



**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ**

**Захарова О. И.**

## **БАЗЫ ДАННЫХ**

**Методические указания  
по выполнению курсовых работ  
по направлению подготовки  
38.03.05 – «Бизнес-информатика»**

**Самара - 2016**

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

Кафедра «Информационные системы и технологии»

**Захарова О.И.**

## **БАЗЫ ДАННЫХ**

**Методические указания**  
по выполнению курсовых работ  
по направлению подготовки  
38.03.05 – «Бизнес информатика»

Самара 2016

УДК 004.65+ 004.43  
ББК 32.973

Рекомендовано к изданию методическим советом ПГУТИ,  
протокол № 38, от 30.05.2016 г.

**Захарова, О. И.**

**Базы данных:** методические указания по выполнению курсовых работ. / О. И. Захарова, – Самара : ПГУТИ, 2016. – 32 с.

Методические указания по выполнению курсовых работ для дисциплины «Базы данных» содержат задания к выполнению курсовых работ, теоретический материал, необходимый для их выполнения.

Задача курсовой работы состоит в том, чтобы создать систему управления данными (базу данных), отражая, таким образом, полученные в пройденном курсе знания.

Методические указания разработаны в соответствии с Федеральным государственным образовательным стандартом высшего профессионального образования по направлению подготовки 38.03.05 – «Бизнес информатика» и предназначено для студентов третьего курса факультета информационных систем и технологий, а также для студентов других специальностей, изучающих и использующих базы данных.

© Захарова О.И., 2016

## Содержание

1. Основные этапы выполнения КР по дисциплине «БД» .....	5
2. Содержание пояснительной записки .....	5
2.1 Постановка задачи .....	5
2.2 Инфологическая модель .....	6
2.3 Нормализация отношений .....	10
2.4 Разработка таблиц и схемы базы данных .....	12
2.5 Запросы системы .....	12
2.6 Выводы .....	12
3. Требования к БД .....	12
4. Дополнительное задание .....	13
5. Варианты предметных областей для выполнения КР .....	14
Список рекомендуемой литературы .....	18

## 1. Основные этапы выполнения КР по дисциплине «БД»

### 1. Изучение темы курсовой работы.

На этом этапе студент получает тему из числа предложенных.

### 2. Анализ предметной области и составление словесного описания модели проектируемой системы и правил ее функционирования.

Студент при составлении словесной модели может задавать любые ограничения на свою предметную область, но эти ограничения обязательно должны учитываться в дальнейшем при проектировании структуры базы данных и, аналогично, любое ограничение в структуре базы данных должно быть описано в словесной модели.

### 3. Разработка структуры баз данных.

Студент составляет проект структуры базы данных, оформляет его в виде ER-диаграммы (концептуальная схема). После этого, на основе концептуальной диаграммы, составляется физическая диаграмма. Проводится нормализация.

### 4. Создание структуры базы данных и заполнение их тестовой выборкой данных.

Студент создает необходимые таблицы и ограничения к ним (первичные ключи, внешние ключи и собственные ограничения). Заполняет их тестовыми данными.

### 5. Написание требуемых SQL-запросов.

Студент составляет необходимые по заданию SQL-запросы и проверяет их на тестовой выборке данных.

### 6. Оформление пояснительной записки о курсовой работе.

## 2. Содержание пояснительной записки

1. Постановка задачи.
2. Инфологическая модель.
3. Нормализация отношений.
4. Разработка таблиц и схемы базы данных.
5. Запросы системы.
6. Выводы.
7. Список литературы

В ходе защиты курсовой работы необходимо продемонстрировать работу приложения и выполнение запросов.

### 2.1 Постановка задачи

Задача курсовой работы состоит в том, чтобы создать систему управления данными (базу данных), отражая, таким образом, полученные в пройденном курсе знания.

**База данных** - центральная часть информационной системы, совокупность данных для машинной обработки, которая отражает информационную модель некоторой предметной области. Современные БД (базы данных) хранят данные в виде таблиц, которые определенным образом связаны между собой, по этому свойству БД называются **реляционными**.

БД создается с помощью СУБД (системы управления БД). **СУБД** - программная инструментальная система, основные назначения которой:

- 1) описать БД, таблицы и связи между ними, описать операции над данными в таблицах;
- 2) контролировать целостность и непротиворечивость данных;
- 3) автоматически отображать описанную информационную модель в физическую БД на

магнитных носителях компьютера.

### Основы проектирования реляционных БД

Жизненный цикл БД можно разбить на три основные стадии:

- 1) проектирование (на бумаге или с помощью специальных программ);
- 2) программная реализация;
- 3) эксплуатация.

На этапе **проектирования** решаются следующие вопросы:

- 1) изучение задачи (обследование предметной области), выделение объектов и связей, о которых надо хранить информацию;
- 2) составление исходных таблиц БД;
- 3) нормализация (декомпозиция) таблиц и назначение ключевых полей.

На этапе **реализации** происходит:

- 1) описание полученных таблиц средствами СУБД и ввод их в компьютер;
- 2) разработка отчетов, экранных форм, запросов, макросов и программ;
- 3) отладка и тестирование программ из ИС и обучение персонала.

На стадии **эксплуатации** происходит наполнение ИС реальными данными, использование, доработка и сопровождение.

### 2.2 Инфологическая модель

Цель инфологического этапа проектирования состоит в получении семантических (концептуальных) моделей, отражающих предметную область и информационные потребности пользователей. Поэтому этот этап называют еще как семантическое моделирование. Семантическое моделирование представляет собой моделирование структуры данных, опираясь на смысл этих данных.

Понятие “Предметная область” - базисное в теории БД и не имеет строгого определения. Оно вытекает из понятий “объект” и “предмет”. Предметная область (ПО) – часть реального мира, подлежащая изучению с целью организации управления и, в конечном итоге, автоматизации. ПО представляется множеством фрагментов, которые характеризуются множеством объектов, множеством процессов, использующих объекты, а также множеством пользователей, характеризующихся единым взглядом на предметную область.

Объектом называется явление внешнего мира. Это либо нечто реально существующее - человек, товар, изделие, либо процесс - учет рождаемости, получение товаров, выпуск изделий. Каждый объект обладает огромным количеством свойств.

Предмет – модель реального объекта, в котором зафиксированы лишь выделенные для ИС свойства и связи. Совокупность отобранных предметов образует объектное ядро предметной области, а совокупность их взаимосвязей - структуру фрагмента действительности. Т.о. понятие “Предметная область” соответствует точке зрения потребителя на объектное ядро: в ней выделены только те объекты, свойства объектов и связи между объектами, которые представляют ценность для ИС и должны быть сохранены в БД.

Имеется целый ряд методик моделирования предметной области. Одна из наиболее популярных в настоящее время методик базируется на использовании графических диаграмм, включающих небольшое число разнородных компонентов ERD (Entity-Relationship Diagrams). В русскоязычной литературе эти диаграммы называют "объект – отношение" либо "сущность - связь".

Модель ERD была предложена в 1976 г. Питером Пин-Шэн Ченом. В дальнейшем многими авторами были разработаны свои варианты подобных моделей: нотация (notation – система обозначения, записи) Мартина, нотация IDEF1X, нотация Баркера), но все они базируются на графических диаграммах, предложенных Ченом.

На использовании разновидностей ER-модели основано большинство современных подходов к проектированию реляционных баз данных.

По сути, все варианты диаграмм сущность-связь исходят из одной идеи - рисунок всегда нагляднее текстового описания. Все такие диаграммы используют графическое изображение сущностей предметной области, их свойств (атрибутов), и взаимосвязей между сущностями.

Мы познакомимся с ER-диаграммами в нотации Баркера, как довольно легкой в понимании основных идей.

### Основные понятия ER-диаграмм

Основными понятиями ER-модели являются сущность, связь и атрибут.

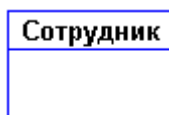
Для большей выразительности и лучшего понимания имя сущности может сопровождаться примерами конкретных объектов этого типа.

*Определение 1. Сущность* - это реальный или представляемый объект, информация о котором должна сохраняться и быть доступна. Сущностями могут быть люди, места, самолеты, рейсы, вкус, цвет и т.д.

Каждая сущность должна иметь наименование, выраженное существительным в единственном числе. При этом имя сущности - это имя типа, а не некоторого конкретного экземпляра этого типа. Понятие тип сущности относится к набору однородных личностей, предметов, событий или идей, выступающих как целое.

Примерами сущностей могут быть такие классы объектов как "Поставщик", "Сотрудник", "Накладная".

Каждая сущность в модели изображается в виде прямоугольника, содержащего имя сущности:



*Определение 2. Экземпляр сущности* - это конкретный представитель данной сущности. Например, представителем сущности "Сотрудник" может быть "Сотрудник Иванов".

Экземпляры сущностей должны быть различимы, т.е. сущности должны иметь некоторые свойства, уникальные для каждого экземпляра этой сущности.

*Определение 3. Атрибут сущности* - это поименованная характеристика сущности. Его наименование должно быть уникальным для конкретного типа сущности, но может быть одинаковым для различного типа сущностей (например, ЦВЕТ может быть определен для многих сущностей: СОБАКА, АВТОМОБИЛЬ, КРАСКА и т.д.). Атрибуты используются для определения того, какая информация должна быть собрана о сущности. Примерами атрибутов для сущности АВТОМОБИЛЬ являются ТИП, МАРКА, НОМЕРНОЙ ЗНАК, ЦВЕТ и т.д.

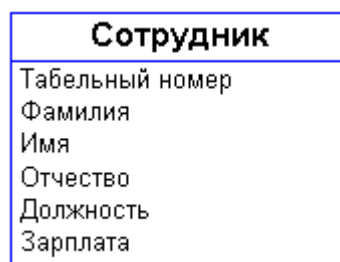
Здесь также существует различие между типом атрибута и экземпляром. Тип атрибута ЦВЕТ имеет много экземпляров или значений: Красный, Синий, Банановый, Белая ночь и т.д., однако каждому экземпляру сущности присваивается только одно значение атрибута.

Абсолютное различие между типами сущностей и атрибутами отсутствует. Атрибут является таковым только в связи с типом сущности. В другом контексте атрибут может выступать как самостоятельная сущность. Например, для автомобильного завода цвет - это только атрибут продукта производства, а для лакокрасочной фабрики цвет - тип сущности.

Каждый атрибут обеспечивается именем, уникальным в пределах сущности. Наименование атрибута должно быть выражено существительным в единственном числе (возможно, с характеризующими прилагательными).

Примерами атрибутов сущности "Сотрудник" могут быть такие атрибуты как "Табельный номер", "Фамилия", "Имя", "Отчество", "Должность", "Зарплата" и т.п.

Атрибуты изображаются в пределах прямоугольника, определяющего сущность:



Атрибуты могут классифицироваться по принадлежности к одному из трех различных типов: описательные, указывающие, вспомогательные.

Описательные атрибуты представляют факты, внутренне присущие каждому экземпляру сущности.

Указывающие атрибуты используются для присвоения имени или обозначения экземплярам сущности.

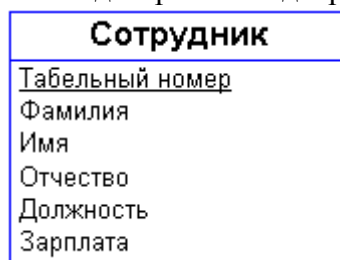
Вспомогательные атрибуты используются для связи экземпляра одной сущности с экземпляром другого. Атрибуты подчиняются строго определенным правилам.

**Определение 4. Ключ сущности** - минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности. Минимальность означает, что исключение из набора любого атрибута не позволяет идентифицировать сущность по оставшимся.

Например, для сущности Расписание ключом является атрибут Номер\_рейса или набор: Пункт\_отправления, Время\_вылета и Пункт\_назначения (при условии, что из пункта в пункт вылетает в каждый момент времени один самолет).

Сущность может иметь несколько различных ключей.

Ключевые атрибуты изображаются на диаграмме подчеркиванием:

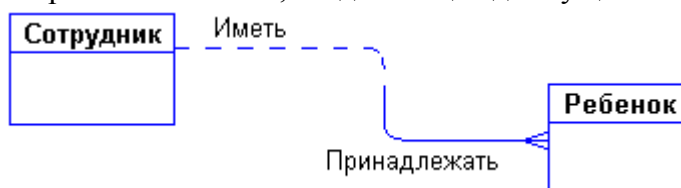


**Определение 5. Связь** - это некоторая ассоциация между двумя сущностями. Одна сущность может быть связана с другой сущностью или сама с собою. Связи позволяют по одной сущности находить другие сущности, связанные с ней.

Если бы назначением базы данных было только хранение отдельных, не связанных между собой данных, то ее структура могла бы быть очень простой. Однако одно из основных требований к организации базы данных – это обеспечение возможности отыскания одних сущностей по значениям других, для чего необходимо установить между ними определенные связи. А так как в реальных базах данных нередко содержатся сотни или даже тысячи сущностей, то теоретически между ними может быть установлено более миллиона связей. Наличие такого множества связей и определяет сложность инфологических моделей.

Например, связи между сущностями могут выражаться следующими фразами - "СОТРУДНИК может иметь несколько ДЕТЕЙ", "каждый СОТРУДНИК обязан числиться ровно в одном ОТДЕЛЕ".

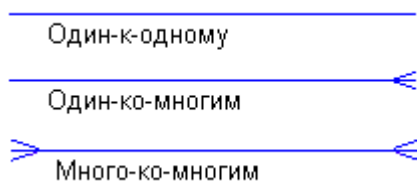
Графически связь изображается линией, соединяющей две сущности:





Каждая связь имеет два конца и одно или два наименования. Наименование обычно выражается в неопределенной глагольной форме: "иметь", "принадлежать" и т.п. Каждое из наименований относится к своему концу связи. Иногда наименования не пишутся ввиду их очевидности.

Каждая связь может иметь один из следующих типов связи:

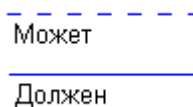


Связь типа один-к-одному означает, что один экземпляр первой сущности (левой) связан с одним экземпляром второй сущности (правой). Связь один-к-одному чаще всего свидетельствует о том, что на самом деле мы имеем всего одну сущность, неправильно разделенную на две.

Связь типа один-ко-многим означает, что один экземпляр первой сущности (левой) связан с несколькими экземплярами второй сущности (правой). Это наиболее часто используемый тип связи. Левая сущность (со стороны "один") называется родительской, правая (со стороны "много") - дочерней. (см. рис. графического изображения связи)

Связь типа много-ко-многим означает, что каждый экземпляр первой сущности может быть связан с несколькими экземплярами второй сущности, и каждый экземпляр второй сущности может быть связан с несколькими экземплярами первой сущности. Тип связи много-ко-многим является временным типом связи, допустимым на ранних этапах разработки модели. В дальнейшем этот тип связи должен быть заменен двумя связями типа один-ко-многим путем создания промежуточной сущности.

Каждая связь может иметь одну из двух модальностей связи:



Модальность "может" означает, что экземпляр одной сущности может быть связан с одним или несколькими экземплярами другой сущности, а может быть и не связан ни с одним экземпляром.

Модальность "должен" означает, что экземпляр одной сущности обязан быть связан не менее чем с одним экземпляром другой сущности.

Связь может иметь разную модальность с разных концов.

Описанный графический синтаксис позволяет однозначно читать диаграммы, пользуясь следующей схемой построения фраз:

<Каждый экземпляр СУЩНОСТИ 1> <МОДАЛЬНОСТЬ СВЯЗИ> <НАИМЕНОВАНИЕ СВЯЗИ> <ТИП СВЯЗИ> <экземпляр СУЩНОСТИ 2>.

Каждая связь может быть прочитана как слева направо, так и справа налево. Например, связь, представленная на рисунке выше 4 читается так:

Слева направо: "каждый сотрудник может иметь несколько детей".

Справа налево: "Каждый ребенок обязан принадлежать ровно одному сотруднику".

### Получение реляционной схемы из ER-схемы

**Шаг 1.** Каждая простая сущность превращается в таблицу. Простая сущность - сущность, не являющаяся подтипом и не имеющая подтипов. Имя сущности становится именем таблицы.

**Шаг 2.** Каждый атрибут становится возможным столбцом с тем же именем; может выбираться более точный формат. Столбцы, соответствующие необязательным атрибутам, мо-

гут содержать неопределенные значения; столбцы, соответствующие обязательным атрибутам, - не могут.

**Шаг 3.** Компоненты уникального идентификатора сущности превращаются в первичный ключ таблицы. Если имеется несколько возможных уникальных идентификатора, выбирается наиболее используемый. Если в состав уникального идентификатора входят связи, к числу столбцов первичного ключа добавляется копия уникального идентификатора сущности, находящейся на дальнем конце связи (этот процесс может продолжаться рекурсивно). Для именованя этих столбцов используются имена концов связей и/или имена сущностей.

**Шаг 4.** Связи многие-к-одному (и один-к-одному) становятся внешними ключами. Т.е. делается копия уникального идентификатора с конца связи "один", и соответствующие столбцы составляют внешний ключ. Необязательные связи соответствуют столбцам, допускающим неопределенные значения; обязательные связи - столбцам, не допускающим неопределенные значения.

**Шаг 5.** Индексы создаются для первичного ключа (уникальный индекс), внешних ключей и тех атрибутов, на которых предполагается в основном базировать запросы.

### 2.3 Нормализация отношений

Нормализация - процесс, в результате которого можно избавиться от дефектов проектирования базы данных. В процессе нормализации мы, получаем ряд нормальных форм, используя набор правил, описывающих то, что следует, и что не следует делать со структурой нашей таблицы. Процесс нормализации состоит из разбиения таблиц на меньшие, в результате чего формируется лучшая структура.

Чтобы выполнить нормализацию, структура базы данных последовательно приводится к различным формам. Вообще говоря, каждая последующая форма относится к предыдущей категории. Например, чтобы схема базы данных соответствовала второй нормальной форме, она должна также соответствовать и первой нормальной форме. Чтобы схема соответствовала третьей нормальной форме, она должна соответствовать и второй нормальной форме, и т.д. На каждой стадии добавляется все больше правил, которым должна соответствовать схема.

Первая нормальная форма требует, чтобы каждое значение атрибута или столбца было атомарным. Это значит, что каждый атрибут должен содержать единственное значение, а не множество значений или другую строку базы данных. Рассмотрим таблицу 1.

Таблица 1. Client

id	f_sob	fio	otv	ur_adr	fiz_adr	tel	vid_doc
3	Физ. лицо	Аверин А.С.	Леонов А.Ю.	8 Марта 38	Советская 45	2-24-09	Вод. удост.
2	Физ. лицо	Петров П.П.	Синегубов М.С.	Калинина 4	Пролеткая 24	72-80-21	Воен. билет
5	Юр. лицо	Ронжин Д.С.	Карикова Т.Н.	Пролет-кая 20	Маяков-го 150	55-12-33	паспорт

Это - ненормализованная версия таблицы клиент. Здесь размешены такие столбцы как form\_sob, vid\_doc, которые надо привести к первой нормальной форме. Чтобы привести схему к первой нормальной форме, необходимо в столбце товар, покупателя и форма расчета разместить атомарные значения. Это можно сделать различными способами. Первая, наверное, самая очевидная возможность, показана в таблице 2.

Таблица 2. Client

id	Id_f_sb	fio	otv	ur_adr	fiz_adr	tel	Id_v _dc
3	2	Аверин А.С.	Леонов А.Ю.	8 Марта 38	Советская 45	2-24-09	3
2	2	Петров П.П.	Синегубов М.С.	Калинина 4	Пролетарская 24	72-80-21	2
5	1	Ронжин Д.С.	Карикова Т.Н.	Пролетарская 20	Маяковского 150	55-12-33	1

В данной таблице мы выделили по одной строке на каждый элемент в столбцах форма собственности, вид документа. Это решение далеко от идеального, по сколько оно порождает очевидную избыточность данных. Лучший и самый правильный вариант приведения наших данных к нормальной форме показан на рисунке 2.

В этом случае мы выделили данные формы собственности в отдельную таблицу, что позволяет решить проблему избыточности.

После приведения схемы к нормальной форме можно перейти к формам старших порядков, которые немного труднее для понимания. Говорят, что схема находится во второй нормальной форме, если все атрибуты, не являющиеся частью первичного ключа, а сама схема уже находится в первой нормальной форме. Что это значит? Это значит, что каждый атрибут, не является ключом, должен функционально зависеть от всех составляющих ключа. Так, если первичный ключ составлен из нескольких столбцов, то всякий другой атрибут в таблице должен зависеть от комбинации всех этих столбцов.

В таблице 2 Client имеет по одной строке на каждый элемент квалификации типов. Таблица находится в первой нормальной форме, но не удовлетворяет второй нормальной форме. Другими словами, мы можем определить наименование vid\_doc, используя только кодовый номер типа (Id\_vid\_doc). Это значит, что указанные атрибуты функционально зависимы только от части первичного ключа, а не от всего первичного ключа. Таким образом, я могу определить эти атрибуты по части первичного ключа, и для этого совсем не нужен весь первичный ключ. Следовательно, указанная схема не находится во второй нормальной форме.

Чтобы привести эту схему ко второй нормальной форме необходимо разбить таблицу на такие таблицы, в которых все не ключевые атрибуты будут полностью функционально зависеть от ключа. Очевидно, что в данном случае этого можно достичь, разделив исходную таблицу на две:

client (Id\_vid\_doc)

Vid\_doc (vid)

Пример показан в таблице 3.

Таблица 3. Vid\_doc

id	Vid
1	Паспорт
3	Водительское удостоверение
2	Военный билет

Как уже говорилось, эта схема находится в первой нормальной форме, поскольку все значения являются атомарными. Но она также находится и во второй нормальной форме, поскольку каждый атрибут, не являющийся ключом, теперь функционально зависит от всех частей ключа.

## 2.4 Разработка таблиц и схемы базы данных

Для создания данной системы управления данными понадобятся следующие таблицы (перечислить названия таблиц), которые необходимо связать для обеспечения непротиворечивости данных.

Для создания таблицы в СУБД MySQL используется команда CREATE TABLE. С ее помощью описывается структура таблицы, включая названия полей, их тип данных, описание и значения некоторых свойств.

Описать структуру каждой таблицы и представить данные, которые они содержат.

## 2.5 Запросы системы

**Запросы** позволяют отбирать необходимую информацию из таблиц и представлять ее в табличном виде. Запросы создаются с помощью команды SELECT.

Существует следующие типа запросов для различных целей:

*Запросы на выборку* отображают данные из одной или нескольких таблиц в виде таблицы.

*Перекрестные запросы* собирают данные из одной или нескольких таблиц в формате, похожем на формат электронной таблицы. Эти запросы используются для анализа данных и создания диаграмм, основанных на суммарных значениях числовых величин из некоторого множества записей.

*Запросы на изменение* используются для создания новых таблиц из результатов запроса и для внесения изменений в данные существующих таблиц. С их помощью можно добавлять или удалять записи из таблицы и изменять записи согласно выражениям, задаваемым в режиме конструктора запроса.

Многотабличные запросы, в свою очередь, делятся по типу используемого соединения таблиц:

- внутреннее соединение по одному полю,
- внутреннее соединение по нескольким полям,
- косвенное соединение таблиц,
- внешнее соединение
- рекурсивное соединение
- соединение по отношению

Для данной системы было создано N запросов.

Запрос 1 создается по данным таблиц (название таблиц) и является запросом (указать тип запроса). Представить запрос в виде команды SELECT и результаты его выполнения.

Таким образом описываются все запросы.

## 2.6 Выводы

Содержат основные выводы о проделанной работе:

- в данной работе была исследована предметная область по теме - название темы;
- построена инфологическая модель;
- разработана даталогическая модель базы данных средствами MySQL, включающая N таблиц и связи между ними;
- разработаны N запросов для отображения данных из таблиц.

## 3. Требования к БД

- наличие таблиц и связей между ними в соответствии с инфологической моделью проектируемой предметной области;
- приведение БД к третьей НФ, описание процесса приведения;

- наличие не менее 10 запросов **всех** типов, реализуемых средствами СУБД MySQL:
  1. С использованием внутреннего соединения по одному полю.
  2. С использованием косвенно связанных таблиц.
  3. С использованием таблиц, связанных более чем по одному полю.
  4. С использованием внешнего соединения таблиц.
  5. С использованием рекурсивного соединения.
  6. С использованием соединений по отношению.
  7. С использованием функций агрегирования.
  8. Перекрестный запрос.
  9. С параметрами.
  10. С вычисляемым полем.

При выполнении работы в среде операционной системы Windows рекомендуется применять пакет разработчика «denever».

«Джентльменский» набор Web-разработчика («Д.н.в.р», читается «Денвер») — проект Дмитрия Котерова, локальный сервер (Apache, PHP, MySQL, Perl и т.д.) и программная оболочка, используемые Web-разработчиками для разработки сайтов на «домашней» (локальной) Windows-машине без необходимости выхода в Интернет. Главная особенность Денвера — удобство при удаленной работе сразу над несколькими независимыми проектами и возможность размещения на Flash-накопителе. Загрузить проект можно с сайта <http://www.denwer.ru/>

#### 4. Дополнительное задание

Для сдачи курсовой работы «автоматом» без защиты.

Разработать программу, реализующую созданную БД в виде готового приложения для пользователей согласно варианту.

Для разработки ПО потребуется создать программу. При выполнении задания можно воспользоваться любым доступным и удобным для студента языком программирования и средством разработки.

## 5. Варианты предметных областей для выполнения КР.

### *Вариант 1*

Предметная область ИС: Библиотека

Минимальный список характеристик:

- Автор книги, название, год издания, цена, является ли новым изданием, краткая аннотация;
- номер читательского билета, ФИО, адрес и телефон читателя, дата выдачи книги читателю и дата сдачи книги читателем.

### *Вариант 2*

Предметная область ИС: Университет

Минимальный список характеристик:

- Номер, ФИО, адрес и должность преподавателя;
- код, название, количество часов, тип контроля и раздел предмета;
- код, название, номер заведующего кафедрой;
- номер аудитории, где преподаватель читает свой предмет.

### *Вариант 3*

Предметная область ИС: Оптовая база

Минимальный список характеристик:

- Код товара, название товара, количество на складе, стоимость единицы товара, примечания - описание товара;
- номер и ФИО поставщика товара, срок поставки и количество товаров в поставке.

### *Вариант 4*

Предметная область ИС: Производство

Минимальный список характеристик:

- Код изделия, название изделия, является ли типовым, примечание - для каких целей предназначено;
- код, название, адрес и телефон предприятий, выпускающих изделия;
- год выпуска и объем выпуска данного изделия предприятием.

### *Вариант 5*

Предметная область ИС: Сеть магазинов

Минимальный список характеристик:

- Номер, ФИО, адрес, телефон и капитал владельцев магазинов;
- номер, название, адрес и телефон магазина;
- номер, ФИО, адрес, телефон поставщика, а также стоимость поставки данного поставщика в данный магазин.

### *Вариант 6*

Предметная область ИС: Авторемонтные мастерские

Минимальный список характеристик:

- Номер водительских прав, ФИО, адрес и телефон владельца автомобиля;
- номер, ФИО, адрес, телефон и квалификация механика;
- номер, марка, мощность и цвет автомобиля;
- номер, название, адрес и телефон ремонтной мастерской.

### *Вариант 7*

Предметная область ИС: Сессия

Минимальный список характеристик:

- Наименование специальности, код группы, ФИО, дата рождения, домашний адрес, телефон слушателя, примечания - автобиография слушателя;
- код, название, количество часов и вид контроля предметов, код сессии и оценки каждого слушателя каждому предмету в каждую сессию.

### *Вариант 8*

Предметная область ИС: Договорная деятельность организации

Минимальный список характеристик:

- Шифр договора, наименование организации, сроки выполнения, сумма договора, примечания вид договора
- номер, ФИО, адрес, телефон, должность, оклад сотрудников, сроки работы данного сотрудника по данному договору.

#### ***Вариант 9***

Предметная область ИС: Поликлиника

Минимальный список характеристик:

- Номер, фамилия, имя, отчество, дата рождения пациента;
- ФИО, должность и специализация лечащего врача,
- диагноз, поставленный данным врачом данному пациенту, необходимо ли амбулаторное лечение, срок потери трудоспособности, состоит ли на диспансерном учете, примечание

#### ***Вариант 10***

Предметная область ИС: Телефонная станция

Минимальный список характеристик:

- Номер абонента, фамилия абонента, адрес, дата установки, наличие блокиратора, задолженность, примечание
- Код АТС, код района, количество номеров.

#### ***Вариант 11***

Предметная область ИС: Спорт

Минимальный список характеристик:

- Фамилия спортсмена, дата рождения, вид спорта, команда, спортивный разряд
- Название соревнования, дата проведения, место проведения, показанный результат
- Название вида спорта, мировой рекорд и его дата, единица измерения.

#### ***Вариант 12***

Предметная область ИС: Сельскохозяйственные поставки

Минимальный список характеристик:

- Наименование с/х предприятия, дата регистрации, вид собственности, число работников
- Название продукции, единица измерения, закупочная цена
- Название поставщика, дата поставки, объем, себестоимость поставщика

#### ***Вариант 13***

Предметная область ИС: Городской транспорт

Минимальный список характеристик:

- Вид транспорта, средняя скорость движения, стоимость проезда, количество машин в парке
- номер маршрута, количество пассажиров в день, количество остановок в пути, количество машин на маршруте
- начальный пункт, конечный пункт, расстояние.

#### ***Вариант 14***

Предметная область ИС: География

Минимальный список характеристик:

- Название страны, столица, площадь территории, является ли страна развитой в экономическом отношении, количество населения
- национальность, язык, общая численность

#### ***Вариант 15***

Предметная область ИС: Домоуправление

Минимальный список характеристик:

- Номер подъезда, номер квартиры, общая площадь, полезная площадь, количество комнат,

- фамилия квартиросъемщика, дата прописки, количество членов семьи, количество детей в семье, есть ли задолженность по квартплате, примечание

#### ***Вариант 16***

Предметная область ИС: Аэропорт

Минимальный список характеристик:

- Номер рейса, пункт назначения, дата рейса, тип самолета, время вылета, время в пути, является ли маршрут международным,
- сведения о пассажире, примечание

#### ***Вариант 17***

Предметная область ИС: Рынок ПК

Минимальный список характеристик:

- Фирма-изготовитель, тип процессора, тактовая частота, объем RAM, объем жесткого диска, дата выпуска,
- Сведения о фирмах - реализаторах: Наименование, адрес, телефон, примечание

#### ***Вариант 18***

Предметная область ИС: Деканат.

Минимальный список характеристик:

- Фамилия и инициалы студента, дата рождения, группа, сумма стипендии, год зачисления
- факультет, специальность, число мест

#### ***Вариант 19***

Предметная область ИС: Микросхемы памяти

Минимальный список характеристик:

- Обозначение, разрядность, емкость, дата начала выпуска, время доступа, является ли широко используемой, стоимость, примечание
- Дата продажи, количество, цена

#### ***Вариант 20***

Предметная область ИС: Шахматы

Минимальный список характеристик:

- Фамилия спортсмена, страна, рейтинг
- Город, страна, дата проведения турнира, квалификационный уровень
- Стартовый номер, место участников.

#### ***Вариант 21***

Предметная область ИС: Судоходство

Минимальный список характеристик:

- Название корабля, водоизмещение, порт прописки, капитан
- Номер причала, дата прибытия, дата убытия, цель посещения
- Порт, страна, категория.

#### ***Вариант 22***

Предметная область ИС: Автотранспортное предприятие

Минимальный список характеристик:

- Марка автомобиля, номер, пробег, грузоподъемность
- Номер рейса, пункт назначения, расстояние, дата выезда, дата возвращения
- ФИО водителя, его табельный номер и категория.

#### ***Вариант 23***

Предметная область ИС: Научные конференции

Минимальный список характеристик:

- ФИО ученого, ученая степень, организация, страна
- Тема доклада, тип участия, наличие публикаций
- Название конференции, место проведения, дата



#### ***Вариант 24***

Предметная область ИС: Программные продукты

Минимальный список характеристик:

- Название, класс продукта, фирма, версия, дата выпуска, стоимость одной лицензии
- Дата инсталляции, дата деинсталляции, стоимость инсталляции, количество
- Пользователь, регион, сфера применения.

#### ***Вариант 25***

Предметная область ИС: Операционная система

Минимальный список характеристик:

- Идентификатор процесса, идентификатор владельца, приоритет, время выполнения
- Название ресурса, запрошенное количество, выделенное количество

#### ***Вариант 26***

Предметная область ИС: Автотранспортное предприятие

Минимальный список характеристик:

- номерной знак автомобиля, марка автомобиля, его техническое состояние, местонахождение автомобиля, средняя скорость, грузоподъемность, расход топлива,
- табельный номер водителя, фамилия, имя, отчество, дата рождения, стаж работы, оклад,
- дата выезда, дата прибытия, место назначения, расстояние, расход горючего, масса груза

#### ***Вариант 27***

Предметная область ИС: Каталог видео клипов

Минимальный список характеристик:

- Код видеоленты, дата записи, длительность, тема, стоимость,
- Сведения об актере: Фамилия, имя отчество, амплуа.

#### ***Вариант 28***

Предметная область ИС: Заказы

Минимальный список характеристик:

- Фамилия, имя, отчество клиента, номер счета, адрес, телефон,
- номер заказа, дата исполнения, стоимость заказа, название товара, его цена и количество

#### ***Вариант 29***

Предметная область ИС: Рецепты

Минимальный список характеристик:

- Наименование блюда, описание рецепта, категория, вес порции
- перечень ингредиентов, их количество, очередность добавления, предварительная обработка, количество порций
- название продукта, калорийность, единица измерения, цена за единицу.

#### ***Вариант 30***

Предметная область ИС: Отдел кадров

Минимальный список характеристик:

- Фамилия, имя, отчество, домашний адрес, телефон, дата рождения, образование, табельный номер
- должность, дата зачисления, оклад
- наименование подразделения, количество штатных единиц, фонд заработной платы за месяц и за год.

### Список рекомендуемой литературы

1. Андон Ф. Язык запросов SQL. Учебный курс [Text] / Андон Ф. - СПб.: Питер, 2006. - 416 с. : ил. - 30 экз. - ISBN 5-469-00394-9 : Б. ц. УДК 004.6004.65(075)
2. Атре Ш. «Microsoft SQL Server 6.5». М.: Комплект документации, 2004 г. -200 с.
3. Белью К., Браун М., Ливингстон Д. «Perl Web\_профессионалам». Киев: BHV, 2004 г. - 300 с.
4. Голицина ОЛ, Максимов НВ, Попов ИИ. «Базы данных». М.: Диалектика, 2005 г. - 350 с.
5. Диго С. М. Базы данных: проектирование и использование [Text] : учебник для вузов / Диго С. М. - М. : Финансы и статистика, 2005. - 592 с. : ил. - 20 экз. - ISBN 5-279-02571-2
6. Дюбуа П. «Применение MySQL и Perl в Web\_приложениях». М.: Вильямс, 2004 г. - 400 с.
7. Игнатович И. «Семейство реляционных баз данных IBM DB2». М.: СУБД 2/05, 2006 г.-250 с.
8. Клинтон П. «Освой самостоятельно Perl за 24 часа». Киев: Мир 2005 г.-150 с.
9. Конноли Т.К. «Базы данных проектирование реализация и сопровождение». Берг: Диалектика, 2005 г.-180 с.
10. Конноли Т.К. «Базы данных проектирование реализация и сопровождение» Берг: Диалектика, 2005 г.-300 с.
11. Леонтьев Б.К. «Язык программирования Perl для Интернет»; «Новый издательский дом» - 2006 г.-350 с.
12. Малыхина М.П. Базы данных: основы, проектирование, использование, 2-е изд. перераб. и доп. – СПб.: БХВ-Петербург, 2008. 528 с.
13. Маслов В.В. «Введение в Perl». М.: Наука, 2005 г.-200 с.
14. Местерн П. «Perl: интеграция с Интернет». Берг: Диалектика, 2005 г.-200
15. Нагао М., Катаяма Т., Уэмура С. «Реализация языка SQL в СУБД MySQL». М.: Мир, 2006 г.-350 с.
16. Паттерсон Л. «Использование HTML». М.: Изд. дом Вильямс, 2004 г.-400 с.
17. Прохоров А. «Microsoft SQL Server 6.5 DBA Survival Guide, by Mark Spenik & Orryn Sledge». Хабаровск: ISBN, 2005 г.-500 с.
18. Уайтхэд П «Perl: наглядный курс программирования»; М.: «Диалектика», 2005 г.-350 с.
19. Ульман Д. «Теория и практика БД». М.: Машиностроение, 2005 г.-250 с.
20. Фролов П.С. «Базы данных в Интернет» М.: СУБД, 2005 г.-400 с.
21. Хомоненко А. Д., Цыганков В. М., Мальцев М. Г. Базы данных: Учебник для высших учебных заведений /Под. ред. проф. А. Д. Хомоненко. 5-е изд., доп.- М:Бином-Пресс;СПб.: КОРОНА принт, 2009. - 736 с.