



Министерство науки и высшего образования  
Российской Федерации

**Братский педагогический колледж**  
федерального государственного бюджетного  
образовательного учреждения высшего  
образования  
«Братский государственный университет»

# **ОСНОВЫ ПРОЕКТИРОВАНИЯ БАЗ ДАНЫХ**

**методические рекомендации  
по выполнению лабораторных занятий**

для студентов 3 курса  
очной формы обучения  
специальности

09.02.07 Информационные системы и программирование

Автор: Л.Д. Разумова

Братск, 2020

Основы проектирования баз данных. Методические рекомендации по выполнению лабораторных занятий . /Сост. Л.Д. Разумова.- Братск, 2020.- 61с.

Методические рекомендации предназначены для студентов третьего курса специальности 09.02.07 Информационные системы и программирование при изучении дисциплины «Основы проектирования баз данных»

Печатается по решению научно-методического совета  
Братского педагогического колледжа ФГБОУ ВО «БрГУ»  
665709, г. Братск, ул. Макаренко 40

## СОДЕРЖАНИЕ

Введение .....	4
Лабораторное занятие № 1. Сбор сведений и системный анализ предметной области.....	7
Лабораторное занятие № 2. Разработка концептуальной ER-модели предметной области. ....	18
Лабораторное занятие № 3. Создание логической реляционной модели базы данных.....	23
Лабораторное занятие № 4. Создание физической модели БД средствами СУБД Microsoft Access.....	31
Лабораторное занятие № 5. Создание SQL запросов в среде СУБД Microsoft Access (на выборку).....	56
Лабораторное занятие № 6 Создание SQL запросов в среде СУБД Microsoft Access (создание, обновление, добавление, удаление).....	57
Список использованных источников.....	59

## ВВЕДЕНИЕ

Лабораторные занятия являются одним из основных видов учебных занятий, направленных на экспериментальное подтверждение теоретических положений и формирование учебных и профессиональных практических умений. Они составляют важную часть теоретической и профессиональной практической подготовки.

Лабораторные занятия направлены на экспериментальное подтверждение теоретических положений и формирование учебных и профессионально-значимых умений обучающихся.

Выполнение лабораторных занятий проводится с целью:

- формирования практических умений в соответствии с требованиями к уровню подготовки студентов, установленными рабочей программой дисциплины по конкретным разделам (темам);

- обобщение, систематизацию, углубление, закрепление полученных теоретических знаний;

- совершенствование умений применять полученные знания на практике, реализацию единства интеллектуальной и практической деятельности;

- развитие интеллектуальных умений у будущих специалистов: аналитических, проектировочных, конструктивных и др.;

- выработку при решении поставленных задач таких профессионально значимых качеств, как самостоятельность, ответственность, точность, творческая инициатива.

*В результате освоения учебной дисциплины обучающийся должен уметь:*

- проектировать реляционную базу данных;

- использовать язык запросов для программного извлечения сведений из баз данных.

*В результате освоения учебной дисциплины обучающийся должен знать:*

- основы теории баз данных;

- модели данных;

- особенности реляционной модели и проектирование баз данных, изобразительные средства, используемые в ER-моделировании;

- основы реляционной алгебры;

- принципы проектирования баз данных, обеспечение непротиворечивости и целостности данных;
- средства проектирования структур баз данных;
- язык запросов SQL.

*В результате освоения учебной дисциплины у обучающегося должны быть сформированы следующие профессиональные компетенции:*

ПК 1.1. Сбирать данные для анализа использования и функционирования информационной системы, участвовать в составлении отчетной документации, принимать участие в разработке проектной документации на модификацию информационной системы.

ПК 1.2. Взаимодействовать со специалистами смежного профиля при разработке методов, средств и технологий применения объектов профессиональной деятельности.

ПК 1.3. Производить модификацию отдельных модулей информационной системы в соответствии с рабочим заданием, документировать произведенные изменения.

ПК 1.7. Производить инсталляцию и настройку информационной системы в рамках своей компетенции, документировать результаты работ.

ПК 1.9 Выполнять регламенты по обновлению, техническому сопровождению и восстановлению данных информационной системы, работать с технической документацией.

### **Общие указания по выполнению работ на лабораторных занятиях**

О проведении лабораторного занятия студентам сообщается заблаговременно: когда предстоит практическое занятие, какие вопросы нужно повторить, чтобы ее выполнить. Просматриваются задания, оговаривается объем и время выполнения. Критерии оценки сообщаются перед выполнением каждого лабораторного занятия.

Студенты получают распечатанный или электронный вариант задания с описанием этапов выполнения работы.

При выполнении лабораторного занятия студент придерживается следующего алгоритма:

1. Записать в тетради дату, тему и цель занятия.
2. Ознакомиться с правилами и условиями выполнения практического задания.
3. Повторить теоретические задания, необходимые для рациональной работы и других практических действий.
4. Выполнить работу по предложенному алгоритму действий.
5. Обобщить результаты работы, сформулировать выводы по работе.
6. Записать в тетрадь ответы на контрольные вопросы.

Задания студентами сдаются в электронном виде. Все выполненные работы нумеруются, в соответствии с номером практического задания, и сохраняются в отдельной папке.

# ЛАБОРАТОРНАЯ РАБОТА № 1

## ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ. СБОР СВЕДЕНИЙ И СИСТЕМНЫЙ АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.

**Цель:** получить знания и умения по основным этапам проектирования базы данных.

**Задание 1.** Изучить конспект лекций и ответить письменно на контрольные вопросы:

1. Дайте определение банка данных.
2. Дайте определение СУБД.
3. Дайте определение базы данных.
4. Что означает термин «самодокументированность»?
5. Какие четыре типа данных содержит база данных?
6. Основные этапы проектирования БД.
7. Дайте определение предметной области.
8. Дайте определение «данные».
9. Дайте определение «модель данных»
10. Классификация моделей данных. Назовите первый уровень классификации .
11. Назовите два подхода к выбору состава и структуры предметной области.

**Задание 2.** Проанализировать предложенные бизнес-правила (Варианты лабораторной работы), определить объекты предметной области, провести подробное словесное (текстовое) описание объектов предметной области и реальных связей, присутствующих между реальными объектами.

### Отчет по лабораторной работе

Содержит ответы на контрольные вопросы и текстовое описание предметной области. Отчет может быть оформлен в рукописном или печатном варианте.

## Пример выполнения задания 2.

### Предложенные бизнес-правила «АЭРОПОРТ».

Создаваемая информационная система предназначена для учета движения самолетов и пассажиропотока. В аэровокзале имеется расписание движения самолетов, которое включает: Номер рейса, Тип самолета, Маршрут, Пункты промежуточной посадки, Время отправления, Дни полета. В системе ведется учет: Количество свободных мест на каждом рейсе, Общий вес пассажиров, Вес ручной клади, Вес багажа. Система формирует посадочную ведомость с учетом веса багажа и ручной клади. В системе имеется справочник типов самолетов, в котором учитываются: Количество мест, Суммарная грузоподъемность.

### Текстовое описание предметной области

#### Задача «Аэропорт»:

1. Каждый аэропорт обслуживает рейсы разных авиакомпаний и имеет международный код и название.

2. Авиакомпания характеризуется названием. У каждой авиакомпании есть несколько рейсов, проходящих через этот аэропорт.

3. Каждому рейсу соответствует один самолёт («воздушное судно»), каждому самолёту — несколько рейсов.

4. Самолёт характеризуется номером, маркой, моделью, вместимостью.

5. Каждому рейсу соответствует несколько экипажей, выполняющих их в разное время согласно расписанию.

6. Рейсы могут быть терминальные — завершающиеся или начинающиеся в аэропорту и транзитные, которые используют аэропорт только для временной посадки для заправки и отдыха.

7. Рейс характеризуется номером, типом (терминальные/транзитные), аэропортом отправления, аэропортом назначения, временем отправления, временем прибытия, временем в пути, дальностью, периодичностью (по нечётным дням, по чётным, по выходным, каждый N понедельник/вторник/.../воскресенье).

8. Экипаж состоит из нескольких сотрудников авиакомпании. Каждый член экипажа имеет ФИО, должность (командир, пилот, стюардесса) и лётный стаж, исчисляющийся в количестве вылетов.



9. В аэропорту происходит 2 типа событий — вылет и посадка. Каждое событие — вылет или посадка — характеризуется состоянием: «ожидается»/«состоялось»/«отменён», датой, временем задержки/опережения \_\_\_\_\_.

## **ВАРИАНТЫ ЛАБОРАТОРНОЙ РАБОТЫ**

### **Вариант 1**

#### **Бизнес-правила «РЕСТОРАН».**

Постоянным клиентам предоставляется возможность заказать столик заранее. Официант указывает столик, открывает гостевой счет и вводит заказы в соответствии с меню. Далее заказ автоматически обрабатывается, формируются марки на приготовление выбранных блюд и направляют их на производство, в соответствующие цеха кухни, в бар. Расчеты с посетителем сводятся к простой операции: на бланке печатается итоговый счет. Если клиент – постоянный посетитель, то соответствующие привилегии рассчитываются автоматически, затем указываются способ оплаты и полученная от клиента сумма.

### **Вариант 2**

#### **Бизнес-правила «КИНОТЕАТР».**

Продажа и бронирование билетов, а также резервирование мест для постоянных посетителей — основные технологические процессы работы кинотеатра. Важную роль здесь играет качество предоставления информации и контроль выполнения операций. Клиент в момент покупки билета должен видеть план зала и свободные места. Постоянные клиенты имеют возможность зарезервировать билеты по телефону или через Интернет. Формирование билета и его печать. Выводить анонс сеансов с указанием времени и кратким описанием.

### **Вариант 3**

#### **Бизнес-правила «ГОСТИНИЦА».**

Номера в гостинице имеют разный уровень обслуживания и соответственно разную стоимость, (предоставление информации о свободных номерах и их стоимости). Клиенты могут бронировать номера по телефону или Интернету. За номерами прикреплен

обслуживающий персонал. Необходимо вести учет обслуживания и оплаты номеров, (заказы в номер, телефонные звонки и т. д. ) . Клиент может несколько раз останавливаться в гостинице в разных номерах.

#### **Вариант 4**

##### **Бизнес-правила «ФИТНЕС – КЛУБА».**

Они предлагают пакеты услуг – абонементы. Подразумевая предоплату определенного набора услуг. Абонемент позволяет пользоваться ими в течение определенного времени. Для идентификации владельца абонемента используются клубные карты. Комплекс позволяет быстро и просто осуществлять резервирование ресурсов по просьбе постоянного клиента предприятия: как тренера, так и места — спортзала, солярия, бассейна для персональных тренировок или занятий.

#### **Вариант 5**

##### **Бизнес-правила «ОПТОВЫЙ СКЛАД».**

Создаваемая информационная система предназначена для учета деятельности оптового склада. Оптовый склад состоит из нескольких складских помещений, каждое помещение имеет наименование, адрес и кладовщика. Склад принимает партии товаров от поставщиков и отпускает его клиентам мелкими партиями. Требуется вести (количественный и стоимостной) учет поступающих и отпускаемых товаров, поставщиков и клиентов, формировать приходные и расходные накладные. Сведения о товаре: Артикул, Наименование полное, Наименование сокращенное, Производитель, Поставщик, Количество, Цена. Сведения о поставщике и клиенте: Наименование, Адрес, Телефон. Накладная включает: Номер, Дата, Клиент, Список товаров, Общая сумма, Кладовщик. В системе формируются отчеты о поступлении и отпуске товаров на складе за произвольный период.

#### **Вариант 6**

##### **Бизнес-правила «РЕКЛАМНОЕ АГЕНТСТВО».**

Создаваемая информационная система должна вести учет деятельности рекламного агентства. Рекламное агентство регистрирует заявки от рекламодателей и публикует рекламы в печатных изданиях. О рекламодателе регистрируются следующие

данные: Наименование, Адрес, Руководитель, Телефон, Заявка, Оплата, Издание, Место размещения рекламы.

Заявка включает: Вид рекламы, Объем, Желаемые издания, Количество выходов рекламы, Дополнительная информация. Заявка от рекламодателя может содержать публикацию в несколько печатных изданиях и на различные даты выхода. Справочник печатных изданий включает: Наименование, Виды реклам, Стоимость рекламы. Требуется вести списки печатных изданий с их расценками на рекламу, списки рекламодателей, заявок. Система должна обеспечить оперативный просмотр списка заявок (печатные издания, рекламодатель, стоимость) на любую вводимую дату, а также формирование отчета о заявленных и выполненных рекламах.

### **Вариант 7**

#### **Бизнес-правила «КИНОТЕАТР».**

Создаваемая информационная система предназначена для учета проданных билетов в кинотеатре. Кинотеатр имеет несколько залов. Сеансы планируются для каждого зала отдельно. Система формирует базу данных, включающую следующую информацию: Место и сеанс, Справочник кинозалов, Справочник сеансов и стоимость, Справочник фильмов, Справочник жанров. Система формирует отчеты: Отчет о посещаемости по месяцам, Отчет о популярности фильмов, Отчет о популярности жанров. Необходимо предусмотреть возврат билетов и денег.

### **Вариант 8**

#### **Бизнес-правила «МАГАЗИН “ЦВЕТЫ”».**

Создаваемая информационная система предназначена для учета деятельности магазина по продаже цветов. В системе формируется база данных отдельных цветов и готовых букетов: Наименование цветка или букета, Поставщик цветов, Состав букета, Стоимость, Срок поступления, Срок и место хранения (выставочный зал, склад), Дата продажи. В системе ведется учет бракованных и увядших цветов. Формируется отчет о движении товара за заданный период времени.

## **Вариант 9**

### **Бизнес-правила «АДМИНИСТРАТОР ГОСТИНИЦЫ».**

Создаваемая информационная система предназначена для учета деятельности гостиницы. В гостинице имеется список номеров: Место нахождения номера, Класс, Число мест, Признак занятости места, Дата освобождения номера. Каждый гость проходит регистрацию: Паспортные данные, Даты приезда и отъезда, Номер, Место, Цель приезда, Организация, в которую прибыл (в случае командировки). Администратор гостиницы осуществляется поселение гостя: выбор подходящего номера (при наличии свободных мест), регистрация, оформление квитанции. В системе автоматически формируется квитанция об оплате услуг гостиницы. Система должна предусмотреть оформление дополнительной квитанции в случае продления гостем срока проживания в гостинице. В системе имеется возможность поиска гостя по произвольному признаку и формируется отчет о текущем состоянии номеров гостиницы (номер, место, не занят/ занят и кем, дата отъезда).

## **Вариант 10**

### **Бизнес-правила «МУЗЫКАЛЬНЫЙ МАГАЗИН».**

Создаваемая информационная система предназначена для учета музыкальных произведений в магазине. В системе формируются: База групп и исполнителей, База песен, База дисков с перечнем песен (в виде ссылок). База групп и исполнителей содержит: Наименование группы или исполнителя, Страна, Год образование группы или год начала творческого пути, Краткое содержание творческого пути. База песен содержит: Название, Автор текста, Автор музыки, Время звучания. База дисков содержит: Название диска, Перечень песен (название, исполнитель, время звучания, номер трека). Система имеет возможность поиска всех песен заданной группы (исполнителя). Имеется возможность выбора всех дисков, где встречается заданная песня.

## **Вариант 11**

### **Бизнес-правила «АВТОЗАПРАВКА (АЗС)».**

Создаваемая информационная система предназначена для учета деятельности автозаправки. На автозаправке имеются несколько колонок для заправки топливом: АИ-98, АИ-95, АИ-92,

АИ-80, АИ-76, Дизельное топливо. В базе данных должна быть информация: О колонках, О видах бензина, О ценах и остатках. Необходимо учитывать отпуск топлива по чеку: Номер колонки, Тип топлива, Количество, Цена за литр, Стоимость.

Предусмотреть отпуск топлива по дисконтной карточке со скидкой, при этом необходимо учитывать: Номер карточки, Общее количество отпущенного топлива, Скидка в %. Размер скидок зависит от общего количества заправленного топлива. В 19 часов – “пересменка” операторов АЗС, печатается отчет об отпуске топлива за время от 19 часов предыдущего дня до 19 часов текущего дня.

### **Вариант 12**

#### **Бизнес-правила «САЛОН КРАСОТЫ».**

Создаваемая база данных предназначена для учета деятельности салона красоты. База данных салона красоты включает данные об оказываемых услугах, мастерах и оказанных услугах. В системе имеется график работы мастеров, и расписание на день с разделением на интервалы времени (например, по полчаса). О мастере учитываются следующие данные: Время работы, Обслуживаемые клиенты, Оказанные услуги, Сумма оказанных услуг. О клиенте учитываются следующие данные: Фамилия, Инициалы, Телефон. Осуществляется предварительная запись клиентов к мастерам. Случайный клиент может обслуживаться свободным мастером. В системе формируются отчеты по деятельности мастеров (процент их занятости, оказанные услуги).

### **Вариант 13**

#### **Бизнес-правила «ТУРИСТИЧЕСКАЯ ФИРМА».**

Создаваемая информационная система предназначена для учета результатов деятельности туристической фирмы. В туристической фирме ведется учет путевок: Страна, Место пребывания, Сроки, Цена и скидки, Вид транспорта, Маршрут (дата, место пребывания), Количество мест, Гид, Данные о туристах (ФИО, дата рождения, паспорт, адрес, страховой полис, оплата), Менеджер. В системе формируются отчеты о реализации путевок. Необходимо предусмотреть возможность возврата путевок.

### **Вариант 14**

#### **Бизнес-правила «МАГАЗИН ИГРУШЕК».**

Создаваемая база данных предназначена для учета товара в магазине игрушек. В системе формируется база данных магазина игрушек: Наименование товара, Изготовитель, Страна, По какой накладной поступил товар, Для какого возраста, Цена, Место нахождения (склад, торговый зал, бракованная группа), Факт продажи (дата, номер кассового чека), Факт возврата брака. Товары в магазин поступают по накладной, в которой отражены: Номер и дата накладной, Поставщик, Перечень товаров (наименование, количество, цена), Общая сумма товаров по накладной. В системе формируется отчет о движении товаров за заданный период.

### **Вариант 15**

#### **Бизнес-правила «КОНДИТЕРСКОЕ ПРЕДПРИЯТИЕ».**

Создаваемая информационная система предназначена для учета изделий кондитерского предприятия. В системе ведется учет выпускаемых изделий: Конфеты, Печенье, Вафли, Пирожные, Торты, Напитки. Каждое изделие имеет соответствующий: Вид упаковки, Вес, Единицы измерения, Количество, Цену, Дату изготовления, Срок реализации, Дату отгрузки оптовым покупателям (магазины, фирмы и т.п.). В системе формируются отчеты о финансово-хозяйственной деятельности кондитерского предприятия, в котором отражаются данные о произведенных и реализованных кондитерских изделиях.

### **Вариант 16**

#### **Бизнес-правила «БИБЛИОТЕКА».**

Создаваемая информационная система предназначена для учета книг публичной библиотеки. В библиотеке ведется картотека книг. По каждой книге учитываются данные: Авторы, Название, Экземпляр, Издательство, Год издания, Количество экземпляров, Раздел библиотеки (специальная литература, хобби, домашнее хозяйство, беллетристика и так далее), Происхождение книги (приобретена, подарена и т.п.), Наличие книги в данный момент (книгу может взять читатель), Оценка книги читателями. Учет читателей ведется по следующим данным: Читательский билет, Фамилия, Имя, Отчество, Адрес, Телефон, Книги, Дата возврата (планируемая и фактическая) Система обрабатывает данные и

выдает результат: Перечень выданных книг на текущую дату, Статистику (кто чаще берет книги, кто возвращает в срок, кто нарушает сроки возврата?)

### **Вариант 17**

#### **Бизнес-правила «ТРАНСПОРТНОЕ ПРЕДПРИЯТИЕ».**

Создаваемая информационная система предназначена для учета путевых листов. Транспортное предприятие ведет учет путевых листов, выданных водителям (от одного дня до 14 дней). Данные путевого листа:

- номер путевого листа
- дата выдачи
- автомобиль (марка, госномер)
- водитель (может быть два водителя)
- маршрут
- объем бака
- заправка топливом (тип топлива, остаток в баке, количество заправленного топлива)
- показания спидометра на моменты выдачи путевого листа и сдачи путевого листа на обработку.

В системе имеется база данных норм расхода топлива по каждой марке автомобиля. При обработке путевого листа производится расчет расхода топлива по спидометру (плановый расход) и сравнение с фактическим (по остатку топлива). Ежедневно выводятся результаты обработки путевых листов по расходу топлива.

### **Вариант 18**

#### **Бизнес-правила «КОММУНАЛЬНЫЕ УСЛУГИ».**

Создаваемая информационная система предназначена для учета оплат коммунальных услуг жителями микрорайона. В ЖЭКе ведется учет оплаты жильцами коммунальных услуг:

- дом
- квартира
- основной квартиросъемщик
- квартплата
- электроэнергия
- газ, вода (горячая и холодная)
- вывоз мусора, лифт.

Учет электроэнергии ведется по соответствующим показаниям счетчиков. Учет газа и воды у части жильцов осуществляется по показаниям счетчиков, у части по тарифам. В системе имеются справочники по тарифам для расчета суммы оплаты, а так же льгот по оплате коммунальных услуг. Формируются отчеты об оплате коммунальных услуг.

### **Вариант 19**

#### **Бизнес-правила «МЕБЕЛЬНЫЙ САЛОН».**

Разрабатываемая информационная система предназначена для учета деятельности мебельного салона. Мебельный салон имеет несколько магазинов. При получении мебели в магазин в базу данных заносятся данные: Название мебели, Тип мебели, Производитель мебели, Дата добавления мебели, Стоимость мебели, Количество мебели. При покупке мебели в базу данных заносятся данные: Код магазина, Номер мебели, Код типа мебели, Количество выбранной мебели, Код покупателя.

По требованию покупателя кассир распечатывает чек об оплате. При добавлении нового магазина в базу данных заносятся данные: Код магазина, Название магазина, ФИО заведующего магазином, Номер телефона, Местонахождение магазина. При возврате мебели в базу данных заносятся данные: Код магазина, Номер типа мебели, Номер мебели, Количество мебели, Причина возврата мебели. По требованию кассира покупатель показывает чек и причину возврата мебели. Данные по названию и виду мебели заносятся с чека.

### **Вариант 20**

#### **Бизнес-правила «СПРАВОЧНАЯ АПТЕКА».**

Создаваемая информационная система предназначена для учета медицинских препаратов в сети аптек города с возможностью поиска препаратов через сеть Интернет. В системе формируется база данных, содержащая данные: Название аптеки, Адрес аптеки, Телефон, Проезд к аптеке. Перечень медицинских препаратов: Группа товара, Наименование, Единица измерения, Область применения, Производитель, Страна, Поставщик, Дата поставки, Дата изготовления, Срок годности, Место нахождения (торговый зал, склад и т.п.), Количество и цена, Наличие препарата на данный



момент. В системе формируются отчеты о движении медицинских препаратов.

### **Вариант 21**

#### **Бизнес-правила «АПТЕКА».**

Предположим вы владелец аптеки. В аптеке имеются различные препараты различного назначения (от головной боли, сердечной, желудочной и т.д.) и различного применения (внутреннего, наружного и т.д.) соответственно по различным оптовым ценам. Вы работаете, в основном, с постоянными поставщиками и для этого вам необходимо знать наименование их фирм, фамилию, имя и отчество руководителя, адрес, номер телефона. Каждую операцию по поставке товаров вы заносите в книгу учета, где регистрируете дату поставки, вид оплаты (наличный, безналичный, по кредитной карточке), наименование поступающего медикамента, номер сопровождающего документа, количество.

### **Вариант 22**

#### **Бизнес-правила «МЕЖДУГОРОДНИМИ ПОСТАВКАМИ ГРУЗОВ».**

Предположим, вы занимаетесь междугородними поставками грузов. У вас в наличии несколько грузовых машин. При оформлении рейса ведется составление регистрационной записи, содержащей следующую информацию: регистрационный номер, дату отправки, место назначения, ФИО водителя, бортовой номер машины, вид груза и его количество \_\_

**ЛАБОРАТОРНОЕ ЗАНЯТИЕ № 2**  
**ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ**  
**ИНФОЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ**  
**РАЗРАБОТКА КОНЦЕПТУАЛЬНОЙ ER-МОДЕЛИ**  
**ПРЕДМЕТНОЙ ОБЛАСТИ.**

**Цель:** получить знания и умения по основным этапам проектирования базы данных.

**Задание 1.** Изучить конспект лекций и ответить письменно на контрольные вопросы:

1. Дайте определение: типы сущностей, сущность, слабый и сильный тип сущности.
2. Дайте определение: атрибут; домен атрибута; простой, составной, однозначный, многозначный, производный атрибут.
3. Дайте определение: потенциальный, первичный, составной ключ.
4. Дайте определение: тип связи, связь, степень связи.
5. Какие типы связи существуют в инфологическом проектировании.
6. Обозначьте графические обозначения основных элементов модели.
7. Определите правила трансляции текстовой модели в ER-модель.

**Задание 2.**

На основе анализа проведенного подробного словесного (текстового) описания предметной области и реальных связей между реальными объектами (лабораторная работа № 1- согласно варианту работы):

1. Транслировать текстовую модель предметной области в концептуальную ER- модель предметной области.

**Отчет по лабораторной работе**

Содержит ответы на контрольные вопросы, ER-модель предметной области (согласно варианта работы) . Отчет может быть оформлен в рукописном или печатном варианте.

## Теоретические сведения - основные концепциями ER-моделирования.

### *Типы сущностей*

Типы сущностей — объект или концепция, которые характеризуются на данном предприятии как имеющие независимое существование.

Сущность — экземпляр типа сущности, который может быть идентифицирован уникальным образом.

Слабый тип сущности — тип сущности, существование которого зависит от какого-то другого типа сущности.

Сильный тип сущности — тип сущности, существование которого не зависит от какого-то другого типа сущности.

### *Атрибуты*

Атрибут — свойство типа сущности или связи.

Домен атрибута — набор значений, которые могут быть присвоены атрибуту.

Простой атрибут — атрибут, состоящий из одного компонента с независимым существованием.

Составной атрибут — атрибут, состоящий из нескольких компонентов, каждый из которых характеризуется независимым существованием.

Однозначный атрибут — атрибут, который содержит одно значение для одной сущности.

Многозначный атрибут — атрибут, который содержит несколько значений для одной сущности.

Производный атрибут — атрибут, который представляет значение, производное от значения связанного с ним атрибута или некоторого множества атрибутов, принадлежащих некоторому (не обязательно данному) типу сущности.

### *Ключи*

Потенциальный ключ — атрибут или набор атрибутов, который уникально идентифицирует отдельные экземпляры типа сущности.

Первичный ключ — потенциальный ключ, который выбран в качестве первичного ключа.

Составной ключ — потенциальный ключ, который состоит из двух или больше атрибутов.

### *Типы связей*

Тип связи — осмысленная ассоциация между сущностями разных типов.

Связь — ассоциация между сущностями, включающая по одной сущности из каждого участвующего в связи типа сущности.

Степень связи — количество сущностей, которые охвачены данной связью.

### *Графические обозначения основных элементов модели:*

1. Сущности обозначаются с помощью прямоугольников.
2. Атрибуты обозначаются в виде овалов, связанных с сущностями, к которым они принадлежат.
3. Связи обозначаются с помощью ромбов, соединённых линиями с участвующими в них сущностями.
4. Имена ключевых атрибутов подчёркиваются.
5. Овалы производных атрибутов отображаются прерывистой линией.

### *Правила трансляции текстовой модели в ER-модель*

1. Существительное, образующее некоторое независимое понятие, отображается в сущность-прямоугольник.
2. Существительное или фраза, представляющая собой некоторое свойство понятия, отображается в атрибут-овал этого понятия.
3. Глаголы, описывающие взаимосвязи между понятиями, отображаются в связи-ромбы.

## **Пример выполнения задания 2.**

### **Предложенные бизнес-правила «АЭРОПОРТ».**

#### **(один из вариантов задания)**

Создаваемая информационная система предназначена для учета движения самолетов и пассажиропотока. В аэровокзале имеется расписание движения самолетов, которое включает: Номер рейса, Тип самолета, Маршрут, Пункты промежуточной посадки, Время отправления, Дни полета. В системе ведется учет: Количество свободных мест на каждом рейсе, Общий вес пассажиров, Вес ручной клади, Вес багажа. Система формирует посадочную ведомость с учетом веса багажа и ручной клади. В системе имеется справочник типов самолетов, в котором учитываются: Количество мест, Суммарная грузоподъемность.

## **Текстовое описание предметной области (результат Лабораторной работы № 1)**

### **Задача «Аэропорт»:**

1. Каждый аэропорт обслуживает рейсы разных авиакомпаний и имеет международный код и название.

2. Авиакомпания характеризуется названием. У каждой авиакомпании есть несколько рейсов, проходящих через этот аэропорт.

3. Каждому рейсу соответствует один самолёт («воздушное судно»), каждому самолёту — несколько рейсов.

4. Самолёт характеризуется номером, маркой, моделью, вместимостью.

5. Каждому рейсу соответствует несколько экипажей, выполняющих их в разное время согласно расписанию.

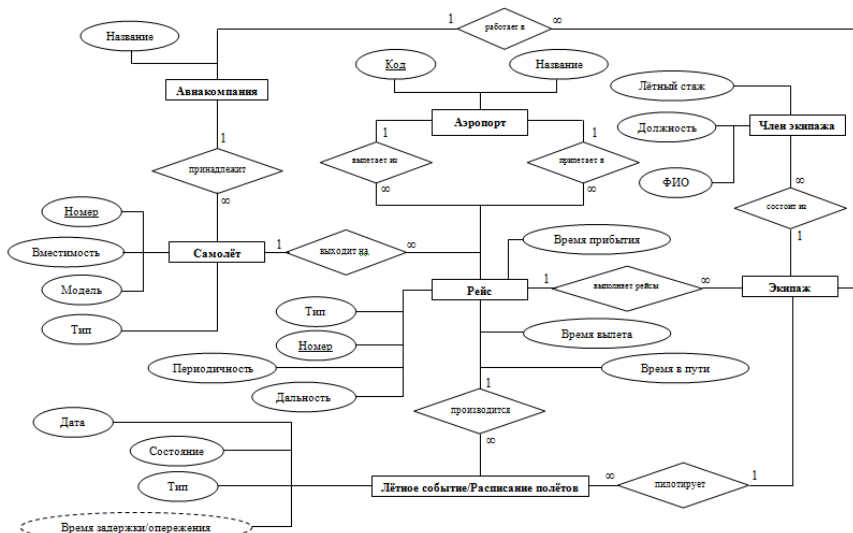
6. Рейсы могут быть терминальные — завершающиеся или начинающиеся в аэропорту и транзитные, которые используют аэропорт только для временной посадки для заправки и отдыха.

7. Рейс характеризуется номером, типом (терминальные/транзитные), аэропортом отправления, аэропортом назначения, временем отправления, временем прибытия, временем в пути, дальностью, периодичностью (по нечётным дням, по чётным, по выходным, каждый N понедельник/вторник/.../воскресенье).

8. Экипаж состоит из нескольких сотрудников авиакомпании. Каждый член экипажа имеет ФИО, должность (командир, пилот, стюардесса) и лётный стаж, исчисляющийся в количестве вылетов.

9. В аэропорту происходит 2 типа событий — вылет и посадка. Каждое событие — вылет или посадка — характеризуется состоянием: «ожидается»/«состоялось»/«отменён», датой, временем задержки/опережения\_\_\_\_\_.

## Концептуальная ER –модель предметной области «Аэропорт». (результат данной лабораторной работы № 2)



# **ЛАБОРАТОРНОЕ ЗАНЯТИЕ № 3**

## **ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ**

### **ДАТАЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ**

#### **СОЗДАНИЕ ЛОГИЧЕСКОЙ РЕЛЯЦИОННОЙ МОДЕЛИ**

#### **БАЗЫ ДАННЫХ**

**Цель:** получить знания и умения по основным этапам проектирования базы данных.

**Задание 1.** Изучить конспект лекций и ответить письменно на контрольные вопросы:

1. Основные элементы реляционной модели.
2. Реляционные ключи.
3. Реляционная целостность.
4. Правила преобразования ER- модели в логическую модель.
5. Что такое процесс нормализации?
6. Какие нормальные формы вы знаете?

#### **Задание 2.**

Создать логическую реляционную модель базы данных из концептуальной ER-модели предметной области(на основе лабораторной работы № 2- согласно варианту работы), используя правила преобразования.

Проверить логическую модель на соответствие 1-3 нормальным формам последовательно и , при необходимости, провести нормализацию.

#### **Отчет по лабораторной работе**

Содержит ответы на контрольные вопросы, логическую реляционную модель базы данных (согласно варианту работы), проверку логической модели на соответствие 1-3 нормальным формам и заключение по результатам проверки.

Отчет может быть оформлен в рукописном или печатном варианте.

#### **Теоретические сведения**

С точки зрения теории реляционных баз данных, существуют следующие понятия.

### *Основные элементы реляционной модели*

Отношение — плоская таблица, состоящая из столбцов и строк.

Атрибут — поименованный столбец отношения.

Домен — набор допустимых значений для одного или нескольких атрибутов.

Кортеж — строка отношения.

Степень — количество атрибутов в отношении.

Кардинальность — количество кортежей, которое содержит отношение.

Реляционная БД — набор нормализованных отношений.

#### *Реляционные ключи*

Суперключ — атрибут или множество атрибутов, которое единственным образом идентифицирует кортеж данного отношения.

Потенциальный ключ — суперключ, который не содержит подмножества, также являющегося суперключом данного отношения.

Первичный ключ — потенциальный ключ, который выбран для уникальной идентификации кортежей внутри отношения.

Внешний ключ — атрибут или множество атрибутов внутри отношения, которое соответствует потенциальному ключу некоторого (возможно, того же самого) отношения.

#### *Реляционная целостность*

Определитель **NULL** — значение атрибута в данный момент неизвестно или неприемлемо.

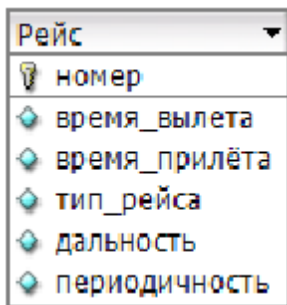
Целостность сущностей — в базовом отношении ни один атрибут первичного ключа не может содержать отсутствующих значений, обозначаемых оператором **NULL**.

Ссылочная целостность — значение внешнего ключа должно соответствовать значению потенциального ключа некоторого кортежа либо задаваться определителем **NULL**.

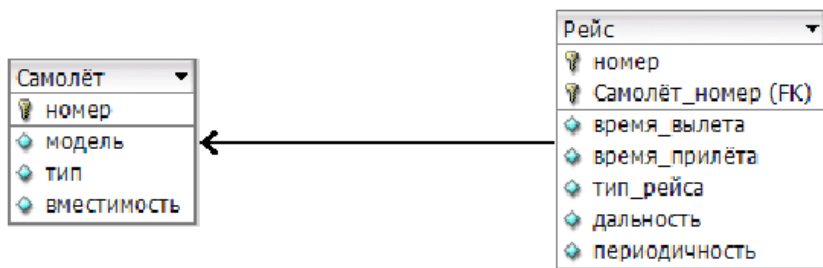
*На диаграмме логической модели базы данных присутствуют следующие элементы.*

Таблицы, представленные в виде прямоугольника, в верхней части которого располагается название таблицы, отчёркнутое линией, а в нижней — список полей таблицы с указанием ключей:





Помимо первичных ключей, рассмотренных нами в ER-моделировании, логическая модель включает в себя внешние, либо вторичные ключи, которые необходимы для реализации связей между таблицами. Связи между таблицами изображаются в виде прямых, ломаных или кривых линий со стрелками, ведущих от внешних к первичным ключам связываемых таблиц:



### Правила преобразования *ER*-диаграммы в логическую модель

Для построения такой диаграммы существуют следующие правила:

1. Сущность становится таблицей с соответствующим именем.
2. Атрибут становится столбцом с таким же именем.
3. Связи типа 1:1 преобразуются одним из следующих вариантов:
  - 3.1. Связанные сущности-таблицы сливаются в одну таблицу.

3.2. В одну из таблиц добавляется столбец – внешний ключ, содержащий ссылку-значение на первичный ключ другой таблицы.

3.3. В обе таблицы добавляются столбцы – внешние ключи, содержащие ссылки на первичные ключи других таблиц.

4. Связи типа 1:N реализуются в виде добавления столбца – внешнего ключа в ту таблицу, которой соответствует N единиц сущностей отношения. Такая таблица называется подчинённой, или дочерней, а таблица, соответствующая одной сущности отношения — родительской или главной.

5. Связи типа M:N реализуются с помощью дополнительно вводимой вспомогательной таблицы, используемой лишь для хранения пар внешних ключей, ссылающихся на первичные ключи связываемых таблиц. Обычно такая таблица получает составное название из названий связываемых ею таблиц вида «Таблица1\_Таблица2».

### ПРИМЕР

**Полная логическая реляционная модель базы данных для задачи «Аэропорт» (смотри лабораторные работы № 1 и № 2) будет выглядеть следующим образом.**

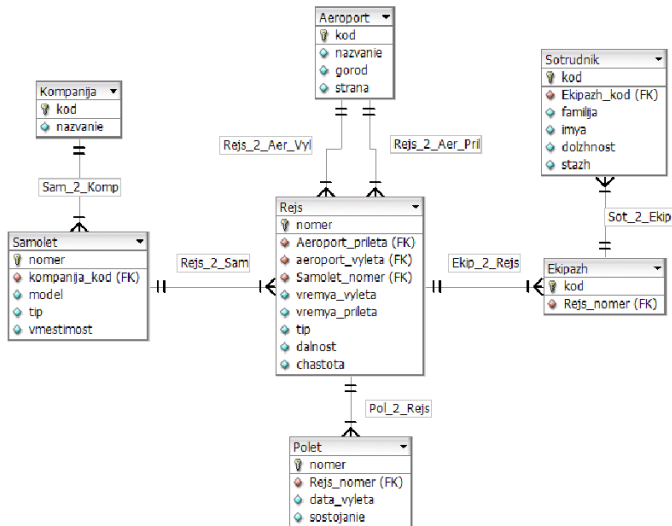


Рис.1 Полная логическая реляционная модель базы данных для задачи «Аэропорт»

## Нормализация логической реляционной модели базы данных

Согласно определению, реляционная база данных представляет собой набор нормализованных отношений (таблиц).

Под нормализованностью понимают соответствие отношений определённому набору правил, а процесс приведения модели БД в нормализованную форму называют нормализацией

*Нормализация* — метод создания набора отношений с заданными свойствами на основе требований к данным, проистекающих из логики организации предметной области.

### *Избыточность данных и аномалии обновления*

Известно, что одну и ту же предметную область можно описать в виде различных логических моделей. Как определить, какая из моделей будет более удачной, а какая — менее? Неправильные модели приводят к ряду проблем при работе с базой данных, созданной на основании такой модели, главными из которых являются неоднозначность и

большой объём лишней работы при обработке, вызванные избыточностью или наоборот, потерей части информации в модели:

1. Аномалии вставки,
2. Аномалии удаления,
3. Аномалия обновления,
4. Свойство соединения без потерь,
5. Свойство сохранения зависимости.

### *Функциональные зависимости*

Функциональная зависимость — описывает связь между атрибутами отношения. Например, если в отношении R, содержащем атрибуты A и B, атрибут B функционально зависит от атрибута A (что обозначается как  $A \rightarrow B$ ), то каждое значение атрибута A связано только с одним значением атрибута B.

*Детерминант* — детерминантом функциональной зависимости называется атрибут или группа атрибутов, расположенная на диаграмме функциональной зависимости слева от символа стрелки.

### *Процесс нормализации*

*Ненормализованная форма* — таблица, содержащая одну или несколько повторяющихся групп данных.

*Первая нормальная форма* — отношение, в котором на пересечении каждой строки и каждого столбца содержится только одно значение.

*Полная функциональная зависимость* — в некотором отношении атрибут В называется полностью функционально зависимым от атрибута А, если атрибут В функционально зависит от полного значения атрибута А и не зависит ни от какого подмножества атрибута А.

*Частичная функциональная зависимость* — такая зависимость  $A \rightarrow B$ , если в А есть некий атрибут, при удалении которого эта зависимость сохраняется.

*Вторая нормальная форма* — отношение, которое находится в первой нормальной форме и каждый атрибут которого, не входящий в состав первичного ключа, характеризуется полной функциональной зависимостью от этого ключа.

*Транзитивная зависимость* — если для атрибутов А, В и С некоторого отношения существуют зависимости вида  $A \rightarrow B$  и  $B \rightarrow C$ , то говорят, что атрибут С транзитивно зависит от атрибута А через атрибут В (при условии, что атрибут А функционально не зависит ни от атрибута В, ни от атрибута С).

*Третья нормальная форма* — отношение, которое находится в первой и второй нормальных формах и не имеет не входящих в первичный ключ атрибутов, которые находились бы в транзитивной функциональной зависимости от этого первичного ключа.

*Нормальная форма Бойса-Кодда (НФБК)* — отношение находится в НФБК тогда и только тогда, когда каждый его детерминант является потенциальным ключом.

## ПРИМЕР

*Последовательно разберём полученные в результате преобразования таблицы (смотри рис1.) на соответствие 1-3 нормальным формам.*

*Приведение модели в 1-ю нормальную форму*

Поскольку ещё в процессе анализа мы избегали использования многозначных атрибутов, то в нашей модели нет таблиц, в ячейках которых хранилось бы более одного значения. Следовательно, наша модель уже находится в 1-й нормальной форме (1НФ).

*Приведение модели во 2-ю нормальную форму*

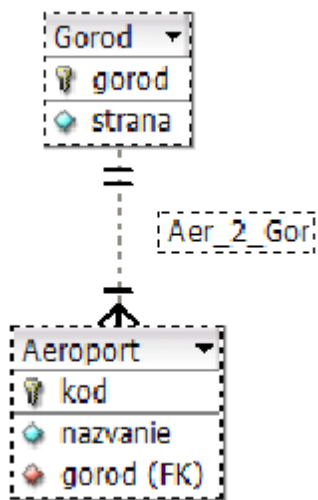
Согласно определению 2НФ, каждый неключевой атрибут каждой таблицы должен функционально зависеть от первичного ключа. Последовательно рассматривая каждую таблицу, убеждаемся, что это действительно так.

*Приведение модели в 3-ю нормальную форму*

Рассмотрим, зависят ли атрибуты таблиц ТОЛЬКО от первичного ключа, или встречаются ещё и зависимости от других, неключевых атрибутов, называемые транзитивными.

Последовательно рассматривая каждую таблицу, обнаруживаем, что в таблице «Аэропорт» наблюдается транзитивная зависимость, а именно — атрибут «страна» зависит не только от первичного ключа, но и от атрибута «город», т.е.

налицо транзитивная зависимость: «код аэропорта»->«город»->«страна». Для её исключения необходимо произвести декомпозицию таблицы «Аэропорт» на две, а именно — вынести повторяющиеся группы «город», «страна» в таблицу «Город»:



Таким образом, мы привели нашу схему в 3-ю нормальную форму: (рис.2)

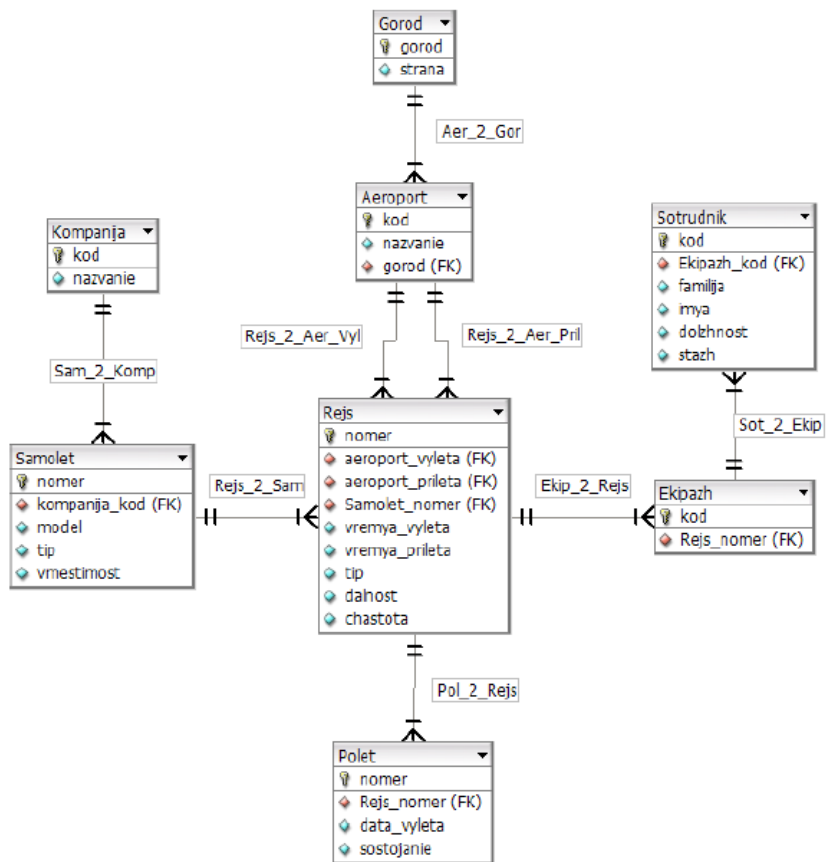


Рис.2 Нормализованная логическая модель

## **ЛАБОРАТОРНОЕ ЗАНЯТИЕ № 4 СОЗДАНИЕ ФИЗИЧЕСКОЙ МОДЕЛИ БД СРЕДСТВАМИ СУБД MICROSOFT ACCESS**

**Цель** -обеспечить студентов конкретным инструментарием для практической реализации методов проектирования приложений клиент-сервер:

1. Научиться создавать и редактировать структуру таблицы в MS Access при помощи Конструктора таблиц, вводить в таблицу данные, а также их модифицировать.

2. Научиться создавать индексы для таблиц базы данных, организовывать простую и сложную сортировку записей таблицы.

### **Физическая модель данных зависит от выбранной СУБД.**

В данном цикле лабораторных работ будут рассмотрено создание физической модели БД средствами СУБД Microsoft Access и сервера баз данных Microsoft SQL Server

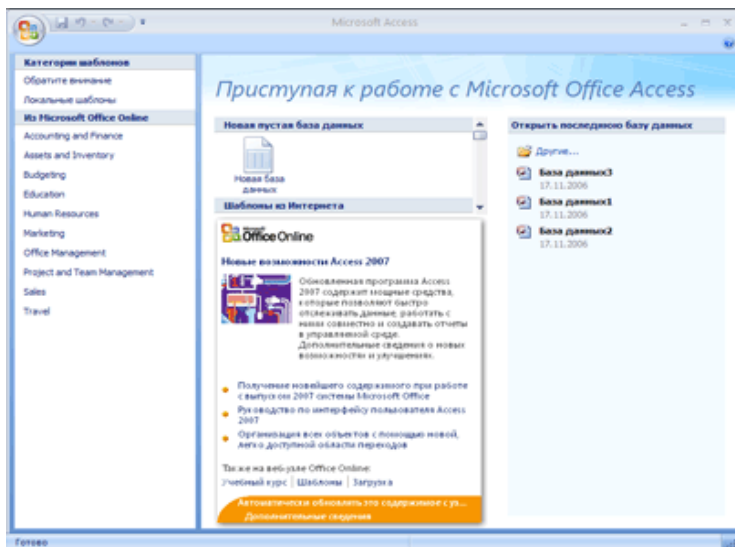
### **ОБРАТИТЕ ВНИМАНИЕ.**

Лабораторная работа состоит из двух разделов. Задание на лабораторную работу приведено в каждом разделе. Отчет по лабораторной работе представить по результатам выполнения задания по разделу № 2.

**РАЗДЕЛ 1. «СОЗДАНИЕ ТАБЛИЦ И ВВОД ИСХОДНЫХ  
ДАННЫХ. РЕДАКТИРОВАНИЕ ДАННЫХ ТАБЛИЦ.»  
«МОДИФИКАЦИЯ СТРУКТУРЫ ТАБЛИЦ»**

### **ИНФОРМАЦИОННАЯ ЧАСТЬ**

**Создание баз данных**



**Рисунок 1- Стартовая страница**

При первом запуске программы Access или закрытии базы данных без выхода из Access отображается окно **Приступая к работе с Microsoft Office Access**.

Это отправной пункт, из которого можно создать новые базы данных, открыть существующую базу данных или ознакомиться с информацией на веб-узле Microsoft Office Online.


### **Создание базы данных**

В приложении Access предусмотрены разнообразные шаблоны, с помощью которых можно быстро создать базу данных. Шаблон – это уже готовая к использованию база данных, включающая все необходимые таблицы, запросы, формы и отчеты для выполнения определенной задачи. Например, предусмотрены шаблоны, которые можно использовать для отслеживания вопросов управления контактами или учета расходов.

Если один из этих шаблонов точно соответствует потребностям, с его помощью обычно проще и быстрее всего создать необходимую базу данных.

В средней части страницы **Приступая к работе с Microsoft Office Access** отобразится несколько шаблонов. Щелкните ссылки в области **Категории шаблонов**, чтобы отобразить другие шаблоны.



Выберите шаблон, который необходимо использовать. В поле **Имя файла** предлагается имя файла для базы данных. Его можно заменить на любое другое имя. Чтобы сохранить эту базу данных в другой папке, отличной от отображаемой под полем имени файла, нажмите кнопку , перейдите к папке, в которой необходимо сохранить базу данных, и нажмите кнопку **ОК**. Нажмите кнопку **Создать** (или **Загрузить**).

Приложение Access создаст или загрузит, а затем откроет базу данных. Отображается форма, в которой можно начать ввод данных.

### **Создание пустой базы данных**

Если использовать шаблон не имеет смысла, можно создать базу данных с нуля. Для этого нужно создать таблицы, формы, отчеты и другие объекты базы данных. В большинстве случаев необходимо выполнить одно или оба следующих действия:

Ввод, вставка или импорт данных в таблицу, которая создана при создании новой базы данных, и последующее повторение этой процедуры для новых таблиц, которые создаются с помощью команды Таблица, расположенной на вкладке Создать.

Импорт данных из других источников и создание новых таблиц в этом процессе.

На странице **Приступая к работе с Microsoft Office Access** в разделе Новая пустая база данных выберите команду Новая база данных.

#### **Новая пустая база данных**



Новая база данных

В области Новая база данных в поле Имя файла введите имя файла. Если имя файла указано без расширения, расширение будет добавлено автоматически. Нажмите кнопку Создать.

### **Создание таблиц**

База данных может включать множество таблиц, в которых хранятся данные по различным темам. Каждая таблица может состоять из множества полей различного типа, включая текст, числа, даты и рисунки.

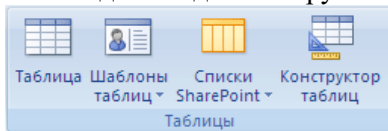
Необходимо тщательно спланировать и правильно разработать базу данных, чтобы впоследствии избежать внесения значительных

изменений.

Таблицу можно создать с помощью создания новой базы данных, вставки таблицы в существующую базу данных, а также импорта или создания ссылки на таблицу из другого источника данных, такого как книга Microsoft Office Excel 2007, документ Microsoft Office Word 2007, текстовый файл или другая база данных. При создании новой базы данных в нее автоматически вставляется новая пустая таблица. Затем можно ввести данные, чтобы начать определение полей.

### **Создание новой таблицы в существующей базе данных**

На вкладке Создание в группе Таблицы щелкните Таблица.



В базу данных вставляется новая таблица, которая открывается в режиме таблицы.

### **Создание таблицы на основе шаблона таблицы**

Для создания таблиц «Контакты», «Задачи», «Вопросы», «События» и «Основные фонды» можно использовать шаблоны таблиц, которые включены в Office Access 2007. Шаблоны таблиц совместимы со списками Службы Microsoft Windows SharePoint Services 3.0 с теми же именами.

На вкладке Создание в группе Таблицы щелкните Шаблоны таблицы и затем выберите из списка один из доступных шаблонов.

Будет вставлена новая таблица на основе выбранного шаблона таблицы.

### **Использование импорта или связи для создания таблицы**

Для создания таблицы можно использовать импорт данных, сохраненных в другом месте, или связь с ними. Например, можно использовать импорт данных или связь с данными электронной таблицы Excel, списка SharePoint, файла XML, другой базы данных Access, папки Microsoft Office Outlook 2007 и ряда других источников. При импорте данных в новой таблице текущей базы данных создается копия этих данных. Напротив, при использовании связи с данными в текущей базе данных создается таблица, динамически связанная с данными, хранящимися в другом месте.

Таким образом, данные, изменяемые в связанной таблице, изменяются и в источнике исходных данных. Когда данные в источнике изменяются с помощью другой программы, это изменение будет отображаться и в связанной таблице.

Для создания таблиц таким способом на вкладке Внешние данные в группе Импорт выберите один из доступных источников данных.



Следуйте инструкциям в диалоговых окнах.

Будет создана новая таблица, имя которой появится в области переходов.

### **Создание таблицы на основе списка SharePoint**

Для создания таблицы базы данных можно использовать импорт списка SharePoint или связь с ним. Изучите при помощи справки MS Access 2007 самостоятельно создание таблиц этим способом.

### **Добавление полей в таблицу в режиме таблицы**

В полях (также называемых столбцами) сохраняются элементы отслеживаемых данных.

Поля определяются рядом характеристик. Например, у каждого поля есть имя, однозначно определяющее это поле в таблице. Кроме того, у поля есть тип данных, выбираемый в соответствии с сохраняемыми в этом поле данными. Этот тип данных определяет значения, которые можно сохранить, и операции, которые можно выполнить с данными, а также объем памяти, выделяемой для каждого значения. С каждым полем также связана группа параметров, называемых свойствами, которые определяют внешний вид и функциональные характеристики этого поля. Например, свойство «Формат (Format)» определяет структуру отображения данных в поле, то есть вид, в котором они должны отображаться.

Создаваемая новая таблица открывается в режиме таблицы. Для быстрого добавления нового поля можно ввести данные в столбец Добавить поле.

Состояние заказа			
	Код состояния ▾	Название состояния ▾	Добавить поле
+	0	Создать	1
+	1	Выписан счет	
+	2	Отгружен	
+	3	Закрыт	

### Добавление нового поля в новую таблицу

На вкладке Создание в группе Таблицы щелкните Таблица.

Приложение Access вставляет новую таблицу в базу данных и открывает ее в режиме таблицы.

Введите данные в ячейку под заголовком столбца Добавить поле.

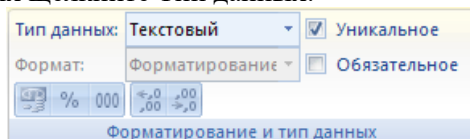
При вводе данных в новый столбец Office Access 2007 автоматически распознает соответствующий тип данных для этого поля. Например, при вводе в столбец даты «1.01.2017» Office Access 2007 распознает, что введена дата, и задает для этого поля тип данных «Дата/время». Если на основании введенных данных приложение Access не может точно определить тип данных, задается тип данных «Текстовый».

Кроме определения типа данных, Office Access 2007 может задавать значение для свойства Формат (Format) в зависимости от вводимых данных. Например, если ввести 10:50, будет задан тип данных «Дата/время» и значение «Средний формат времени» для свойства Формат (Format).

Чтобы явным образом задать тип данных и формат для поля, переопределив тип, назначенный Office Access 2007, используйте команды в группе Форматирование и тип данных на вкладке Режим таблицы.

### Явное задание типа данных

На вкладке Режим таблицы в группе Форматирование и тип данных щелкните Тип данных.



Выберите нужный тип данных.

## Явное задание формата

На вкладке Режим таблицы в группе Форматирование и тип данных щелкните Формат.

Выберите нужный формат. Если поле добавляется путем ввода данных в ячейке под заголовком Добавить поле, Office Access 2007 автоматически назначает имя этому полю: «Поле1» для первого поля, «Поле2» для второго поля и т. д. Рекомендуется использовать описательные имена полей. Для изменения имени поля щелкните его заголовок правой кнопкой мыши и выберите в контекстном меню команду Переименовать столбец.

Имена полей могут содержать до 64 знаков (цифр или букв), включая пробелы. Рекомендуется присваивать полям описательные имена, чтобы легко различать их при просмотре или изменении записей.

**Таблица 1-Типы данных**

<b>Тип данных</b>	<b>Хранит</b>	<b>Размер</b>
Текстовый	Алфавитно-цифровые знаки Используется для текста или текста и чисел, не применяемых в расчетах (например, кода товара).	До 255 знаков.
Поле MEMO	Алфавитно-цифровые знаки (более 255 знаков) или форматированный текст. Используется для текста длиннее 255 знаков или форматированного текста. Типичные примеры использования поля типа «Поле MEMO» — примечания, длинные описания и абзацы с полужирным шрифтом или курсивом.	До 1 гигабайта знаков или 2 гигабайт памяти (2 байта на знак), из которых в элементе управления можно отобразить 65 535 знаков.
Числовой	Числовые значения (целые или дробные). Используется для хранения числовых данных, используемых в вычислениях, за исключением денежных значений (для	1, 2, 4 и 8 байт

Тип данных	Хранит	Размер
	денежных значений используется тип данных «Денежный»).	
Дата/время	Даты и время. Используется для хранения значений даты и времени. Обратите внимание, что каждое сохраняемое значение содержит одновременно и компонент даты, и компонент времени.	8 байт.
Денежный	Денежные значения. Используется для хранения денежных значений (валюты).	8 байт.
Счетчик	Уникальное числовое значение, которое автоматически вводит Office Access 2007 при добавлении записи. Используется для создания уникальных значений, например для первичного ключа. Обратите внимание, что в поле с типом данных «Счетчик» числа могут последовательно увеличиваться на указанное приращение или выбираться случайно.	4 байта
Логический	Логические значения. Используется для полей, которые могут содержать одно из двух значений, например «Да» и «Нет» или True и False.	1 бит (8 бит = 1 байт).
Поле объекта OLE	OLE-объекты или другие двоичные данные. Используется для хранения OLE-объектов других приложений Microsoft Windows.	До 1 Гбайт.
Вложение	Рисунки, изображения, двоичные файлы, файлы Microsoft Office. Стандартный тип данных для сохранения цифровых	Для сжатых вложений — 2 гигабайта. Для несжатых

Тип данных	Хранит	Размер
	изображений и любого типа двоичных файлов.	вложений примерно 700 Кбайт в зависимости от степени возможного сжатия вложения.
Гиперссылка	Гиперссылки. Используется для хранения гиперссылок вызова веб-страниц одним щелчком с помощью URL-адреса или файлов с помощью формата универсального имени UNC. Кроме того, можно использовать ссылку на объекты Access, хранящиеся в базе данных.	До 1 гигабайта знаков или 2 гигабайт памяти (2 байта на знак), из которых в элементе управления можно отобразить 65 535 знаков.
Мастер подстановок	Фактически типом данных не является, а вызывает мастер подстановок. Используется для запуска мастера подстановок, с помощью которого можно создать поле, позволяющее выбрать значение из другой таблицы, запроса или списка значений, используя поле со списком.	На основе таблицы или запроса — размер привязанного столбца. На основе значения — размер текстового поля, содержащего значение.

Для телефонных, инвентарных и других номеров, которые не используются в математических вычислениях, вместо числового необходимо выбрать текстовый тип данных.

Для текстового и числового типа данных можно более точно указать тип данных и размер поля с помощью значения поля

свойства Размер поля (FieldSize)

### **Сохранение таблицы**

После добавления полей в таблицу необходимо сохранить ее структуру. При первом сохранении новой таблицы необходимо присвоить имя, описывающее содержащиеся в ней данные. Можно использовать до 64 знаков (букв или цифр), включая пробелы. Например, можно назвать таблицу «Клиенты», «Перечень запасных частей» или «Товары».

### **Определение первичного ключа**

Первичный ключ таблицы состоит из одного или нескольких полей, однозначно определяющих каждую строку в этой таблице. Часто в качестве первичного ключа используется уникальный инвентарный номер, порядковый номер или код.

В качестве первичного ключа удобно использовать поле, которое обладает следующими характеристиками. Во-первых, оно должно однозначно определять каждую строку. Во-вторых, оно не должно быть пустым — в нем всегда должно быть значение. В-третьих, оно должно изменяться крайне редко (лучше всего — никогда). Приложение Access использует поля первичного ключа для быстрого объединения данных нескольких таблиц.

Всегда следует определять для таблицы первичный ключ. Для первичного ключа автоматически создается индекс, ускоряющий выполнение запросов и операций. Кроме того, приложение Access проверяет наличие и уникальность значений в поле первичного ключа.

При создании новой таблицы в режиме таблицы Access автоматически создает первичный ключ и назначает полю имя «Код» и тип данных «Счетчик». Это поле скрыто в режиме таблицы, для его просмотра необходимо переключиться в режим конструктора.

Переключение в режим конструктора:

- Щелкните правой кнопкой мыши вкладку документа и выберите команду Конструктор.
- Щелкните правой кнопкой мыши имя таблицы в области переходов и выберите команду Конструктор.
- Нажмите кнопку Конструктор в строке состояния Access.

Для изменения или удаления первичного ключа и для задания первичного ключа таблицы, у которой его еще нет, необходимо



использовать режим конструктора.

### Определение или изменение первичного ключа

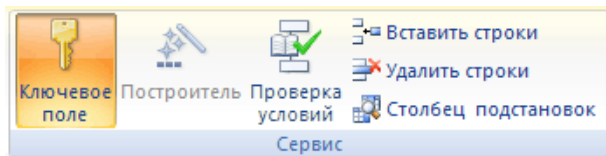
Откройте таблицу в режиме конструктора.

Выберите поле или поля, которые требуется использовать в качестве первичного ключа.

Чтобы выделить одно поле, щелкните область выделения строки нужного поля.

Для выделения нескольких полей щелкните область выделения каждого поля, удерживая нажатой клавишу CTRL.

На вкладке Структура в группе Элементы нажмите кнопку ключевое поле.



Индикатор ключа будет добавлен слева от поля или полей, определенных как первичный ключ.

### Удаление первичного ключа

Откройте таблицу в режиме конструктора.

Щелкните область выделения строки для текущего первичного ключа.

Если первичный ключ состоит из одного поля, щелкните область выделения для этого поля.

Если первичный ключ состоит из нескольких полей, выделите все поля первичного ключа.

На вкладке Структура в группе Элементы нажмите кнопку ключевое поле.

### Задание свойств полей

С помощью значений свойств полей можно управлять отображением данных, предотвращать ввод неверных значений, задавать значения по умолчанию, ускорять поиск и сортировку, а также управлять другими функциональными характеристиками и внешним видом полей.

Таблица 2- Свойства полей

Свойство поля	Используется
---------------	--------------

<b>Свойство поля</b>	<b>Используется</b>
Размер поля (FieldSize)	Задания максимального размера данных, сохраняемых в полях с типом данных Текстовый, Числовой или Счетчик.
Формат	Настройки формата данных поля для отображения или печати.
Число десятичных знаков	Задания количества отображаемых знаков в дробной части для числовых значений.
Новые значения	Определения способа присвоения значений для поля «Счетчик»: последовательное увеличение или случайные значения.
Маска ввода	Отображения специальных знаков для управления вводом данных.
Подпись	Определения текста, отображаемого по умолчанию в надписях для форм, отчетов или запросов.
Значение по умолчанию	Автоматического назначения значения по умолчанию полю при добавлении новых записей.
Условие на значение	Задания выражения, которое должно быть истинно при каждом добавлении или изменении значения в этом поле.
Сообщение об ошибке	Ввода текста, который будет отображаться при нарушении значением правила «Условие на значение».
Обязательное	Требования обязательного ввода данных в поле.
Пустые строки	Разрешения ввода пустой строки (путем задания значения «Да») в поле с типом данных «Текстовый» или «Поле МЕМО».
Индексированное поле	Ускорения доступа к данным в этом поле путем создания и использования индекса.
Сжатие Юникод	Сжатия текста, если в этом поле содержится > 4 096 знаков.
Режим IME	Управления преобразованием знаков в восточноазиатских версиях Windows.
Режим предложений IME	Управления преобразованием знаков в восточноазиатских версиях Windows.
Смарт-теги	Добавления смарт-тега к этому полю.

<b>Свойство поля</b>	<b>Используется</b>
Только добавление	Включения управления версиями (путем установки значения «Да») для поля «Поле МЕМО».
Формат текста	Выбора «Формат RTF» для хранения текста в виде HTML с разрешением форматирования. Чтобы сохранить только текст, выберите «Обычный текст».
Выравнивание текста	Задания выравнивания текста по умолчанию в элементе управления.
Точность	Задания суммарного допустимого числа сохраняемых знаков по обе стороны от десятичного разделителя.
Масштаб	Задания максимально допустимого числа сохраняемых знаков справа от десятичного разделителя.

**Таблица 3- Свойство "Размер поля"**

<b>Тип данных</b>	<b>Описание</b>
Текстовый	Введите значение от 1 до 255. Текстовые поля могут содержать от 1 до 255 знаков. Для сохранения текста большого объема используется поле с типом данных «Поле МЕМО».
Числовой	Выберите одно из следующих значений: Байт. Для числовых значений в диапазоне от 0 до 255 знаков. Целое. Для числовых значений в диапазоне от - 32 768 до +32 768. Длинное целое. Для числовых значений в диапазоне от -2 147 483 648 до +2 147 483 647. Одинарное с плавающей точкой. Для числовых значений с плавающей запятой в диапазоне от -3,4 x 10 <sup>38</sup> до +3,4 x 10 <sup>38</sup> , включающих до семи значащих разрядов. Двойное с плавающей точкой. Для числовых значений с плавающей запятой в диапазоне от -1,797 x 10 <sup>308</sup> до +1,797 x 10 <sup>308</sup> Действительное. Для числовых значений в диапазоне от -9,999 x 10 <sup>27</sup> до +9,999 x 10 <sup>27</sup> .

Тип данных	Описание
Счетчик	<p>Выберите одно из следующих значений:</p> <p>Длинное целое Для уникальных числовых значений в диапазоне от 1 до +2 147 483 648, если для свойства поля «Новые значения» задано значение «Последовательные», и от -2 147 483 648 до +2 147 483 647, если для свойства поля «Новые значения» задано значение «Случайные».</p>

**Таблица 4- Свойство "Формат поля"**

Тип данных	Описание
Текстовый	<p>Можно задать свой формат. Дополнительные сведения см. в разделе Форматирование данных в таблицах, формах и отчетах.</p>
Числовой	<p>Выберите одно из следующих значений:</p> <p>Основной. Число отображается в том виде, в каком было введено. Например, 3456,789 отображается как 3456,789.</p> <p>Денежный. При отображении числа используется разделитель групп разрядов и параметры, обозначения денежной единицы, разделителя целой и дробной части и количества дробных знаков. Например, 3456,789 отображается как 3456,79р.</p> <p>Евро. При отображении числа используется обозначение денежной единицы евро.</p> <p>Фиксированный. Отображается хотя бы одна цифра и применяются параметры для отрицательных значений, обозначения денежной единицы, разделителя целой и дробной части и количества дробных знаков. Например, 3456,789 отображается как 3456,79.</p> <p>Стандартный. При отображении числа используется разделитель групп разрядов и параметры для отрицательных значений,</p>

Тип данных	Описание
	<p>разделителя целой и дробной части и количества дробных знаков. В этом формате обозначение денежной единицы не используется.</p> <p>Например, 3456,789 отображается как 3 456,79.</p> <p>Процентный. Значение умножается на 100, и к полученному значению добавляется знак процентов. Например, 0,3456 отображается как 35%</p> <p>Научный. Значение отображается в стандартном экспоненциальном представлении.</p> <p>Например, значение 3 456,789 отображается как 3,46E+03</p>
Дата/время	<p>Выберите один из следующих встроенных форматов отображения:</p> <p>Полный формат даты. При отображении значения используется сочетание форматов «Краткий формат даты» и «Длинный формат времени».</p> <p>Длинный формат даты. При отображении значения используется полный формат даты, заданный в компоненте «Язык и региональные стандарты» панели управления.</p> <p>Средний формат даты. При отображении значения используется формат дд-ммм-гг (например, 14-июл-06).</p> <p>Краткий формат даты. При отображении значения используется краткий формат даты, заданный в компоненте «Язык и региональные стандарты» панели управления.</p> <p>Длинный формат времени. При отображении значения используется формат времени, заданный в компоненте «Язык и региональные стандарты» панели</p>

Тип данных	Описание
	<p>управления.</p> <p>Средний формат времени. При отображении значения используется формат ЧЧ:ММ РМ, где ЧЧ — часы, ММ — минуты и РМ — АМ или РМ. Значение часов может находиться в диапазоне от 1 до 12. Значение минут может находиться в диапазоне от 0 до 59.</p> <p>Краткий формат времени. При отображении значения используется формат ЧЧ:ММ, где ЧЧ — часы, ММ — минуты. Значение часов может находиться в диапазоне от 1 до 23, значение минут — в диапазоне от 0 до 59.</p>
Да/Нет	<p>Выберите одно из следующих значений: Истина/Ложь. Отображается значение «Истина» или «Ложь».</p> <p>Да/Нет . Отображается значение «Да» или «Нет».</p> <p>Вкл/Выкл. Отображается значение «Вкл.» или «Выкл.».</p> <p>Примечание. В вышеперечисленном значения «Вкл.», «Истина» и «Да» эквивалентны. Также эквивалентны значения «Ложь», «Нет» и «Выкл.».</p>

### Свойство Маска ввода

Свойство **Маска ввода (InputMask)** облегчает ввод данных и определяет значения, которые можно ввести в элемент управления поля. Значение свойства **Маска ввода** может содержать до трех разделов, разделенных точкой с запятой.

**Таблица 5 Маска ввода**

Раздел	Описание
Первый	Представляет саму маску ввода (например, !(999) 000-0000). Перечень знаков, используемых для определения масок ввода, приводится ниже в таблице.
Второй	<p>Определяет режим занесения в таблицу текстовых констант добавляемых к знакам, вводимым пользователем.</p> <p><b>0</b> в данном компоненте указывает, что текстовые</p>

	<p>константы (например, скобки и дефисы в маске ввода телефонных номеров) сохраняются вместе с введенными пользователем значениями;</p> <p><b>1</b> или пустое значение данного раздела указывает, что сохраняются только знаки, введенные пользователем.</p>
Третий	<p>Определяет знак, используемый для изображения пустых позиций в маске ввода, в которые помещаются вводимые пользователем знаки. В этом разделе можно указать любой знак; чтобы отобразить пустую строку, введите пробел, заключенный в кавычки (" ").</p>

При создании маски ввода можно использовать специальные знаки, показывающие что текущие данные нужно обязательно ввести (например, региональный код для телефонных номеров), а другие данные являются необязательными (например, добавочный номер телефона). Эти знаки определяют тип данных, например число или знак, которые необходимо ввести для каждого знака маски ввода. С помощью следующих знаков можно определить маску ввода.

**Таблица 6-Символы маски ввода**

<b>Символ</b>	<b>Описание</b>
0	Цифра (0-9, обязательный знак; знаки (+) и (-) не разрешены).
9	Цифра или пробел (необязательный знак; знаки (+) и (-) не разрешены).
#	Цифра или пробел (необязательный знак; незаполненные позиции выводятся как пробелы в режиме редактирования, но удаляются при сохранении данных; знаки (+) и (-) не разрешены).
L	Буква (от А до Я, обязательный знак).
?	Буква (от А до Я, необязательный знак).
A	Буква или цифра (обязательный знак).
a	Буква или цифра (необязательный знак).
&	Любой знак или пробел (обязательный знак).
C	Любой знак или пробел (необязательный знак).
., : ; - /	Десятичный разделитель, разделители групп разрядов, времени или даты.
<	Преобразует все знаки к нижнему регистру.
>	Преобразует все знаки к верхнему регистру.

Символ	Описание
!	Указывает заполнение маски ввода справа налево, а не слева направо. Восклицательный знак в маске ввода можно помещать в любую позицию.
\	Указывает, что следующий знак будет отображаться как текстовая константа .

- Свойство Подпись (Caption) определяет текст, который выводится в подписях объектов в различных режимах.

- Свойство Значение по умолчанию (DefaultValue) Определяет значение типа String, которое автоматически вводится в поле при создании новой записи.

### **ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ**

1. Создать новую базу данных и сохранить в личном каталоге на своем флэш-накопителе. **(ВАМ работать с этой базой до конца семестра).**

2. В БД определить три новые таблицы со структурой полей согласно номеру варианта задания.

3. Определить требуемые индексы для таблиц БД (первичный ключ и вторичные ключи).. Установить связи между таблицами БД.

4. Внести в таблицы БД по 10-15 записей, содержание которых соответствует семантике и типу полей.

### **КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Какими различными способами можно создавать таблицы в MS Access 2007?

2. Какие типы данных определены в MS Access 2007?

3. Для каких целей применяется свойство поля «Маска ввода»?

4. Какие поля нужно задавать как обязательные?

5. Для чего нужно применять индексирование полей?

6. Опишите создание таблиц в MS Access 2007 при помощи списков Share Point.

7. Для чего нужны шаблоны таблиц?

8. Какими характеристиками определяются поля?

9. При вводе данных в новый столбец, какой тип данных присваивается полю.



## РАЗДЕЛ 2. «ИНДЕКСИРОВАНИЕ И СОРТИРОВКА ТАБЛИЦ»

### ИНФОРМАЦИОННАЯ ЧАСТЬ

#### Индексирование полей и записей в базе данных Access

При помощи индексов ускоряется сортировка и поиск записей. Индексы таблиц Microsoft Access используются так же, как и предметные указатели в книгах: при поиске данных выполняется их поиск в индексе. Индексы можно создавать по одному или нескольким полям. Составные индексы позволяют пользователю различать записи, в которых первые поля могут иметь одинаковые значения.

#### Выбор полей для индексирования

В основном, требуется индексировать поля, в которых часто осуществляется поиск, поля сортировки или поля, объединенные с полями из других таблиц в запросах. Однако индексы могут замедлить выполнение некоторых запросов на изменение, например, запросов на добавление, при выполнении которых требуется обновление индексов многих полей.

Поля первичного ключа таблиц индексируются автоматически, а поля с типом данных «Поле объекта OLE» индексировать нельзя. Для остальных полей индексирование используется, если выполняются следующие условия.

- Поле имеет тип данных «Текстовый», «Числовой», «Денежный» или «Дата/время».
- Предполагается выполнение поиска значений в поле.
- Предполагается выполнение сортировки значений в поле.
- Предполагается выполнение сортировки большого числа различных значений в поле. Если поле содержит много одинаковых значений, то применение индекса незначительно ускорит выполнение запросов.

#### Составные индексы

Если предполагается частое выполнение одновременной сортировки или поиска в нескольких полях, можно создать для этих полей составной индекс. Например, если в одном и том же запросе часто задаются условия для полей «Имя» и «Фамилия», то для этих

двух полей имеет смысл создать составной индекс.

При сортировке таблицы по составному индексу Microsoft Access сначала выполняет сортировку по первому полю, определенному для данного индекса. Если в первом поле содержатся записи с повторяющимися значениями, то выполняется сортировка по второму полю, определенному для данного индекса, и так далее. В составной индекс можно включить до 10 полей.

### Создание индекса

Перед созданием индекса необходимо решить, следует ли создать индекс для одного поля или составной индекс. Индекс для одного поля создается с помощью установки свойства Индексированное поле. В следующей таблице приведены возможные параметры свойства Индексированное поле.

**Таблица 1- Параметры свойства "Индексированное поле"**

Параметр свойства	Значение
Нет	Не создавать индекс для этого поля (или удалить существующий индекс)
Да (Допускаются совпадения)	Создать индекс для этого поля
Да (Совпадения не допускаются)	Создать уникальный индекс для этого поля

При создании уникального индекса невозможно ввести новое значение в определенном поле, если такое значение уже существует в том же поле другой записи. В Access уникальный индекс автоматически создается для первичных ключей, однако может понадобиться, чтобы создание значений, совпадающих со значениями в других полях, было невозможным. Например, можно создать уникальный индекс для поля, в котором содержатся серийные номера, чтобы двум продуктам не мог быть присвоен один и тот же серийный номер.

### Создание индекса для одного поля

1. В области переходов щелкните правой кнопкой мыши название таблицы, в которой необходимо создать индекс, затем в контекстном меню выберите Конструктор.

2. Щелкните Имя поля для поля, которое следует индексировать.
3. В разделе Свойства поля откройте вкладку Общие.
4. В свойстве Индексированное поле щелкните значение Да (Допускаются совпадения), если следует разрешить повторяющиеся значения, или значение Да (Совпадения не допускаются), чтобы создать уникальный индекс.
5. Чтобы сохранить изменения, щелкните Сохранить на панели быстрого доступа или нажмите сочетание клавиш CTRL+S.

### **Создание составного индекса**

Чтобы создать составной индекс, необходимо включить в него строку для каждого поля в индексе и поместить индекс только в самой первой строке. Все строки обрабатываются как часть одного индекса до тех пор, пока не будет обнаружена строка с другим названием индекса. Чтобы вставить строку, щелкните правой кнопкой мыши место, куда следует вставить строку, затем в контекстном меню щелкните команду Вставить строки.

1. В области переходов щелкните правой кнопкой мыши название таблицы, в которой необходимо создать индекс, затем в контекстном меню выберите Конструктор.

2. На вкладке Конструктор в группе Показать или скрыть щелкните Индексы.

Появится окно «Индексы». Измените размеры этого окна, чтобы отображались пустые строки и свойства индекса.

3. В первой пустой строке столбца Индекс введите имя индекса. Для индекса можно использовать либо имя одного из индексируемых полей, либо другое подходящее имя.

4. В столбце Имя поля щелкните стрелку, затем щелкните первое поле, которое следует использовать в индексе.

5. Следующую строку столбца Индекс оставьте пустой, затем в столбце Имя поля укажите второе индексируемое поле. Повторите этот шаг для всех полей, которые необходимо включить в индекс.

6. Чтобы изменить порядок сортировки значений полей, в столбце Порядок сортировки окна «Индексы» щелкните По возрастанию или По убыванию.

7. В окне Свойства индекса окна Индексы установите свойства индекса для строки в столбце Имя индекса, содержащем индекс. Установите свойства в соответствии со следующей таблицей.

**Таблица 2-Свойства индексов**

<b>Надпись</b>	<b>Значение</b>
Первичный	Если Да, то индекс является первичным ключом.
Уникальный	Если Да, то каждое индексируемое значение должно быть уникальным.
Пропуск пустых полей	Если Да, то записи с пустыми значениями в индексируемых полях будут исключены из индекса.

8. Чтобы сохранить изменения, щелкните Сохранить на панели быстрого доступа.

### **Сортировка записей**

Существует два вида сортировки, которые можно выполнить: простая сортировка и сложная сортировка.

- **Простая сортировка.** При сортировке в режиме формы, в режиме таблицы выполняется простая сортировка, то есть все записи поля сортируются по возрастанию или по убыванию (но не в том и другом порядке сортировки одновременно).

- **Сложная сортировка.** Если нужно провести сортировку записей в режиме конструктора запроса, в окне расширенного фильтра, в режиме конструктора отчета, в режиме конструктора страницы, в режиме сводной диаграммы или сводной таблицы, можно выполнить сложную сортировку. Это означает, что по некоторым полям допускается сортировка по возрастанию, а по другим полям сортировка по убыванию.

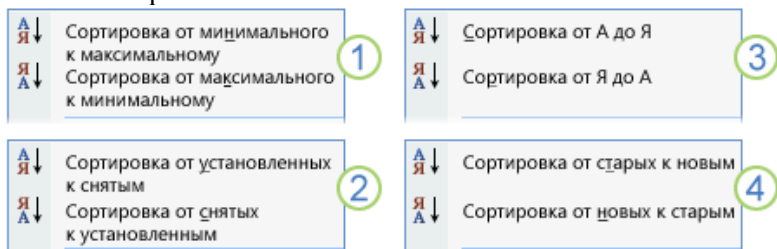
### **Сортировка таблицы, запроса или формы**

1. Укажите поля для сортировки. Чтобы выполнить сортировку по двум или более полям, укажите, какие из полей будут использоваться в качестве внутренних и внешних полей сортировки.

2. Щелкните правой кнопкой мыши столбец или элемент управления, соответствующий внутреннему полю, и выберите одну из команд сортировки. Команды зависят от типа данных, содержащихся в выбранном поле.

Примечание. При сортировке данных по полю с логическим типом данных значения «Да», «Истина» или «Включено» считаются «выбранными», а значения «Нет», «Ложь» или «Отключено» — «снятыми». По умолчанию этот тип поля отображается в виде флажка, но пользователь может настроить отображение поля в виде

текстового поля или поля со списком. При смене вида отображения поля на текстовое поле или поля со списком сортировка происходит по признакам «выбрано» или «снято».



- 1 Числовой, Денежный, Счетчик
- 2 Текстовый, Поле МЕМО, Гиперссылка
- 3 Логический
- 4 Дата/время

3. Повторите предыдущий шаг для каждого поля сортировки, включая последнее внешнее поле сортировки.

Записи переупорядочиваются в соответствии с порядком сортировки.

Если текстовое поле содержит значения Null и пустые строки, при сортировке по возрастанию сначала отображаются записи со значением Null, потом записи с пустыми строками, а затем записи с непустыми значениями.

Если значение в поле начинается со специального знака, такого как дефис, скобки или другого символа, при сортировке по возрастанию соблюдаются следующие правила:

- Значения, начинающиеся с пробела, отображаются перед алфавитно-цифровыми значениями.
- Значения в скобках отображаются после значений, начинающихся с пробелов, но перед алфавитно-цифровыми значениями.
- Значения, начинающиеся со знака «минус» (-), отображаются перед значениями со знаком «плюс» (+).
- Для всех других знаков порядок сортировки определяется на основе кодов ASCII этих знаков. Например, для знака доллара (\$) используется код 36, а для знака равенства (=) — 61, поэтому значения, начинающиеся с \$, отображаются перед значениями, начинающимися с =.

Для переопределения этого порядка можно проигнорировать первый знак для всех значений в этом поле. Этот метод удобно

использовать, если значения в поле всегда начинаются с одного специального знака, например знака «минус (-)», или с одинакового количества специальных знаков — то есть заранее известно, сколько знаков игнорировать. Если количество знаков, которое необходимо игнорировать, изменяется, можно определить специальный (пользовательский) порядок сортировки.

Имейте в виду, что нельзя удалить порядок сортировки только в одном поле. Чтобы отменить сортировку во всех полях сортировки, на вкладке Главная в группе Сортировка и фильтр нажмите кнопку Очистить все сортировки, а затем примените необходимый порядок сортировки.

#### Дополнительные сценарии сортировки

Если применяются команды сортировки, для определения порядка записей используется тип данных для поля и полные значения каждого поля. Однако иногда требуется отсортировать текстовые значения, такие как понедельник, вторник и т.д. не в алфавитном, а в специальном порядке, или отсортировать текстовое поле, содержащее IP-адреса. Если требуются особые способы сортировки, для которых не предусмотрены команды сортировки, см. следующие разделы:

- Если новая форма или отчет основываются на таблице или запросе, порядок сортировки которых был сохранен вместе с ними, то он наследуется и новой формой или отчетом.

- В одном или нескольких полях результатов запроса или расширенного фильтра может быть отсортировано до 255 знаков.

- Если бланк запроса или фильтра содержит знак «звездочка» из списка полей, то определить в нем порядок сортировки можно, только добавив в него поля, которые необходимо отсортировать.

- Для сортировки значений дат и времени от более ранних к более поздним используйте порядок сортировки по возрастанию. Для сортировки от более поздних значений к более ранним используйте сортировку по убыванию.

- Числа, хранящиеся в текстовых полях, сортируются как строки знаков, а не как числовые значения. Поэтому для выполнения их сортировки в числовом порядке все текстовые строки должны иметь одинаковую длину. Например, результатом сортировки по возрастанию текстовых строк «1», «2», «11» и «22» будет «1», «11», «2», «22». В начало строк с меньшим количеством знаков следует

добавить незначащие нули, например: «01», «02», «11», «22». Еще одним решением данной проблемы для полей, не содержащих значения **Null**, будет использование функции **Val** для сортировки числовых значений строк. Например, если столбец «Возраст» является текстовым полем, содержащим числовые значения, то для расположения записей этого столбца в должном порядке можно указать в ячейке **Поле** функцию **Val([Возраст])**, а в ячейке **Сортировка** указать нужный порядок сортировки. Если числовые значения или значения дат хранятся только в текстовом поле, рекомендуется изменить тип данных этого поля таблицы на числовой, денежный или даты/времени. После выполнения сортировки по этому полю числа или даты будут располагаться в надлежащем порядке без ввода дополнительных нулей.

### **ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ (РАЗДЕЛ 2.)**

1. Отсортируйте таблицу по убыванию одного из атрибутов. Скопируйте полученную таблицу в документ MS Word.
2. Отсортируйте таблицу по возрастанию одного из атрибутов. Скопируйте полученную таблицу в документ MS Word.
3. Добавьте в таблицу несколько записей:
4. Переиндексируйте таблицу (**Конструктор- Показать или скрыть-Индексы**) следующим образом, чтобы сортировка и поиск записей сначала осуществлялись по заданному атрибуту (по возрастанию), затем по заданному атрибуту (по возрастанию), и в последнюю очередь по заданному атрибуту (по убыванию).
5. В конструкторе таблиц для таблицы отмените Первичный ключ таблицы.

### **КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Что такое индекс и для чего применяется индексирование полей?
2. Какие поля таблицы индексируются автоматически?
3. Какие типы данных нельзя индексировать?
4. В каком порядке выполняется сортировка записей в сложных индексах?
5. Сколько существует видов сортировки, и в чем их отличие?
6. Каким образом сортируются записи типа Дата\время?
7. Как сортируются числа, хранящиеся в текстовых полях?
8. Каким образом сортируются пустые записи?

## ЛАБОРАТОРНАЯ РАБОТА № 5 СОЗДАНИЕ SQL ЗАПРОСОВ В СРЕДЕ СУБД MICROSOFT ACCESS

**Цель** –изучение понятия запроса к базе данных, выборки данных, классификации запросов, назначения и синтаксиса языка структурированных запросов SQL, а также практика создания запросов различной сложности в режиме SQL.

### **Задание на лабораторную работу**

1. Изучить лекционный материал по теме «Язык структурированных запросов» («Операторы запросов»), теоретический материал к лабораторной работе («Основы SQL»).

2. Создать SQL -запросы, обеспечивающие возможность выполнения запросов к базе данных, согласно полученному варианту(варианты задания получить у преподавателя).

3. В отчете представить скриншоты полученных SQL-конструкций и результатов запуска в виде таблицы . Отчет представить в письменном виде.

<b>Описание запроса (согласно полученному варианту)</b>
<b>SQL-конструкция для создания запроса</b>
<b>Результат запуска</b>

4. Ответить письменно на контрольные вопросы.

### **Контрольные вопросы**

1. Перечислите способы создания запросов в среде СУБД Microsoft Access. Охарактеризуйте достоинства и недостатки каждого из них.

### **Порядок выполнения работы**

1. Для создания нового SQL- запроса в окне базы данных нажмите на кнопку «Создание запроса в режиме конструктора» и не выбирая таблицы для запроса нажмите кнопку Вид на панели инструментов и выберите SQL.

2. Используя команду SELECT, создайте запрос на выборку записей согласно полученному заданию.



**ЛАБОРАТОРНАЯ РАБОТА № 6**  
**СОЗДАНИЕ SQL ЗАПРОСОВ В СРЕДЕ СУБД**  
**MICROSOFT ACCESS**  
**(создание, обновление, добавление, удаление)**

**Цель** – создать SQL-запросы на создание таблицы, на обновление записей, на удаление записей, на добавление данных, на удаление таблицы, на создание индексов.

**Задание на лабораторную работу**

1. Изучить лекционный материал по теме «Язык структурированных запросов» («Операторы запросов»), теоретический материал к лабораторной работе («Основы SQL»).
2. Создать SQL -запросы, на создание таблицы, на обновление записей, на удаление записей, на добавление данных, на удаление таблицы, на создание индексов.
3. В отчете представить скриншоты полученных SQL-конструкций и результатов запуска в виде таблицы . Отчет представить в письменном виде.

<b>Описание запроса</b>
<b>SQL-конструкция для создания запроса</b>
<b>Результат запуска</b>

**Порядок выполнения работы**

1. Для создания нового SQL- запроса в окне базы данных нажмите на кнопку «Создание запроса в режиме конструктора» и не выбирая таблицы для запроса нажмите кнопку Вид на панели инструментов и выберите SQL.

2. Используя инструкцию **CREATE TABLE**, создайте запрос на создание новой таблицы для предметной области (*выданной ранее преподавателем ;таблицу для создания выберите одну, но содержащую обязательно простые и составные индексы*), определив в запросе первичный ключ и проиндексировав соответствующие поля, используя предложение **CONSTRAINT**.

После чего нажмите кнопку Запуск. Результат в виде скриншотов в отчете.

3. Создайте запрос на создание еще одной таблицы, содержащей внешний ключ по отношению к первичному ключу предыдущей таблицы. Запустите запрос, после чего проверьте, отразились ли изменения в схеме данных. Результат в виде скриншотов в отчете.

4. Используя инструкцию **ALTER TABLE** измените структуру таблицы несколькими способами:

-добавить новое поле **ADD COLUMN**;

-добавить составной индекс **ADD CONSTRAINT**;

-удалить поле **DROP COLUMN**;

-удалить составной индекс **DROP CONSTRAINT**. Результат в виде скриншотов в отчете.

5. Используя инструкцию **UPDATE**, создайте запрос на обновление данных в созданных ранее таблицах. Результат в виде скриншотов в отчете.

6. Используя инструкцию **INSERT INTO**, создайте запросы на добавление группы записей и одной записи в существующую таблицу. Результат в виде скриншотов в отчете.

7. Используя инструкцию **CREATE INDEX**, создайте запрос на создание нового индекса, используя различные условия на значения индексов (**IGNORE NULL**, **DISALLOW NULL**, **PRIMARY**), а также типы сортировки. Результат в виде скриншотов в отчете.

8. Используя инструкцию **DROP**, создайте запросы на удаление таблицы и индекса, созданных ранее в БД. Результат в виде скриншотов в отчете.

9. Сохраните все созданные запросы в базе данных.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

### Основные источники:

1. Агальцов В.П. Базы данных. Распределенные и удаленные базы данных: учебник. – М.: ФОРУМ: ИНФРА-М, 2017. -271с.
2. Баженова И.Ю. Основы проектирования приложений баз данных / И.Ю. Баженова. - 2-е изд., испр. - Москва: Национальный Открытый Университет «ИНТУИТ», 2016. - 238 с.: ил. - Библиогр. в кн. - ISBN 5-94774-539-9; [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=428933>.
3. Баженова И.Ю. SQL и процедурно-ориентированные языки / И.Ю. Баженова. - 2-е изд., испр. - Москва: Национальный Открытый Университет «ИНТУИТ», 2016. - 167 с.: ил. - Библиогр. в кн. - ISBN 5-94774-539-9; [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=428934>.
4. Кумскова И.А. Базы данных: учебник для СПО / И. А. Кумскова.- М.: КНОРУС, 2016.-488 с.
5. Лазецкас Е.А. Базы данных и системы управления базами данных: учебное пособие / Е.А. Лазецкас, И.Н. Загумённикова, П.Г. Гилевский. - Минск: РИПО, 2016. - 257 с.: ил. - Библиогр. в кн. - ISBN 978-985-503-558-0; [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=463305>.

### Дополнительные источники:

1. Базы данных в высокопроизводительных информационных системах: учебное пособие / авт.-сост. Е.И. Николаев; Министерство образования и науки РФ, Федеральное государственное автономное образовательное учреждение высшего образования «Северо-Кавказский федеральный университет». - Ставрополь: СКФУ, 2016. - 163 с.: ил. - Библиогр.: с.161.; [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=466799>.
2. Кузнецов С. Введение в реляционные базы данных / С. Кузнецов. - 2-е изд., исправ. - Москва: Национальный Открытый Университет «ИНТУИТ», 2016. - 248 с.: ил. - (Основы информационных технологий). - Библиогр. в кн.; [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429088>.
3. Кузнецов С. Введение в модель данных SQL: курс / С. Кузнецов. - 2-е изд., исправ. - Москва: Национальный Открытый Университет «ИНТУИТ», 2016. - 351 с.: ил. - (Основы

информационных технологий). - Библиогр. в кн. - ISBN 5-9556-00028-0; [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429087>.

4. Мартишин С.А. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQL Workbench: учебное пособие. – М.: ФОРУМ: ИНФРА-М, 2017. – 160 с.

5. Системы управления базами данных: лабораторный практикум / сост. Д.Л. Осипов, М.Г. Огур; Министерство образования и науки РФ, Федеральное государственное автономное образовательное учреждение высшего образования «Северо-Кавказский федеральный университет». - Ставрополь: СКФУ, 2017. - 148 с. - Библиогр. в кн.; [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=483760>.

### **Периодические издания:**

1. Проблемы информатики. Издательство «Федеральное государственное бюджетное учреждение науки Институт вычислительной математики и математической геофизики Сибирского отделения Российской академии наук»;

2. Проблемы информационной безопасности. Компьютерные системы. Издательство «Федеральное государственное автономное образовательное учреждение высшего образования Санкт-Петербургский политехнический университет Петра Великого»;

3. Системный администратор: ежемесячный журнал / изд. ООО «Синдикат 13»; гл. ред. Г. Положевец - Москва: Синдикат 13, - ISSN 1813-5579; [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=430336>;

4. Компоненты и технологии: Медиа КиТ.

### **Интернет-ресурсы:**

1. Компьютерные книги. Режим доступа: [<http://computers.plib.ru/programming/Books.VBasic6/index.html> 06.05.2020].

2. On-line библиотека свободно доступных материалов по информационным технологиям. Режим доступа: [<http://digitland.ru> 06.05.2020].

3. Открытые системы. Режим доступа: [<http://www.osp.ru> 06.05.2020].

4. ComputerBild. Режим доступа: [http://www.computerbild.ru/06.05.2020].
5. Мир ПК. Режим доступа: [http://www.pcworld.ru/06.05.2019].
6. Мобильные компьютеры. Режим доступа: http://www.mconline.ru/ 06.05.2020].
7. Компьютерра. Режим доступа: [http://www.computerra.ru/06.05.2020].