

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
КЫРГЫЗСКОЙ РЕСПУБЛИКИ



*У. Бертганов*

СПО (КОЛЛЕДЖ)  
Кыргызского государственного технического университета  
им. И Раззакова

# Учебно-методический комплекс

по дисциплине:

## «Базы данных»

для студентов 2-3курса  
специальности шифр 230109

«Программное обеспечение вычислительных программ и  
автоматизированных систем»

Составила: Акишова Мира Карыбековна

Содержание:

### 1. Стандарт дисциплины:

- обоснование
- цели и задачи
- ожидаемый результат

### 2. Рабочая программа

- Силабус
- Анкета дисциплины

### 3. Календарно-тематический план (КТП)

### 4. График выполнения СРС

### 5. Методические указания для преподавателя и студентов

### 6. Фонд оценочных средств, включающий в себя контрольные вопросы, тесты с ключами

### 7. Экзаменационные билеты

### 8. Рекомендованная литература, методические указания практических занятий

### 9. Конспекты лекций, практических занятий (либо тезисы)

### 10. Экспертная оценка УМК от работодателей или от ведущих специалистов

## **Обоснование необходимости учебной дисциплины**

Дисциплина «Базы данных» является одной из дополнительных

дисциплин базовой части рабочего учебного плана специальности

**230109 «Программное обеспечение вычислительных программ и автоматизированных систем»**

среднего профессионального образования, необходимая в формировании техника - программиста по программному обеспечению вычислительной техники и автоматизированных систем.

Изучение курса «Базы данных» должно позволить будущему специалисту правильно проектировать модели данных и реализации реляционной базы данных.

В связи с этим увеличивается значение данной дисциплины в подготовке специалистов по проектированию базы данных. Курс знакомит студентов с теоретическими основами проектирования модели данных (ER-модель), создание объектов базы данных и их реализация.

По данному предмету учебным планом предусматривается: проведение лекционных, практических занятий и курсовой проект студентов.

Вид аттестации студентов по данной дисциплине – экзамен.

### **1.2. Цели и задачи изучения дисциплины**

- владеть технологией проектирования и управления базами данных;
- владеть основными методами и способами разработки алгоритмов поставленных задач, а также средствами разработки программных приложений;
- использовать информационно-коммуникационные технологии в профессиональной деятельности;

### **1.3. Ожидаемые результаты изучения дисциплины**

В результате изучения курса «Базы данных» и по его успешному завершению в стенах колледжа, студент должен приобрести следующие компетенции:

В результате изучения дисциплины студент должен:

#### **знать:**

- основные положения теории баз данных, хранилищ данных, баз знаний;
- основные принципы построения концептуальной, логической и физической модели данных;
- современные инструментальные средства разработки схемы базы данных;
- структуры данных СУБД, общий подход к организации представлений, таблиц и индексов;
- методы организации целостности данных;
- способы контроля доступа к данным и управления привилегиями; – модели и структуры информационных систем.

#### **уметь:**

- создавать объекты базы данных в современных системах управления базами данных и управлять доступом к этим объектам;
- формировать и настраивать схему базы данных;
- применять стандартные методы для защиты объектов баз данных.

#### **владеть:**

- работы с объектами базы данных в конкретной системе управления базами данных;
- использования средств заполнения базы данных;
- использования стандартных методов защиты объектов баз данных.

**Рабочая программа дисциплины «БАЗЫ ДАННЫХ»**  
**сост. Акишова М.К.**  
**Колледж КГТУ имени И.Раззакова 2022**

Рабочая программа предназначена для преподавания дисциплины студентам 2 курса в 4семестре и 3 курса 5семестре среднего профессионального образования очной формы

обучения специальности **230109 «Программное обеспечение вычислительных программ и автоматизированных систем»**

Рабочая программа предусматривает изучение курса «БАЗЫ ДАННЫХ». В ней содержится темы лекционных, лабораторных занятий, вопросы СРС, список основной и дополнительной литературы.

Рабочая программа составлена с учетом Государственного образовательного стандарта среднего профессионального образования Кыргызской Республики по специальности **230109 «Программное обеспечение вычислительных программ и автоматизированных систем»** утвержденного

Министерством образования и науки Кыргызской Республики.

Составитель                      Акишова М.К..



**СИЛЛАБУС**  
**по дисциплине " БАЗЫ ДАННЫХ "**

**КОЛЛЕДЖ**  
**КГТУ им. И.Раззакова**

**Политика курса – опоздания, пропуски, поведение в аудитории, позднее предоставление работ, отсутствие на экз амене.**

- Обязательно посещать занятия.
- Активно участвовать в учебном процессе.
- Старательно выполнять домашние задания.
- Не опаздывать на занятия.
- Не разговаривать во время занятий, не читать газеты, отключить сотовый телефон, не жевать резинку.
- На занятия приходиться в деловой одежде.
- Не пропускать занятия.
- Избегать пропуска занятий по неуважительным причинам.
- Исключить курение в помещениях университета.
- Исключить телефонные разговоры по сотовому телефону на занятиях.
- Пропущенные занятия отрабатывать в определенное преподавателем время.
- В случае невыполнения заданий итоговая оценка снижается.

**Политика академического поведения и этики.**

- Быть толерантным, уважать мнение окружающих.
- Возражения формулировать в корректной форме.
- Конструктивно поддерживать обратную связь на всех занятиях.
- Содействовать коллективной работе и вовлечению в дискуссию более застенчивых студентов.
- Быть пунктуальным и обязательным.
- Плагиат и другие формы нечестной работы недопустимы. К плагиату относится следующее: отсутствие ссылок при использовании печатных и электронных материалов, цитат, мыслей других авторов.
- Недопустимы подсказывание и списывание во время тестов, экзаменов, занятий; сдача экзамена за другого студента, неразрешенное копирование материалов.
- В случае нарушения одного из вышеперечисленных пунктов студент не аттестовывается по курсу.

**Цель курса (дисциплины):**

- владеть технологией проектирования и управления базами данных;
- владеть основными методами и способами разработки алгоритмов поставленных задач, а также средствами разработки программных приложений;
- использовать информационно-коммуникационные технологии в профессиональной деятельности;



Формы самостоятельной учебной работы студентов, ее тематика и содержание определяются учебными планами и преподавателем по дисциплине «БАЗЫ ДАННЫХ» самостоятельно.

№	Название темы	Аудиторные занятия, часы	
		Лекции	Практические занятия
1	<b>Введение.</b> Основные понятия теории баз данных.	2	2
2	<b>Информационные модели.</b> Взаимосвязи в моделях и реляционный подход к построению моделей.	4	4
3	<b>Нормализация отношений.</b> Этапы проектирования баз данных	4	4
4	<b>Классификация и характеристика СУБД.</b> Обзор современных СУБД.	4	4
5	Постреляционные СУБД	4	4
6	<b>Работа с объектами</b> (таблицы, запросы, формы, отчеты и макросы)	4	4
7	<b>Краткая характеристика языка SQL.</b> Операторы языка SQL для работы с реляционной базой данных.	4	4
8	Организация запросов к базе данных на языке SQL	4	4
<b>Итого:</b>		<b>30</b>	<b>30</b>

**Перечень компетенций:**

Код компетенции	Формулировка компетенций
ПК-4	Владеть технологией проектирования и управления базами данных;

**Правила рейтинговой системы оценивания результатов обучения студентов по дисциплине «БАЗЫ ДАННЫХ»  
МОДУЛЬНО-РЕЙТИНГОВАЯ КАРТА**

Вид контроля	Вид учебных поручений	Формы отчётности и контроля	Рейтинг-баллы	
			mi	x

Текущий рейтинг-контроль (ТРК)	<b>Лекционные занятия</b>	активность 0-10	6	10
	Практические занятия	1.устный ответ, активность на занятиях- 0-10 2. выполнение домашнего задания 0-10	12	20
	Самостоятельная работа студента	1. реферат, доклад 2.исследование 3. РГЗ 0	7	12
Промежуточный рейтинговый контроль (ПРК)	контрольная работа 0-4; тестирование 0-4		5	8
<b>Итого по дисциплинарному модулю 1</b>			<b>30</b>	<b>50</b>
Текущий рейтинг-контроль	<b>Лекционные занятия</b>	активность 0-10	6	10
	Практические занятия	1.устный ответ, активность на занятиях- 0-10 2. выполнение домашнего задания 0-10	12	20
	Самостоятельная работа студента	1. реферат, доклад 0-6 2.исследование 3. РГЗ	8	12
Рубежный контроль	контрольная работа 0-4; тестирование 0-4		5	8
<b>Итого по дисциплинарному модулю 2</b>			<b>31</b>	<b>50</b>
Итоговый рейтинговый контроль			24	40
<b>Итого</b>			<b>61</b>	<b>100</b>
<b>Премияльные баллы</b> (участие в олимпиаде-3-5; выступление на конференциях – 3-5; участие в конкурсах научных работ – 3-5)			3	14
<b>Штрафы</b> (нарушение сроков сдачи работ - минус 0,5-5; отказ от ответа на практическом занятии – минус 0,5-2; не выполнение домашнего задания – минус 0,5-2)			-0,5	- 21
<b>Рейтинговая оценка по дисциплине</b>			<b>61</b>	<b>100</b>
Шкала перевода дисциплинарного рейтинга в академические оценки				
Дисциплинарный рейтинг		Академическая оценка		
61 – 73		3 (удовлетворительно)		
74- 87		4 (хорошо)		
88- 100		5 (отлично)		

**СПО (КОЛЛЕДЖ)  
КГУСТА им. Н.Исанова Анкета дисциплины  
«БАЗЫ ДАННЫХ»**

202 -20 уч.гг. 4-5семестр \_\_\_\_\_

**Краткое содержание дисциплины:**

Дисциплина предназначена для преподавания студентам 2- 3 курса в 4- 5 семестре среднего

профессионального образования очной формы обучения специальности

**«230109 «Программное обеспечение вычислительных программ и автоматизированных систем»**

предусматривает изучение курса «БАЗЫ

*ДАННЫХ*». В ней содержится темы лекционных, лабораторных занятий, вопросы СРС, список основной и дополнительной литературы.

**Кредитная стоимость дисциплины: 4 кредита**

- владеть технологией проектирования и управления базами данных;
- владеть основными методами и способами разработки алгоритмов поставленных задач, а также средствами разработки программных приложений;
- использовать информационно-коммуникационные технологии в профессиональной деятельности

В результате изучения курса «Базы данных» и по его успешному завершению в стенах колледжа, студент должен приобрести следующие компетенции:

**Перечень компетенций, которые должны быть сформулированы у студента по итогам обучения дисциплины:**

Код компетенции	Формулировка компетенций
	Владеть технологией проектирования и управления базами данных;

Рассмотрено на методическом совете , Протокол № \_\_\_ от \_\_\_ \_\_\_\_\_ год.

Составитель

Акишова М.К.

## **ФОРМЫ ПРОВЕДЕНИЯ УЧЕБНЫХ ЗАНЯТИЙ**

Содержание дисциплины реализуется в различных формах организации учебного процесса – лекциях, лабораторных занятиях студентов.

## **ФОРМЫ КОНТРОЛЯ ЗНАНИЙ**

В процессе изучения дисциплины целесообразно применять следующие виды контроля: входной рейтинг, текущий рейтинг-контроль, промежуточный рейтинг-контроль, модульный рейтинг, итоговый рейтинг-контроль, итоговый рейтинг по дисциплине.

Текущий контроль осуществляется выборочно на занятиях в любое время изучения темы. Промежуточный контроль – оценка знаний по освоенному материалу каждого дисциплинарного модуля, одновременно для всех студентов.

В каждом модуле определяется минимальное и максимальное количества баллов. Сумма максимальных баллов по всем модулям равняется 100%-ному освоению материала. Минимальное количество баллов в каждом модуле является обязательным и не может быть заменено набором баллов в других модулях.

Для получения положительной оценки необходимо набрать не менее 60 баллов, предусмотренных по дисциплине (при условии набора всех обязательных минимальных баллов). Перевод баллов в академическую оценку осуществляется по следующей схеме: оценка «удовлетворительно» – 61-73 баллов, «хорошо» – 74-87 баллов, «отлично» – 88-100 баллов. Сумма баллов минимальных границ диапазонов всех дисциплинарных модулей должна составлять 61, а максимальных – 100. Рейтинговые баллы распределяются по дисциплинарным модулям в зависимости от значимости и трудоемкости модулей.



**КАЛЕНДАРНО-ТЕМАТИЧЕСКИЙ ПЛАН ДИСЦИПЛИНЫ  
«БАЗЫ ДАННЫХ»**

№	Название темы	Аудиторные занятия,			Самостоятельные работы
		лекции	часы		
			Лекции	Лаб. работа	
	4-5 семестр				
1	<b>Введение.</b> Основные понятия теории баз данных.	2	2	4	6
	<b><u>Информационные модели.</u></b> <u>Взаимосвязи в моделях и реляционный подход к построению моделей.</u>	2	2	4	8
	<b>Нормализация отношений.</b> Этапы проектирования баз данных	2	2	4	8
	<b>Классификация и характеристика СУБД.</b> Обзор современных СУБД.	2	2	4	8
	Постреляционные СУБД	2	2	4	2
	<b>Работа с объектами</b> (таблицы, запросы, формы, отчеты и макросы)	2	2	4	8
	<b>Краткая характеристика языка SQL.</b> Операторы языка SQL для работы с реляционной базой данных.	4	2	4	8
	Организация запросов к базе данных на языке SQL	2	4	4	6
	<b>Итого:</b>		<b>30</b>	18	<b>60</b>

## График самостоятельной работы студентов.

Виды самостоятельной работы	Учебные недели																Всего
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Всего СРС	4	4	4	4	4	4	6		4	4	4	4	4	4	6		60
в т.ч. СРСП			2		2		2	Сдача СРС			2		2		2	Сдача СРС	12

В качестве самостоятельной работы обучающихся предусмотрено:

1. самостоятельное изучение литературы;
2. оформление отчетов по лабораторным работам;
3. подготовка к защите отчетов по лабораторным работам;
4. подготовка к тестированию.

Лабораторные работы выполняются в соответствии с лабораторным практикумом, предоставляемым студентам при изучении курса. Оформление отчетов по лабораторным работам должно соответствовать требованиям, которые также описаны в данном лабораторном практикуме. Тематика лабораторных работ должна соответствовать вариантам, которые согласуются с преподавателем.

Подготовка к защите отчетов по лабораторным работам осуществляется на основе знаний, приобретенных в ходе изучения отдельных тем учебных дисциплин и самостоятельного изучения литературы. Подготовка к тестированию производится с использованием основной и дополнительной литературы, материалов лекций и теоретического материала лабораторного практикума.

### ТЕМАТИКА САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Студент выполняет самостоятельную работу для закрепления теоретических знаний. Суть работы – письменные иллюстрированные схемами ответы на вопросы. Объем 5-25 страниц.

№ недели	Тема для изучения	Форма выполнения	Приложение	Кол-во часов
1	Средства автоматизации проектирование	Конспекты лекций. Рефераты	Степанов А.Н. Информатика. Уч. для вузов. / Степанов А.Н. - СПб. 2010.	6
2	Реляционная алгебра	Рефераты.	SQL Server 2008 R2. Black book. Published by: Dreamtech PRESS, 19-A, Ansari Road, Daryaganj, New Delhi, 2012	9
3	Публикация баз данных в Internet	Самостоятельное конспектирование материала Доклад.	В.В.Дунаев. Базы данных Язык SQL для студента. 2-е издание. – СПб.: БХВ-Петербург, 2012. -320 с. 4. С.Д.Кузнецов. Базы данных. Модели данных. Учебник – М.: ООО «Бином-Пресс», 2008 г.	9
4	Современные СУБД и их применение (MS Access, MS SQL Server,	Рефераты. Доклады..	3. В.В.Дунаев. Базы данных Язык SQL для студента. 2-е издание. – СПб.: БХВ-	9

	Visual Fox Pro, Borland C++ и др.)		Петербург, 2012. -320 с. 4. С.Д.Кузнецов. Базы данных. Модели данных. Учебник – М.: ООО «Бином-Пресс», 2008 г.	
5	Объектно-ориентированное программированные СУБД	Рефераты, Доклады. Наличие конспектов лекций	2. SQL Server 2008 R2. Black book. Published by: Dreamtech PRESS, 19-A, Ansari Road, Daryaganj, New Delhi, 2012	9
6	Системы баз данных, основанные на правилах	Рефераты. Доклады.	3. В.В.Дунаев. Базы данных Язык SQL для студента. 2-е издание. – СПб.: БХВ-Петербург, 2012. -320 с. 4. С.Д.Кузнецов. Базы данных. Модели данных. Учебник – М.: ООО «Бином-Пресс», 2008 г.	9
7	Основные категории команд SQL	Доклады. Рефераты.	3. В.В.Дунаев. Базы данных Язык SQL для студента. 2-е издание. – СПб.: БХВ-Петербург, 2012. -320 с. 4. С.Д.Кузнецов. Базы данных. Модели данных. Учебник – М.: ООО «Бином-Пресс», 2008 г.	9
<b>Итого:</b>				<b>60</b>

#### СРС по модулям

	Название тем	Часы	Сроки выполнения
<b>МОДУЛЬ 1</b>			
1	Средства автоматизации проектирование	6	2,3 неделя
2	Реляционная алгебра	9	4-6 неделя
3	Публикация баз данных в Internet	9	6,7 неделя
4	Современные СУБД и их применение (MS Access, MS SQL Server, Visual Fox Pro, Borland C++ и др.)	9	8 неделя
<b>МОДУЛЬ 2</b>			
6	Объектно-ориентированное программированные СУБД	9	8-11 неделя
7	Системы баз данных, основанные на правилах	9	12-14-неделя
8	Основные категории команд SQL	9	15 –неделя
	<b>ВСЕГО:</b>	<b>60</b>	

## МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ (МАТЕРИАЛЫ) ДЛЯ ПРЕПОДАВАТЕЛЯ И СТУДЕНТА

№ п/п	Темы занятий	Применяемые средства обучения	Применяемые формы обучения	Используемые методы обучения
1.	<b>Введение.</b> Основные понятия теории баз данных.	методические указания к лабораторным занятиям	Лабораторные занятия на ПК	тесты, опрос
2.	<b>Информационные модели.</b> Взаимосвязи в моделях и реляционный подход к построению моделей.	методические указания к лабораторным занятиям	Лабораторные занятия на ПК	тесты, опрос
3.	<b>Нормализация отношений.</b> Этапы проектирования баз данных	методические указания к лабораторным занятиям	Лабораторные занятия на ПК	тесты, опрос
4.	<b>Классификация и характеристика СУБД.</b> Обзор современных СУБД.	методические указания к лабораторным занятиям	Лабораторные занятия на ПК	тесты, опрос
5.	Постреляционные СУБД	методические указания к лабораторным занятиям	Лабораторные занятия на ПК	тесты, опрос
6.	<b>Работа с объектами</b> (таблицы, запросы, формы, отчеты и макросы)	методические указания к лабораторным занятиям	Лабораторные занятия на ПК	тесты, опрос
7.	<b>Краткая характеристика языка SQL.</b> Операторы языка SQL для работы с реляционной базой данных.	методические указания к лабораторным занятиям	Лабораторные занятия на ПК	тесты, опрос
8.	Организация запросов к базе данных на языке SQL	методические указания к лабораторным занятиям	Лабораторные занятия на ПК	тесты, опрос

## МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ СТУДЕНТОВ

### *Промежуточная аттестация*

Каждый учебный семестр заканчивается сдачей зачетов и экзаменов (в период экзаменационной сессии). Подготовка к сдаче зачетов и экзаменов является также самостоятельной работой студента. Основное в подготовке к промежуточной аттестации по дисциплине (модулю) – повторение всего учебного материала дисциплины, по которому необходимо сдавать зачет или экзамен.

Только тот студент успевает, кто хорошо усвоил учебный материал. Если студент плохо работал в семестре, пропускал лекции, слушал их невнимательно, не конспектировал, не изучал рекомендованную литературу, то в процессе подготовки к сессии ему придется не повторять уже знакомое, а заново в короткий срок изучать весь учебный материал. Все это зачастую невозможно сделать из-за нехватки времени.

Для такого студента подготовка к зачету или экзамену будет трудным, а иногда непосильным делом, а конечный результат – академическая задолженность, и как следствие, возможное отчисление.

### *Лекции*

Главное в период подготовки к лекционным занятиям – научиться методам самостоятельного умственного труда, сознательно развивать свои творческие способности и овладевать навыками творческой работы. Для этого необходимо строго соблюдать дисциплину учебы и поведения. Четкое планирование своего рабочего времени и отдыха является необходимым условием для успешной самостоятельной работы.

В основу его нужно положить рабочие программы изучаемых в семестре дисциплин. Ежедневной учебной работе студенту следует уделять не менее 9 часов своего времени, т.е. при шести часах аудиторных занятий самостоятельной работе необходимо отводить не менее 3 часов.

Каждому студенту следует составлять еженедельный и семестровый планы работы, а также план на каждый рабочий день. С вечера всегда надо распределять работу на завтрашний день. В конце каждого дня целесообразно подводить итог работы: тщательно проверить, все ли выполнено по намеченному плану, не было ли каких-либо отступлений, а если были, по какой причине это произошло. Нужно осуществлять самоконтроль, который является необходимым условием успешной учебы. Если что-то осталось невыполненным, необходимо изыскать время для завершения этой части работы, не уменьшая объема недельного плана.

### *Самостоятельная работа на лекции*

Слушание и запись лекций – сложный вид аудиторной работы. Внимательное слушание и конспектирование лекций предполагает интенсивную умственную деятельность студента. Краткие записи лекций, их конспектирование помогает усвоить учебный материал. Конспект является полезным тогда, когда записано самое существенное, основное и сделано это самим студентом.

Не надо стремиться записать дословно всю лекцию. Такое «конспектирование» приносит больше вреда, чем пользы. Запись лекций рекомендуется вести по возможности собственными формулировками. Желательно запись осуществлять на одной странице, а следующую оставлять для проработки учебного материала самостоятельно в домашних условиях.

Конспект лекции лучше подразделять на пункты, параграфы, соблюдая красную строку. Этому в большой степени будут способствовать пункты плана лекции, предложенные преподавателям. Принципиальные места, определения, формулы и другое следует сопровождать замечаниями «важно», «особо важно», «хорошо запомнить» и т.п. Можно делать это и с помощью разноцветных маркеров или ручек. Лучше если они будут собственными, чтобы не приходилось просить их у однокурсников и тем самым не отвлекать их во время лекции.

Целесообразно разработать собственную «маркографию» (значки, символы), сокращения слов. Не лишним будет и изучение основ стенографии. Работая над конспектом лекций, всегда необходимо использовать не только учебник, но и ту литературу, которую дополнительно рекомендовал лектор. Именно такая серьезная, кропотливая работа с лекционным материалом позволит глубоко овладеть знаниями.

### *Практические занятия*

Подготовку к каждому практическому занятию каждый студент должен начать с ознакомления с планом занятия, который отражает содержание предложенной темы. Практическое задание необходимо выполнить с учетом предложенной преподавателем инструкции (устно или письменно). Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса.

Результат такой работы должен проявиться в способности студента свободно ответить на теоретические вопросы практического занятия и участия в коллективном обсуждении вопросов изучаемой темы, правильном выполнении практических заданий.

### *Структура практического занятия*

В зависимости от содержания и количества отведенного времени на изучение каждой темы практическое занятие состоит из трех частей:

1. Обсуждение теоретических вопросов, определенных программой дисциплины.
2. Выполнение практического задания с последующим разбором полученных результатов или обсуждение практического задания, выполненного дома, если это предусмотрено рабочей программой дисциплины.
3. Подведение итогов занятия.

Обсуждение теоретических вопросов проводится в виде фронтальной беседы со всей группой и включает выборочную проверку преподавателем теоретических знаний студентов.

Преподавателями определяется его содержание практического задания и дается время на его выполнение, а затем идет обсуждение результатов.

Подведением итогов заканчивается практическое занятие. Студентам должны быть объявлены оценки за работу и даны их четкие обоснования.

### *Работа с литературными источниками*

В процессе подготовки к практическим занятиям, студентам необходимо обратить особое внимание на самостоятельное изучение рекомендованной учебно-методической (а также научной и популярной) литературы. Самостоятельная работа с учебниками, учебными пособиями, научной, справочной и популярной литературой, материалами периодических изданий и Интернета, статистическими данными является наиболее эффективным методом получения знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у студентов свое отношение к конкретной проблеме.

Более глубокому раскрытию вопросов способствует знакомство с дополнительной литературой, рекомендованной преподавателем по каждой теме практического занятия, что позволяет студентам проявить свою индивидуальность, выявить широкий спектр мнений по изучаемой проблеме.

<b>№ п/п</b>	<b>Тема занятий</b>	<b>Источники (список литературы)</b>	<b>Форма контроля</b>
1	2	4	5
1.	Проектирование структуры базы данных	Методические указания к практическим занятиям	Устный опрос, проверка задания
2.	Создание таблиц с помощью мастера таблиц и конструктора таблиц	Методические указания к практическим занятиям	Устный опрос, проверка задания
3.	Заполнение и редактирование базы данных	Методические указания к практическим занятиям	Устный опрос, проверка задания
4.	Создание форм для ввода данных.	Методические указания к практическим занятиям	Устный опрос, проверка задания
5.	Создание и преобразование отчетов Разработка кнопочной формы-меню для работы с базами данных	Методические указания к практическим занятиям	Устный опрос, проверка задания
6.	Создание запросов в режиме SQL.	Методические указания к практическим занятиям	Устный опрос, проверка задания



**Паспорт  
фонда оценочных средств  
ПО ДИСЦИПЛИНЕ  
БАЗЫ ДАННЫХ**

**1. Модели контролируемых компетенций:**

Код компетенции	Формулировка компетенций
ПК-4	Владеть технологией проектирования и управления базами данных;

1.1. Компетенции формируемые в процессе изучения дисциплины (5 семестр):

1.2. Сведения об иных дисциплинах (преподаваемых в том числе на других кафедрах) участвующих в формировании данных компетенций;

- «Основы алгоритмизации и программирования»,
- «Технология разработки программных продуктов»,

1.2.1. Компетенция ПК-4 формируется в процессе изучения дисциплин (прохождения практик):

**2. В результате изучения дисциплины «Базы данных» студенты должны:**

**знать:**

- основные положения теории баз данных, хранилищ данных, баз знаний;
- основные принципы построения концептуальной, логической и физической модели данных;
- современные инструментальные средства разработки схемы базы данных;
- структуры данных СУБД, общий подход к организации представлений, таблиц и индексов;
- методы организации целостности данных;
- способы контроля доступа к данным и управления привилегиями;
- модели и структуры информационных систем.

**уметь:**

- создавать объекты базы данных в современных системах управления базами данных и управлять доступом к этим объектам;
- формировать и настраивать схему базы данных;
- применять стандартные методы для защиты объектов баз данных.

**владеть:**

- работы с объектами базы данных в конкретной системе управления базами данных;
- использования средств заполнения базы данных;
- использования стандартных методов защиты объектов баз данных.

\* Уровни обученности определяются ГОС ВПО по соответствующему направлению подготовки.

### 3. Программа оценивания контролируемой компетенции:

№	Контролируемые модули, разделы (темы) дисциплины*	Код контролируемой компетенции (или ее части)	Наименование оценочного средства**
1	<b>Введение.</b> Основные понятия теории баз данных.	ПК-4	Тест, контрольные вопросы
2	<b>Информационные модели.</b> Взаимосвязи в моделях и реляционный подход к построению моделей.	-//-/-	-//-/-
3	<b>Нормализация отношений.</b> Этапы проектирования баз данных	-//-/-	-//-/-
4	<b>Классификация и характеристика СУБД.</b> Обзор современных СУБД.	-//-/-	-//-/-
5	Постреляционные СУБД	-//-/-	-//-/-
6	<b>Работа с объектами</b> (таблицы, запросы, формы, отчеты и макросы)	-//-/-	-//-/-
7	<b>Краткая характеристика языка SQL.</b> Операторы языка SQL для работы с реляционной базой данных.	-//-/-	-//-/-
8	Организация запросов к базе данных на языке SQL	-//-/-	-//-/-

\* Наименование темы (раздела) или тем (разделов) берется из рабочей программы (силлабуса) дисциплины.

\*\* В графу наименование оценочного средства в обязательном порядке входит способ осуществления оценки компетенции (части контролируемой компетенции) (устно, письменно, компьютерные технологии и др.).

## Экспертное заключение

о соответствии разработанного фонда оценочных средств по дисциплине «БАЗЫ ДАННЫХ» специальности (направления подготовки) 230109 «Программное обеспечение вычислительных программ и автоматизированных систем»

требованиям ГОС СПО, основной образовательной программе по направлению подготовки, целям и задачам обучения, сформулированным в рабочей программе дисциплины

**Акишова Мира Карыбековна**

(Ф.И.О. разработчика ФОС)

Преподаватель колледж КГТУ

(должность, место работы)

№ пп.	Наименования показателей	Максимальный балл	Значение показателей/баллы	Оценка экспертов
<b>1</b>	<b>Соответствие ФОС установленным требованиям</b>			
		<b>6</b>		
1.1	ФГОС ВПО	<b>2</b>	да – 2; частично – 1; нет – 0;	
1.2	Основной образовательной программе по направлению подготовки	<b>2</b>	да – 2; частично – 1; нет – 0;	
1.3	Целям и задачам обучения, сформулированным в рабочей программе дисциплины	<b>2</b>	да – 2; частично – 1; нет – 0;	
<b>2.</b>	<b>Критерии качества ФОС</b>			
		<b>24</b>		
2.1	Соответствие разработанного ФОС результатам обучения	<b>4</b>	- полностью соответствует – 4; - частично соответствует – 2; - не соответствует – 0;	
2.2	Наличие индикаторов оценивания	<b>2</b>	да – 2; нет – 0;	
2.3	Полнота раскрытия контрольно-измерительных материалов, их вариативность, обеспечение контроля междисциплинарной траектории формирования компетенций у студентов при освоении ООП	<b>4</b>	- полностью соответствует – 4; - частично соответствует – 2; - не соответствует – 0;	
2.4	Наличие уровней освоения и развития компетенций, наличие уровней трудности тестовых заданий	<b>2</b>	да – 2; нет – 0;	
2.5	Наличие разнообразных форм заданий, практико-ориентированных заданий (постановка проблемных и перспективных вопросов,	<b>4</b>	да – 4; нет – 0;	

	советующих профессиональной деятельности) видам			
2.6	Использование компетентностного подхода, объективной внешней оценки, автоматизированной процедуры и т.п.)	2	- обосновано – 2; - частично обосновано – 1; - не обосновано – 0;	
2.7	Наличие качественных показателей заданий (надежность и валидность результатов контроля)	4	да – 4; нет – 0;	
2.8	Объем, полнота охвата и оригинальность материалов, в т.ч. результаты собственных научных исследований и т.д	2	- полностью соответствует – 2; - частично соответствует – 1; - не соответствует – 0;	
<b>Всего</b>				

«\_\_\_» \_\_\_\_\_ 20\_\_ г.

основной образовательной программе по направлению подготовки, целям и задачам обучения, сформулированным в рабочей программе дисциплины.

Подпись УМС колледжа \_\_\_\_\_

## КОНТРОЛЬНЫЕ ВОПРОСЫ:

- 1) Основные понятия теории баз данных
- 2) Виды информационных моделей: информационная модель данных и информационная модель предприятия.
- 3) Информационная модель данных. Концептуальная, логическая, внешняя и физическая модели. Условия первого и второго уровней независимости данных
- 4) Типы логических моделей: иерархическая, сетевая и реляционная
- 5) Основы реляционной алгебры
- 6) Взаимосвязи в моделях и реляционный подход к построению моделей
- 7) Нормализация отношений. Условия 1,2 и 3 НФ
- 8) Требования, предъявляемые к базе данных
- 9) Этапы проектирования баз данных
- 10) Классификация СУБД
- 11) Основные характеристики СУБД
- 12) Современные системы управления базами данных: Access, SQL-Server, Visual Basic, Visual C++, Visual Fox Pro
- 13) Основные направления совершенствования баз данных. Постреляционные СУБД.
- 14) Способы создания таблицы
- 15) Открытие таблицы, модификация структуры таблицы
- 16) Добавление новой записи в таблицу
- 17) Удаление записи из таблицы
- 18) Создание и открытие базы данных
- 19) Добавление таблиц в базу данных, освобождение таблиц
- 20) Установление взаимосвязи «один-к-одному»
- 21) Установление взаимосвязи «один-ко-многим»
- 22) Управление данными в базах данных
- 23) Типы реляционных языков
- 24) Язык запросов SQL

## Тест по дисциплине БАЗЫ ДАННЫХ

### Вариант №1

#### Тесты по дисциплине «Базы данных»

#### 1. Уровни проектирования БД

- А. Первый, второй и третий
- Б. Проектировочный и пользовательский
- В. Реляционный, иерархический и сетевой
- Г. Конструкторский и мастер

#### 2. База данных – это

- А. Совместно используемое единое хранилище для некоторого набора логически связанных данных по определенной предметной области.
- Б. это представление реальности, отражающего лишь избранные детали.
- В. хранения специально организованной информации по определенной предметной области
- Г. это значение, которое однозначно определяет элемент объектного множества.

#### 3. Внешний ключ – это

- А. Древовидные иерархические структуры широко используются в повседневной человеческой деятельности
- Б. Идентификатор
- В. Система баз данных
- Г. Набор лексических атрибутов, значения которых всегда однозначно определяют элемент объектного множества

#### 4. Сетевые модели данных

- А. организует и представляет данные в виде таблиц или реляций.
- Б. базируются на использовании графовой формы представления данных. Вершина графа используются для интерпретации типов сущностей
- В. приведения реляционных таблиц к стандартному виду. Правила нормализации:
- Г. Противоречивость данных, вызванная их избыточностью и частичным обновлением.

#### 5. язык DDL (Data Definition Language) это

- А. предназначенный, для выборки и обновления данных.
- Б. разделения таблицы на несколько таблиц в целях избавления от аномалий и поддержания целостности данных
- В. Повторение данных в базе данных
- Г. предназначенный для определения структур базы данных;

#### 6. В базе данных используется следующие способы создания таблиц в MS ACCESS:

- А. через Конструктор;
- Б. Через Мастер;
- В. через поле ввода данных;
- Г. все вышеперечисленные

#### 7. Запросы служат:

- А. Для выбора более чем одной таблицы одновременно
- Б. служат для селекции и фильтрации набора данных
- В. выполняет роль хранилища данных
- Г. Вывода данных на экран

#### 8. Типы данных MS ACCESS:

- А. Запросы, таблицы и формы
- Б. Отчеты, макросы и модули
- В. Тестовый, числовой, логический, дата/время, OLE, счетчик
- Г. Конструктор и мастер

#### 9. Макросы - это

- А. последовательность операций, записанных в виде инструкций на специальном языке.
- Б. MS ACCESS
- В. Базы данных
- Г. совокупность объявлений (деклараций) и последовательностей исполняемых команд

#### 10. Нормализация.

- А. нормальные формы
- Б. Согласованность данных в базе данных
- В. избавления от аномалий и поддержания целостности данных
- Г. Процесс приведения реляционных таблиц к стандартному виду.

*Тесты по дисциплине «Базы данных»*  
**ВАРИАНТ №2**

**1. Главными элементами модели данных**

- А. запросы и таблицы
- Б. объекты и отношения.
- В. Сетевая и реляционная модели
- Г. Текстовый и логический

**2. Системы базы данных состоят из главных компонента:**

- А. Аномалия обновления, удаления и ввода
- Б. один к одному, один ко многим, и многие ко многим
- В. данные, аппаратное обеспечение, программное обеспечение и пользователи.
- Г. DDL (Data Definition Language) и DML (Data Manipulation Language)

**3. Реляционная модель данных**

- А. Количество атрибутов реляции
- Б. это термин, пришедший из математики
- В. организует и представляет данные в виде таблиц или реляций.
- Г. Количество НЕ атрибутов реляции

**4. Модель «Клиент-сервер» – технология**

- А. Графические возможности оболочки производят большое впечатление при изготовлении высококачественных отчетов и распечаток.
- Б. технология, разделяющая приложение- СУБД на две части: клиентскую и сервер,
- В. Они позволяют выбрать из базы только необходимую информацию, т.е. ту, которая соответствует определенному критерию
- Г. когда размеры баз данных велики

**5. В данном примере поле адрес ул. Душанбинская №45 относится к типу данных MS Access:**

- А. дата/время
- Б. числовой
- В. текстовый
- Г. логический

**6. Сколько форм нормализации:**

- А. 5
- Б. 4
- В. 2
- Г. 3

**7. Что означает данная запись: 1-∞**

- А. связи
- Б. один – к - одному
- В. связь одни – ко - многим
- Г. Многие – ко - многим

**8. Реляционная таблица находится в первой нормальной форме (1НФ)**

- А. если никакие неключевые атрибуты не являются функционально зависимыми лишь от части ключа
- Б. если для любой ФЗ: X → Y X является ключом
- В. условие, обеспечивающее независимость атрибутов путем обязательного повторения значений
- Г. если значения в таблице являются атомарными для каждого атрибута таблицы

**9. Последовательная организация файла означает:**

- А. Записи обычным образом физически упорядочены по значениям первичного ключа
- Б. Записи обычным образом физически упорядочены по значениям не первичного ключа
- В. что записи упорядочены по значению ключа
- Г. в качестве способа адресации записей избавляет от необходимости поддерживать и просматривать индексы

**10. SQL – запросы это:**

- А. Представляют Ваши данные в компактном суммированном формате.
- Б. Пользователь формулирует их с использованием инструкций и функций, выстраивая описание
- В. Находят дубликатные записи в выбранной Вами таблице или запросе.
- Г. Копируют записи из существующей таблицы в новую и затем удаляют (по желанию) эти записи из таблицы оригинала

Группа \_\_\_\_\_ ФИО \_\_\_\_\_



**Тесты по дисциплине «Базы данных»**  
**ВАРИАНТ №3**

**1. Рекурсивное отношение это-**

- А. Столбец реляции
- Б. значение, которое однозначно определяет элемент объектного множества
- В. Древовидные иерархические структуры
- Г. Отношение, связывающее объектное множество с ним самим.

**2. Составной ключ**

- А. Любой набор атрибутов, который может быть выбран в качестве ключа таблицы.
- Б. Ключ, содержащий два или более атрибута.
- В. Потенциальный ключ, выбранный для преимущественного использования в целях однозначного определения строк таблицы
- Г. Потенциальный ключ

**3. Ограничительное условие.**

- А. Строки реляционной таблицы представляют в базе данных элементы конкретных объектов реального мира или категорий
- Б. Список, содержащий имена реляционных таблиц, имена атрибутов, ключевые атрибуты и внешние ключи
- В. Правило, ограничивающее значения данных в базе данных.
- Г. Никакой ключевой атрибут строки не может быть пустым.

**4. Процесс приведения реляционных таблиц к стандартному виду - это**

- А. Стандартизация
- Б. Типы данных
- В. Схема данных
- Г. Нормализация.

**5. Повторение данных в базе данных**

- А. Целостность данных
- Б. Избыточность данных
- В. Противоречивость данных, вызванная их избыточностью и частичным обновлением
- Г. Аномалия удаления

**6. Функциональная зависимость.**

- А. Атрибут (ы) в левой части функциональной зависимости
- Б. Никакие неключевые атрибуты не являются функционально зависимыми лишь от части ключа
- В. Значение атрибута в кортеже однозначно определяет значение другого атрибута в кортеже
- Г. Таблица, состоящая из нескольких выбранных атрибутов

**7. Транзитивная зависимость.**

- А. Первая нормальная форма запрещает таблицам иметь неатомарные
- Б. если таблица удовлетворяет НФБК
- В. Неключевой атрибут функционально зависит от одного или более неключевых атрибутов
- Г. если связи между независимыми атрибутами можно исключить

**8. Реляционная модель данных**

- А. Основана на логических отношениях данных
- Б. Модель данных, представляющая данные в виде таблиц.
- В. Основана на концептуальной модели данных
- Г. Реляционная модель данных организует и представляет данные

**9. Объекты MS Access:**

- А. Таблицы, запросы, формы, отчеты
- Б. Текстовое поле, логическое поле
- В. Атрибуты
- Г. Выше перечисленные

**10. Создание новых объектов базы данных: объекта формы**

- А. Лот № Наименование товара, Единица измерения, Склад №
- Б. Текстовое поле, логическое поле
- В. В столбец, Ленточная, Табличная, Сводная таблица
- Г. Выше перечисленные

*Тесты по дисциплине «Базы данных»*  
**ВАРИАНТ №4**

**1. В данной таблице «Крупа манная» имеет тип данных**

Лот №	Наименование товара	Единица измерения	Склад №
101	Сахарный песок	50	1
102	Крупа манная	25	2
103	Крупа гречневая	50	1

- А. логический
- Б. текстовый
- В. числовой
- Г. Наименование товара

**2. Какое поле может быть ключевым (верхняя таблица):**

- А. Наименование
- Б. Единица измерения
- В. Лот №
- Г. Склад №

**3. Разбиение — это процесс**

- А. значения в таблице являются атомарными для каждого атрибута таблицы
- Б. разделения таблицы на несколько таблиц в целях избавления от аномалий и поддержания целостности данных
- В. введенные в атрибут BLDG-ID
- Г. идентификатор конкретного здания

**4. Запрос: Отобразить продажи в октябре**

- А.  $>=01.10.2005$  \_\_ And \_\_  $<=31.11.2005$
- Б. Is \_\_ not \_\_ null
- В. LIKE \_\_ [введите название]
- Г. LIKE \_\_ [введите наименование товара]

**5 Запрос: Создать запрос на вычисление суммы к оплате**

- А. = 0
- Б. [Цена отгрузки]\*1,05
- В. LIKE \_\_ [введите название]
- Г. К оплате: [Количество продажи]\*[Цена отгрузки]

**6. Таблица «Клиенты» включает поля:**

- А. Лот №, Наименование товара, Единица измерения, Склад №, Дата поставки, Поставщик, Количество поставки, Цена поставки, Цена отгрузки
- Б. Название, ИНН, Обращаться к, Должность, Адрес, Телефон, Факс
- В. Накладная №, Дата продажи, ИНН, Лот №, Количество продажи
- Г. Параметрический, выборка

**7. Поле «Количество поставки» имеет тип данных:**

- А. текстовый
- Б. числовой
- В. денежный
- Г. Дата/ время

**8. Составное объектное множество**

- А. Максимальное количество элементов одного объектного множества, связанных с одним элементом другого объектного множества.
- Б. Мощность отношения конкретизации
- В. Отношение, рассматриваемое как объектное множество.
- Г. Отношение, максимальная мощность которого как минимум в одном направлении равна одному

**9. Атрибуты это-**

- А. запросы и формы
- Б. базы данных
- В. имя, дата рождения, номер страховки, рост, вес, пол,
- Г. таблицы

**10. Мощность отношения обозначает**

- А. число элементов одного множества, связанных посредством отношения с одним элементом другого объектного множества
- Б. Модель данных состоит из объектных множеств
- В. Отношение устанавливает связи между элементами двух объектных множеств
- Г. Объектные множества представляют категории

Группа \_\_\_\_\_ ФИО \_\_\_\_\_

**Тесты по дисциплине «Базы данных»**  
**ВАРИАНТ №5**

**1. Запрос с параметрами -**

- А. это запрос, создаваемый при помощи языка SQL
- Б. это запрос, при выполнении отображающий в собственном диалоговом окне приглашение ввести данные
- В. объекты, которые идут на печать
- Г. С помощью макросов можно выполнить практически все действия над объектами

**2. База данных - это**

- А. - совокупность языковых и программных средств, предназначенных для создания, ведения и эксплуатации баз данных;
- Б. - Банк данных (БнД) – совокупность баз данных и системы управления базами данных.
- В. – формализованное описание, отражающее состав и типы данных, а также взаимосвязь между ними.
- Г. -совокупность структурированных данных.

**3.Кортеж в реляционной модели это -**

- А. -строка;
- Б. -столбец;
- В. –таблица
- Г. отношения

**4. Язык T-SQL делиться на:**

- А. DML, DDL, DOL,
- Б. DDL, DKL, DML
- В. DCL,DML, DDL,
- Г. DOL, DML,DDL

**5. Select - это команда**

- А. DML
- Б. DDL
- В. DCL
- Г. KDL

**6. Предложение order by**

- a. сортирует
- b. группирует
- c. выбирает

**7. Главными элементами модели данных**

- А. запросы и таблицы
- Б. объекты и отношения.
- В. Сетевая и реляционная модели
- Г. Текстовый и логический

**8. Системы базы данных состоят из главных компонента:**

- А. Аномалия обновления, удаления и ввода
- Б. один к одному, один ко многим, и многие ко многим
- В. данные, аппаратное обеспечение, программное обеспечение и пользователи.
- Г. DDL (Data Definition Language) и DML (Data Manipulation Language)

**9. Реляционная модель данных**

- А. Количество атрибутов реляции
- Б. это термин, пришедший из математики
- В. организует и представляет данные в виде таблиц или реляций.
- Г. Количество HE атрибутов реляции

**10. Модель «Клиент-сервер» – технология**

- А. Графические возможности оболочки производят большое впечатление при изготовлении высококачественных отчетов и распечаток.
- Б. технология, разделяющая приложение- СУБД на две части: клиентскую и сервер,
- В. Они позволяют выбрать из базы только необходимую информацию, т.е. ту, которая соответствует определенному критерию
- Г. когда размеры баз данных велики

**Тесты по дисциплине «Базы данных»  
ВАРИАНТ №6**

**1. В данном примере поле адрес ул. Душанбинская №45 относится к типу данных MS Access:**

- А. дата/время
- Б. числовой
- В. текстовый
- Г. логический

**2. Сколько форм нормализации:**

- А. 5
- Б. 4
- В. 2
- Г. 3

**3. Процесс приведения реляционных таблиц к стандартному виду - это**

- А. Стандартизация
- Б. Типы данных
- В. Схема данных
- Г. Нормализация.

**4. Повторение данных в базе данных**

- А. Целостность данных
- Б. Избыточность данных
- В. Противоречивость данных, вызванная их избыточностью и частичным обновлением
- Г. Аномалия удаления

**5. Функциональная зависимость.**

- А. Атрибут (ы) в левой части функциональной зависимости
- Б. Никакие неключевые атрибуты не являются функционально зависимыми лишь от части ключа
- В. Значение атрибута в кортеже однозначно определяет значение другого атрибута в кортеже
- Г. Таблица, состоящая из нескольких выбранных атрибутов

**6. Транзитивная зависимость.**

- А. Первая нормальная форма запрещает таблицам иметь неатомарные
- Б. если таблица удовлетворяет НФБК
- В. Неключевой атрибут функционально зависит от одного или более неключевых атрибутов
- Г. если связи между независимыми атрибутами можно исключить

**7. Реляционная модель данных**

- А. Основана на логических отношениях данных
- Б. Модель данных, представляющая данные в виде таблиц.
- В. Основана на концептуальной модели данных
- Г. Реляционная модель данных организует и представляет данные

**8. Объекты MS Access:**

- А. Таблицы, запросы, формы, отчеты
- Б. Текстовое поле, логическое поле
- В. Атрибуты
- Г. Выше перечисленные

**9. Создание новых объектов базы данных: форм**

- А. Лот № Наименование товара, Единица измерения, Склад №
- Б. Текстовое поле, логическое поле
- В. В столбец, Ленточная, Табличная, Сводная таблица
- Г. Выше перечисленные

**10. Внешний ключ – это**

- А. Древовидные иерархические структуры широко используются в повседневной человеческой деятельности
- Б. Идентификатор
- В. Система баз данных
- Г. Набор лексических атрибутов, значения которых всегда однозначно определяют элемент объектного множества

Группа \_\_\_\_\_ ФИО \_\_\_\_\_

## *Экзаменационные билеты*

<b>Экзаменационный билет по «Базы данных» Билет № 1</b>
1) Понятие о базах данных и о системах управления базами данных. Базовые уровни представления БД (логический, концептуальный, физический). 2) Порядок в домене. Частичный порядок. Полный порядок.
Директор колледжа КГТУ <span style="float: right;">Бейшеева <u>Г.С</u></span>
<b>Экзаменационный билет по «Базы данных» Билет № 2.</b>
1) Понятие о модели данных. Множественность моделей. Типы моделей. 2) Нормальные формы (НФ) реляционных баз данных: НФ1, НФ2.
Директор колледжа КГТУ <span style="float: right;">Бейшеева <u>Г.С</u></span>
<b>Экзаменационный билет по «Базы данных» Билет № 3.</b>
1) Индексирование таблиц. Время поиска как функция от размера таблиц. 2) Нормальные формы (НФ) реляционных баз данных: НФ3, НФ3БК.
Директор колледжа КГТУ <span style="float: right;">Бейшеева <u>Г.С</u></span>
<b>Экзаменационный билет по «Базы данных» Билет № 4.</b>
1) Применение функций расстановки ("хэширование"). 2) Нормальные формы (НФ) реляционных баз данных: НФ4.
Директор колледжа КГТУ <span style="float: right;">Бейшеева <u>Г.С.</u></span>
<b>Экзаменационный билет по «Базы данных» Билет № 5</b>
1) Понятие о транзакции, о программно-независимости, целостности и защищенности данных. Оператор GRANT в SQL. 2) Нормальные формы (НФ) реляционных баз данных: НФ5.
Директор колледжа КГТУ <span style="float: right;">Бейшеева <u>Г.С</u></span>

**Экзаменационный билет по «Базы данных»  
Билет № 6**

- 1) Сопоставление понятий файл, набор данных, запись, отношение, таблица, домен, столбец, строка, атрибут, поле, ключ в соответственных контекстах.
- 2) Принцип дедукции.

Директор колледжа КГТУ

Бейшеева Г.С

**Экзаменационный билет по «Базы данных»  
Билет № 7**

- 1) Алгоритмы поиска и время поиска как функция от размера таблиц.
- 2) Хорновские дизъюнкты.

Директор колледжа КГТУ

Бейшеева Г.С

**Экзаменационный билет по «Базы данных»  
Билет № 8**

- 1) Аксиомы Армстронга.
- 2) Реляционные таблицы. Операция Set relation над таблицами.

Директор колледжа КГТУ

Бейшеева Г.С

**Экзаменационный билет по «Базы данных»  
Билет № 9**

- 1) Рекурсивно-перечислимые отношения, частично рекурсивные функции (в терминах канонических систем Поста).
- 2) Принцип резолюции.

Директор колледжа КГТУ

Бейшеева Г.С.

**Экзаменационный билет по «Базы данных»  
Билет № 10**

- 1) Отношение в терминах теории множеств. Понятие о решетке. Понятие о полных решетках.
- 2) Принципы реляционности СУБД Кодда.

Директор колледжа КГТУ

Бейшеева Г.

**Экзаменационный билет по «Базы данных»**

**Билет № 11**

- 1) Вложенные SELECT в SQL.
- 2) Реляционная алгебра: объединение, разность, пересечение отношений.

**Директор колледжа КГТУ**

Бейшеева Г.С

**Экзаменационный билет по «Базы данных»**

**Билет № 12**

- 1) Расширенное декартово произведение отношений.
- 2) Выразить на SQL запрос: "имена поставщиков красных деталей".

**Директор колледжа КГТУ**

**Бейшеева Г.С**

**Экзаменационный билет по «Базы данных»**

**Билет № 13**

- 1) Реляционная алгебра: проекция и селекция отношения.
- 2) Выразить на SQL запрос: "имена поставщиков всех деталей".

**Директор колледжа КГТУ**

Бейшеева Г.С

**Экзаменационный билет по «Базы данных»**

**Билет № 14**

1. Производственный метод.
2. Понятие накладных расходов.
3. Оценка товарно-материальных запасов.

**Директор колледжа КГТУ**

Бейшеева Г.С

**Экзаменационный вопрос по «Базы данных»**

**Билет № 15**

- 1) Представление о теореме Кодда о редукции.
- 2) Теорема о существовании неразрешимых рекурсивно перечислимых отношений.

**Директор колледжа КГТУ**

Бейшеева Г.С



## **Список используемой литературы**

### **Основная:**

№	Название литературы	Автор	Место изд-во, год издания	Кол-во экз. имеющ. в биб. КГУСТА
1	SQL Server 2008	Степанов А.Н.	- СПб. 2010.	
2	Базы данных. Модели данных. Учебник – М.: ООО	С.Д.Кузнецов.	«Бином-Пресс», 2018 г.	4
3	Базы данных. Язык SQL для студента. 2-е издание.	В.В.Дунаев.	– СПб.: БХВ-Петербург, 2019.	4
4	R2. Black book.	Dreamtech Ansari	Road, Daryaganj,PRESS, 19-A, New Delhi, 2018	4

### **Дополнительная:**

№	Название литературы	Автор	Место изд-во, год издания	Кол-во экз. имеющ. в биб. КГУСТА
1	Обеспечение высокой доступности систем на основу MySql	Телманн Г., Киндалл М., Белл Ч.	БХВ – Петербург г. .,2018.	5

# КОНСПЕКТ ЛЕКЦИЙ

## Тема №1. Введение. Основные понятия теории баз данных.

Рассмотрим основные понятия, связанные с развитием концепций баз данных.

**База данных (БД)** – структурированная совокупность данных, организованную по определенным правилам.

БД используются для хранения информации об объектах какой-либо предметной области.

**Предметная область** - часть реального мира, подлежащая изучению с целью автоматизации. Предметную область можно представить как множество взаимосвязанных объектов.

**Объект (сущность)** - это выделенный элемент предметной области, подлежащий хранению в БД. Другими словами - это «нечто, о чем мы хотим хранить информацию в БД». Объект может быть реальным (человек, населенный пункт, какой-либо предмет) и абстрактным (событие, счет покупателя).

Для каждого объекта выделяют набор признаков (характеристик, свойств или атрибутов) которые позволяют описать объект в рамках выбранной предметной области.

Если рассматривать человека как объект, о котором мы хотим хранить информацию в БД, то можно заметить что для предметных областей связанных с медициной наиболее значимыми наборами характеристик человека могут оказаться: рост, вес, пол и т.д. Для производства набор значимых характеристик человека иной: возраст, должность, рейтинг и т.д.

**Характеристики (свойства, атрибуты)** – набор признаков определяющих объект для выбранной предметной области.

**Данные** (в концепции БД) – это набор конкретных значений, параметров, характеризующих объект.

Не следует путать характеристики и данные, например, «ВЕС» – это характеристика объекта, а 120кг – это конкретное значение (данные).

**Класс объектов** - совокупность объектов, обладающих одинаковым набором свойств (характеристик).

**Пример.** Секретарь учебного заведения должен учитывать контактную информацию об учащихся (адрес, телефон и т.д.). Эту информацию удобно расположить в таблице (таблица 1.1). Столбцы таблиц обычно называют **полями**, а строки – **записями**. Каждая запись таблицы (строка) содержит индивидуальные **данные** конкретного ученика – **объекта**. Заголовок таблицы представляет собой набор **свойств (характеристик)**. Все однотипные объекты (ученики) составят **класс объектов**. Можно сказать, что мы получили однотоличную **БД**.

Таблица 1 – Контактная информация учащихся

№ п/п	Фамилия	Имя	Телефон	Адрес
1	Александров	Саша	5-55-57	пр. Ленина, д. 25, кв. 17
2	Андреев	Степан	5-15-23	ул. Чапаева, д. 18, кв. 8
3	Башкирова	Ольга	5-50-44	ул. Северная, д. 117
...	...	...	...	...

Система управления базами данных (СУБД) – это совокупность языковых и программных средств, предназначенных для создания, ведения и использования БД.

СУБД позволяют структурировать, систематизировать и организовывать данные для компьютерного хранения и обработки. Системы управления базами данных (DataBase Management System – DBMS) являются основами практически любой информационной системы. Использование современных СУБД позволяет дать следующие преимущества:

1. Может быть сокращена избыточность в хранимых данных.

2. Может быть устранена возможность возникновения противоречивости хранимых данных.
3. Централизованное управление обеспечивает соблюдение стандартов в представлении данных, принятых в данной предметной области.
4. Могут быть выполнены условия безопасности данных.
5. Может поддерживаться целостность данных. Благодаря централизованному управлению, могут быть определены процедуры проверки, выполняющиеся при операциях запоминания.
6. Может быть обеспечена независимость данных и приложений, т.е. возможность расширения приложений независимо от базы данных и наоборот возможность расширения базы данных без влияния на приложения.

Информационная система (ИС) - взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели. В большинстве случаев сегодня под ИС понимают автоматизированные информационные системы (АИС).

АИС позволяют автоматизировать деятельность, повысить качество и достоверность обрабатываемой информации. Информационную основу ИС составляют хранящиеся в ней данные. В большинстве случаев это БД, а для управления данными используют СУБД.

## Тема №2. Информационные модели. Взаимосвязи в моделях и реляционный подход к построению моделей.

### Понятие и классификация моделей данных

Ядром любой базы данных является модель данных. С помощью модели данных могут быть представлены объекты предметной области и взаимосвязи между ними.

*Модель данных – совокупность структур данных и операций их обработки.*

Рассмотрим 3 основных типа моделей данных: иерархическую, сетевую и реляционную.

а) иерархическая модель данных

Иерархическая модель базы данных представляет собой совокупность элементов, расположенных в порядке их подчинения от общего к частному и образующих перевернутое дерево (граф).

К основным понятиям иерархической структуры относятся: уровень, элемент (узел), связь.

Узел – это информационная модель элемента, находящегося на данном уровне иерархии. На схеме иерархического дерева узлы представляются вершинами графа.

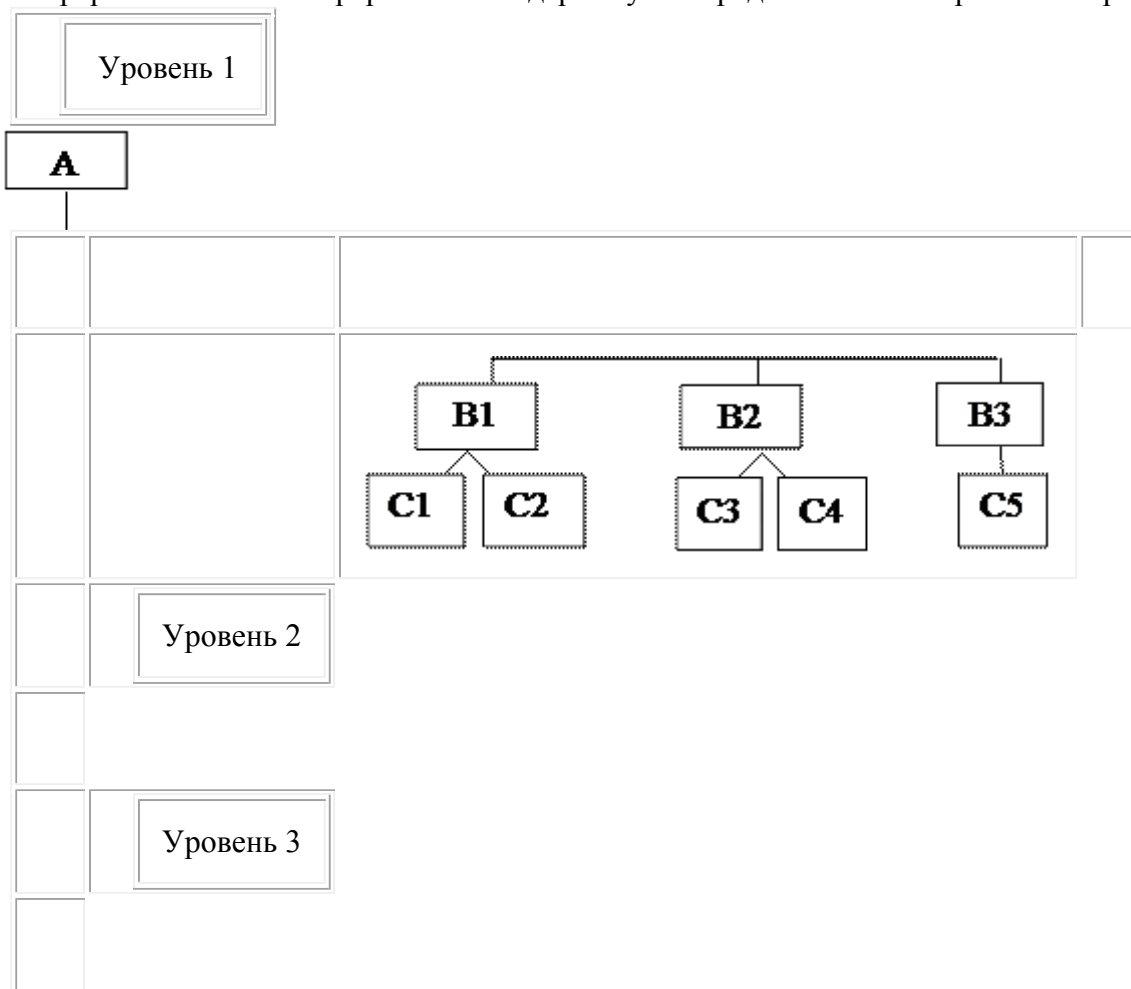
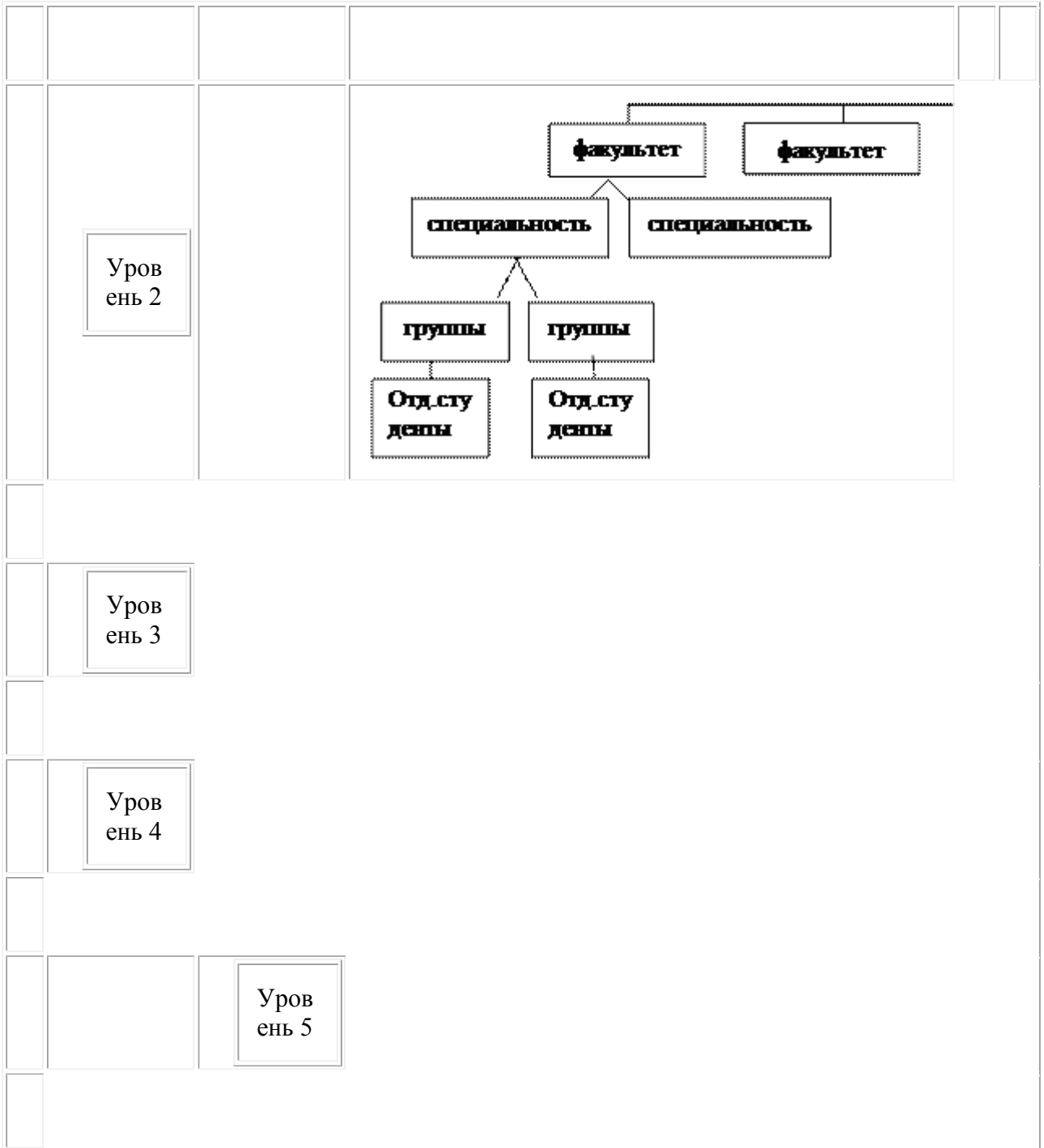


Рис. 1 Графическое изображение иерархической структуры БД

Пример. Иерархическая модель «ВУЗ».

Уровень 1

**ВУЗ**



Свойства иерархической модели:

- несколько узлов низшего уровня связано только с одним узлом высшего уровня;
- иерархическое дерево имеет только одну вершину (корень дерева), не подчиненный никакой другой вершине и находящуюся на самом верхнем (первом) уровне. Зависимые (подчиненные) узлы находятся на втором, третьем и т.д. уровнях.
- каждый узел имеет свое имя (идентификатор).

- количество деревьев в базе данных определяется числом корневых записей;

б) сетевая модель данных:

Сетевая модель имеет те же основные составляющие (узел, уровень, связь). Однако в ней принята свободная связь между элементами разных уровней, т.е. каждый элемент может быть связан с любым другим элементом.

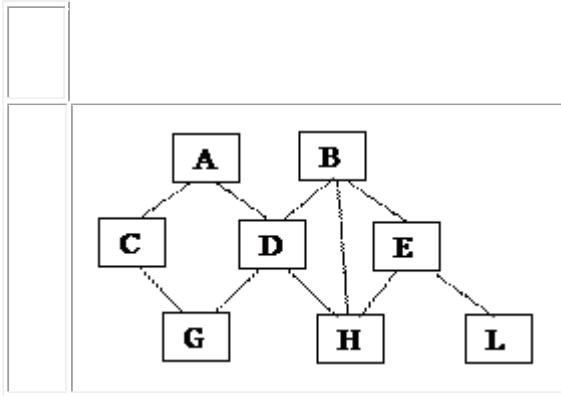
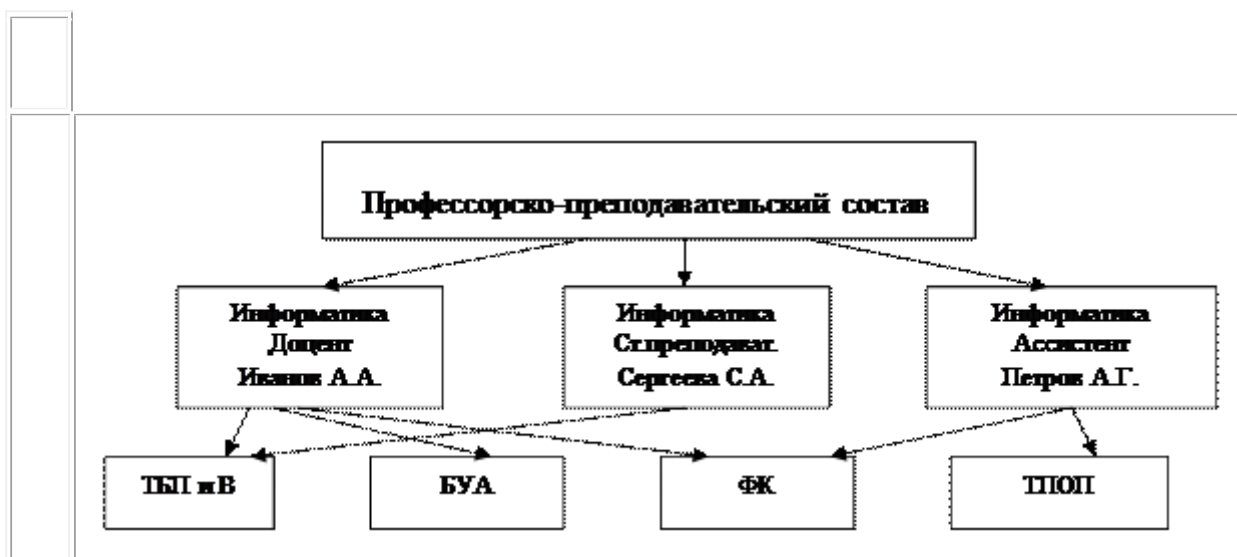


Рис. 2 Графическое изображение сетевой структуры БД

Пример. Сетевая модель «Профессорско-преподавательский состав»



в) реляционная модель данных (табличная)

Термин «реляционный» произошел от англ. слова relation – отношение.

Отношение – математическое понятие, но в терминологии моделей данных отношения удобно изображать в виде таблицы.

Теоретической основой этой модели стала теория отношений американца Чарльза Пирса и немца Эрнеста Шредера. Ими было показано, что множество отношений замкнуто относительно некоторых специальных операций и образует вместе с ними абстрактную алгебру. Американский математик Э.Ф. Кодд в 1970 г. впервые сформулировал основные понятия и ограничения реляционной модели, ограничив набор операций в ней семью основными и одной дополнительной.

Реляционная модель хранения данных построена на взаимоотношении составляющих ее частей. В простейшем случае она представляет собой двухмерный массив или двухмерную таблицу, а при создании сложных информационных моделей составляет совокупность взаимосвязанных таблиц.

Пример реляционной таблицы:

№ личного дела	Фамилия	Имя	Отчество	Дата рождения	Группа
	Костин	Владимир	Владимирович	01.03.78	БУА
	Антонов	Юрий	Петрович	18.09.80	ФК

Реляционная модель базы данных имеет следующие свойства:

- 1) каждый элемент таблицы – один элемент данных;
- 2) все столбцы в таблице являются однородными, т.е. имеют один тип (числа, текст, дата и т.д.)
- 3) каждый столбец (поле) имеет уникальное имя;

4) одинаковые строки в столбце отсутствуют;

5) порядок следования строк и столбцов может быть произвольным.

Отношения представлены в виде таблиц, строки которых соответствуют кортежам или записям, а столбцы – атрибутам отношений, доменам, полям.

Если реляционная модель данных состоит из нескольких таблиц, то они связываются между собой ключами.

Ключ – поле, которое однозначно определяет соответствующую запись (ключевое поле).

В данном примере в качестве ключа может служить номер личного дела студента.

План лекции:

1. Понятие «информационный объект»
2. Нормализация отношений
3. Построение инфологической модели

Информационный объект – это описание некоторой сущности (реального объекта, явления, процесса, события) в виде совокупности логически связанных реквизитов (информационных элементов). Такими сущностями для информационных объектов могут служить: цех, склад, вуз, студент и т.д.

Информационный объект определенного реквизитного состава и структуры образует класс (тип), которому присваивается уникальное имя (символьное обозначение), например Студент, Сессия, Стипендия.

Информационный объект имеет множество реализаций – экземпляров, каждый из которых представлен совокупностью конкретных значений реквизитов и идентифицируется значением ключа (простого – один реквизит или составного – несколько реквизитов). Остальные реквизиты информационного объекта являются описательными.

Пример. 1. На рис. 1 представлен пример структуры и экземпляров информационного объекта Студент

В информационном объекте Студент ключом является реквизит Номер (№ личного дела), к описательным реквизитам относятся: Фамилия (фамилия студента), Имя (имя студента), Отчество (отчество студента), Дата (дата рождения), Группа (№ группы). Если отсутствует реквизит Номер, то для однозначного определения характеристик конкретного студента необходимо использование составного ключа из трех реквизитов: Фамилия + Имя + Отчество.

Структура	№ личного дела	Фамилия	Имя	Отчество	Дата	Группа
Экземпляры инф. объекта Студент		Костин	Владимир	Владимирович	01.03.78	БУА
	Антонов	Юрий	Петрович	18.09.80	ФК	

Рис. 1 - Пример структуры и экземпляров информационного объекта

### Тема №3. Нормализация отношений. Этапы проектирования баз данных

При создании базы данных следует определить количество таблиц (отношений) БД и их атрибутивный состав, который должен быть рациональным. Суть рациональности заключается в минимизации дублирования данных и упрощении процедуры их обработки и обновления. Классическая технология создания реляционных БД связана с теорией нормализации, которая позволяет улучшить характеристики БД и основана на анализе функциональной



зависимости между атрибутами отношений. Процесс нормализации заключается в разложении (декомпозиции) исходных отношений БД на более простые отношения. Каждая ступень этого процесса приводит схему отношений в последовательные нормальные формы. Для каждой ступени нормализации имеются наборы ограничений, которым должны удовлетворять отношения БД. Нормализация позволяет удалить из таблиц базы избыточную неключевую информацию.

При этом можно выделить следующую последовательность нормальных форм:

- первая нормальная форма (1 НФ);
- вторая нормальная форма (2НФ);
- третья нормальная форма (3НФ);
- нормальная форма Бойса—Кодда (4НФ),
- четвертая нормальная форма (4НФ);
- пятая нормальная форма, или нормальная форма проекции- соединения (5НФ или 5НФ).

Основные свойства нормальных форм состоят в том, что каждая следующая нормальная форма лучше предыдущей и при переходе к следующей нормальной форме свойства предыдущих сохраняются.

Первая нормальная форма (1НФ). Отношение называется нормализованным или приведенным к первой нормальной форме, если все его атрибуты простые (атомарные или неделимые). Примеры

1. Отношение Товар = (Код товара: Название; Цвет; Вес; Цена\_за\_единицу) находится в 1НФ, так как все атрибуты в данном отношении являются простыми.

2. Отношение Сотрудник = (Таб номер: ФИОсотрудника; Дата рождения; Дети) не находится в 1НФ, так как атрибуты «ФИО сотрудника», «Дата\_рождения», «Дети» могут быть сложными:

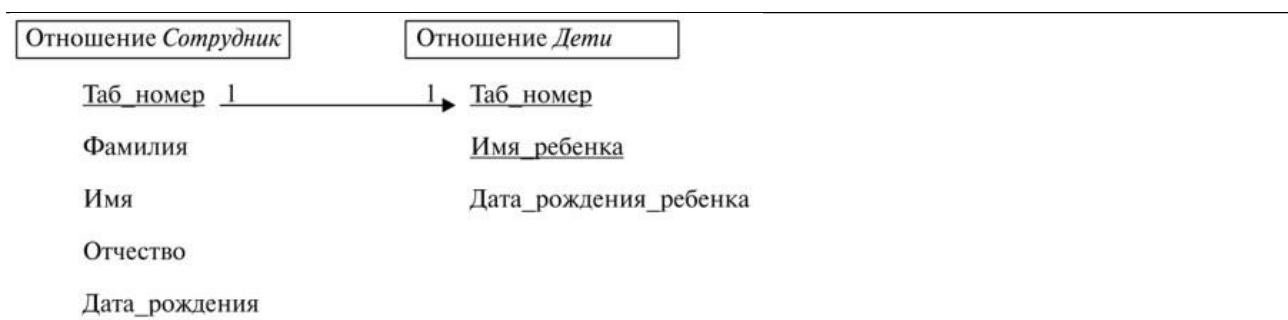
- атрибут «ФИО\_сотрудника» содержит три части — фамилию, имя и отчество сотрудника фирмы, поэтому должен быть разбит на три атомарных атрибута, каждый из которых может использоваться самостоятельно;
- атрибут «Дата\_рождения», как и атрибут «ФИО\_сотрудника», не является атомарным, поскольку включает в себя три понятия — «Год», «Месяц» и «День». Однако в БД существуют типы данных — Date и Date Time, которые позволяют выполнять над датами специальные операции, например, <дата\_1> — <дата\_2> = <количество\_дней\_между\_ними>. В связи с этим атрибуты дат в базах данных считаются атомарными;
- атрибут «Дети» также является сложным, так как может содержать сведения об имени ребенка и его дату рождения.

Для приведения отношения Сотрудник к 1НФ необходимо декомпозировать его на следующие два отношения:

Сотрудник = (Таб номер: Фамилия; Имя; Отчество; Дата\_рождения)

Дети = (Таб номер: Имя ребенка; Дата\_рождения\_ребенка)

При этом связь между этими отношениями осуществляется через ключевое поле «Таб\_номер», тип связи «один к одному».



Вторая нормальная форма (2НФ). Вторая нормальная форма к требованию атомарности атрибутов добавляет еще одно — каждый неключевой атрибут должен функционально полно зависеть от первичного ключа (не должен зависеть от части составного ключа).

Отношение находится во второй нормальной форме, если оно удовлетворяет определению 1НФ и все ее поля, не входящие в первичный ключ, связаны полной функциональной зависимостью с первичным ключом.

Примеры

1. Отношение Товар = (Код товара: Название; Цвет; Вес; Цена\_за единицу) находится в 1НФ и во 2НФ одновременно, так как все атрибуты в данном отношении однозначно определены и функционально зависят от ключа «Код\_товара».

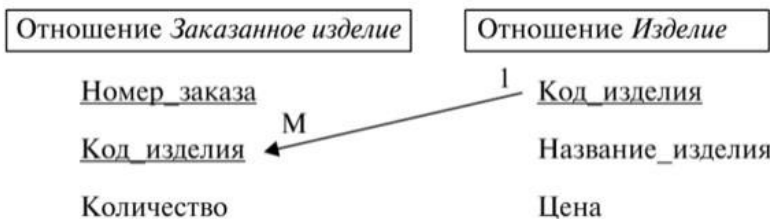
2. Отношение Заказанное изделие = (Номер заказа: Код изделия: Название\_изделия; Цена; Количество) не находится во 2НФ, так как атрибут «Количество» зависит от составного ключа (Номер заказа и Код изделия), а атрибуты «Название\_изделия» и «Цена» зависят только от части составного ключа «Код\_изделия».

Для приведения отношения Заказанное изделие к 2НФ необходимо декомпозировать его на следующие два отношения:

Заказанное\_изделие = (Номер заказа: Код изделия: Количество)

Изделие = (Код изделия: Название\_изделия; Цена)

Связь между этими отношениями осуществляется через ключевое поле «Код\_изделия».



Третья нормальная форма (3НФ) подразумевает атомарность и функционально полную зависимость атрибутов каждой сущности от ее первичного ключа. Кроме того, между неключевыми атрибутами сущности должны отсутствовать транзитивные зависимости, т.е. они должны быть взаимно независимы.

Отношение находится в 3НФ, если оно находится во 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

Поясним понятие транзитивной зависимости.

Имеем отношение  $O$  с атрибутами  $A, B, C$ . Если  $C$  зависит от  $B$ , а  $B$ , в свою очередь, зависит от  $A$ , то считается, что  $C$  транзитивно зависит от  $A$  (рис. 2.13).

Преобразование в 3НФ состоит в декомпозиции исходного отношения на два отношения (рис. 2.14).

Пример

Отношение Сотрудник = (Таб номер: Фамилия; Должность; Оклад; Дата\_зачисления; Дата\_увольнения) не находится в 3НФ, так

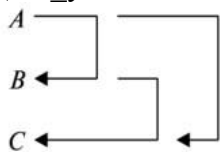


Рис. 2.13. Транзитивная зависимость

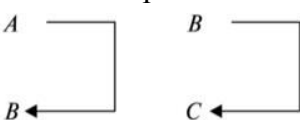


Рис. 2.14. Декомпозиция отношения

как атрибуты «Должность» и «Оклад» находятся в транзитивной зависимости. В этом случае несколько человек могут работать в одной и той же должности и, следовательно, при изменении должностного оклада необходимо менять данные в каждой записи. Также если имеются уникальные должности, то при увольнении сотрудника потеряется информация об окладе по этой должности. Кроме того, сотрудник может переходить с одной должности на другую.

Для приведения отношения Сотрудник к 3НФ необходимо декомпозировать его на следующие отношения:

Сотрудник = (Таб номер: Фамилия)

Должность = (Код должности: Должность; Оклад)

Должностные\_перемещения = (Таб номер: Код должности:

Датазачисления; Дата\_увольнения)

Отношение Сотрудник Отношение Должность



Связь между отношениями Сотрудник и Должностные перемещения осуществляется через ключевое поле «Таб\_номер», между отношениями Должность и Должностные\_перемещения — через ключевое поле «Код\_должности».

## Тема №4. Классификация и характеристика СУБД. Обзор современных СУБД.

*Системы управления базами данных можно классифицировать:*

по используемому языку общения:

- замкнутые, имеющие собственные самостоятельные языки общения пользователей с БД. Они обеспечивают непосредственное общение с системой в режиме диалога, позволяют работать без программистов;
- открытые, в которых для общения с БД используется язык программирования, «расширенный» операторами языка манипулирования данными (ЯМД). В этом случае необходимо присутствие квалифицированного программиста.

по числу поддерживаемых СУБД уровней моделей данных: одно-, двух-, трехуровневые системы. Теоретически обоснован выбор трехуровневой архитектуры данных, однако на практике СУБД для персональных ЭВМ часто объединяют концептуальный и внутренний уровни представления.

по выполняемым функциям:

- операционные, предполагающие иные виды обработки по получению информации, не хранящейся в явном виде в БД;
- информационные, позволяющие организовать хранение данных, поиск и выдачу нужных данных из БД и поддерживать их целесообразность и актуальность;

по сфере применения:

- универсальные, настраиваемые на любую предметную область путём создания соответствующей БД и прикладных программ;
- проблемно-ориентированные на определенные процедуры обработки данных, присущих конкретной области применения;

по допустимым режимам работы:

- пакетные;
- с использованием телеобработки.

Программные средства баз данных

Оболочки информационных систем (системы программирования ИС) представляют гибкие программные комплексы, настраиваемые на задачи пользователя. Наиболее распространёнными классами данных программных средств являются СУБД и оболочки автоматизированных информационно-поисковых систем (АИПС).

### *Информационно-поисковые системы*

В узком смысле под АИПС принято понимать открытый (обычно) или замкнутый (реже) программный продукт, предназначенный для реализации практически большинства функций (процессов): ввод, обработка, хранение, поиск, представление данных (организованных в записи или документы, находящиеся в БД). В этом смысле часто отождествляют АИПС с АИС, и это трудно оспаривать.

Среди АИПС в узком смысле принято выделять:

- фактографические системы с фиксированной структурой данных или записей, для разработки которых, как правило, используются СУБД, поддерживающие табличные (реляционные) БД;
- документальные системы с неопределённой или переменной структурой данных или документов, для разработки которых часто (но не обязательно) применяют оболочки АИПС.

В более широком смысле под АИПС подразумеваются также программные оболочки, ориентированные на разработку продуктов типа АИПС. Это связано с тем фактом, что первые системы типа СУБД и оболочек АИПС были предложены в 1960–1970-е гг. фирмой ИВМ (и сотрудничавшими с ней организациями) и включали в себя:

- IMS/360 (Information Management System) – по-видимому, первая реальная СУБД, поддерживавшая т.н. иерархическую модель данных (понятие появилось позже, в связи с необходимостью систематизации СУБД), нашедшая достаточно широкое применение, в частности, для информационного обеспечения проекта Apollo, завершившегося, как известно, высадкой граждан США на Луну в 1969 г.;
- DPS/360 (Document Processing System) – первый промышленный ППП, предназначенный для реализации документальных АИПС. В дальнейшем путём развития принципов DPS, фирма в 1972 г. выпустила пакет STAIRS (STorage And Information Retrieval System), предназначенный для диалогового обслуживания множества (удалённых) пользователей;
- IRMS (Information Retrieval and Management System), TEXT-RAC и другие аналогичные пакеты.

Как следует из наименований продуктов, разработчики понимали под АИПС именно ППП-оболочки.

Системы управления базами данных и программирования АИС

Среди различных программных средств данного класса различают три типа:

- СУБД в «чистом виде» (IMS, SETOP и пр.);
- СУБД с элементами систем программирования АИС (ADABAS/NATURAL, ORACLE);
- системы программирования АИС с элементами СУБД (FoxBase / FoxPro, Clipper).

Первый тип фактически относится к начальному этапу развития систем второго (реже – третьего) типов. В этом случае СУБД состоит только из системы интерпретации вызовов (обращений) из пользовательской программы (call-interface) на выборку (корректировку, занесение) информации из/в БД, причём программа написана на одном из таких универсальных языков программирования (ЯП), как Кобол, Фортран, Паскаль и пр., получивших название включающие языки СУБД. Данная система в последующих СУБД (второй тип) получила наименование ядра. Соглашения о форматах и структурах такого взаимодействия обычно пытаются оформить в виде некоторого формального языка (языка ядра).

Второй тип представляет расширение первого в направлении создания универсальной системы разработчика АИС, включающей также специализированные языковые средства. В этом случае СУБД представляет совокупность специализированных программных средств, вспомогательных файлов и управляющих таблиц (иногда находящихся в составе БД, реже это файлы ОС), обеспечивающих доступ пользователей к БД при соблюдении следующих существенных критериев: целостность и непротиворечивость данных, описывающих различные аспекты объектов реального мира, защиту информации от несанкционированного доступа на чтение/обновление содержимого БД, установление и поддержание связей между зависимыми данными, удобство использования данных.

Третий тип представляют (разработанные обычно для ПК) системы, содержащие как элементы непроцедурного типа (язык запросов), так и процедурного (язык программирования) во входном языке, предназначенном для управления данными и обработки информации. Элементы СУБД здесь также заключаются в наличии простейшего словаря данных, возможности создания модели предметной области в форме совокупности таблиц, связанных между собой простейшим образом, а также в наличии средств генерации отчетов и управления доступом пользователей.

## Тема №5. Постреляционные СУБД

Постреляционная модель данных представляет собой расширенную реляционную модель, в которой отменено требование атомарности атрибутов. Поэтому постреляционную модель называют "не первой нормальной формой" (NF2) или "многомерной базой данных". Она использует трехмерные структуры, позволяя хранить в полях таблицы другие таблицы. Тем самым расширяются возможности по описанию сложных объектов реального мира. В

качестве языка запросов используется несколько расширенный SQL, позволяющий извлекать сложные объекты из одной таблицы без операций соединения.

Существует несколько коммерческих постреляционных СУБД, более подробные сведения о них можно получить на веб-серверах фирм-производителей. Пожалуй, самыми известными из них являются системы Adabas, Picki Universe.

### *Объектно-ориентированные СУБД.*

Термин "объект" в программной индустрии впервые был введен в языке Simula (1967 г.) и означал какой-либо аспект моделируемой реальности. Сейчас под объектом понимается "нечто, имеющее четко определенные границы" (определение известного американского специалиста Г.Буча). Объекты, обладающие одинаковыми свойствами, составляют классы (например, курица, пингвин и чайка - объекты класса "птицы"). Обычно класс описывается как новый тип данных, а объекты (экземпляры класса) - определенные на его основе переменных.

### *Объектно-ориентированная парадигма.*

Сразу же необходимо заметить, что общепринятого определения "объектно-ориентированной модели данных" не существует. Сейчас можно говорить лишь о некоем "объектном" подходе к логическому представлению данных и о различных объектно-ориентированных способах его реализации.

Мы знаем, что любая модель данных должна включать три аспекта: структурный, целостный и манипуляционный. Посмотрим, как они реализуются на основе объектно-ориентированной парадигмы программирования:

*Структура объектной модели описывается с помощью трех ключевых понятий:*

- инкапсуляция- каждый объект обладает некоторым внутренним состоянием (хранит внутри себя запись данных), а также набором методов - процедур, с помощью которых (итолькотаким образом) можно получить доступ к данным, определяющим внутреннее состояние объекта, или изменить их. Таким образом, объекты можно рассматривать как самостоятельные сущности, отделенные от внешнего мира.

Пример:

- Class Point { // вводим новый тип данных - объект "точка"
- X,Y : int; // данные объекта - координаты точки
- .....
- Point(X : int, Y : int); // конструктор объекта - процедура, вызываемая при
- // определении переменной на базе объекта и
- // присваивающая значения его данным
- .....
- Draw(); // метод "нарисовать точку"
- Erase(); // метод "стереть точку"
- Move(newX,newY); // метод "переместить точку" (изменяет данные объекта)
- int GetX(); // метод "получить значение поля X"
- int GetY(); // метод "получить значение поля Y"
- .....
- // все методы должны быть описаны, например
- // реализация метода Move:
- 
- Move(newX : int, newY : int) {
- X=newX; // запись новых данных в объект
- Y=newY; //



### *Целостность данных:*

Для поддержания целостности объектно-ориентированный подход предлагает использовать следующие средства:

- автоматическое поддержание отношений наследования
- возможность объявить некоторые поля данных и методы объекта как "скрытые", не видимые для других объектов; такие поля и методы используются только методами самого объекта
- создание процедур контроля целостности внутри объекта

### *Средства манипулирования данными:*

К сожалению, в объектно-ориентированном программировании отсутствуют общие средства манипулирования данными, такие как реляционная алгебра или реляционное счисление.

Работа с данными ведется с помощью одного из объектно-ориентированных языков программирования общего назначения, обычно это SmallTalk, C++ или Java.

### *Подведем теперь некоторые итоги:*

В объектно-ориентированных базах данных, в отличие от реляционных, хранятся не записи, а объекты. ОО-подход представляет более совершенные средства для отображения реального мира, чем реляционная модель:

- естественное представление данных. В реляционной модели все отношения принадлежат одному уровню, именно это осложняет преобразование иерархических связей модели "сущность-связь" в реляционную модель (см. параграф 5.5.3). ОО-модель можно рассматривать послойно, на разных уровнях абстракции.
- имеется возможность определения новых типов данных и операций с ними.

В то же время, ОО-модели присущ и ряд недостатков:

- отсутствуют мощные непроцедурные средства извлечения объектов из базы. Все запросы приходится писать на процедурных языках, проблема их оптимизации возлагается на программиста.
- вместо чисто декларативных ограничений целостности (типа явного объявления первичных и внешних ключей реляционных таблиц с помощью ключевых слов PRIMARY KEY и REFERENCES) или полудекларативных триггеров для обеспечения внутренней целостности приходится писать процедурный код.

Очевидно, что оба эти недостатка связаны с отсутствием развитых средств манипулирования данными. Эта задача решается двумя способами - расширение ОО-языков в сторону управления данными (стандарт ODMG), либо добавление объектных свойств в реляционные СУБД (SQL-3, а также так называемые объектно-реляционных СУБД).



## Тема №6. Работа с объектами(таблицы, запросы, формы, отчеты и макросы)

Компьютерная база данных — это хранилище объектов. В одной базе данных может быть больше одной таблицы. Например, система отслеживания складских запасов, в которой используются три таблицы, — это не три базы данных, а одна. В базе данных Access (если ее специально не настраивали для работы с данными или кодом, принадлежащими другому источнику) все таблицы хранятся в одном файле вместе с другими объектами, такими как формы, отчеты, макросы и модули. Для файлов баз данных, созданных в формате Access 2007 (который также используется в Access 2016, Access 2013 и Access 2010), используется расширение ACCDB, а для баз данных, созданных в более ранних версиях Access, — MDB. С помощью Access 2016, Access 2013, Access 2010 и Access 2007 можно создавать файлы в форматах более ранних версий приложения (например, Access 2000 и Access 2002–2003).

Использование Access позволяет:

- добавлять новую информацию в базу данных, например новый артикул складских запасов;
- изменять информацию, уже находящуюся в базе, например перемещать артикул;
- удалять информацию, например если артикул был продан или утилизирован;
- упорядочивать и просматривать данные различными способами;
- обмениваться данными с другими людьми с помощью отчетов, сообщений электронной почты, внутренней сети или Интернета.

Элементы базы данных Access:

### Таблицы

Таблица базы данных похожа на электронную таблицу — и там, и там информация расположена в строках и столбцах. Поэтому импортировать электронную таблицу в таблицу базы данных обычно довольно легко. Основное различие заключается в том, как данные структурированы.

Чтобы база данных была как можно более гибкой и чтобы в ней не появлялось излишней информации, данные должны быть структурированы в виде таблиц. Например, если речь идет о таблице с информацией о сотрудниках компании, больше одного раза вводить данные об одном и том же сотруднике не нужно. Данные о товарах должны храниться в отдельной таблице, как и данные о филиалах компании. Этот процесс называется *нормализацией*.

Строки в таблице называются записями. В записи содержатся блоки информации. Каждая запись состоит по крайней мере из одного поля. Поля соответствуют столбцам в таблице. Например, в таблице под названием "Сотрудники" в каждой записи находится информация об одном сотруднике, а в каждом поле — отдельная категория информации, например имя, фамилия, адрес и т. д. Поля выделяются под определенные типы данных, например текстовые, цифровые или иные данные.

Записи и поля можно описать по-другому. Представьте старый библиотечный карточный каталог. Каждой карточке в шкафу соответствует *запись* в базе данных. Блоки информации на карточке (автор, название книги и т. д.) соответствуют *полям* в базе данных.

### Формы

С помощью форм создается пользовательский интерфейс для ввода и редактирования данных. Формы часто содержат кнопки команд и другие элементы управления, предназначенные для выполнения различных функций. Можно создать базу данных, не

используя формы, если просто отредактировать уже имеющуюся информацию в таблицах Access. Тем не менее, большинство пользователей предпочитает использовать формы для просмотра, ввода и редактирования информации в таблицах.

С помощью кнопок команд задаются данные, которые должны появляться в форме, открываются прочие формы и отчеты и выполняется ряд других задач. Например, есть "Форма клиента", в которой вы работаете с данными о клиентах. И в ней может быть кнопка, нажатием которой открывается форма заказа, с помощью которой вы вносите информацию о заказе, сделанном определенным клиентом.

Формы также дают возможность контролировать взаимодействие пользователей с информацией базы данных. Например, можно создать форму, в которой отображаются только определенные поля и с помощью которой можно выполнять только ограниченное число операций. Таким образом обеспечивается защита и корректный ввод данных.

### *Отчеты*

Отчеты используются для форматирования, сведения и показа данных. Обычно отчет позволяет найти ответ на определенный вопрос, например "Какую прибыль в этом году принесли нам наши клиенты?" или "В каких городах живут наши клиенты?" Отчеты можно форматировать таким образом, чтобы информация отображалась в наиболее читабельном виде.

Отчет можно сформировать в любое время, и в нем всегда будет отображена текущая информация базы данных. Отчеты обычно формируются таким образом, чтобы их можно было распечатать, но их также можно просматривать на экране, экспортировать в другие программы или вкладывать в сообщения электронной почты.

### *Запросы*

Запросы могут выполнять множество функций в базе данных. Одна из их основных функций — находить информацию в таблицах. Нужная информация обычно содержится в нескольких таблицах, но, если использовать запросы, ее можно просматривать в одной. Кроме того, запросы дают возможность фильтровать данные (для этого задаются критерии поиска), чтобы отображались только нужные записи.

Используются и так называемые "обновляемые" запросы, которые дают возможность редактировать данные, найденные в основных таблицах. При работе с обновляемым запросом помните, что правки вносятся в основные таблицы, а не только в таблицу запроса.

Есть два основных вида запросов: запросы на выборку и на изменение. Запрос на выборку только находит данные и предоставляет к ним доступ. Результаты такого запроса можно просмотреть на экране, распечатать или скопировать в буфер обмена, а также использовать в качестве источника записей для формы или отчета.

С помощью запроса на изменение, как видно из названия, можно выполнять определенные операции с найденными данными: создавать таблицы, добавлять информацию в уже существующие таблицы, а также обновлять или удалять данные.

### *Макросы*

Макросы в Access — это нечто вроде упрощенного языка программирования, с помощью которого можно сделать базу данных более функциональной. Например, если к кнопке команды в форме добавить макрос, то он будет запускаться всякий раз при нажатии этой кнопки. Макросы состоят из команд, с помощью которых выполняются определенные задачи: открываются отчеты, выполняются запросы, закрывается база данных и т. д. Используя макросы, можно автоматизировать большинство операций, которые в базе данных вы делаете вручную, и, таким образом, значительно сэкономить время.

## *Модули*

Подобно макросам, модули — это объекты, с помощью которых базу данных можно сделать более функциональной. Но если макросы в Access составляются путем выбора из списка макрокоманд, модули создаются на языке Visual Basic для приложений (VBA). Модули представляют собой наборы описаний, инструкций и процедур. Существуют модули класса и стандартные модули. Модули класса связаны с конкретными формами или отчетами и обычно включают в себя процедуры, которые работают только с этими формами или отчетами. В стандартных модулях содержатся общие процедуры, не связанные ни с каким объектом. Стандартные модули, в отличие от модулей класса, перечисляются в списке **Модули** в области навигации.

## Тема №7. Краткая характеристика языка SQL. Операторы языка SQL для работы с реляционной базой данных.

### Типы данных языка SQL

Типы данных, используемые в языке SQL для хранения информации в столбцах таблиц БД, весьма разнообразны. К сожалению, производители конкретных реляционных СУБД считают своим долгом ?улучшить? множество типов данных, регламентируемых стандартом, реализуя свои собственные версии и расширения.

В качестве базовых предлагается считать следующие типы данных:

- INT[(len)] - целое число длиной 4 байта, представляемое при выводе максимально len цифрами;
- SMALLINT[(len)] - целое число длиной 2 байта, представляемое при выводе максимально len цифрами;
- FLOAT[(len,dec)] - действительное число, представляемое при выводе максимально len символами с dec цифрами после десятичной точки;
- CHAR(size) - строка символов фиксированной длины размером size символов;
- VARCHAR(size) - строка символов переменной длины максимальным размером до size символов;
- BLOB (Binary Large Object) - массив произвольных (двоичных) байтов (максимальный размер зависит от реализации, обычно это 65535 байт); этот тип данных может использоваться, например, для хранения изображений;
- DATE - астрономическая дата;
- TIME - астрономическое время.

Символьные константы (типа CHAR и VARCHAR) записываются как последовательности символов, заключенные в одиночные апострофы, например ?brass? (латунь).

Десятичные константы (типа FLOAT) могут записываться в ?научной? нотации как последовательности следующих компонент:

- знак числа;
- десятичное число с точкой;
- символ ?e?;
- знак (?+? или ?-?) показателя степени;
- целое число, играющее роль показателя степени числа 10.

Например, десятичное число -0,123 может быть записано как -12.3e-2.

Отличие типов данных CHAR и VARCHAR заключается в том, что для хранения в таблице строк символов типа CHAR используется точно size байт (хотя содержание хранимых строк может быть значительно короче), в то время как для строк типа VARCHAR незанятые символами строк (?пустые?) байты в таблице не хранятся.

Подчеркнем, что величины len и dec (в отличие от size) не влияют на размер хранения данных в таблице, а только форматируют вывод данных из таблицы.

Примечание. Тип данных BLOB поддерживается непосредственно не всеми СУБД, однако каждая из них предлагает его аналог (например, BINARY или IMAGE).

Рекомендация. Разрабатывая мобильное приложение (рассчитанное на работу в среде различных СУБД), старайтесь без необходимости избегать использования необязательных возможностей в описании типов данных.

#### *Манипулирование таблицами*

Для создания, изменения и удаления таблиц в SQL БД используются операторы CREATE TABLE, ALTER TABLE и DROP TABLE.

#### *Создание таблицы*

Создание таблицы в БД реализуется оператором CREATE TABLE, имеющим следующий синтаксис

CREATE TABLE имя\_табл (с\_спецификация, ...);

где с\_спецификация имеет разнообразный синтаксис. Здесь же рассматриваются наиболее часто используемые ее формы.

*Описание столбца таблицы*

имя\_столбца тип\_данных [NULL]

где имя\_столбца - имя столбца таблицы, а тип\_данных - спецификация одного из типов данных, рассмотренных в разделе ?Типы данных языка SQL?. Необязательное ключевое слово NULL означает, что ячейкам данного столбца разрешено быть пустыми (т.е. не содержать какого-либо значения).

*Описание столбца таблицы*

имя\_столбца тип\_данных NOT NULL [DEFAULT по\_умолч] [PRIMARY KEY]

где конструкция NOT NULL запрещает иметь в таблице пустые ячейки в данном столбце.

Конструкция PRIMARY KEY указывает, что содержимое столбца будет играть роль первичного ключа для создаваемой таблицы. Конструкция

DEFAULT по\_умолч переопределяет имеющееся для столбцов каждого типа данных

значение ?по умолчанию? (например, 0 для числовых типов), используемое при добавлении в таблицу оператором INSERT INTO строк, не содержащих значений в этом столбце.

*Описание первичного ключа*

PRIMARY KEY имя\_ключа (имя\_столбца, ...)

Эта спецификация позволяет задать первичный ключ для таблицы в виде композиции содержимого нескольких столбцов.

*Описание вторичного ключа*

KEY имя\_ключа (имя\_столбца, ...)

Примеры

Ниже приводятся примеры использования оператора CREATE TABLE для создания четырех таблиц учебной БД.

CREATE TABLE nodes (

id SMALLINT NOT NULL PRIMARY KEY, # номер узла  
x FLOAT NOT NULL, # x-координата  
y FLOAT NOT NULL); # y-координата

CREATE TABLE elements (

id SMALLINT NOT NULL PRIMARY KEY, # номер КЭ  
n1 SMALLINT NOT NULL, # номер первой вершины  
n2 SMALLINT NOT NULL, # номер второй вершины  
n3 SMALLINT NOT NULL, # номер третьей вершины  
props CHAR(12) NOT NULL DEFAULT 'steel');

Столбец props таблицы elements предназначен для хранения названия материала КЭ и не может содержать ?пустых? полей, его значением ?по умолчанию? является строка символов ?steel? (сталь).

CREATE TABLE materials (

name CHAR(12) NOT NULL PRIMARY KEY, # название материала  
density FLOAT NOT NULL, # плотность  
elastics FLOAT NOT NULL, # модуль Юнга  
poisson FLOAT NOT NULL, # к-т Пуассона  
strength FLOAT NOT NULL); # прочность

CREATE TABLE loadings (

type CHAR(1) NOT NULL, # тип граничного условия  
direction CHAR(1), # направление действия  
node SMALLINT NOT NULL, # номер узла приложения  
value FLOAT, # числовое значение  
KEY key\_node (node); # вторичный ключ

В таблице граничных условий loadings поля столбцов direction и value могут быть пустыми (иметь значение NULL), поскольку не все виды нагрузок имеют направление действия и/или

величину. Номер узла node приложения граничного условия определяется как ключ поиска в таблице, т.к. типичный запрос на поиск в таблице loadings - это запрос на определение граничных условий для конкретного узла. Однако этот ключ не может быть первичным, поскольку к одному узлу допустимо приложение нескольких граничных условий (например, момент внешних сил в шарнире). Следует отметить, что в этой таблице первичный ключ может быть сконструирован только составным из столбцов type, direction и node.

#### *Модификация таблицы*

Модификация существующей таблицы в БД реализуется оператором ALTER TABLE, имеющим следующий синтаксис

```
ALTER TABLE имя_табл м_спецификация [,м_спецификация ...]
```

где м\_спецификация имеет различные формы. Ниже рассматриваются наиболее часто используемые.

#### *Добавление нового столбца*

```
ADD COLUMN с_спецификация
```

где с\_спецификация - описание добавляемого столбца в том виде, как оно используется для создания таблицы оператором CREATE TABLE.

#### *Удаление первичного ключа для таблицы*

```
DROP PRIMARY KEY
```

Изменение/удаление значения ?по умолчанию?

```
ALTER COLUMN имя_столбца SET по_умолч
```

или

```
ALTER COLUMN имя_столбца DROP DEFAULT
```

#### *Пример*

Предположим в нашей задаче моделирования состояния плоского механического объекта возникла необходимость учесть дополнительно тепловые эффекты. Для этого, в частности, необходимо иметь сведения о теплофизических параметрах материала объекта (теплоемкости и теплопроводности). Включение дополнительных сведений в таблицу materials требует ее расширения двумя новыми столбцами, что можно реализовать таким оператором языка SQL:

```
ALTER TABLE materials
```

```
  ADD COLUMN capacity FLOAT NOT NULL,      # теплоемкость
```

```
  ADD COLUMN conductivity FLOAT NOT NULL;  # теплопроводность
```

#### *Удаление таблицы*

Удаление одной или сразу нескольких таблиц из БД реализуется оператором DROP TABLE, имеющим следующий простой синтаксис

```
DROP TABLE имя_табл, ...
```

Подчеркнем, что оператор DROP TABLE удаляет не только все содержимое таблицы, но и само описание таблицы из БД. Если требуется удалить только содержимое таблицы, то необходимо использовать оператор DELETE FROM.

#### *Добавление строк в таблицу*

Для добавления строк в таблицу SQL базы данных используется оператор INSERT INTO.

Основные его синтаксические формы описываются ниже.

Добавление строки перечислением значений всех ее ячеек

```
INSERT INTO имя_табл VALUES (знач, ...);
```

где знач - константное значение ячейки строки. Значения ячеек в списке должны соответствовать порядку перечисления спецификаций столбцов таблицы в операторе CREATE TABLE. Допустимо в качестве знач указывать ключевое слово NULL, что означает отсутствие значения для соответствующей ячейки строки.

Перед добавлением новой строки в таблицу СУБД проверяет допустимость перечисленных значений, используя описание столбцов таблицы из оператора CREATE TABLE.

#### *Добавление строки с использованием списка имен столбцов*

```
INSERT INTO имя_табл (имя_столбца, ...) VALUES (знач, ...);
```

Здесь списки имен столбцов и значений ячеек добавляемой строки должны быть согласованы, хотя нет никаких требований к их порядку. Допустимо опускать в списках информацию о некоторых ячейках строки, при этом ячейки, соответствующие столбцам со спецификацией NULL в операторе CREATE TABLE, будут пустыми;

ячейки, соответствующие столбцам со спецификацией NOT NULL в операторе CREATE TABLE, заполняются значениями по умолчанию?.

*Добавление строк по результатам запроса к БД*

```
INSERT INTO имя_табл [(имя_столбца, ...)] SELECT ...
```

Такой оператор дает возможность добавить в таблицу 0, 1 или сразу несколько новых строк, полученных в результате запроса к базе данных, реализуемого оператором SELECT.

Пример

Добавление информации о новом узле КЭ-сетки в таблицу nodes:

```
INSERT INTO nodes VALUES (25, 6.3, 1.8);
```

Отметим, что добавление новой строки будет удачным только в том случае, если узла с таким же идентификатором в таблице nodes еще нет - дело в том, что столбец id этой таблицы объявлен первичным ключом и, следовательно, значения всех его ячеек должны быть уникальны.

Пример

Добавление информации о новом КЭ в таблицу elements:

```
INSERT INTO elements  
  (n1, n2, n3, id) VALUES  
  (14, 25, 18, 46);
```

В результате в таблице elements появится новая строка, содержащая в поле props значение 'steel', как умолчательное значение, определенное при создании таблицы.

Пример

Включение в таблицу materials сведений о новом материале:

```
INSERT INTO materials VALUES (  
  'wood', 0.6, 2.0, 0.12, 50);
```

Пример

Добавление в таблицу граничных условий loadings информации об ориентированном горизонтально 'катке' в узле 2:

```
INSERT INTO loadings VALUES (  
  'r', 'x', 2, NULL);
```

## Тема №8. Организация запросов к базе данных на языке SQL

### *Выборка данных из таблиц*

Для извлечения данных, содержащихся в таблицах SQL БД, используется оператор SELECT, имеющий в общем случае сложный и многовариантный синтаксис. В данном учебном пособии рассматриваются только несложные и наиболее часто используемые примеры конструкций оператора SELECT.

Упрощенно оператор SELECT выглядит следующим образом:

```
SELECT [ALL | DISTINCT] в_выражение, ...  
FROM имя_табл [син_табл], ...  
[WHERE сложн_условие]  
[GROUP BY полн_имя_столбца|ном_столбца, ...]  
[ORDER BY полн_имя_столбца|ном_столбца [ASC|DESC], ...]  
[HAVING сложн_условие];
```

Результатом работы оператора является выводимая на стандартный вывод (экран дисплея) вновь построенная таблица, для которой

- количество и смысл (семантика) столбцов определяется списком элементов в\_выражение;
- содержимое строк определяется содержимым исходных таблиц из списка FROM и критерием выборки, задаваемым сложн\_условие.

При описании синтаксиса оператора SELECT использованы следующие обозначения:  
син\_табл - необязательный синоним имени таблицы, используемый для сокращения длины записи выражений и условий в операторе SELECT.

полн\_имя\_столбца - полное имя столбца в виде

[имя\_табл|син\_табл.]имя\_столбца

Конкретизирующий таблицу префикс в имени столбца необходим только для различения столбцов, имеющих одинаковое имя в разных таблицах из списка FROM.

ном\_столбца - номер столбца результирующей таблицы.

Описание столбцов результирующей таблицы

1. Специальным (и часто используемым) видом в\_выражение является символ ??, имеющий смысл ?все столбцы таблиц из списка FROM?.

Пример

Вывод всего содержимого таблицы materials.



```
SELECT * FROM materials;
```

name	density	elastics	poisson	strength
steel	7.80	200.00	0.25	1000.00
aluminium	2.70	65.00	0.34	600.00
concrete	5.60	25.00	0.12	300.00
duraluminium	2.80	70.00	0.31	700.00
titanium	4.50	116.00	0.32	950.00
brass	8.50	93.00	0.37	300.00

**2.** Простым (и также часто используемым) случаем в `_выражение` является полное имя столбца одной из таблиц списка FROM.

Пример

Пусть необходимо определить идентификаторы всех узлов КЭ-сетки, к которым приложено какое-либо граничное условие, при этом необходимо знать тип приложенного условия. Эта задача может быть решена с помощью следующего оператора:

```
SELECT node, type FROM loadings;
```

node	type
1	r
2	r
3	r
14	h
27	f
27	f

Полученная результирующая таблица содержит дублирующие строки для узла 27. Избежать этого можно, добавив в оператор ключевое слово `DISTINCT`, запрещающее включение в итоговую таблицу одинаковых строк.

```
SELECT DISTINCT node, type FROM loadings;
```

node	type
1	r
2	r
3	r
14	h
27	f

**3.** В общем случае в `_выражение` может представлять собой сложное скобочное выражение над содержимым столбцов таблицы, использующее арифметические, строковые, логические операции и функции.

**ЭКСПЕРТНАЯ ОЦЕНКА**  
**на учебно-методический комплекс для 2-3 курса по**  
**дисциплине «Базы данных»**

Направление подготовки: **230109 «Программное обеспечение вычислительных программ и автоматизированных систем»**

Учебно-методический комплекс разработан для СПО 2-3 курса в 4-5 семестре для обеспечения выполнения требований Государственного образовательного стандарта среднего профессионального образования по подготовке техник - программистов по направлению **230109 «Программное обеспечение вычислительных программ и автоматизированных систем»**

УМК дисциплины «Базы данных», представленный на рецензию, выполнен старшим преподавателем Акишовой М.К

УМК дисциплины включает: рабочую программу дисциплины, syllabus, описание дисциплины, конспект лекций, тестовые задания, фонд оценочных средств, график и методические указания по СРС и СРСП.

Фонды оценочных средств включают тестовые задания (текущего и итогового контроля); задания по проверке освоения практических навыков с критериями оценки; материалы к экзамену (тестовые задания, контрольные вопросы по проверке освоения практических навыков, список вопросов к экзамену, билеты к экзамену).

Перечень рекомендуемой основной и дополнительной литературы включает источники 2014-2019 гг., ресурсы информационно-телекоммуникационной сети «Интернет». Имеются ссылки на базы данных, сайты, справочные системы, сетевые ресурсы т.д.

Содержание учебно-методического комплекса дисциплины обеспечивает использование современных образовательных технологий для реализации основной цели - формирования знаний, умений и навыков общекультурных, общепрофессиональных и профессиональных компетенций.

Материалы УМК дисциплины «Базы данных» разработаны в строгом соответствии с разделами рабочей программы.

Учебно-методический комплекс составлен в полном соответствии с требованиями по реализации компетентностного подхода в обучении техник-программистов.

\_\_\_\_\_ Дата \_\_\_\_\_

**КАЛЕНДАРНО-ТЕМАТИЧЕСКИЙ ПЛАН ДИСЦИПЛИНЫ**  
**«БАЗЫ ДАННЫХ»**

№	Название темы	Аудиторные занятия			
		Лекция	практика	лабораторные	СРС
1	<b>Введение.</b> Основные понятия теории баз данных	2	2	2	6
2	<b>Информационные модели.</b> Взаимосвязи в моделях и реляционный подход к построению моделей.	2	4	4	6
3	<b>Нормализация отношений.</b> Этапы проектирования баз данных	2	2	4	6
4	<b>Классификация и характеристика СУБД.</b> Обзор современных СУБД.	2	2	6	6
5	Постреляционные СУБД	4	2	4	6
6	<b>Работа с объектами</b> (таблицы, запросы, формы, отчеты и макросы)	2	2	4	6
7	<b>Краткая характеристика языка SQL.</b> Операторы языка SQL для работы с реляционной базой данных.	2	2	6	6
8	Организация запросов к базе данных на языке SQL	2	2	6	6
	<b>Итого:</b>	<b>18</b>	<b>18</b>	<b>36</b>	<b>48</b>