;

date – дата;

mark – оценка.

### Порядок выполнения работы

1. Для указанного варианта задания выполнить нормализацию отношений методом декомпозиции с учетом выделенных функциональных зависимостей.

2. Представить результат в виде *нормализованной реляционной*

модели.

3. Оформить отчет по работе.

### Варианты заданий

#### Вариант 1 – отношение «Морские перевозки»

Номер судна	Название	Номер рейса	Дата погрузки	Порт погрузки	Дата прибытия	Порт прибытия	Ф.И.О. капитана	Вид судна	Грузоподъемность, тонны
526	Japan Bear	9201W	5/31/22	SFO	6/6/92	NL	Емелин А.О.	Сухогруз	500
603	Korea Bear	9202W	5/05/22	ОАК	6/19/92	ОСА	Крылов О.Б.	Ролкер	1000
531	China Bear	9203W	6/20/22	LAX	7/10/92	РАР	Мушин Е.А.	Универсал	1500
526	Japan Bear	9204W	8/20/22	SFO	8/27/92	NL	Емелин А.О.	Сухогруз	500

*Вариант 2 – отношение «Контрагенты»*

Наименование контрагента	Город	Адрес	Вид контрагента	Должность контактного лица	Ф.И.О. контактного лица	Код города	Телефон
Поршневой завод	Владимир	ул. Кольцевая, 17	Поставщик	зам. дир.	Иванов И.И.	3254	76-15-95
Поршневой завод	Владимир	ул. Кольцевая, 17	Поставщик	нач. отд. сбыта	Петров П.П.	3254	76-15-35
ООО «Вымпел»	Курск	ул. Гоголя, 25	Клиент, Поставщик	директор	Сидоров С.С.	7634	66-65-38
ИП «Альфа»	Владимир	ул. Пушкинская, 37	Клиент, Поставщик	директор	Васильев В.В.	3254	74-57-45

*Вариант 3 – отношение «Отдел кадров»*

Код сотрудника	ФИО	Должность	Номер отдела	Наименование отдела	Квалификация
7513	Иванов И.И.	Программист	120	Отдел проектирования	C, Java
9842	Петров А.А.	Администратор БД	30	Финансовый отдел	MS SQL Server
6651	Сорокин А.П.	Программист	120	Отдел проектирования	VB, Java
9006	Ворнов Г.Р.	Системный администратор	120	Отдел проектирования	Windows, Linux

		ор			
--	--	----	--	--	--

*Вариант 4 – отношение «Ведомость расходов»*

Номер проекта	Наименование проекта	Номер работника	Ф.И.О. работника	Должность	руб/час	трудозатраты в часах	Общие расходы
15	Alpha Edit	101	Семен Иванов	Программист	200	120	24000
		102	Андрей Петров	Программист	200	100	20000
		110	Антон Сидоров*	Сист. аналитик	300	40	12000
18	Beta Base	103	Федот Антонов	Прграммист	200	250	50000
		102	Андрей Петров	Программист	200	280	56000
		111	Петр Семенов*	Проектировщик БД	250	80	20000
22	Delta CAD	104	Сидор Федотов	Программист	200	180	36000
		105	Иван Андреев	Программист	200	150	30000
		110	Антон Сидоров*	Системный аналитик	300	60	18000

*Вариант 5 – отношение «Железнодорожные перевозки»*

Номер поезда	Номер рейса	Дата отправления	Станция отправки	Дата прибытия	Станция прибытия	Ф.И.О. капитана	Вид поезда	Количество вагонов
276	821А	04.12.22	Москва	08.12.22	Дербент	Исаев А.О.	Скорый	10
653	822А	07.11.22	Тюмень	10.11.22	Махачкала	Магомедов О.Б.	Грузовой	12
781	823А	08.11.22	Сургут	11.11.22	Астрахань	Иванов Е.А.	Комбинированный	15
585	824А	14.09.22	Москва	18.09.22	Дербент	Исаев А.О.	Скорый	10

## **Практическая работа №3 «Создание проекта БД. Создание БД. Редактирование и модификация таблиц»**

**Цель работы:** освоить основные приемы заполнения и редактирования таблиц; познакомиться с простой сортировкой данных и с поиском записей по образцу; научиться сохранять и загружать базы данных.

### **Теоретические сведения**

#### **Microsoft Office Access**

Access является наиболее сложной программой из всех офисных приложений Microsoft Office. Чтобы начать работу с этой программой, вначале необходимо создать структуру базы данных, подробно ее описать, а затем создать различные формы.

ACCESS – это реляционная СУБД. Это означает, что с ее помощью можно работать одновременно с несколькими таблицами базы данных, эти таблицы между собой связаны. Таблицу ACCESS можно связать с данными, хранящимися на другом компьютере. Данные ACCESS очень просто комбинировать с данными EXCEL, WORD и другими программами Office.

Access во многом похож на Excel. Основное различие между таблицей БД и электронной таблицей – в системе адресации: в электронной таблице адресуется каждая ячейка, а в таблице БД – только поля текущей записи.

Таблицы – основные объекты базы данных (БД). В них хранятся данные. Реляционная база данных может иметь много взаимосвязанных таблиц. Сведения по разным вопросам следует хранить в разных таблицах.

Запрос – это средство, с помощью которого извлекается из базы данных информация, отвечающая определенным критериям. Результаты запроса представляют не все записи из таблицы, а только те, которые удовлетворяют запросу.


Формы – Обеспечивают более наглядную работу с таблицами, с помощью форм в базу вводят новые данные или просматривают имеющиеся.

Отчеты – средство представления данных таблиц. Отчеты могут быть оформлены надлежащим образом и распечатаны в том виде, в котором требуется пользователю.

Макросы – набор из одной или более макрокоманд, выполняющих определенные операции (открытие форм, печать отчетов)

Модули - это программы, написанные на языке программирования Visual Basic.

## Практическая часть

1. Вызвать программу Access 2007 (Access 2016).
2. В окне системы управления базы данных щелкнуть по значку «Новая база данных». Справа в появившемся окне дать имя новой базе данных «Анкета ИТ-23» и щелкнуть по значку папки, находящемуся справа от окна названия . Откроется окно сохранения, найдите свою папку и сохраните в нее новый файл базы данных «Анкета ИТ-23». Затем нажмите на кнопку «Создать».
3. Появится окно «Таблица» (Рисунок 3.1).

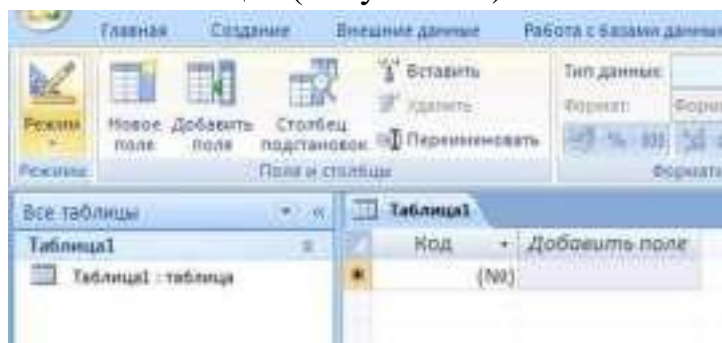





Рисунок 3.1. Окно пустой базы данных

4. В появившемся окне откройте меню команды <Режим> и выберите вариант <Конструктор>  и сохраните будущую таблицу под названием <Ведомость успеваемости>. Появится окно Конструктора.
5. Заполните поля в Конструкторе данными из *рисунка 3.2*. Тип данных можно выбрать из меню, появившемся при нажатии на кнопку  в ячейке справа.

Обратите внимание: ключевое поле «Счетчик» внесен в таблицу автоматически. Если напротив поля отсутствует значок ключа, то на панели инструментов щелкните по этому значку.



Имя поля	Тип данных
Код	Счетчик
Фамилия	Текстовый
Имя	Текстовый
Математика	Числовой
Менеджмент	Числовой
Сервисная деятельность	Числовой
Информационные технологии	Числовой
Стандартизация	Числовой
Гостиничная индустрия	Числовой
Пропуски по неуважительной причине	Числовой
Пропуски по уважительной причине	Числовой

Рисунок 3.2. Создание таблицы через конструктор

6. Перейдите в режим таблицы, щелкнув по кнопке **Режим** на панели инструментов. Введите данные в этом режиме, заполняя клетки таблицы. Значение поля **Код** будет меняться автоматически.

7. Заполните базу данных значениями из *таблицы 3.1*. Напротив каждой фамилии выставьте по всем дисциплинам оценки от 2 до 5

Таблица 3.1.

Код	Фамилия	Имя	Математика	Менеджмент	Сервисная деятельность	Информационные технологии	Стандартизация	Гостиничная индустрия	Пропуски по неуважительной причине	Пропуски по уважительной причине
1	Иваникова	Анна								
2	Баранова	Ирина								
3	Корнилова	Ольга								
4	Воробьев	Алексей								
5	Воробьев	Олег								
6	Скоркин	Алекс								
7	Володина	Нина								
8	Новоселов	Алексей								




9	Петрова	Елена								
10	Чернова	Кристина								
11	Терещинка	Инна								
12	Истратов	Максим								
13	Бондарь	Ольга								
14	Ревин	Олег								
15	Шарова	Оксана								

8. Выполните редактирование ячеек:

– Замените фамилию Иванникова на Иванова.

9. Отсортируйте:


а) *фамилии* – по алфавиту (поставьте маркер на любую фамилию в столбце

Фамилия и щелкните мышкой по кнопке  на панели инструментов или произведите сортировку с помощью контекстного меню)

б) *имя* – по алфавиту

10. Сохраните текущую таблицу, щелкнув по кнопке «крестик» в правом верхнем углу окна таблицы.

11. Откройте снова свою базу данных.

12. Выполните поиск записей по образцу: *найти студентку по фамилии Володина*. Для этого установите курсор в поле фамилия, щелкните на кнопке  <Бинокль> на панели инструментов меню Главная и в появившемся диалоговом окне введите в поле <Образец> фамилию *Володина* и щелкните по кнопке <Найти>.

Примечание: Если требуется найти следующую подобную запись, то щелкните мышкой по кнопке <Найти далее>. По окончании работы щелкните по кнопке <Отмена>.

13. Переименуйте поле «Математика» на «Информатика» с помощью контекстного меню. (Верните все как было назад).

14. Скройте столбец «Пропуски по неважной причине», потом отобразите его назад.

15. Войдите в режим *Конструктора* и назначьте полю Пр н/пр и Пр ув/пр. *маску ввода* 00 «часов». Заполните эти поля данными от 0 до 99.

16. Завершите работу с Access.

## Практическая работа №4

### «Создание ключевых полей. Задание индексов. Установление и удаление связей между таблицами»

**Цели работы:** научиться самостоятельно создавать ключевое поле; закрепить навыки по удалению, добавлению, заполнению и редактированию таблиц; научиться использовать фильтр в таблице.

### Теоретические сведения

#### Технология ввода и редактирования данных

Созданную таблицу открывают в окне База данных двойным щелчком левой кнопки мыши на её значке. Новая таблица не имеет записей - только названия столбцов, характеризующие структуру таблицы. Заполнение таблицы данными производится обычным порядком. Очередная запись вводится в конец таблицы.

**Ввод** в определённую ячейку таблицы (выделенную курсором) осуществляется путём набора информации на клавиатуре и последующим нажатием клавиши <Enter> или <Tab>. При окончании ввода данных в последнее поле записи MS ACCESS сам переходит на первое поле новой записи и ожидает ввода данных.

Редактировать данные в ячейке таблицы можно с полной или с частичной их заменой. Для полной замены данных необходимо подвести курсор к редактируемой ячейке, а затем набрать (ввести) новую информацию. При частичной замене данных можно использовать курсор мыши или функциональную клавишу <F2> с целью изменения положения курсора ввода.

Для удаления записи её необходимо выделить (щелкнуть по области маркера записи) и нажать клавишу <Del>, или выполнить команду меню ПРАВКА-УДАЛИТЬ. В выводимом на экран запросе подтвердить удаление.


### Практическая часть


1. Откройте учебную базу данных <Анкета ИТ-23>.
2. Создайте таблицу <Преподаватели > в *Режиме таблицы*. Для этого в меню Создание выберите кнопку Таблица. В появившейся таблице сделайте следующее:
  - Добавьте два поля – Поле 1 и Поле 2, выполнив команду через контекстное меню.

- Переименуйте <Поле 1> на <Предмет>. Для этого поставьте курсор в любую ячейку столбца <Поля 1> и выполните команду *Переименовать столбец* из контекстного меню. Или щелкните два раза по имени поля, удалите старое название и впечатайте новое.

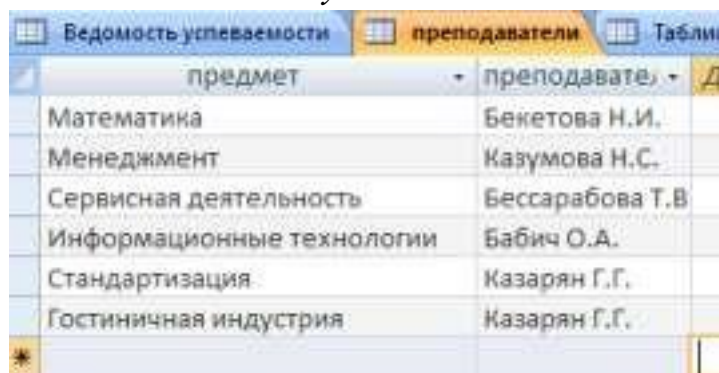
- Переименуйте аналогично <Поле 2> на <Преподаватель>.

3. Сохраните таблицу с именем <Преподаватели>, щелкнув по кнопке <Сохранить>

(дискетка  на панели инструментов).

4. Перейдите в режим <Конструктор> и удалите строку с ключевым словом Счетчик. Посмотрите как заданы поля. Сделайте поле <Предмет> ключевым, поместив курсор на имя этого поля и щелкнув по кнопке  - *Ключевое поле*. Тип данных поля задайте *текстовым*.

5. Перейдите в *Режим таблицы* и заполните таблицу <Преподаватели> записями из *Рисунок 4.1*.



предмет	преподаватель
Математика	Бекетова Н.И.
Менеджмент	Казумова Н.С.
Сервисная деятельность	Бессарабова Т.В.
Информационные технологии	Бабич О.А.
Стандартизация	Казарян Г.Г.
Гостиничная индустрия	Казарян Г.Г.

Рисунок 4.1. Таблица «Преподаватели»

6. Закройте таблицу <Преподаватели>, сохранив все изменения.

7. Используя <Шаблон таблиц>, создайте таблицу <Личные данные> студентов с ключевым полем. Для этого:

- Находясь на закладке <Создание> щелкните по кнопке <Шаблоны таблиц>, <Контакты>. Появится таблица уже с готовыми полями.


- Переименуйте предложенные поля на следующие поля: <Код студента>, <Фамилия>, <Имя>, <Город>, <Адрес>, <Телефон>, <Дата рождения>, <Фотография>, <Любимый предмет>, лишние поля удалите.

- Сохраните полученную таблицу под названием <Личные данные>. Ключевое поле задано автоматически.

8. Внесите данные в новую таблицу, заполнив поля <Фамилия>, <Имя>, <Город>, <Адрес>, <Телефон>, <Дата рождения>.

ПРИМЕЧАНИЕ. Поля <Фамилия> и <Имя> можно скопировать из таблицы <Ведомость успеваемости>. В поле <Город> внесите четыре разных города (например, Новороссийск, Геленджик, Анапа, Крымск)


9. Перейдите в режим <Конструктор> и назначьте типы данных: для поля <Телефон> - *числовой*, для поля <Дата рождения> - *дата/время*, для поля <Фотография> – *поле объекта OLE*, для остальных – *текстовый*.

Для поля <Любимый предмет> выполните свойство выбор предмета из списка с помощью *Мастера подстановок*. Для этого в строке <Любимый предмет> в поле *Тип данных – текстовый* щелкните по кнопке  и в ниспадающем меню выберите команду <Мастер подстановок>.


- В диалоговом окне <Создание подстановки> поставьте флажок напротив способа <Будет введен фиксированный набор значений> и нажмите <Далее>.

- В следующем окне внесите в столбец все предметы (предметы из таблицы <Преподаватели>), нажмите <Далее>.

- В последнем окне, не изменяя имени столбца нажмите <Готово>.

10. Перейдите в режим таблицы и выберите для каждого студента с помощью кнопки  из списка любимый предмет.

11. Создайте *схему данных*, т.е. установите связи между таблицами.

- Щелкните по кнопке  - *Схема данных* на панели инструментов меню <Работа с базами данных>. В окне <Отобразить таблицу> выделите таблицу <Ведомость успеваемости> и щелкните по кнопке <Добавить>. Также добавьте таблицы <Преподаватели> и <Личные данные>. В окне <Схема данных> появится условный вид этих таблиц. Закройте окно <Добавление таблицы>.



- Поставьте мышку на имя поля <Предметы> в таблице <Преподаватели>, и не отпуская кнопку мыши перетащите его на поле <Любимый предмет> таблицы <Личные данные>. Отпустите мышку. Появится диалоговое окно <Связи>, в котором включите значки «*Обеспечение целостности данных*», «*Каскадное обновление связанных полей*» и «*Каскадное удаление связанных полей*». Щелкните по кнопке <Создать>. Появится связь «один-ко-многим».

- Поставьте мышку на имя поля <Код студента> в таблице <Личные данные> и перетащите его, не отпуская мышки, на поле <Код> таблицы

<Ведомость успеваемости>. В появившемся окне <Связи> включите значок «Обеспечение целостности данных» и щелкните по кнопке <Создать>. Появится связь «один-к одному».

- Закройте схему данных, сохранив ее.

12. Произведите фильтрацию данных в таблице <Личные данные> по выделенному.

- Откройте таблицу в режиме таблицы.
- Выберите студентов, проживающих в Новороссийске. Для этого поставьте курсор в одну из первых записей, где есть город Новороссийск и щелкните по кнопке  - *Фильтр по выделенному* на панели инструментов. Выберите команду <Равно «Новороссийск» >. Access отобразит все записи, удовлетворяющие критерию фильтрации.
- Для отображения всех записей выполните команду <Удалить фильтр> для этого щелкните по соответствующей кнопке на панели инструментов .

13. Закончите работу с базой данных Access.

## Практическая работа №5

### «Запросы»

**Цели работы:** закрепить навыки по редактированию таблиц; познакомиться с основными видами запросов; научиться создавать запросы на выборку различными способами; научиться создавать сложные запросы; научиться создавать перекрестные запросы.

**Запрос** – это средство, с помощью которого извлекается из базы данных информация, отвечающая определенным критериям. Результаты запроса представляют не все записи из таблицы, а только те, которые удовлетворяют запросу.

Запросы состоят из ряда условий, каждое условие состоит из трех элементов:

1. поле, которое используется для сравнения;
2. оператор, описывающий тип сравнения;
3. величина, с которой должно сравниваться значение поля.


Таблица 5.1 - Выражения и операторы, применяемые в условиях отбора.

Выражения и операторы	Описание выражений и операторов
<b>Числа</b>	<b>Вводятся без ограничений</b>
Текст	Должен быть заключен в кавычки
Даты	Ограничиваются с двух сторон символами # (например, #01.02.02#)
*; +; -; /; ^	Арифметические операторы, связывающие выражения
<; <=; >; >=; =; <>	Операторы сравнения
And (И); Not (Нет); Or (Или)	Логические операторы
Like	Используется для логики замены в выражениях
In	Для определения, содержится ли элемент данных в списке значений
Between... And...	Для выбора значений из определенного интервала

?	Заменяет один символ (букву или цифру)
*	Заменяет несколько символов

Запросы могут быть простые, сложные перекрестные.

### Задание для практической работы Создайте запросы к вашей базе данных

- 1) Откройте свою учебную базу данных.
- 2) Создайте запрос на выборку студентов, у которых по всем предметам только хорошие оценки с помощью *Мастера запросов*.
  - На панели инструментов выберите команду <Мастер запросов>.
  - В появившемся диалоговом окне выберите <Простой запрос> и щелкните по кнопке <ОК>.
  - В следующем окне выберите таблицу, по которой строится запрос (<Ведомость успеваемости>), и те поля, которые участвуют в запросе. Перенесите их в правую часть окна с помощью кнопки , нажмите <Далее>. В следующем окне тоже нажмите <Далее>.
  - В другом окне дайте название запроса «Хорошисты» и нажмите <Готово>.
  - Появится таблица <Хорошисты>, в которой отражены фамилии всех студентов и изучаемые предметы.
  - Откройте таблицу «Хорошисты», перейдите в режим <Конструктор>. Здесь в поле <Условия отбора> под каждым предметом поставьте условие  $\geq 4$  или  $4OR5$ .

Примечание: Галочки в каждом поле означают, что по вашему выбору можно включить или убрать любое поле на выборку.

- Перейдите в режим таблицы, ответив <Да> на вопрос о сохранении запроса. (В таблице должны остаться фамилии «хорошистов»).

3) С помощью <Конструктора запросов> создайте запрос на выборку по таблице <Личные данные>.

- Щелкните по таблице <Личные данные>, зайдите в меню <Создание>, выберите команду <Конструктор запросов >.
- Добавьте нужную таблицу в поле запроса. Выделите её в списке и щелкните по кнопке <Добавить>. Закройте окно <Добавление таблицы>.

- Выберите студентов, чьи фамилии начинаются на букву «В» и которые проживают в Анапе. Для этого:
- добавьте в строку <Поле> два поля <Фамилия> и <Город>;
- в строке <Условия отбора> в первом столбце укажите значение Like «В \*», а во втором столбце с названием <Город> - «Анапа»;
- закройте запрос, сохранив его под названием — «ВВВ» (у вас должны остаться в списке студенты, проживающие в Анапе). Рисунок 5.1.

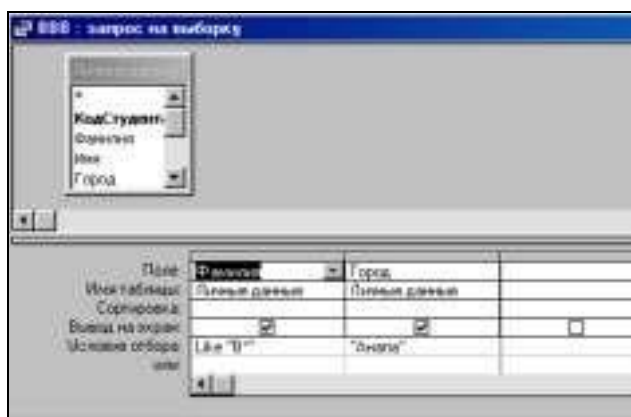


Рисунок 5.1. Запрос на выборку

### Индивидуальные задания к практической работе

а) Составьте запрос с названием <Запрос 1> на базе таблицы <Ведомость успеваемости>, в котором будут указаны студенты, имеющие по первым двум предметам оценки не менее «4».

(Выполните запрос или через *Конструктор запросов*, или через *Мастер запросов*)

б) Составьте <Запрос 2> на базе таблицы <Ведомость успеваемости>, в котором будут указаны студенты, имеющие не более 30 часов пропусков по неуважительной причине. Добавьте в этот запрос поле пропуски по уважительной причине в интервале от 30 часов до 45 часов (используйте оператор *Between... And...*)

в) Составьте <Запрос> на базе таблицы <Личные данные>. Выведите список студентов, которым на данный момент, т.е. на сегодняшнее число, исполнилось уже 17 лет (используйте оператор *Between... And...*)

Примечание: Дата записывается с использованием символа #, например, #01.02.02.#



4) Составьте запрос на базе трех таблиц <Ведомость успеваемости>, <Личные данные> и <Преподаватель>. Выберите студентов, которые проживают в Новороссийске и у которых любимый предмет «Менеджмент». Озаглавьте <Запрос 4>. Используйте <Конструктор запросов>.

- В меню <Создание> выберите <Конструктор запросов>.
- Добавьте все три таблицы в поле запроса. Закройте окно <Добавление таблицы>.
- В первый столбец в строку <Поле> перетащите из первой таблицы с помощью мышки <Фамилия>, из второй таблицы во второй столбец <Город> и из третьей таблицы в третий столбец строки <Поле> - <Предмет> (Рисунок 5.2).

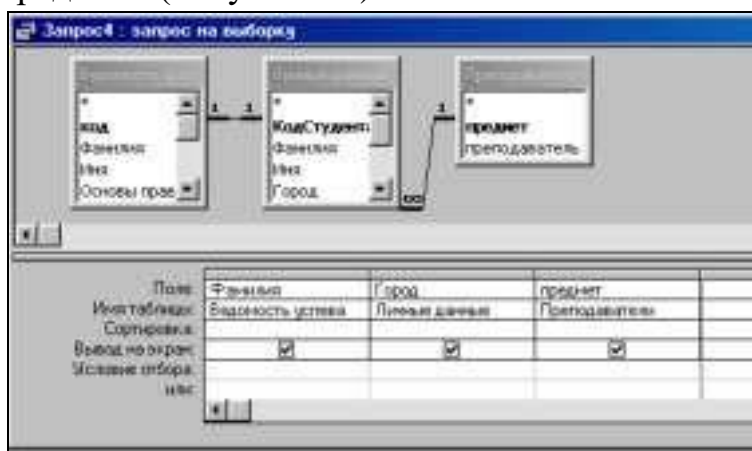


Рисунок 5.2. Запрос на выборку

- В поле <Условия отбора> в столбце <Город> введите город «Новороссийск», в столбец <Предмет> введите «Менеджмент».
- Сохраните запрос под именем <Запрос 4>.
- Откройте запрос и проверьте результат проделанной работы.

5) Выполните запрос на создание новой таблицы, в которой должны быть поля <Фамилия>, <Имя>, <Пропуски по неуважительной причине>, <Город> и <Предмет>.

- В меню <Создание> выберите <Конструктор запросов>.
- Добавьте все три таблицы из списка окна <Добавление таблицы>. Закройте это окно.
- В первую строку <Поле> из первой таблицы перенесите в первый столбец поля <Фамилия>, во второй <Имя> и в третий <Пропуски по

уважительной причине>, в четвертый столбец перетащите поле <Город> из второй таблицы и в последнем столбце будет поле <Предмет> из третьей таблицы.

- Закройте запрос, сохранив его с именем <Запрос 5>.

б) Создайте *перекрестный запрос*.

Допустим, нужно посчитать для ведомости, сколько в группе человек получили по предмету троек, четверок и пятерок. Для этих целей используется *перекрестный запрос*.

- В меню <Создание> выберите <Мастер запросов>.
- В диалоговом окне выберите <Перекрестный запрос>, щелкните по кнопке <ОК>.
- В окне <Создание перекрестных запросов> выделите таблицу <Ведомость успеваемости> и щелкните <Далее>.
- Выберите поля, значения которого будут использоваться в качестве заголовков строк – это <Фамилия> и <Имя>. Щелкните по кнопке <Далее>.
- Выберите поле, значение которого будут использоваться в качестве заголовков столбцов, например, <Менеджмент>. Щелкните по кнопке <Далее>.
- Выберите функцию, по которой будут вычисляться значения ячеек на пересечении столбцов и строк (в данном случае Count – количество). Щелкните по кнопке <Далее>.
- Задайте имя запроса <Итог по менеджменту> и щелкните по кнопке <Готово>. Составьте аналогичные запросы для оценок по трем другим предметам.

7) Предъявите преподавателю все запросы своей базы данных на экране дисплея.

8) Завершите работу с Access.

## Практическая работа №6 и №7

### «Формы. Отчеты»

**Цели работы:** Научиться создавать формы ввода-вывода; научиться создавать кнопочные формы и отчеты.

**Форма** – это средство, упрощающее ввод, редактирование и отображение информации, хранящейся в таблицах базы данных. Она представляет собой окно с набором элементов управления.

Форма сама по себе не хранит информацию, она просто обеспечивает удобный способ доступа к информации, хранящейся в одной или нескольких таблицах. Формы по сравнению с обработкой данных в режиме таблицы обладают следующими преимуществами:

- Форма позволяет в каждый момент сфокусировать внимание на отдельной записи;
- Элементы управления на форме можно расположить логичным образом, облегчающим чтение и работу с данными;
- Отдельные элементы управления обладают возможностями облегчить ввод и изменение отдельных данных;
- Некоторые объекты баз данных, такие как рисунки, анимации, звуки и видеоклипы, могут отображаться только в режиме формы, но не в режиме таблицы. Создание кнопочной формы.




Кнопочное меню представляет собой форму, на которой расположены элементы управления – кнопки с поясняющими надписями. Щелчок на кнопке открывает соответствующую таблицу, запрос, форму или отчет. Меню - удобный инструмент работы с базами данных, и он практически всегда присутствует в базах созданных для предприятий или фирм.

Кнопочное меню создают с помощью Диспетчера кнопочных форм.

Отчет – это гибкое и эффективное средство для организации просмотра и распечатки итоговой информации. В отчете можно получить результаты сложных расчетов, статистических сравнений, а также поместить в него рисунки и диаграммы. Пользователь имеет возможность разработать отчет самостоятельно (в режиме *Конструктора*) или создать отчет с помощью *Мастера*, т.е. полуавтоматически.

## Задание для практической работы

### Задание 1. Создайте формы к базе данных

- 1) Откройте свою базу данных.
- 2) Создайте форму с помощью <Мастера форм> на базе таблицы <Ведомость успеваемости>.
  - Откройте таблицу <Ведомость успеваемости>.
  - Выберите закладку <Формы >, щелкните мышкой по кнопке <Другие формы>. 
  - В появившемся диалоговом окне выберите <Мастер форм>.
  - В поле <Таблицы/Запросы> выберите таблицу <Ведомость успеваемости>, в поле <Доступные поля> выберите поля <Фамилия>, <Имя> и перенесите их стрелкой в поле <Выбранные поля>. Также перенесите поля с названием предметов, щелкните по кнопке <Далее>.
  - Выберите внешний вид формы – Табличный, щелкните по кнопке <Далее>.
  - Выберите требуемый стиль (н-р, Обычная), щелкните по кнопке <Далее>.
  - Задайте имя формы <Успеваемость> и щелкните по кнопке <Готово>. В результате получите форму, в которой можно менять данные и вводить новые значения. □ Закройте форму.
- 4) Создайте форму на основе таблицы <Преподаватели>. Откройте таблицу <Преподаватели>.
  - Выберите закладку <Формы >, щелкните мышкой по кнопке <Другие формы>. 
  - В появившемся диалоговом окне выберите <Мастер форм> .
  - Выберите внешний вид формы - <ленточный>.
  - Выберите любой стиль.
  - Получите готовую форму. Сохраните ее под именем <Преподаватели>.
  - Закройте форму.
  - Создайте форму <Личные данные> с помощью инструмента <Пустая форма>
  - На вкладке Создание в группе Формы щелкните Пустая форма. 

- Access открывает пустую форму в режиме макета и отображает область Список полей.

- В области Список полей щелкните знак плюс (+) рядом с таблицей или таблицами, содержащими поля, которые нужно включить в форму.

- Чтобы добавить поле к форме, дважды щелкните его или перетащите его на форму. Чтобы добавить сразу несколько полей, щелкните их последовательно, удерживая нажатой клавишу CTRL. Затем перетащите выбранные поля на форму.

- Закройте окно списка полей.

- Перейдите в режим Конструктора

**Примечание 1.** Размер окошка для названия поля и для его значений меняются мышкой.

Для этого выделите черный квадратик рамки (рамка станет цветной), установите курсор на границу рамки и с помощью двунаправленной стрелки измените размеры рамки.

**Примечание 2.** С помощью кнопок панели инструментов Шрифт меняйте соответственно цвет фона, текста, линии/границы и т.д.

- Расположите элементы удобно по полю.

- Задайте размер текста поля <Фамилия> равным 24 пт, шрифт - синего цвета.

- Увеличьте в высоту рамку поля <Фотография>.

- Сохраните форму с именем <Данные студентов>.

- Посмотрите все способы представления форм: в режиме Конструктора, режиме Макета и режиме Форм.


- Закройте форму.

4) Добавьте в таблицу <Личные данные> логическое поле <Институт> (т.е., собирается ли в дальнейшем учащийся поступать в институт). Значение этого поля <ДА> или <НЕТ>.

- Откройте таблицу <Личные данные> в режиме Конструктор. Добавьте поле с именем <Институт> и типом Логический.

- Закройте таблицу.

- Перейдите на закладку Формы и откройте форму <Данные студентов> в режиме Конструктор

- Щелкните по кнопке <Список полей> на панели инструментов, выделите название <Институт> и перетащите его мышкой в область данных, появится значок  и надпись <Институт>.

- Расположите новые элементы по правилам оформления формы (с помощью мыши).

- Закройте <Список полей>

**Примечание 3.** Если флажок установлен, поле в таблице имеет значение <ДА>, если снят, то <НЕТ>.

- Перейдите в режим <Раздельная форма> и посмотрите записи. Установите флажки у восьми разных учащихся.

- Закройте форму, ответив утвердительно на вопрос о сохранении.

5) Создайте кнопочную форму <Заставка> с помощью Конструктора.  Щелкните по кнопке <Создать>.

- Выберите <Конструктор>. Появится пустая форма. Задайте мышкой ширину формы, равную 10см, а высоту – 7см.

- Сохраните работу с именем <Заставка>.

- Откройте созданную форму <Заставка> в режиме Конструктора.


- Выберите на панели инструментов <Элементы управления> кнопку Aa – <Надпись>. Курсор мышки примет вид крестика с «приклеенной» буквой А. Щелкните мышкой по месту начала надписи и введите:

*База данных*

*«Гостиница»*

(после слов База данных нажмите одновременно комбинацию клавиш Shift+Enter.)

- Нажмите клавишу <Enter>. Выберите размер букв 18, а выравнивание - по центру. Цвет фона – голубой. Растяните мышкой надпись на ширину окна.

- Выберите на панели элементов значок  - Кнопка. Щелкните мышкой по тому месту области данных, где должна быть кнопка. Появится диалоговое окно <Создание кнопок>.

- Выберите категорию <Работа с формой>, а действие <Открыть форму>, и щелкните по кнопке <Далее>.

- Выберите форму <Успеваемость>, открываемую этой кнопкой щелкните по кнопке <Далее>. В следующем окне также щелкните по кнопке <Далее>.
- В следующем окне поставьте переключатель в положение <Текст>, наберите в поле слово <Успеваемость> (Рисунок 6.1) и щелкните по кнопке <Далее>.

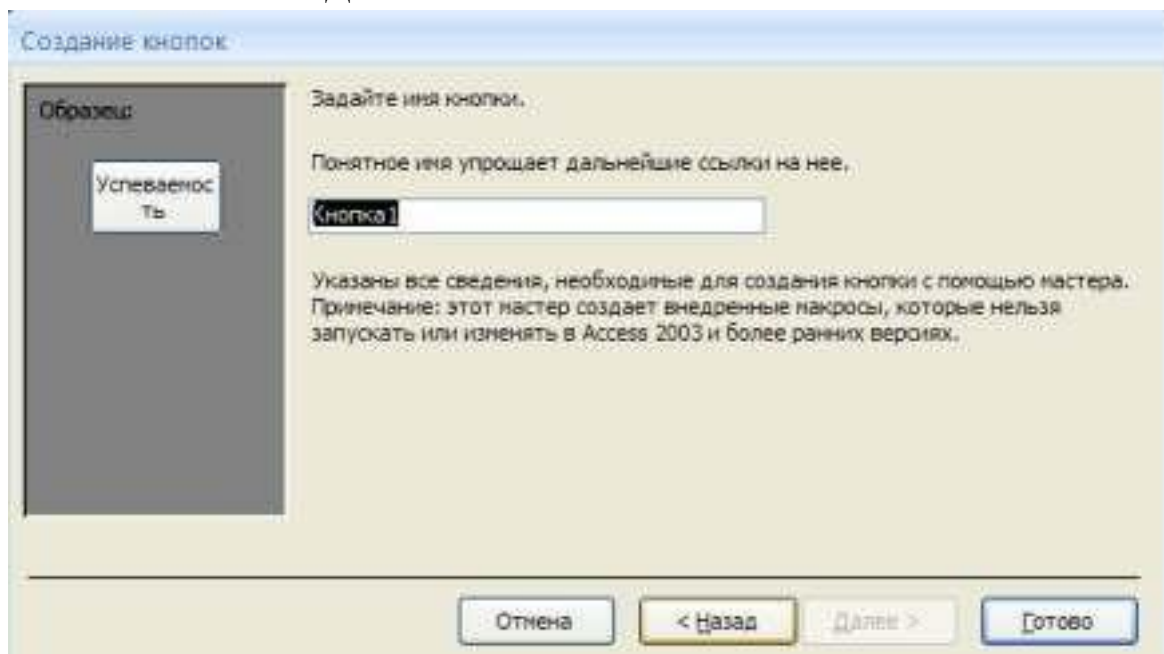


Рисунок 6.1. Создание кнопок

- Задайте имя кнопки <Успеваемость> и щелкните по кнопке <Готово>.

**Примечание 4.** Размер и расположение кнопок можно менять мышкой в режиме Конструктор.

Самостоятельно создайте кнопки для форм <Личные данные> и <Преподаватели>.

- Перейдите в режим формы (Рисунок 6.2). Теперь при щелчке мышью по соответствующим кнопкам будут открываться соответствующие формы для работы.
- Закройте форму.



Рисунок 6.2. Окно формы

б) Создайте кнопочную форму при помощи Диспетчера кнопочных форм.

- Откройте вкладку Работа с базами данных, команда - Диспетчер кнопочных форм. Вы получите диалоговое окно, представленное на Рисунке 6.3.

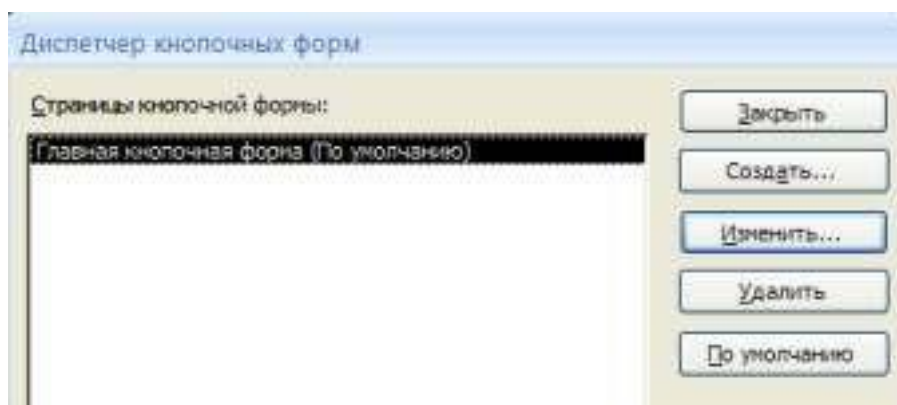


Рисунок 6.3. Диспетчер кнопок

- Щелкните в этом окне по кнопке <Изменить>.
- В следующем окне щелкните по кнопке <Создать> и в появившемся окне измените содержимое полей в соответствии с Рисунком 6.4 (Команду и Форму выбирайте из списка, а не набирайте вручную). Щелкните по кнопке <ОК>.



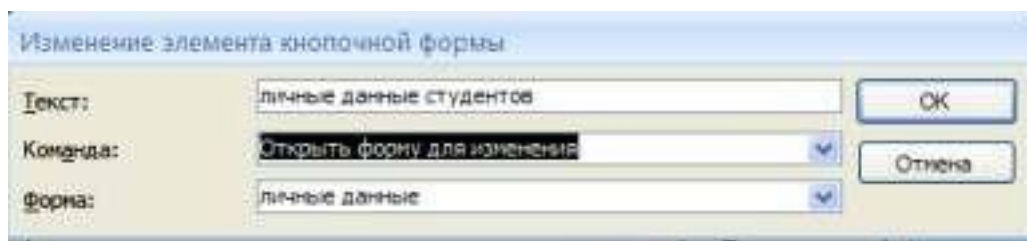


Рисунок 6.4. Изменение элементов кнопочной формы

- Аналогично создайте еще три элемента кнопочной формы: <Успеваемость>, <Преподаватели > и <Заставка>.
- Добавьте кнопку закрытия базы данных. Для этого щелкните по кнопке <Создать>, наберите в поле Текст слово <Выход>, а в поле Команда выберите <Выйти из приложения>. Закройте диалоговые окна.
- Откройте окно <Кнопочная форма> в режиме Конструктора или Макета, измените цвет надписи и название вашей базы данных на ГОСТИНИЦА, сохраните форму.
- Украсьте вашу форму рисунком. Для этого щелкните по значку Эмблема и выберите в открывшемся окне папку с рисунками, выберите понравившийся и вставьте в свою кнопочную форму.
- Перейдите в режим формы, проверьте работу всех кнопок кнопочной формы.

Завершите работу с базой данных, нажав на кнопку <Выход>.

## Задание 2. Создайте отчет с помощью *Мастера отчетов*.

- Откройте вкладку *Создание*, меню *Отчеты*.
- Выберите *Мастер отчетов* и таблицу «Личные данные».
- Выберите нужные поля, которые будут участвовать в отчете, нажмите кнопку «Далее».
- В новом окне выберите поля для группировки так, чтобы сначала было указано поле «Фамилия», нажмите кнопку «Далее».
- На этом шаге отсортируйте данные по алфавиту, нажмите кнопку «Далее».
- Выберите вид макета *Ступенчатый* и щелкните по кнопке «Далее».
- Выберите стиль отчета: *Открытая* и щелкните по кнопке «Далее».

- Задайте имя отчета: «Отчет1» и щелкните по кнопке «Готово». Вы попадете в режим просмотра отчета.

- Закройте отчет согласившись с сохранением.

Самостоятельно. Составьте еще два отчета по запросам – «Запрос 3» и «Запрос 5», выбирая из разных макетов: *блок*; *структура*, выбирая из разных стилей. Сохраните отчеты под именами «Отчет 2» и «Отчет 3».

**Задание 3. Создайте Пустой отчет** в столбец на базе таблицы «Ведомость успеваемости» и сохраните его с именем «Успеваемость».

С помощью Конструктора измените цвет букв заголовка, их размер и шрифт.

**Задание 4. Создайте почтовые наклейки.**

- Откройте вкладку *Создание*, меню *Отчеты*.

- Выберите таблицу «Личные данные», команда Наклейки.

- В следующем окне щелкните по кнопке «Далее».

- В следующем окне выберите шрифт, размер шрифта, насыщенность и цвет, вновь щелкните по кнопке «Далее».

- В следующем окне создайте прототип наклейки, напечатав слово ЛИЧНОСТЬ и выбрав соответствующие поля, щелкните по кнопке «Далее».

- В следующем окне укажите поля для сортировки (Фамилия, Имя), щелкните по кнопке «Далее».

- Введите имя отчета «Наклейки» и щелкните по кнопке «Готово».

- Просмотрите Наклейки (Рисунок 6.5).

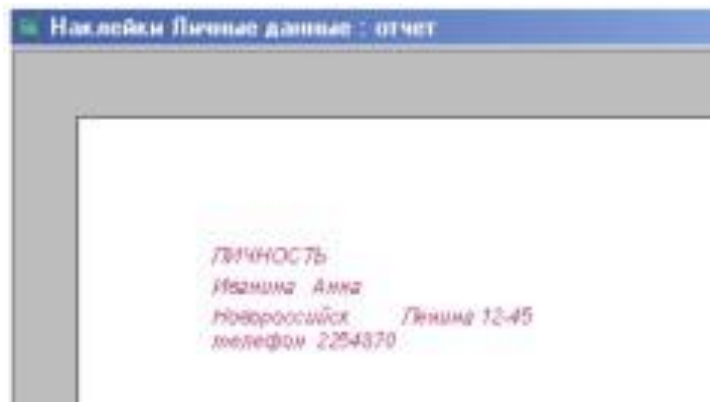


Рисунок 6.5

## Самостоятельная практическая работа №8

**По заданному описанию предметной области построить концептуальную, логическую и физическую модели базы данных**

- Выделите типы сущностей;
- Выделите атрибуты;
- Выделите типы связей между сущностями;
- Определите для отношений показатели кардинальности и степени;
- Определите первичные ключи сущностей;
- Нарисуйте ER-диаграмму предметной области;
- Проанализируйте информационные задачи и группы пользователей;
- Разработайте базу данных средствами MS Access;
- Разработайте формы, запросы и отчеты базы данных.

### Индивидуальные задания к практической работе

#### Вариант 1.

Задача – организация учебного процесса в вузе:

- Студенты: паспортные данные, адрес, дата зачисления, номер приказа, факультет, группа, является ли старостой, кафедра (специализация), изучаемые (изученные) предметы, оценки, задолженности, стипендия.
- Учебные курсы: название, факультет(ы), групп(ы), кафедра, семестр(ы), форма отчётности, число часов.
- Преподаватели: паспортные данные, адрес, телефон, фотография, кафедра, должность, учёная степень, начальник (зав. кафедрой), предмет(ы), число ставок, зарплата.

#### Вариант 2.

Учет и выдача книг в библиотеке вуза:

- Книги: авторы, название, раздел УДК, раздел (техническая, общественно-политическая и т.п.), место и год издания, издательство, количество страниц, иллюстрированность, цена, дата покупки, номер сопроводительного документа (чек, счёт/накладная), вид издания (книги, учебники, брошюры, периодические издания), инвентарный номер (есть только для книг и некоторых учебников), длительность использования читателями (год, две недели, день), электронная версия книги или ее реферата (отсканированный текст).

- Читатели: номер читательского билета, ФИО, год рождения, адрес, дата записи, вид (студент, аспирант, преподаватель, сотрудник), курс, номер группы, названия взятых книг и даты их выдачи.

### **Вариант 3.**

Отдел кадров некоторой компании.

- Сотрудники: ФИО, паспортные данные, фотография, дом. и моб. телефоны, отдел, комната, раб. телефоны (в т.ч. местный), подчинённые сотрудники, должность, тип(ы) работы, задание(я), проект(ы), размер зарплаты, форма зарплаты (почасовая, фиксированная).

- Отделы: название, комната, телефон(ы), начальник, размер финансирования, число сотрудников.

- Проекты: название, дата начала, дата окончания, размер финансирования, тип финансирования (периодический, разовый), задачи и их исполнители, структура затрат и статьи расходов.

### **Вариант 4.**

Отдел поставок некоторого предприятия.

- Поставщики: название компании, ФИО контактного лица, расчётный счёт в банке, телефон, факс, поставляемое оборудование (материалы), даты поставок (по договорам и реальные), метод и стоимость доставки.

- Сырьё: тип, марка, минимальный запас на складе, время задержки, цена, продукты, при производстве которых используется, потребляемые объёмы (необходимый, реальный, на единицу продукции).

### **Вариант 5.**

Пункт проката видеозаписей (внутренний учёт).

- Видеокассеты: идентификационный номер видеокассеты, тип видеокассет, дата его создания, компания-поставщик, число штук данного типа (общее, в магазине, выдано в настоящее время, выдано всего, выдано в среднем за месяц), общая длительность записей; записи видеокассет: название, длительность, категория, год выпуска и производитель (оригинала).

- Клиенты: ФИО, паспортные данные, адрес, телефон; заказы, т.е. взятые видеокассеты (сейчас и в прошлом): номер, дата выдачи, дата возвращения, общая стоимость заказа.

### **Вариант 6.**

Пункт проката видеозаписей (информация для клиентов).

- Видеокассеты: краткое описание, внешний вид (этикетка), марка (пустой видеокассеты, цена за единицу прокатного времени (например: 1 день, 3 дня, неделя), есть ли в наличии, общая длительность записей; записи на видеокассете: название, длительность, жанр (категория), тема, год и страна выпуска (оригинала), кинокомпания, описание, актеры, режиссер.

- Заказы: идентификационные номера и названия выданных видеокассет, дата выдачи, дата возвращения (продления), общая стоимость заказа, возвращены ли кассеты заказа.

### **Вариант 7.**

Кинотеатры (информация для зрителей).

- Фильмы: название, описание, жанр (категория), длительность, популярность (рейтинг, число проданных билетов в России и в мире), показывается ли сейчас (сегодня, на текущей неделе), в каких кинотеатрах показывается, цены на билеты (в т.ч. средние).

- Кинотеатры: название, адрес, схема проезда, описание, число мест (в разных залах, если их несколько), акустическая система, широкоэкранность, фильмы и цены на них: детские и взрослые билеты в зависимости от сеанса (дневной, вечерний и т.п.) и от категории мест (передние, задние и т.п.); сеансы показа фильмов (дата и время начала).

### **Вариант 8.**

Ресторан (информация для посетителей).

- Меню: дневное или вечернее, список блюд по категориям.
- Блюда: цена, название, вид кухни, категории (первое, второе и т.п.; мясное, рыбное, салат и т.п.), является ли вегетарианским, компоненты блюда, время приготовления, есть ли в наличии.
- Компоненты блюд: тип (гарнир, соус, мясо и т.п.), калорийность, цена, рецепт, время приготовления, есть ли в наличии, ингредиенты (продукты) и их расходы на порцию.

### **Вариант 9.**

Задача - информационная поддержка деятельности склада.

База данных должна содержать информацию о наименовании товара, его поставщике, количестве, цене товара, конечном сроке реализации, сроке хранения на складе. Торговый склад производит уценку хранящейся продукции. Если продукция хранится на складе дольше 10 месяцев, то она уценивается в 2 раза, а если срок хранения превысил 6 месяцев, но не достиг 10, то в 1,5 раза. Ведомость уценки товаров должна содержать информацию: наименование товара, количество товара(шт.), цена товара до уценки, срок хранения товара, цена товара после уценки, общая стоимость товаров после уценки.

### **Вариант10.**

Задача – информационная поддержка деятельности адвокатской конторы. БД должна осуществлять:

- ведение списка адвокатов;
- ведение списка клиентов;
- ведение архива законченных дел.

- Необходимо предусмотреть:
- получение списка текущих клиентов для конкретного адвоката;
- определение эффективности защиты (максимальный срок минус полученный срок) с учётом оправданий, условных сроков и штрафов;
- определение неэффективности защиты (полученный срок минус минимальный срок);
- подсчёт суммы гонораров (по отдельным делам) в текущем году;
- получение для конкретного адвоката списка текущих клиентов, которых он защищал ранее (из архива, с указанием полученных сроков и статей).

### **Вариант11.**

Задача – информационная поддержка деятельности гостиницы.

БД должна осуществлять:

- ведение списка постояльцев;
- учёт забронированных мест;
- ведение архива выбывших постояльцев за последний год.
- Необходимо предусмотреть:
- получение списка свободных номеров (по количеству мест и классу);
- получение списка номеров (мест), освобождающихся сегодня и завтра;
- выдачу информации по конкретному номеру;
- автоматизацию выдачи счетов на оплату номера и услуг;
- получение списка забронированных номеров;
- проверку наличия брони по имени клиента и/или названию организации

### **Вариант12.**

Описание предметной области:

- В компании несколько отделов.
- В каждом отделе есть некоторое количество сотрудников, занятых в нескольких проектах и размещающихся в нескольких офисах.
- Каждый сотрудник имеет план работы, т.е. несколько заданий, которые он должен выполнить. Для каждого такого задания существует ведомость, содержащая перечень денежных сумм, полученных сотрудником за выполнение этого задания.
- В каждом офисе установлено несколько телефонов.
- В базе данных должна храниться следующая информация.
- Для каждого отдела: номер отдела (уникальный), его бюджет и личный номер сотрудника, возглавляющего отдел (уникальный).
- Для каждого сотрудника: личный номер сотрудника (уникальный), номер текущего проекта, номер офиса, номер телефона, название выполняемого задания вместе с датой и размером выплат, проведенных в качестве оплаты за выполнение данного задания.

- Для каждого проекта: номер проекта ( уникальный) и его бюджет.
- Для каждого офиса: номер офиса ( уникальный), площадь в квадратных футах, номера всех установленных в нем телефонов.

### **Вариант13.**

Задача – информационная поддержка деятельности спортивного клуба. БД должна осуществлять:

- ведение списков спортсменов и тренеров;
- учёт проводимых соревнований (с ведением их архива);
- учёт травм, полученных спортсменами.

Необходимо предусмотреть:

- возможность перехода спортсмена от одного тренера к другому;
- составление рейтингов спортсменов;
- составление рейтингов тренеров;
- выдачу информации по соревнованиям;
- выдачу информации по конкретному спортсмену;
- подбор возможных кандидатур на участие в соревнованиях (соответствующего уровня мастерства, возраста и без травм).

### **Вариант14.**

Задача – информационная поддержка деятельности аптечного склада.

В аптечном складе хранятся лекарства. Сведения о лекарствах содержатся в специальной ведомости: наименование лекарственного препарата; количество (в шт.); цена; срок хранения на складе (в месяцах). Лекарства поступают на склад ежедневно от разных поставщиков, отпускаются два раза в неделю по предварительным заказам аптек. Выяснить, сколько стоит самый дорогой и самый дешевый препарат; сколько препаратов хранится на складе более 3 месяцев; сколько стоят все препараты, хранящиеся на складе, отыскать препараты, остаток которых равен нулю , ниже требуемого по заказам.

### **Вариант15.**

—Электронный журнал посещаемости"

Предметная область представлена следующими документами:

- Список студентов
- Журнал посещаемости
- Расписание занятий
- Предусмотреть учет пропусков по уважительным, неуважительным причинам.
- Подсчет пропусков по каждому студенту, за неделю, месяц, заданный период, по конкретному предмету.

### **Вариант16.**

«Итоги сессии»

База данных должна содержать информацию о двух последних сессиях студентов. Источником информации являются экзаменационные ведомости. Необходимо проводить анализ успеваемости по специальностям, формам обучения, курсам, группам, предметам, вычислять средний балл по указанным критериям, а также число каждой оценки .

## Практическая работа №9 «Защита БД»

Цель работы: изучение способов защиты информации в СУБД Microsoft Access

### Создание, изменение и удаление паролей

Частичную защиту базы данных обеспечивает добавление пароля, ограничивающего пользователей, которые могут открывать базу данных.

*Microsoft Access* хранит пароль базы данных в незашифрованном виде. Если это нарушает безопасность защищаемой паролем базы данных, не следует использовать пароль. Вместо этого необходимо определить *защиту на уровне пользователей*.

Защита на уровне пользователей предполагает задание администратором базы данных определенных разрешений отдельным пользователям и группам пользователей на следующие объекты: таблицы, запросы, формы, отчеты и макросы.

#### Внимание!!!

- Если Вы потеряли или забыли пароль, восстановить его будет невозможно, и вы не сможете открыть базу данных.
- Нельзя установить пароль базы данных, если для базы данных была определена защита на уровне пользователей, а у Вас нет разрешений администратора для базы данных. Кроме того, пароль базы данных определяется дополнительно к защите на уровне пользователей.
- Если была определена защита на уровне пользователей, то любые ограничения, основанные на разрешениях на доступ на уровне пользователей, остаются в силе.

#### Соглашения о паролях

Имена учетных записей могут иметь длину от 1 до 20 знаков и могут состоять из букв, цифр, пробелов и символов из расширенных наборов, за исключением следующих:

- знаки " \ [ ] : | < > + = ; , . ? \*
- пробелы в начале имени;
- управляющие знаки (с кодами ASCII от 10 до 31).

**Примечание.** В паролях учитывается регистр знаков.

#### Практическая часть

1. Закройте базу данных. Если база данных открыта для общего доступа в сети, убедитесь, что все остальные пользователи закрыли ее.



2. Сделайте резервную копию базы данных и сохраните ее в надежном месте.
3. В меню **Файл** выберите команду **Открыть**.
4. Щелкните стрелку справа от поля **Папка**, выберите диск и папку, в которой расположен файл базы данных *Microsoft Access*, и щелкните значок этого файла.
5. Щелкните стрелку справа от кнопки **Открыть**, выберите вариант **Монопольно** и откройте базу данных.
6. В меню **Файл** выберите команду **Зашифровать с использованием пароля** и в появившемся окне **Задать пароль базы данных** введите пароль .
7. Для подтверждения пароля введите его еще раз в поле **Подтверждение**, а затем нажмите кнопку **ОК**.

Теперь пароль задан. При следующем открытии базы данных появляется диалоговое окно, в которое необходимо ввести пароль.

#### Примечания

- Пароль базы данных сохраняется в базе данных, а не в файле рабочей группы. Файл рабочей группы – это файл, который *Microsoft Access* считывает при запуске, содержащий сведения о пользователях, входящих в рабочую группу. Эти сведения включают имена учетных записей пользователей, их пароли и имена групп, в которые входят пользователи.

- Если таблица из защищенной паролем базы данных является **связанной**, то при установлении связи пароль сохраняется (помещается во временный буфер) в базе данных, с которой связывается таблица. Это позволит любому пользователю видеть ваши данные.

#### Задание

Закройте паролем Вашу базу данных.

### Отображение и скрытие объектов базы данных в окне базы данных

Указание отображения или скрытия объекта базы данных

1. В списке **Объекты окна базы данных** (Окно базы данных. Окно, которое открывается при открытии базы данных *Microsoft Access* или проекта *Microsoft Access*. В окне базы данных выводятся ярлыки для создания новых объектов базы данных и открытия существующих объектов.) выберите тип объекта базы данных, свойства которого нужно изменить.

2. Нажмите кнопку **Свойства** на панели инструментов **База данных**.

3. Установите или снимите флажок **скрытый**.

### Отображение или скрытие объектов, скрытых по умолчанию

Если нужно выполнить какие-либо действия с объектами, которые были определены как скрытые, можно отобразить эти объекты в окне базы данных , не отменяя их атрибута скрытия.

1. В меню **Сервис** выберите команду **Параметры**.

2. Выберите вкладку **Вид**.

3. Установите или снимите флажок **скрытые объекты** в группе **Отображать**. Чтобы показать различие между скрытыми и остальными объектами, скрытые объекты отображаются в виде серых значков.

#### **Удаление пароля в базе данных *Microsoft Access (.mdb)***

1. В меню **Файл** выберите команду **Открыть**.
2. Щелкните стрелку справа от кнопки **Открыть**, выберите команду **Монопольно** и откройте базу данных.
3. В меню **Файл** выберите команду **Расшифровать БД**.
4. В диалоговом окне **Пароль** введите пароль базы данных и нажмите кнопку **ОК**. Пароли учитывают регистр знаков (учет регистра – это учет различия между строчными и прописными буквами. При поиске с учетом регистра требуется совпадение с образцом с точностью до прописных и строчных букв).
5. В меню **Сервис** выберите команды **Защита** и **Удалить пароль базы данных**. Эта команда доступна, когда пароль базы данных уже установлен.
6. В диалоговом окне **Удаление пароля базы данных** введите текущий пароль.

## **Раздел. Администрирование баз данных**

Администрирование БД – это управление информационными ресурсами, включая планирование базы данных, разработку и внедрение стандартов, определение ограничений и процедур, а также концептуальное и логическое проектирование баз данных. Администратор данных отвечает за корпоративные информационные ресурсы, включая и некомпьютеризированные данные. На практике это часто связано с управлением данными, которые являются совместно используемым ресурсом для различных пользователей и прикладных программ данной организации.

СУБД - это совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных. В разных организациях количество сотрудников, выполняющих функции администратора, может отличаться и обычно определяется размерами самой организации. Основная обязанность Администратора состоит в обмене консультациями и советами со старшими менеджерами, а также в слежении за тем, чтобы применение технологий баз данных продолжало соответствовать корпоративным целям. Должность администратора обычно принадлежит отделу информационных систем организации. В настоящее время при обдумывании стратегии планирования информационной системы все больший акцент делается на важности администратора. Организации все в большей и большей степени склонны уделять внимание значению данных, используемых или собранных в их информационной системе, как средству достижения более высокой конкурентоспособности. В результате возникает обязательное требование слияния стратегии построения информационных систем с бизнес - стратегиями организации.

Это позволяет создать организацию с более гибкой структурой, способную адаптироваться к резким изменениям, имеющую более творческую и инновационную внутреннюю среду, обеспечивающую эффективную перестройку бизнес-процессов в случае необходимости. Упомянутый перенос акцентов означает, что администратор во все большей мере должен понимать идеологию развития не только информационных систем, но и бизнес-процессов, и играть ключевую роль в разработке стратегии развития информационной системы, поддерживая ее соответствие деловым стратегиям организации. Это изменение мышления отражает происшедшее в недавнем прошлом драматическое изменение в назначении компьютерных систем: от исходного использования компьютеров для более эффективного управления некоторыми аспектами бизнес-процессов, через последующее повышение эффективности бизнес-процессов, до поддержки и обеспечения изменчивости и инновационности организаций.

**Основные задачи администрирования данных включают:**

- выбор подходящих инструментов разработки;
- помощь в разработке корпоративных стратегий построения информационной системы, развития информационных технологий и бизнес-стратегий;
- предварительная оценка осуществимости и планирование процесса создания базы данных;
- разработка корпоративной модели данных;
- определение требований организации к используемым данным;
- определение стандартов сбора данных и выбор формата их представления;
- оценка объемов данных и вероятности их роста;
- определение способов и интенсивности использования данных;
- определение правил доступа к данным и мер безопасности, соответствующих правовым нормам и внутренним требованиям организации;
- концептуальное и логическое проектирование базы данных;
- взаимодействие с администратором баз данных и разработчиками приложений, с целью обеспечения соответствия создаваемых приложений всем существующим требованиям;
- обучение пользователей — изучение существующих стандартов обработки данных и юридической ответственности за их некорректное применение;
- постоянная модернизация используемых информационных систем и технологий по мере развития бизнес-процессов;
- обеспечение полноты всей требуемой документации, включая корпоративную модель, стандарты, ограничения, процедуры, использование словаря данных, а также элементы управления работой конечных пользователей;
- поддержка словаря данных организации;
- взаимодействие с конечными пользователями для определения новых требований и разрешения проблем, связанных с доступом к данным и недостаточной производительностью их обработки.

## **Обязанности администратора современных систем управления базами данных**

Администратор баз данных в соответствии с возложенной на него задачей выполняет следующие обязанности:

1. Анализирует информационные потребности структурных подразделений предприятия.
2. Осваивает программные средства, предлагаемые отделом информатизации.
3. Согласовывает с отделом информатизации вопросы по совершенствованию процесса хранения и обработки информации с целью обеспечения требуемой достоверности результатов и минимизации времени расчетов.
4. Обеспечивает обмен информацией с отделом информатизации в соответствии с установленным порядком (в том числе с использованием электронных сетей телекоммуникаций).
5. По запросам заинтересованных подразделений организации совместно с отделом информатизации составляет технические задания на разработку программного обеспечения новых задач.
6. Осуществляет комплекс работ по внедрению программных средств.
7. Осуществляет систематическое обучение работников предприятия приемам и навыкам работы на компьютерах, а также работе с внедряемыми программными средствами.
8. Осуществляет контроль за работой компьютеров и своевременно сообщает о сбоях и неисправностях в отдел информатизации.
9. Осуществляет учет и хранение документов, имеющих отношение к автоматизированной обработке информации на компьютерах.
10. Обобщает и анализирует замечания пользователей по результатам эксплуатации задач и передает информацию об отмеченных недостатках в отдел информатизации.
11. Следит за обеспечением целостности, достоверности и сохранности циркулирующих в автоматизированной информационной системе данных.
12. Ведет архив используемых программных средств и нормативно-справочной информации.
13. Выдает руководству предприятия рекомендации по распределению между структурными подразделениями поступающей вычислительной техники с учетом наличия программного обеспечения, подготовленных специалистов и анализа загрузки, находящихся в эксплуатации ПЭВМ.