

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Усынин Максим Валерьевич  
Должность: Ректор  
Дата подписания: 02.05.2024 14:37:48  
Уникальный программный ключ:  
f498e59e83f65dd7c3ce7bb8a25cbbabb33ebc58

**Частное образовательное учреждение высшего образования  
«Международный Институт Дизайна и Сервиса»  
(ЧОУВО МИДиС)**

Кафедра математики и информатики

**ФОНД  
ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО  
КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ  
АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ  
БАЗЫ ДАННЫХ**

Направление подготовки: 09.03.03 Прикладная информатика  
Направленность (профиль): Разработка компьютерных игр и приложений с  
виртуальной и дополненной реальностью  
Квалификация выпускника: Бакалавр  
Год набора – 2024

Автор-составитель: Чеботарев С.С.

Челябинск 2024

## СОДЕРЖАНИЕ

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы .....	3
2. Показатели и критерии оценивания компетенций на различных этапах их формирования, описание шкал оценивания .....	4
3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы .....	8
4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы .....	60

## 1. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ С УКАЗАНИЕМ ЭТАПОВ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Процесс изучения дисциплины «Базы данных» направлен на формирование следующих компетенций:

Код и наименование компетенций выпускника	Код и наименование индикатора достижения компетенций
ОПК-5 Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.1. Знает основы системного администрирования, администрирования СУБД, современные стандарты информационного взаимодействия систем ОПК-5.2. Умеет выполнять параметрическую настройку информационных и автоматизированных систем ОПК-5.3. Владеет навыками инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем
ПК-1 Способен кодировать на языках программирования (объектно-ориентированных, современных структурных языках, языках современных бизнес-приложений)	ПК-1.1. Разрабатывает код информационных систем и баз данных информационных систем. ПК-1.2. Осуществляет верификацию кода, баз данных и структуры баз данных информационных систем ПК-1.3. Устраняет обнаруженные несоответствия с применением методик тестирования разрабатываемых информационных систем
ПК-3 Способен проектировать и разрабатывать информационные системы в соответствии с требованиями заказчика	ПК-3.1. Выполнять действия разработке прототипов информационных систем, мобильных и Web приложений ПК-3.2. Выполнять действия по проектированию, верификации информационных систем, мобильных и Web приложений в соответствии с требованиями заказчика. ПК 3.3. Владеть инструментами и методами разработки и тестирования баз данных информационных систем

№ п/п	Код компетенции	Наименование компетенции	Этапы формирования компетенций
1.	ОПК-5	Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем	<p><i>1 Этап – Знать:</i> ОПК-5.1. Основы системного администрирования, основы администрирования СУБД, современные стандарты информационного взаимодействия систем</p> <p><i>2 Этап – Уметь:</i> ОПК-5.2. Выполнять параметрическую настройку информационных и автоматизированных систем</p> <p><i>3 Этап – Владеть:</i> ОПК-5.3. Навыками инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем</p>
2.	ПК-1	Способен кодировать на языках программирования	<p><i>1 Этап – Знать:</i> ПК-1.1. Способы разработки кода информационных систем;</p>

		(объектно-ориентированных, современных структурных языках, языках современных бизнес-приложений)	<p><i>2 Этап – Уметь:</i> ПК-1.2. Осуществлять верификацию кода, баз данных и структуры баз данных информационных систем;</p> <p><i>3 Этап – Владеть:</i> ПК-1.3. Навыками устранения обнаруженных несоответствий с применением методик тестирования разрабатываемых информационных систем</p>
3.	ПК-3	Способен проектировать и разрабатывать информационные системы в соответствии с требованиями заказчика	<p><i>1 Этап – Знать:</i> ПК-3.1. Способы разработки прототипов информационных систем, мобильных и Web приложений</p>
	<p><i>2 Этап – Уметь:</i> ПК-3.2. Выполнять действия по проектированию, верификации информационных систем, мобильных и Web приложений в соответствии с требованиями заказчика</p>		
	<p><i>3 Этап – Владеть:</i> ПК-3.3. Инструментами и методами разработки и тестирования баз данных информационных систем;</p>		

## 2. ПОКАЗАТЕЛИ И КРИТЕРИИ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ НА РАЗЛИЧНЫХ ЭТАПАХ ИХ ФОРМИРОВАНИЯ, ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ

№ п/п	Код компетенции	Наименование компетенции	Критерии оценивания компетенций на различных этапах формирования	Шкала оценивания
1.	ОПК-5	Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем	<p><i>1 Этап – Знать:</i> ОПК-5.1. Основы системного администрирования, основы администрирования СУБД, современные стандарты информационного взаимодействия систем</p>	<p>«ОТЛИЧНО»:</p> <ol style="list-style-type: none"> <li>1. Глубокое и прочное усвоение программного материала.</li> <li>2. Знание пакетов прикладных программ</li> <li>3. Знание основных принципов построения пакетов прикладных программ.</li> <li>4. Знание основных задач прикладных программ.</li> <li>5. Свободное владение пакетами прикладных программ.</li> </ol>
	<p><i>2 Этап – Уметь:</i> ОПК-5.2. Выполнять параметрическую настройку информационных и автоматизированных систем</p>			
	<p><i>3 Этап – Владеть:</i> ОПК-5.3. Навыками инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем</p>			
2.	ПК-1	Способен	<p><i>1 Этап – Знать:</i></p>	

		кодировать на языках программирования (объектно-ориентированных, современных структурных языках, языках со-временных бизнес-приложений)	<p>ПК-1.1. Способы разработки кода информационных систем;</p> <p><i>2 Этап – Уметь:</i> ПК-1.2. Осуществлять верификацию кода, баз данных и структуры баз данных информационных систем;</p> <p><i>3 Этап – Владеть:</i> ПК-1.3. Навыками устранения обнаруженных несоответствий с применением методик тестирования разрабатываемых информационных систем</p>	<p>6. Точность и обоснованность выводов.</p> <p>7. Безошибочное выполнение практического задания.</p> <p>8. Точные, полные и логичные ответы на дополнительные вопросы.</p> <p>«ХОРОШО»:</p> <p>1. Хорошее знание программного материала.</p> <p>2. Недостаточно полное изложение теоретического вопроса экзаменационного билета.</p>
3.	ПК-3	Способен проектировать и разрабатывать информационные системы в соответствии с требованиями заказчика	<p><i>1 Этап – Знать:</i> ПК-3.1. Способы разработки прототипов информационных систем, мобильных и Web приложений</p> <p><i>2 Этап – Уметь:</i> ПК-3.2. Выполнять действия по проектированию, верификации информационных систем, мобильных и Web приложений в соответствии с требованиями заказчика</p> <p><i>3 Этап – Владеть:</i> ПК-3.3. Инструментами и методами разработки и тестирования баз данных информационных систем;</p>	<p>3. Наличие незначительных неточностей в употреблении терминов, классификаций.</p> <p>4. Знание основных пакетов прикладных программ</p> <p>5. Неполнота представленного иллюстративного материала.</p> <p>6. Точность и обоснованность выводов.</p> <p>7. Логичное изложение вопроса, соответствие изложения научному стилю.</p> <p>8. Негрубая ошибка при выполнении практического задания.</p> <p>9. Правильные ответы на дополнительные вопросы.</p> <p>«УДОВЛЕТВОРИТЕЛЬ»</p>

				<p>НО»:</p> <ol style="list-style-type: none"> <li>1. Поверхностное усвоение программного материала.</li> <li>2. Недостаточно полное изложение теоретического вопроса экзаменационного билета.</li> <li>3. Затруднение в приведении примеров, подтверждающих теоретические положения.</li> <li>4. Наличие неточностей в употреблении терминов, классификаций.</li> <li>5. Неумение четко сформулировать выводы.</li> <li>6. Отсутствие навыков научного стиля изложения.</li> <li>7. Грубая ошибка в практическом задании.</li> <li>8. Неточные ответы на дополнительные вопросы.</li> </ol> <p>«НЕУДОВЛЕТВОРИТЕЛЬНО»:</p> <ol style="list-style-type: none"> <li>1. Незнание значительной части программного материала.</li> <li>2. Неспособность привести примеры пакетов прикладных программ</li> <li>3. Неумение выделить главное, сделать выводы и обобщения.</li> <li>4. Грубые ошибки при выполнении практического задания.</li> <li>5. Неправильные</li> </ol>
--	--	--	--	--

				ОТВЕТЫ ДОПОЛНИТЕЛЬНЫЕ ВОПРОСЫ.	на
--	--	--	--	--------------------------------------	----

3. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

## 1 ЭТАП – ЗНАТЬ

### Устный опрос по темам:

1. Основные термины и определения
2. СУБД. Архитектура представления данных
3. Архитектуры организации работы с СУБД
4. Модели представления данных
5. Этапы проектирования. Инфологическое проектирование
6. Логическое проектирование
7. Команды определения таблиц
8. Команды выборки данных
9. Индексы
10. Команды манипулирования данными
11. Представления и защита данных
12. Транзакции
13. Хранимые процедуры и триггеры

## 2 ЭТАП – УМЕТЬ

### Индивидуальное задание № 1

Создайте ER-диаграмму для своего варианта предметной области.

**Вариант 1.** В базе данных необходимо хранить информацию о туристических путевках: код путевки, дата начала, дата окончания, город назначения, страна назначения, описание страны, отель, адрес отеля, телефон отеля, фамилия и имя туристов. За одной путевкой может быть закреплено несколько туристов. В одном городе может быть несколько отелей. Несколько путевок могут быть закреплены за одним отелем.

**Вариант 2.** В базе необходимо хранить информацию о технике, сданной на ремонт в компьютерный сервис: серийный номер, название (модель), тип проблемы, стоимость устранения проблемы, фамилия и имя владельца, телефон владельца. У каждой единицы техники только один владелец, у одного владельца может быть несколько единиц техники. У одной единицы техники может быть несколько технических проблем, одна и та же проблема может быть у разных единиц техники. Количество типов техники конечно.

**Вариант 3.** В базе необходимо хранить информацию об автомобилях: VIN-код, модель, цвет, владелец, даты начала и окончания регистрации автомобиля за владельцем. В базе может храниться информация о всех моделях, при этом несколько автомобилей могут быть одной модели. У одного автомобиля может быть несколько владельцев в разные периоды времени. Один человек может владеть несколькими автомобилями.

**Вариант 4.** В базе необходимо хранить информацию об автобусных рейсах на междугороднем автовокзале: номер маршрута, конечный пункт назначения рейса, населенные пункты остановок автобуса, время отправления, тип автобуса, количество мест в данном типе автобуса. По одному маршруту организовано один или более рейсов. За каждым рейсом закрепляется определенный тип автобуса, один тип автобуса может быть закреплен за несколькими рейсами. Через один населённый пункт может проходить несколько маршрутов. Каждый маршрут может проходить через несколько населённых пунктов.



**Вариант 5.** В базе данных необходимо хранить информацию о сотрудниках одного предприятия: номер личного дела, фамилия, имя, отчество, адрес, телефон, тип образования (высшее, среднее, незаконченное высшее, ...), должность, должностная инструкция, период занятия должности. Один тип образования может быть у нескольких сотрудников. Один сотрудник может занимать разные должности в разные периоды времени. На одной должности могут работать несколько сотрудников. У каждой должности своя должностная инструкция.

**Вариант 6.** В базе необходимо хранить информацию об успеваемости школьников: фамилия, имя школьника, учебные предметы, количество часов на предметы, учебный класс, оценка, дата оценки. В одном классе учатся несколько учеников, каждый ученик учится в одном классе. Каждый ученик изучает несколько предметов и один предмет изучают несколько учеников. У каждого ученика может быть несколько оценок по одному предмету в разные даты.

**Вариант 7.** В базе необходимо хранить информацию о кредитах: номер кредитного договора, дата договора, процентная ставка, сумма кредита, срок кредита, фамилия, имя и отчество клиента, паспортные данные клиента, фамилия, имя и отчество сотрудника банка. Один клиент может взять несколько кредитов, но каждый кредит закреплен за одним клиентом. Один сотрудник может оформить несколько кредитов, но каждый кредит оформляется одним сотрудником.

**Вариант 8.** В базе необходимо хранить информацию о товарах на складе: артикул товара, название товара, количество товара на складе, зоны склада, цвет зоны, вместительность зоны. На складе может быть несколько зон, в каждой зоне может храниться несколько товаров. Каждый товар может храниться в одной или нескольких зонах в определенном количестве.

**Вариант 9.** В базе необходимо хранить информацию о пациентах клиники: номер полиса, фамилия, имя, отчество, дата рождения, дата посещения врача, тип специалиста (терапевт, кардиолог, ...), фамилия, имя и отчество врача. Каждый врач является врачом конкретной специализации, в больнице может быть несколько врачей одной специализации. Клиент может посещать разных врачей и одного и того же врача по несколько раз.

**Вариант 10.** В базе необходимо хранить информацию о перевозках: номер перевозки, дата перевозки, пункт назначения, номер транспортного средства, тип транспорта, вместительность определенного типа транспорта, артикул товара, название товара, количество товара. У каждой перевозки только один пункт назначения и один тип транспорта. На перевозках работают несколько типов транспорта. Один тип транспорта может быть назначен нескольким перевозкам. В каждой перевозке может перевозиться несколько товаров в разном количестве. Один и тот же товар может перевозиться в разных перевозках.

**Вариант 11.** В базе необходимо хранить информацию о медицинских анализах: тип анализа, срок выполнения данного типа анализа, способ забора, наименование показателя, единица измерения, значение нормы для показателя. Каждому типу анализа соответствует один способ забора, с помощью одного способа забора можно получить разные типы анализов. Одному типу анализа соответствует один или несколько показателей. Один показатель может измеряться в различных типах анализа.

**Вариант 12.** В базе необходимо хранить информацию об опросах клиентов: фамилия, имя и отчество клиента, дата проведения опроса, название опроса, описание опроса, вопросы, ответы. В одном опросе может быть несколько вопросов. Один и тот же опрос проводят среди нескольких клиентов, и один клиент может участвовать в нескольких опросах.

**Вариант 13.** В базе необходимо хранить информацию о заявках службы технической поддержки: тип заявки (инцидент, консультация, ...), описание проблемы, дата заявки, фамилия и имя сотрудника, исполнитель заявки. Один сотрудник может

подать несколько заявок. Один исполнитель может выполнять несколько заявок. Одной заявки соответствует один конкретный тип заявки.

**Вариант 14.** В базе необходимо хранить информацию о заработной плате сотрудников предприятия: фамилия, имя, отчество сотрудника, должность, оклад для должности, количество отработанных дней в месяце, премия в месяце, сумма зарплаты. Сотрудникам зарплата начисляется ежемесячно. На одной должности может работать несколько сотрудников, один сотрудник занимает только одну должность.

**Вариант 15.** В базе необходимо хранить информацию о программе телепередач на день: канал, название передачи, время передачи, тип передачи (фильм, сериал, спорт, ...) Одна передача может идти на нескольких каналах. Каждая передача относится к одному типу, одному типу может соответствовать несколько передач.

**Вариант 16.** В базе необходимо хранить информацию о достопримечательностях города: название места, адрес места, время работы, тип достопримечательности (музей, парк, памятник, ...), экскурсия. Каждое место соответствует одному типу, а одному типу могут соответствовать несколько мест. В план одной экскурсии может входить посещение одного или нескольких мест. Одни и те же места можно посетить в нескольких экскурсиях.

**Вариант 17.** В базе необходимо хранить информацию о производственном плане: производимые продукты, прибыль от продажи одной единицы данного продукта, количество производимой продукции в день в цехе, дата, номер цеха, название цеха. В одном цехе может производиться только один тип продукции, один тип продукции может производиться в нескольких цехах.

**Вариант 18.** В базе необходимо хранить информацию о заказах клиентов: код заказа, дата заказа, фамилия, имя клиента, артикул товара, название товара, количество заказанного товара. Один клиент может сделать несколько заказов. Каждый заказ принадлежит одному клиенту. В одном заказе может быть заказано один или несколько товаров. Один и тот же товар может быть заказан в разных заказах.

### Индивидуальные задания №2

Преобразуйте ER-диаграмму в логическую модель для своего варианта индивидуального задания из практической работы №2.

### Индивидуальные задания №3

Разработать базу данных и составить запросы к ней.

#### Вариант 1. Информационная система Вуза

Студенты, организованные в группы, учатся на одном из факультетов, возглавляемом деканатом, в функции которого входит контроль за учебным процессом. В учебном процессе участвуют преподаватели кафедр, административно относящиеся к одному из факультетов. Преподаватели подразделяются на следующие категории: преподаватели, старшие преподаватели, доценты, профессора.

Учебный процесс регламентируется учебным планом, в котором указывается, какие учебные дисциплины на каких курсах и в каких семестрах читаются для студентов каждого года набора с указанием количества часов на каждый вид занятий по дисциплине (виды занятий: лекции, семинары, лабораторные работы, консультации, курсовые работы и т.д.) и формы контроля (зачет, экзамен, контрольная работа, курсовая работа и т.д.) Перед началом учебного семестра деканаты раздают на кафедры учебные поручения, в которых указываются какие дисциплины и для каких групп должны вести в очередном семестре. Руководствуясь ими, на кафедрах осуществляется распределение нагрузки. Преподаватель может вести занятия по одной или нескольким дисциплинам. Сведения о проведенных экзаменах и зачетах собираются деканатом.

Виды запросов в информационной системе:

1. Получить перечень и общее число студентов указанных групп либо указанного курса (курсов) факультета полностью, по половому признаку, году рождения, возрасту.

2. Получить список и общее число преподавателей указанных кафедр полностью либо указанных категорий (преподаватели, доценты, профессора и т.д.), по половому признаку, году рождения, возрасту, размеру заработной платы.

3. Получить перечень преподавателей, проводящих занятия в указанной группе либо на указанном курсе указанного факультета в указанном семестре, либо за указанный период.

4. Получить список студентов указанных групп с оценками за зачеты и экзамены по всем дисциплинам данного семестра.

5. Получить список и общее число студентов указанных групп или указанного курса указанного факультета, сдавших указанную сессию на отлично, без троек, без двоек.

6. Получить нагрузку преподавателей (название дисциплины, количество часов), ее объем по отдельным видам занятий и общую нагрузку в указанном семестре для конкретного преподавателя либо для преподавателей указанной кафедры.

Пользователи системы:

1. Администратор системы: полные права.

2. Сотрудники деканата: просмотр всех данных; ввод и редактирование данных о студентах, преподавателях, предметах.

3. Зав. кафедрами: просмотр всех данных; распределение нагрузки по преподавателям (закрепление часов по предметам за конкретным преподавателем)

4. Преподаватели: просмотр всех данных; ввод и редактирование оценок за экзамены, зачеты и т.д. только по читаемым ими дисциплинам.

5. Студент: только просмотр данных о своей успеваемости.

## **Вариант 2. Информационная система торговой организации.**

Торговая организация ведет торговлю в торговых точках. Заказы поставщику составляются на основе заявок, поступающих из торговых точек. На основе заявок менеджеры торговой организации выбирают поставщика, формируют заказы, в которых перечисляются наименования товаров и заказываемое их количество. Если указанное наименование товара ранее не поставлялось, оно пополняет справочник номенклатуры товаров. На основе маркетинговых работ постоянно изучается рынок поставщиков, в результате чего могут появляться новые поставщики и исчезать старые. При этом одни и те же товары торговая организация может получать от разных поставщиков и, естественно, по различным ценам. Поступившие товары распределяются по торговым точкам. Продавцы торговых точек ведут продажу товаров, учитывая все сделанные продажи, фиксируя номенклатуру и количество проданного товара

Виды запросов в информационной системе:

1. Получить перечень и общее число поставщиков, поставляющих указанный вид товара, либо некоторый товар за указанный период.

2. Получить номенклатуру и объем товаров в указанной торговой точке.

3. Получить данные об объеме продаж указанного товара за некоторый период по всем торговым точкам, по конкретной торговой точке.

4. Получить сведения о поставках определенного товара указанным поставщиком за все время поставок, либо за некоторый период.

5. Получить сведения о поставках товаров по указанному номеру заказа.

6. Получить данные о товарообороте торговой точки за указанный период.

Пользователи системы:

1. Администратор системы: полные права.

2. Менеджер торговой организации: просмотр всех данных; ввод и редактирование данных о торговых точках, номенклатуре товаров, поставщиках и т.д; формирование заявок поставщикам.

3. Менеджеры торговых точек: просмотр данных о своей торговой точке; формирование заявок на определенные товары, формирование заказов клиентов.

4. Клиенты: просмотр данных о товарах, их ценах, наличии в различных торговых точках.

### **Вариант 3. Информационная система автопредприятия города.**

Автопредприятие города занимается организацией пассажирских и грузовых перевозок внутри города. В ведении предприятия находится автотранспорт различного назначения: автобусы, такси, маршрутные такси, прочий легковой транспорт, грузовой транспорт, транспорт вспомогательного характера, представленный различными марками. Каждая из перечисленных категорий транспорта имеет характеристики, свойственные только этой категории: например, к характеристикам только грузового транспорта относится грузоподъемность, пассажирский транспорт характеризуется вместимостью и т.д. С течением времени, с одной стороны, транспорт стареет и списывается (возможно, продается), а с другой, - предприятие пополняется новым автотранспортом.

Предприятие имеет штат водителей, закрепленных за автомобилями (за одним автомобилем может быть закреплено более одного водителя)

Пассажирский автотранспорт (автобусы, маршрутные такси) перевозит пассажиров по определенным маршрутам, за каждым из них закреплены отдельные единицы автотранспорта. Ведется учет числа перевозимых пассажиров. Учитывается также пробег, число ремонтов и затраты на ремонт по всему автотранспорту, объем грузоперевозок для грузового транспорта, интенсивность использования транспорта вспомогательного назначения.

Виды запросов в информационной системе:

1. Получить перечень и общее число водителей по предприятию, по указанной автомашине.

2. Получить данные о распределении пассажирского автотранспорта по маршрутам.

3. Получить сведения о пробеге автотранспорта определенной категории или конкретной автомашины за указанный день, месяц, год.

4. Получить данные о числе ремонтов и их стоимости для автотранспорта определенной категории, отдельной марки автотранспорта или указанной автомашины за указанный период.

5. Получить сведения о грузоперевозках, выполненных указанной автомашиной за обозначенный период.

6. Получить сведения о полученной и списанной автотехники за указанный период.

Пользователи системы:

1. Администратор системы: полные права.

2. Руководитель организации: просмотр всех данных; ввод и редактирование любых данных.

3. Диспетчер: просмотр всех данных; ввод и редактирование данных о грузовых и пассажирских перевозках; распределение водителей по автомобилям.

4. Отдел обслуживания: просмотр данных о машинах; ввод и редактирование данных ремонте и обслуживании машин.

5. Водитель: просмотр данных только о машине, за которой он закреплен.

### **Вариант 4. Информационная система библиотечного фонда города**

Библиотечный фонд города составляют библиотеки, расположенные на территории города. Каждая библиотека включает в себя абонементы и читальные залы. Пользователями библиотек являются различные категории читателей: студенты, научные работники, преподаватели, школьники, рабочие, пенсионеры и другие жители города.

Каждая категория читателей может обладать непересекающимися характеристиками-атрибутами: для студентов это название учебного заведения, факультет, курс, номер группы, для научного работника - название организации, научная тема и т.д.

Библиотечный фонд (книги, журналы, газеты, сборники статей, сборники стихов, диссертации, рефераты, сборники докладов и тезисов докладов и пр.) размещен в залах-хранилищах различных библиотек на определенных местах хранения (номер зала, стеллажа, полки) и идентифицируется номенклатурными номерами. С одной стороны, библиотечный фонд может пополняться, с другой, - с течением времени происходит его списание.

Сотрудники библиотеки, работающие в различных залах различных библиотек, ведут учет читателей, а также учет размещения и выдачи литературы

Виды запросов в информационной системе:

1. Получить список читателей с заданными характеристиками: студентов указанного учебного заведения, факультета, научных работников по определенной тематике и т.д.
2. Выдать перечень читателей, на руках у которых находится указанное произведение.
3. Выдать список изданий, которые в течение некоторого времени получал указанный читатель из фонда библиотеки, где он зарегистрирован.
4. Получить данные о выработке библиотекарей (число обслуженных читателей в указанный период времени)
5. Получить список читателей с просроченным сроком литературы.
6. Получить перечень указанной литературы, которая поступила (была списана) в течение некоторого периода.

Пользователи системы:

1. Администратор системы: полные права.
2. Руководитель библиотечной системы города: просмотр всех данных; ввод данных о библиотеках, библиотекарях, книгах.
3. Руководители отдельных библиотек: просмотр всех данных; ввод и изменение данных о библиотекарях, работающих в данной библиотеке; ввод и редактирование данных о читальных залах и абонементе этой библиотеки.
4. Сотрудники: просмотр всех данных; ввод и редактирование данных о читателях, книгах и их расположении, данных о выдаче книг.
5. Читатель: только просмотр данных о книгах и их размещении.

### **Вариант 5. Информационная система гостиничного комплекса**

Гостиничный комплекс состоит из нескольких зданий-гостиниц (корпусов) Каждый корпус имеет ряд характеристик, таких, как класс отеля (двух-, пятизвездочные), количество этажей в здании, общее количество комнат, комнат на этаже, местность номеров (одно-, двух-, трехместные и т.д.), наличие служб быта: ежедневная уборка номера, прачечная, химчистка, питание (рестораны, бары) и развлечения (бассейн, сауна, бильярд и пр.) От типа корпуса и местности номера зависит сумма оплаты за него. Химчистка, стирка, дополнительное питание, все развлечения производятся за отдельную плату.

С крупными организациями заключаются договора, позволяющие организациям бронировать номера с большими скидками на определенное время вперед не для одного человека, а для группы людей. В брони указывается класс отеля, этаж, количество комнат и общее количество людей. На основе маркетинговых работ расширяется рынок гостиничных услуг, в результате чего заключаются договора с новыми фирмами. Изучается статистика популярности номеров. Ведется учет долгов постояльца гостинице за все дополнительные услуги.

Новые жильцы пополняют перечень клиентов гостиницы. Ведется учет свободных номеров, дополнительных затрат постояльцев гостиницы и учет расходов и доходов гостиничного комплекса.

Виды запросов в информационной системе:

1. Получить перечень и общее число фирм, забронировавших места в объеме, не менее указанного, за весь период сотрудничества, либо за некоторый период.
2. Получить перечень и общее число постояльцев, заселявшихся в номера с указанными характеристиками за некоторый период.
3. Получить сведения о количестве свободных номеров с указанными характеристиками.
4. Получить список занятых сейчас номеров, которые освобождаются к указанному сроку.
5. Получить сведения о конкретном человеке, сколько раз он посещал гостиницу, в каких номерах и в какой период останавливался, какие счета оплачивал.
6. Получить сведения о конкретном номере: кем он был занят в определенный период.

Пользователи системы:

1. Администратор системы: полные права.
2. Менеджер гостиничного комплекса: просмотр всех данных; ввод и редактирование данных о фирмах, зданиях, номерах, этажах, услугах и т.п.
3. Администраторы отдельных гостиниц: просмотр данных о своем отеле; ввод и редактирование данных о клиентах, о занятии/освобождении комнат, предоставлении/оплате услуг;
4. Клиенты: просмотр всех данных о номерах и услугах гостиничного комплекса.

### **Вариант 6. Информационная система магазина автозапчастей**

Магазин розничной торговли осуществляет заказ запчастей в различных странах. Ведется статистика продаж, отражающая спрос на те или иные детали, и, соответственно, потребность магазина в них (сколько единиц, на какую сумму, какого товара продано за последнее время) и на ее основе составляются заказы на требуемые товары. Выбор поставщика на каждый конкретный заказ осуществляют менеджеры магазина. В заказах перечисляется наименование товара, количество. Если указанное наименование товара ранее не поставлялось, оно пополняет справочник номенклатуры товаров.

В ходе маркетинговых работ изучается рынок поставщиков, в результате чего могут появляться новые поставщики и исчезать старые.

В любой момент можно получить любую информацию о деталях, находящихся на складе, либо о поставляемых деталях. Детали хранятся на складе в определенных ячейках. Все ячейки пронумерованы. Касса занимается приемом денег от покупателей за товар.

Виды запросов в информационной системе:

1. Получить перечень и общее число поставщиков, поставляющих указанный вид товара, либо поставивших указанный товар в объеме, не менее заданного за определенный период.
2. Получить сведения о конкретном виде деталей: какими поставщиками поставляется, их расценки, время поставки.
3. Получить перечень и общее число покупателей, купивших указанный вид товара за некоторый период, либо сделавших покупку товара в объеме, не менее указанного.
4. Вывести в порядке возрастания десять самых продаваемых деталей и десять самых "дешевых" поставщиков.
5. Получить долю товара конкретного поставщика в процентах, деньгах, единицах от всего оборота магазина прибыль магазина за указанный период.
6. Получить инвентаризационную ведомость.

Пользователи системы:

1. Администратор системы: полные права.
2. Менеджер магазина: просмотр всех данных; ввод и редактирование данных о товарах, их характеристиках, ценах, поставщиках; формирование заявок поставщикам.
3. Продавцы: просмотр данных о товаре; ввод данных о продаже.
4. Клиенты: просмотр данных о товарах, их ценах, наличии.

### **Вариант 7. Информационная система проектной организации**

Проектная организация представлена следующими категориями сотрудников: конструкторы, инженеры, техники, лаборанты, прочий обслуживающий персонал, каждая из которых может иметь свойственные только ей атрибуты. Например, конструктор характеризуется числом авторских свидетельств, техники - оборудованием, которое они могут обслуживать, инженер или конструктор может руководить договором или проектом и т.д. Сотрудники разделены на отделы, руководимые начальником так, что каждый сотрудник числится только в одном отделе.

В рамках заключаемых проектной организацией договоров с заказчиками выполняются различного рода проекты, причем по одному договору может выполняться более одного проекта, и один проект может выполняться для нескольких договоров. Суммарная стоимость договора определяется стоимостью всех проектных работ, выполняемых для этого договора. Каждый договор и проект имеет руководителя и группу сотрудников, выполняющих этот договор или проект, причем это могут быть сотрудники только одного отдела.

Ведется учет кадров, учет выполнения договоров и проектов, стоимостной учет всех выполненных работ.

Виды запросов в информационной системе:

1. Получить данные о составе указанного отдела или всей организации полностью, по указанной категории сотрудников, по возрастному составу.
2. Получить перечень договоров или проектов, выполняемых в данный момент или в период указанного интервала времени.
3. Получить информацию о том, какие проекты выполняются (выполнялись) в рамках указанного договора и какие договора поддерживаются указанными проектами.
4. Получить данные о стоимости выполненных договоров (проектов) в течение указанного периода времени.
5. Получить данные о численности и составе сотрудников в целом и по отдельным категориям, участвующих в указанном проекте.
6. Получить сведения об эффективности проектов (стоимость договоров, соотнесенная с затраченным временем или стоимость с учетом привлеченных людских ресурсов)

Пользователи системы:

1. Администратор системы: полные права.
2. Руководитель организации: просмотр всех данных; ввод и редактирование данных об отделах и сотрудниках; формирование договоров и проектов по этим договорам.
3. Инженеры и конструкторах: просмотр и редактирование данных о проектах, закрепленных за ними.
4. Клиенты: просмотр данных только о своих договорах и проектах.

### **Вариант 8. Информационная система автомобилестроительного предприятия**

Структурно предприятие состоит из цехов, которые в свою очередь подразделяются на участки. Выпускаемые изделия предприятия: грузовые, легковые автомобили, автобусы, сельскохозяйственные, дорожно-строительные машины,

мотоциклы и прочие изделия. Каждая категория изделий имеет специфические, присущие только ей атрибуты. Например, для автобусов это вместимость, для сельскохозяйственных и дорожно-строительных машин - производительность и т.д. По каждой категории изделий может собираться несколько видов изделий. Каждой категории инженерно-технического персонала (инженеры, технологи, техники) и рабочих (сборщики, токари, слесари, сварщики и пр.) также характерны атрибуты.

Каждое изделие собирается в своем цехе (в цехе может собираться несколько видов изделий) и в процессе изготовления проходит определенный цикл работ, перемещаясь с одного участка на другой.

Собранное изделие проходит серию испытаний в испытательных лабораториях. Испытательные лаборатории могут обслуживать несколько цехов, в свою очередь цеха могут пользоваться несколькими лабораториями.

Ведется учет движения кадров и учет выпускаемой продукции.

Виды запросов в информационной системе:

1. Получить перечень видов изделий отдельной категории и в целом, собираемых указанным цехом, предприятием.

2. Получить число и перечень изделий отдельной категории и в целом, собранных указанным цехом, предприятием в целом за определенный отрезок времени.

3. Получить данные о кадровом составе цеха, предприятия в целом и по указанным категориям инженерно-технического персонала и рабочих.

4. Получить число и перечень участков указанного цеха, предприятия в целом и их начальников.

5. Получить перечень мастеров указанного цеха.

6. Получить число и перечень изделий отдельной категории и в целом, собираемых указанным цехом, участком, предприятием в настоящее время.

Пользователи системы:

1. Администратор системы: полные права.

2. Руководитель предприятия: просмотр всех данных; ввод и редактирование данных о цехах, участках, выпускаемой продукции, рабочих и сотрудников.

3. Начальники цехов и лаборатории: просмотр данных о продукции и сотрудниках данного цеха/лаборатории; ввод и редактирование данных о производстве данного цеха/лаборатории.

4. Клиенты: просмотр данных о производимых машинах, их количестве, дате сборки.

### **Вариант 9. Информационная система аптеки**

Аптека продает медикаменты. Лекарства могут быть разных типов: таблетки, мази, настойки.

На складе на все медикаменты устанавливается критическая норма, т.е. когда какого-либо лекарства на складе меньше критической нормы, то составляются заявки на данные лекарства и их в срочном порядке привозят с оптовых складов медикаментов.

Для продажи аптекой лекарства, больной должен принести рецепт от лечащего врача. В рецепте должно быть указано: ФИО, подпись и печать врача, ФИО, возраст и диагноз пациента, также количество лекарства и способ применения. Больной отдает рецепт регистратору, он принимает заказ и смотрит, есть ли заказываемое лекарства. Если не все лекарства имеются в наличии, то делает заявки на оптовые склады лекарств и фиксирует ФИО, телефон и адрес необслуженного покупателя, чтобы сообщить ему, когда доставят нужные лекарства. Такой больной пополняет справочник заказов - это те заказы, которые находятся в процессе обработке, с пометкой, что не все лекарства есть для заказа. Если все лекарства имеются, то покупатель выплачивает цену лекарства. Больной также пополняет справочник заказов.



Ведется статистика по объемам используемых медикаментов. Через определенный промежуток времени производится инвентаризация склада. Это делается для того, чтобы определить, есть ли лекарства с критической нормой, или вышел срок хранения или не достача.

Виды запросов в информационной системе:

1. Получить перечень и общее число покупателей, которые ждут прибытия на склад нужных им медикаментов в целом и по указанной категории медикаментов.
2. Получить перечень десяти наиболее часто используемых медикаментов в целом и указанной категории медикаментов.
3. Получить какой объем указанных лекарств использован за указанный период.
4. Получить перечень и типы лекарств, достигших своей критической нормы или закончившихся.
5. Получить полный перечень и общее число заказов находящихся в ожидании.
6. Получить сведения о наиболее часто делающих заказы клиентах на медикаменты определенного типа, на конкретные медикаменты.

Пользователи системы:

1. Администратор системы: полные права.
2. Администратор аптеки: просмотр всех данных; ввод и редактирование данных о лекарствах, их характеристиках, ценах, количестве на складе; формирование заявок на лекарства.
3. Фармацевты-продавцы: просмотр, ввод и редактирование о продажах и заказах больных; просмотр, ввод и редактирование о больных.
4. Клиенты: просмотр данных о лекарствах, их ценах, наличии.

#### **Вариант 10. Информационная система городской телефонной сети**

ГТС представляет собой разветвленную сеть локальных АТС. У каждой АТС есть свои абоненты. За каждым абонентом (у него есть фамилия, имя, отчество, пол, возраст и т.д.) закреплен свой номер телефона. Каждому номеру телефона соответствует адрес (индекс, район, улица, дом, квартира).

Все телефоны городской АТС имеют выход на межгород, но для конкретного абонента он может быть либо открыт, либо закрыт по какой-либо причине (отключен по желанию абонента, за неуплату и т.п.) Сведения о междугородных переговорах собираются и анализируются на ГТС.

Абоненты обязаны платить абонентскую плату. Плата должна вноситься каждый месяц до 20-го числа. При неуплате после письменного уведомления в течение двух суток отключается абонент. При задолженности за междугородние разговоры и неоплате после письменного уведомления производится отключение только возможности выхода на межгород. Включение того и (или) другого производится при оплате стоимости включения, абонентской платы и пени.

Абонентов любой АТС можно подразделить на простых и льготных. К категории льготников относятся пенсионеры, инвалиды и т.д. Льготники платят только 50% абонентской платы. В соответствии со всем этим (тип телефона, льготник или нет, есть ли выход на межгород) рассчитывается размер абонентской платы.

Виды запросов в информационной системе:

1. Получить перечень и общее число абонентов указанной АТС полностью, только льготников, по возрастному признаку, по группе фамилий.
2. Получить перечень и общее число должников на указанной АТС, по всей ГТС, по данному району, абонентов, которые имеют задолженность уже больше недели (месяца), по признаку задолженности за межгород и (или) по абонентской плате, по размеру долга.
3. Определить АТС, на которой самое большое (маленькое) число должников, самая большая сумма задолженности.

4. Найти процентное соотношение обычных и льготных абонентов на указанной АТС, по всей ГТС, по данному району, по типам АТ.

5. Определить десять номеров, с которых происходит большее всего количество междугородных переговоров.

6. Получить полную информацию об абонентах с заданным телефонным номером.

Пользователи системы:

1. Администратор системы: полные права.

2. Руководитель ГТС: просмотр всех данных; ввод и редактирование данных о АТС, телефонах и их абонентах.

3. Фармацевты-продавцы: просмотр, ввод и редактирование о телефонах, абонентах, их оплате только номеров, привязанных к данной АТС.

4. Клиенты: просмотр истории оплаты за свой телефонный номер.

### **Вариант 11. Информационная система фотоцентра**

Фотоцентр имеет главный офис и сеть филиалов и киосков приема заказов, расположенных по определенным адресам. Филиалы и киоски различаются количеством рабочих мест. В киосках осуществляется только прием заказов, поэтому каждый киоск прикреплен к определенному филиалу, в котором эти заказы выполняются. В филиалах имеется необходимое оборудование для печати фотографий. Филиалы и киоски принимают заказы на печать фотографий. В заказе на печать указывается количество фотографий с каждого кадра, общее количество фотографий, формат, тип бумаги и срочность выполнения заказа. При приобретении дисконтной карты клиент получает значительные скидки на печать фотографий.

Фотомагазины и киоски предлагают к продаже различные фототовары: фотопленки, фотоаппараты, альбомы и другие фотопринадлежности. Фотомагазины также предлагают дополнительные виды услуг: фотографии на документы, реставрация фотографий, прокат фотоаппаратов, художественное фото, предоставление услуг профессионального фотографа.

Виды запросов в информационной системе:

1. Получить перечень и общее число пунктов приема заказов на фотоработы по филиалам, по киоскам приема заказов, в целом по фотоцентру.

2. Получить перечень и общее число заказов на фотоработы по филиалам, киоскам приема заказов, в целом по фотоцентру, поступивших в течение некоторого периода времени.

3. Получить сумму выручки с заказов (отдельно простых и срочных) на отдельные виды фоторабот по указанному филиалу, киоску приема заказов, поступивших в течение некоторого периода времени.

4. Получить количество отпечатанных фотографий в рамках простых и срочных заказов по указанному филиалу, киоску приема заказов, фотоцентру в целом за некоторый период времени.

5. Получить список клиентов в целом по фотоцентру, клиентов указанного филиала, имеющих скидки, сделавших заказы определенного объема.

6. Получить перечень реализованных фототоваров и объемы их реализации в целом по фотоцентру, по указанному филиалу, проданных в течение некоторого периода времени.

Пользователи системы:

1. Администратор системы: полные права.

2. Администратор фотоцентра: просмотр всех данных; ввод и редактирование данных о филиалах и киосках, о ценах на печать, услуги и товаров, о перечне товаров и услуг.

3. Администратор филиала/киоска: просмотр, редактирование и ввод данных о товарах, услугах, заказах, клиентах в данном филиале.

4. Клиенты: просмотр данных о товарах и услугах фотоцентра.

### Кейс-задачи по дисциплине «Базы данных»

**Задача 1. Создайте базу данных orders и выполните запросы представленные ниже**

```
CREATE TABLE `user` (
  id_user int(11) PRIMARY KEY,
  fname varchar(30) NOT NULL,
  lname varchar(30) NOT NULL,
  gender char(1) NOT NULL
)
CREATE TABLE `order` (
  id_order int(11) PRIMARY KEY,
  id_user int(11),
  cost float,
  o_date date NOT NULL,
  status varchar(20),
  FOREIGN KEY (id_user) REFERENCES user(id_user) ON DELETE CASCADE ON
UPDATE CASCADE
)
CREATE TABLE `order_history`(
  id_order int(11) PRIMARY KEY,
  id_user int(11),
  cost float,
  o_date date
)
INSERT INTO `user`
VALUES (1,'Иван','Иванов','м'),
(2,'Анна','Фролова','ж'),
(3,'Ирина','Ефремова','ж'),
(4,'Олег','Петров','м'),
(5,'Ольга','Степанова','ж'),
(6,'Андрей','Игнатъев','м'),
(7,'Петр','Фомин','м')

INSERT INTO `order`
VALUES (1,1,45000,'2014/01/30','выполнен'),
(2,2,5500,'2014/01/25','выполнен'),
(3,7,155000,'2014/02/05','выполнен'),
(4,3,60000,'2014/02/05','выполнен'),
(5,2,45000,'2014/02/07','выполнен'),
(6,1,4560,'2014/02/07','выполнен'),
(7,5,33300,'2014/02/09','выполнен'),
(8,4,45120,'2014/02/10','выполнен'),
(9,6,8000,'2014/02/11','выполнен'),
(10,6,78000,'2014/02/15','выполнен'),
(11,1,45061,'2014/02/16','в обработке'),
(12,5,15007,'2014/02/16','в обработке'),
(13,4,45123,'2014/02/17','в обработке'),
```

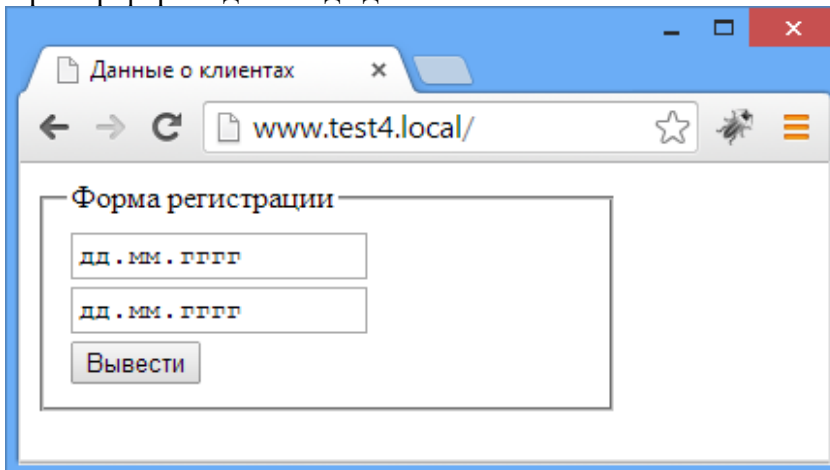
```
(14,6,1302,'2014/02/18','в обработке'),  
(15,6,56996,'2014/02/25','в обработке'),  
(16,4,33420,'2014/02/26','выполнен'),  
(17,2,2395,'2014/03/01','в обработке'),  
(18,6,5090,'2014/03/05','выполнен'),  
(19,4,67129,'2014/03/17','в обработке'),  
(21,7,7994,'2014/03/25','в обработке')
```

```
INSERT INTO `order_history`  
SELECT `id_order`, `id_user`, `cost`, `o_date` FROM `order`  
WHERE `status`='выполнен'
```

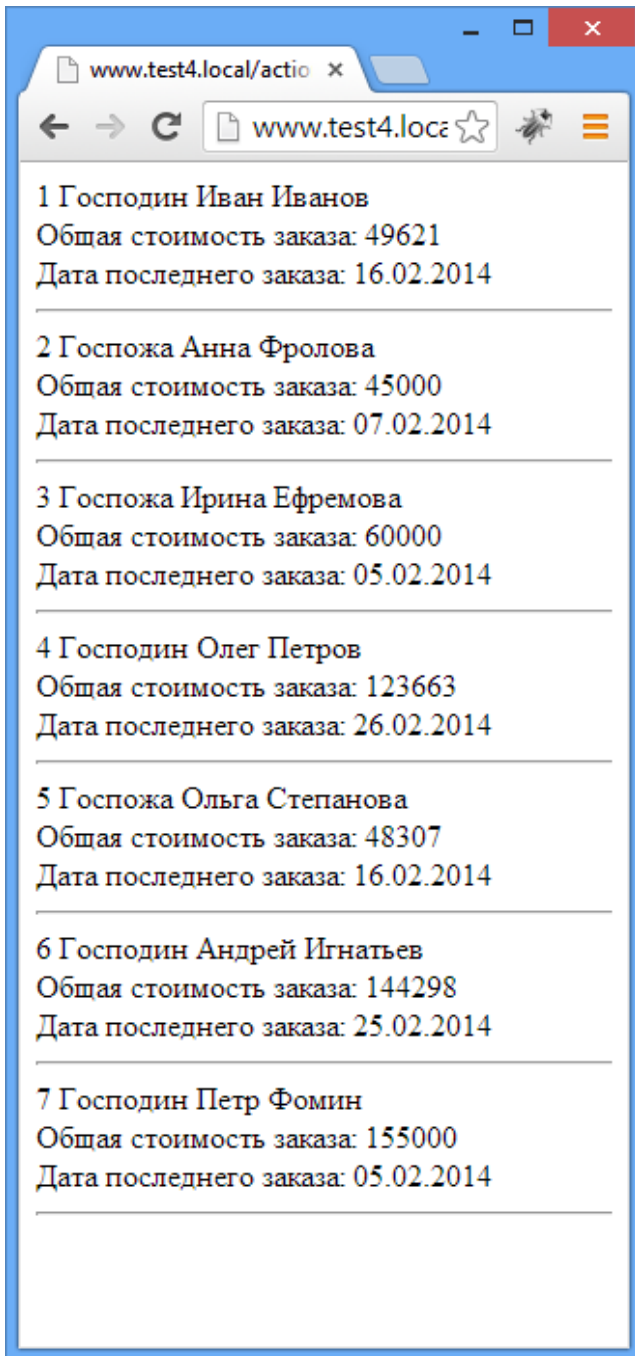
```
DELETE FROM `order` WHERE `status`='выполнен'
```

**Постановка задачи:** Реализовать веб-приложение, которое бы выводила список пользователей (**id**, имя и фамилию) и данных об их заказах (общая сумма всех заказов, дата последнего заказа), осуществлявших заказ в указанный период.

Пример формы для ввода дат:

The image shows a screenshot of a web browser window. The browser's address bar displays 'www.test4.local/'. The page content features a form titled 'Форма регистрации' (Registration Form). Inside the form, there are two text input fields, each containing the placeholder text 'дд.мм.гггг' (dd.mm.yyyy). Below these fields is a button labeled 'Вывести' (Output/Show). The browser window has a single tab titled 'Данные о клиентах' (Client Data).

Форма вывода результатов (для дат с 1.02.2014 по 28.02.2014):



**Задача 2. Создайте базу данных orders и выполните запросы представленные ниже**

```
CREATE TABLE `user` (
  id_user int(11) PRIMARY KEY,
  fname varchar(30) NOT NULL,
  lname varchar(30) NOT NULL,
  gender char(1) NOT NULL
)
CREATE TABLE `order` (
  id_order int(11) PRIMARY KEY,
  id_user int(11),
  cost float,
  o_date date NOT NULL,
  status varchar(20),
```

```
FOREIGN KEY (id_user) REFERENCES user(id_user) ON DELETE CASCADE ON
UPDATE CASCADE
```

```
)
CREATE TABLE `order_history` (
    id_order int(11) PRIMARY KEY,
    id_user int(11),
    cost float,
    o_date date
```

```
)
INSERT INTO `user`
VALUES (1,'Иван','Иванов','м'),
(2,'Анна','Фролова','ж'),
(3,'Ирина','Ефремова','ж'),
(4,'Олег','Петров','м'),
(5,'Ольга','Степанова','ж'),
(6,'Андрей','Игнатъев','м'),
(7,'Петр','Фомин','м')
```

```
INSERT INTO `order`
VALUES (1,1,45000,'2014/01/30','выполнен'),
(2,2,5500,'2014/01/25','выполнен'),
(3,7,155000,'2014/02/05','выполнен'),
(4,3,60000,'2014/02/05','выполнен'),
(5,2,45000,'2014/02/07','выполнен'),
(6,1,4560,'2014/02/07','выполнен'),
(7,5,33300,'2014/02/09','выполнен'),
(8,4,45120,'2014/02/10','выполнен'),
(9,6,8000,'2014/02/11','выполнен'),
(10,6,78000,'2014/02/15','выполнен'),
(11,1,45061,'2014/02/16','в обработке'),
(12,5,15007,'2014/02/16','в обработке'),
(13,4,45123,'2014/02/17','в обработке'),
(14,6,1302,'2014/02/18','в обработке'),
(15,6,56996,'2014/02/25','в обработке'),
(16,4,33420,'2014/02/26','выполнен'),
(17,2,2395,'2014/03/01','в обработке'),
(18,6,5090,'2014/03/05','выполнен'),
(19,4,67129,'2014/03/17','в обработке'),
(21,7,7994,'2014/03/25','в обработке')
```

```
INSERT INTO `order_history`
SELECT `id_order`, `id_user`, `cost`, `o_date` FROM `order`
WHERE `status`='выполнен'
```

```
DELETE FROM `order` WHERE `status`='выполнен'
```

**Постановка задачи:** Реализовать веб-приложение, которое бы выводила список пользователей (id, имя и фамилию) и данных об их заказах (общая сумма всех заказов, дата последнего заказа), осуществлявших заказ в указанный период. Реализовать все только через одно обращение к базе данных (за один запрос – используя только одну команду mysql\_query)

Пример формы для ввода дат:

Форма регистрации

дд . мм . гggg

дд . мм . гggg

Вывести

Форма вывода результатов (для дат с 1.02.2014 по 28.02.2014):

www.test4.local/actio

1 Господин Иван Иванов  
Общая стоимость заказа: 49621  
Дата последнего заказа: 16.02.2014

2 Госпожа Анна Фролова  
Общая стоимость заказа: 45000  
Дата последнего заказа: 07.02.2014

3 Госпожа Ирина Ефремова  
Общая стоимость заказа: 60000  
Дата последнего заказа: 05.02.2014

4 Господин Олег Петров  
Общая стоимость заказа: 123663  
Дата последнего заказа: 26.02.2014

5 Госпожа Ольга Степанова  
Общая стоимость заказа: 48307  
Дата последнего заказа: 16.02.2014

6 Господин Андрей Игнатьев  
Общая стоимость заказа: 144298  
Дата последнего заказа: 25.02.2014

7 Господин Петр Фомин  
Общая стоимость заказа: 155000  
Дата последнего заказа: 05.02.2014

**Задача 3. Создайте базу данных orders и выполните запросы представленные ниже**

```
CREATE TABLE `user` (
  id_user int(11) PRIMARY KEY,
  fname varchar(30) NOT NULL,
  lname varchar(30) NOT NULL,
  gender char(1) NOT NULL
)
```

```
CREATE TABLE `order` (
  id_order int(11) PRIMARY KEY,
  id_user int(11),
  cost float,
  o_date date NOT NULL,
```

```

        status varchar(20),
        FOREIGN KEY (id_user) REFERENCES user(id_user) ON DELETE CASCADE ON
UPDATE CASCADE
    )
CREATE TABLE `order_history`(
    id_order int(11) PRIMARY KEY,
    id_user int(11),
    cost float,
    o_date date
)
INSERT INTO `user`
VALUES (1,'Иван','Иванов','м'),
(2,'Анна','Фролова','ж'),
(3,'Ирина','Ефремова','ж'),
(4,'Олег','Петров','м'),
(5,'Ольга','Степанова','ж'),
(6,'Андрей','Игнатъев','м'),
(7,'Петр','Фомин','м')

INSERT INTO `order`
VALUES (1,1,45000,'2014/01/30','выполнен'),
(2,2,5500,'2014/01/25','выполнен'),
(3,7,155000,'2014/02/05','выполнен'),
(4,3,60000,'2014/02/05','выполнен'),
(5,2,45000,'2014/02/07','выполнен'),
(6,1,4560,'2014/02/07','выполнен'),
(7,5,33300,'2014/02/09','выполнен'),
(8,4,45120,'2014/02/10','выполнен'),
(9,6,8000,'2014/02/11','выполнен'),
(10,6,78000,'2014/02/15','выполнен'),
(11,1,45061,'2014/02/16','в обработке'),
(12,5,15007,'2014/02/16','в обработке'),
(13,4,45123,'2014/02/17','в обработке'),
(14,6,1302,'2014/02/18','в обработке'),
(15,6,56996,'2014/02/25','в обработке'),
(16,4,33420,'2014/02/26','выполнен'),
(17,2,2395,'2014/03/01','в обработке'),
(18,6,5090,'2014/03/05','выполнен'),
(19,4,67129,'2014/03/17','в обработке'),
(21,7,7994,'2014/03/25','в обработке')

INSERT INTO `order_history`
SELECT `id_order`, `id_user`, `cost`, `o_date` FROM `order`
WHERE `status`='выполнен'

DELETE FROM `order` WHERE `status`='выполнен'

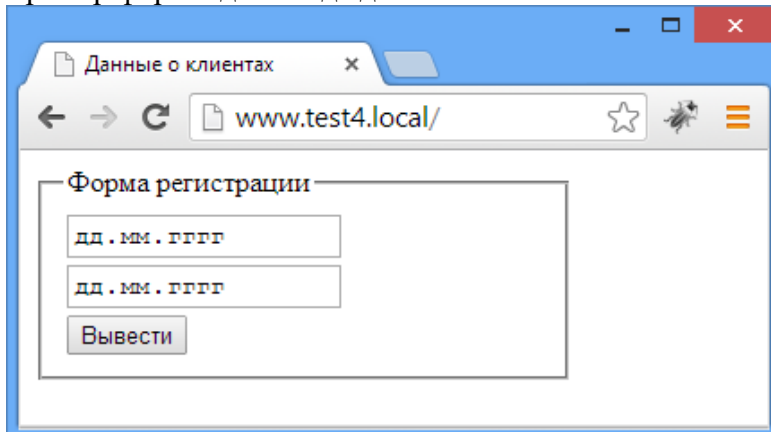
```

**Постановка задачи:** Реализовать веб-приложение, которое бы выводила список пользователей (id, имя и фамилию) и данных об их заказах (общая сумма всех заказов, дата последнего заказа), осуществлявших заказ в указанный период. Реализовать все только через одно обращение к базе данных (за один запрос – используя только одну команду mysql\_query) В коде программы не должно быть



условных операторов, вся информация должна формироваться в теле запроса. К стоимости добавить вывод денежных единиц (например, 456 рублей, 8901 рубль)

Пример формы для ввода дат:



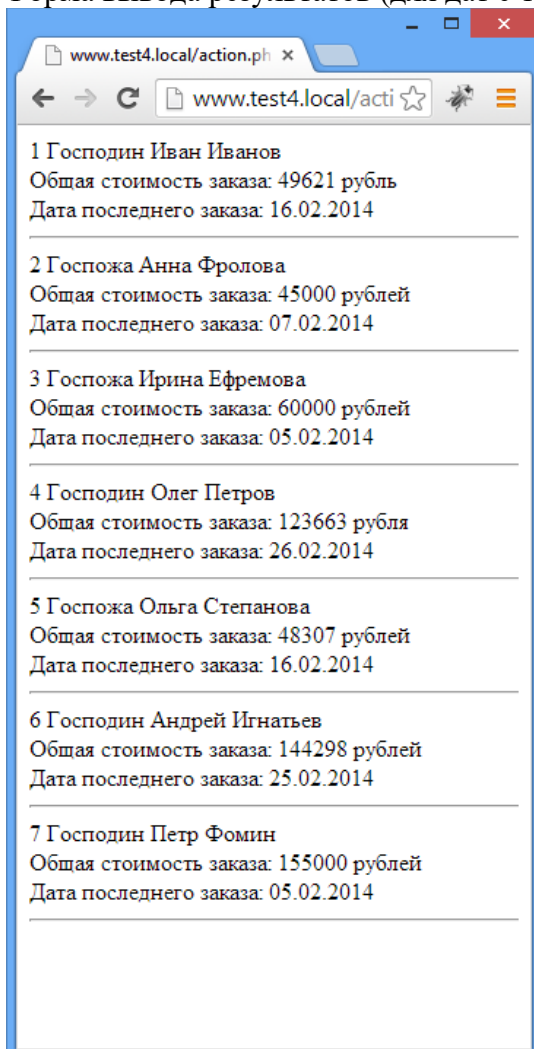
Форма регистрации

дд . мм . гггг

дд . мм . гггг

Вывести

Форма вывода результатов (для дат с 1.02.2014 по 28.02.2014):



www.test4.local/action.ph x

www.test4.local/acti

1 Господин Иван Иванов  
Общая стоимость заказа: 49621 рубль  
Дата последнего заказа: 16.02.2014

2 Госпожа Анна Фролова  
Общая стоимость заказа: 45000 рублей  
Дата последнего заказа: 07.02.2014

3 Госпожа Ирина Ефремова  
Общая стоимость заказа: 60000 рублей  
Дата последнего заказа: 05.02.2014

4 Господин Олег Петров  
Общая стоимость заказа: 123663 рубля  
Дата последнего заказа: 26.02.2014

5 Госпожа Ольга Степанова  
Общая стоимость заказа: 48307 рублей  
Дата последнего заказа: 16.02.2014

6 Господин Андрей Игнатьев  
Общая стоимость заказа: 144298 рублей  
Дата последнего заказа: 25.02.2014

7 Господин Петр Фомин  
Общая стоимость заказа: 155000 рублей  
Дата последнего заказа: 05.02.2014

### 3 ЭТАП – ВЛАДЕТЬ

#### Темы групповых и/или индивидуальных творческих проектов по дисциплине «Базы данных»

##### Темы проектов:

1. Разработка базы данных для прототипа бизнес-приложения “Банкомат” с использованием базы данных.
2. Разработка базы данных для прототипа бизнес-приложения “Кинотеатр” с использованием базы данных.
3. Разработка базы данных для прототипа бизнес-приложения “Авиакасса” с использованием базы данных.
4. Разработка базы данных для прототипа бизнес-приложения “Поисковая система” с использованием базы данных.
5. Разработка базы данных для прототипа бизнес-приложения “Интернет-магазин” с использованием базы данных.
6. Разработка базы данных для прототипа бизнес-приложения “Учет пациентов клиники” с использованием базы данных.
7. Разработка базы данных для прототипа бизнес-приложения “Учет техники” с использованием базы данных.
8. Разработка базы данных для прототипа бизнес-приложения “Отдел кадров” с использованием базы данных.

#### Вопросы к экзамену

1. Общие понятия реляционного подхода к организации БД.
2. Системы хранения данных на основе файлов.
3. Понятие базы данных.
4. Назначение и основные функции СУБД.
5. Типовая организация СУБД.
6. Последовательность действий при выполнении запроса.
7. Понятие модели данных.
8. Иерархические системы.
9. Сетевые системы.
10. Общие понятия реляционного подхода к организации БД.
11. Основные этапы проектирования БД.
12. Проектирование реляционных БД с использованием нормализации.
13. Назначение нормализации отношений.
14. Проектирование БД.
15. Основные этапы проектирования БД.
16. Проектирование реляционных БД с использованием нормализации.
17. Назначение нормализации отношений.
18. Проектирование БД.
19. Семантические модели данных: ER. (Сущность-Связь)
20. Физическое представление структур данных.
21. Методы хранения данных и доступа к ним.
22. Создание баз данных Microsoft SQL Server.
23. Многотабличные базы данных.
24. Общие сведения об SQL.
25. Состав языка SQL.
26. Средства DDL.
27. SQL. Реляционные операции.

28. Команды языка манипулирования данными: выборка данных.
29. Команды языка манипулирования данными: внесение изменений.
30. Команды языка манипулирования данными: создание базы данных.
31. Объектно-ориентированные СУБД.

### **Примерный перечень тем курсовых работ:**

1. Проектирование и разработка информационной системы для работника склада (варианты: склад торговой организации, занимающейся сбытом как продукции собственного производства, так и продукции внешних поставщиков; склад оптовой торговой организации; склад готовой продукции; склад сырья и материалов и др.).
2. Проектирование и разработка информационной системы для контроля выполнения нагрузки преподавателей ВУЗа.
3. Проектирование и разработка информационной системы для контроля сессионной успеваемости студентов ВУЗа.
4. Проектирование и разработка информационной системы для учета контингента студентов ВУЗа.
5. Проектирование и разработка информационной системы для организации дипломного проектирования в ВУЗе.
6. Проектирование и разработка информационной системы для организации курсового проектирования.
7. Проектирование и разработка информационной системы для студенческого профкома ВУЗа.
8. Проектирование и разработка информационной системы для начисления стипендии в ВУЗе.
9. Проектирование и разработка информационной системы для библиотеки ВУЗа.
10. Проектирование и разработка информационной системы для управления работой компьютерных аудиторий учебного заведения.
11. Проектирование и разработка информационной системы для управления работой класса свободного доступа.
12. Проектирование и разработка информационной системы для начисления заработной платы преподавателей.
13. Проектирование и разработка информационной системы для контроля успеваемости школьников.
14. Проектирование и разработка информационной системы детского сада.
15. Проектирование и разработка информационной системы спортивной школы.
16. Проектирование и разработка информационной системы центра детского творчества.
17. Проектирование и разработка информационной системы партнеров клининговой фирмы.
18. Проектирование и разработка информационной системы коммерческого учебного центра для дошкольников.
19. Проектирование и разработка информационной системы коммерческого учебного центра для изучения иностранного языка.
20. Проектирование и разработка информационной системы для расчета заработной платы преподавателей ВУЗа.
21. Проектирование и разработка информационной системы для учета домашних финансов.
22. Проектирование и разработка информационной системы для домашней библиотеки.
23. Проектирование и разработка информационной системы для районной библиотеки.
24. Проектирование и разработка информационной системы для домашней видеотеки.
25. Проектирование и разработка информационной системы для пункта проката видеофильмов.
26. Проектирование и разработка информационной системы кинотеатра.

27. Проектирование и разработка информационной системы драматического театра.
28. Проектирование и разработка информационной системы для домашней аудиотеки.
29. Проектирование и разработка информационной системы тренера спортивной команды.
30. Проектирование и разработка информационной системы агентства по аренде квартир.
31. Проектирование и разработка информационной системы риэлтерского агентства.
32. Проектирование и разработка информационной системы для учета услуг, оказываемых юридической консультационной фирмой.
33. Проектирование и разработка информационной системы для автосервисной фирмы.
34. Проектирование и разработка информационной системы для автозаправочной станции.
35. Проектирование и разработка информационной системы центра по продаже автомобилей.
36. Проектирование и разработка информационной системы таксомоторного парка.
37. Проектирование и разработка информационной системы по подсистеме «Кадры» для ВУЗа.
38. Проектирование и разработка информационной системы по подсистеме «Кадры» для школы.
39. Проектирование и разработка информационной системы по подсистеме «Кадры» для промышленного предприятия.
40. Проектирование и разработка информационной системы по подсистеме «Кадры» для торговой фирмы.
41. Проектирование и разработка информационной системы районной поликлиники. Подсистема «Работа с пациентами».
42. Проектирование и разработка информационной системы автоматизированной системы инспектора ГИБДД.
43. Проектирование и разработка информационной системы информационной системы военного округа.
44. Проектирование и разработка информационной системы для гостиничного комплекса.
45. Проектирование и разработка информационной системы для автомобилестроительного предприятия.
46. Проектирование и разработка информационной системы для представительства туристической фирмы в зарубежной стране.
47. Проектирование и разработка информационной системы для аптеки.
48. Проектирование и разработка информационной системы для городской телефонной сети.
49. Проектирование и разработка информационной системы для аэропорта.
50. Проектирование и разработка информационной системы для гостиниц.
51. Проектирование и разработка информационной системы для фотоателье.

**Тестовые задания**  
по разделам «Введение в базы данных» и «Проектирование баз данных»

№	Фамилия	Имя	Отчество	Адрес	Телефон
1	Иванов	Иван	Иванович	Ул. Володарского, 73-6	255-67-03
2	Петров	Петр	Евгеньевич	Ул. Калинина, 13-25	345-76-89
3	Сидоров	Константин	Викторович	Ул. Ф. Горелова, 43-879	326-98-05

## 1) Выделенный фрагмент таблицы является

- А). Атрибутом *неверно*  
 В). Отношением *неверно*  
 С). Значением атрибутов *неверно*  
 D). Схемой отношения *неверно*  
 E). Кортежем *верно*

## 2) Выделенный фрагмент таблицы является

№	Фамилия	Имя	Отчество	Адрес	Телефон
1	Иванов	Иван	Иванович	Ул. Володарского, 73-6	255-67-03
2	Петров	Петр	Евгеньевич	Ул. Калинина, 13-25	345-76-89
3	Сидоров	Константин	Викторович	Ул. Ф. Горелова, 43-879	326-98-05

- А). Атрибутом *неверно*  
 В). Отношением *верно*  
 С). Записью *неверно*  
 D). Значением атрибутов *неверно*  
 E). Схемой отношения *неверно*  
 F). Кортежем *неверно*

№	Фамилия	Имя	Отчество	Адрес	Телефон
1	Иванов	Иван	Иванович	Ул. Володарского, 73-6	255-67-03
2	Петров	Петр	Евгеньевич	Ул. Калинина, 13-25	345-76-89
3	Сидоров	Константин	Викторович	Ул. Ф. Горелова, 43-879	326-98-05

## 3) Выделенные фрагменты таблицы являются

- А). Атрибутами *верно*  
 В). Отношениями *неверно*  
 С). Записями *неверно*  
 D). Значениями атрибутов *неверно*  
 E). Схемой отношения *неверно*  
 F). Кортежем *неверно*

№	Фамилия	Имя	Отчество	Адрес	Телефон
1	Иванов	Иван	Иванович	Ул. Володарского, 73-6	255-67-03
2	Петров	Петр	Евгеньевич	Ул. Калинина, 13-25	345-76-89
3	Сидоров	Константин	Викторович	Ул. Ф. Горелова, 43-879	326-98-05

## 4) Выделенный фрагмент таблицы является

- А). Атрибутом *неверно*  
 В). Отношением *неверно*  
 С). Записью *неверно*  
 D). Значениями атрибута *верно*  
 E). Схемой отношения *неверно*  
 F). Кортежем *неверно*

## 5) Число атрибутов в отношении называется:

- A). Степенью *верно*  
 B). Мощностью *неверно*  
 C). Схемой отношения *неверно*  
 D). Кorteжем *неверно*

**6) Текущее число corteжей называется:**

- A). Степенью *неверно*  
 B). Мощностью *верно*  
 C). Схемой отношения *неверно*  
 D). Кorteжем *неверно*

**7) Первичный ключ**

- A). Атрибут (набор атрибутов) однозначно идентифицирующий каждый из его corteжей *верно*  
 B). Совокупность уникальных corteжей *неверно*  
 C). Любое уникальное поле таблицы *неверно*  
 D). Поле, которое используется для организации поиска данных *неверно*

**8) Первичный ключ должен обладать следующими свойствами (выберите НЕВЕРНЫЙ ответ)**

- A). Компактный *неверно*  
 B). Существует для любого отношения *неверно*  
 C). Содержит неопределенные значения (NULL) только в исключительных случаях *верно*  
 D). Имеет уникальное значение *неверно*

Табельный №	Фамилия	Имя	Отчество	Группа
111233211	Иванов	Иван	Иванович	Ул. Володарского, 73-6
111233651	Петров	Петр	Евгеньевич	Ул. Калинина, 13-25
111233215	Сидоров	Константин	Викторович	Ул. Ф.Горелова 43-879

**9) Первичным ключом данного отношения будет**

- A). Табельный\_№ *верно*  
 B). Фамилия *неверно*  
 C). Имя *неверно*  
 D). Отчества *неверно*  
 E). Группа *неверно*  
 F). Фамилия Имя Отчество *неверно*  
 G). Фотография *неверно*

**10) Таблицу можно считать отношением, если (выберите ЛИШНЕЕ):**

- A). не существует столбцов с одинаковыми именами *неверно*  
 B). все столбцы имеют одинаковую структуру *неверно*  
 C). все строки таблицы уникальны *неверно*  
 D). таблицы должны быть отсортированы по ключевому полю *верно*  
 E). нет неверных высказываний *неверно*

**11) Простая структура, организация данных в виде двумерных таблиц, использование формальной алгебры отношений – все это характеризует**

- A). иерархическую модель данных *неверно*  
 B). сетевую модель данных *неверно*  
 C). реляционную модель данных *верно*

D). структурную модель данных *неверно*

**12) Какая база данных состоит из взаимосвязанного набора таблиц**

- A). Сетевая *неверно*  
 B). Иерархическая *неверно*  
 C). Реляционная *верно*  
 D). Объектная *неверно*  
 E). Нет правильных ответов *неверно*

**13) База данных – это**

- A). Именованная структурированная совокупность данных об одной предметной области *верно*  
 B). Совокупность данных, хранящихся на компьютере *неверно*  
 C). Комплекс программных и языковых средств, необходимых для хранения данных *неверно*  
 D). Совокупность данных, организованных по определенным правилам *неверно*

**14) Описание структуры базы данных называется:**

- A). Структурой *неверно*  
 B). Описанием *неверно*  
 C). Методанными *верно*  
 D). Суперданными *неверно*  
 E). Нет правильных ответов *неверно*

**15) Множество допустимых значений определенного поля**

- A). Таблица *неверно*  
 B). Домен *верно*  
 C). Кортёж *неверно*  
 D). Триггер *неверно*  
 E). Ограничение *неверно*

**16) Последовательность действий над базой данной, в которой либо все действия выполняются успешно, либо не выполняется ни одно из них, называется**

- A). Последовательность *неверно*  
 B). Транзакция *верно*  
 C). Операция *неверно*  
 D). Протокол *неверно*  
 E). Нет верных определений *неверно*

**17) Несуществующая архитектура многопользовательского приложения баз данных**

- A). Архитектура «клиент-сервер» *неверно*  
 B). Архитектура «заказчик-исполнитель» *верно*  
 C). Архитектура «файл-сервер» *неверно*

**18) Один или несколько атрибутов, которые уникальным образом идентифицируют строки таблицы**

- A). Ключ *верно*  
 B). Замок *неверно*  
 C). Метка *неверно*  
 D). Индекс *неверно*  
 E). Нет правильных ответов *неверно*

**19) Кортёжем реляционной базы данных является:**

- A). корень дерева; *неверно*
- B). столбец таблицы; *неверно*
- C). строка таблицы; *верно*
- D). ветви дерева; *неверно*
- E). дерево *неверно*

**20) Первичный ключ в реляционной базе данных служит для:**

- A). организации новой структуры данных; *неверно*
- B). указания типа поля *неверно*
- C). связи между различными структурами данных *неверно*
- D). связи между различными таблицами в реляционной базе данных; *неверно*
- E). однозначного выделения записи в базе данных *верно*

**21) База данных содержит информацию об учениках компьютерной школы: имя, номер группы, балл за тест, балл за задание, общее количество баллов. Какого типа должно быть поле ОБЩЕЕ КОЛИЧЕСТВО БАЛЛОВ?**

- A). символьного; *неверно*
- B). логического; *неверно*
- C). числового; *верно*
- D). любого типа; *неверно*
- E). числового или логического. *неверно*

**22) Система управления базами данных — это:**

- A). программная система, поддерживающая наполнение и манипулирование данными в файлах баз данных; *верно*
- B). набор программ, обеспечивающий работу всех аппаратных устройств компьютера и доступ пользователя к ним; *неверно*
- C). прикладная программа для обработки текстов и различных документов; *неверно*
- D). оболочка операционной системы, позволяющая более комфортно работать с файлами. *Неверно*

**23) Что такое транзакция в теории БД (выберите один наиболее подходящий и полный ответ)?**

- A). Любое действие над базой данных *неверно*
- B). Любое действие на физическом уровне над хранимыми данными *неверно*
- C). Некоторое действие над базой данных, которое переводит БД из одного непротиворечивого состояния в другое *верно*
- D). Любое атомарное действие над хранимыми данными в БД *неверно*

**24) Модель представления данных в виде дерева с одним корневым объектом называется:**

- A). Реляционная *неверно*
- B). Иерархическая *верно*
- C). Сетевая *неверно*
- D). Постреляционная *неверно*

**25) Согласно сетевой модели данных**

- A). У одного потомка может быть несколько предков *верно*
- B). У одного потомка может быть только один предок *неверно*

**26) Согласно сетевой модели данных**

- A). У одного потомка может быть несколько предков *неверно*
- B). У одного потомка может быть только один предок *верно*



- 27) Многозначные поля и объединение полей являются отличительной особенностью**
- |                              |                |
|------------------------------|----------------|
| A). Реляционная              | <i>неверно</i> |
| B). Иерархическая            | <i>неверно</i> |
| C). Сетевая                  | <i>неверно</i> |
| D). Объектно-ориентированная | <i>неверно</i> |
| E). Постреляционная          | <i>верно</i>   |
- 28) Модель представления данных в виде дерева, узлами которого являются объекты**
- |                              |                |
|------------------------------|----------------|
| A). Реляционная              | <i>неверно</i> |
| B). Иерархическая            | <i>неверно</i> |
| C). Сетевая                  | <i>неверно</i> |
| D). Объектно-ориентированная | <i>верно</i>   |
| E). Постреляционная          | <i>неверно</i> |
- 29) В веб-приложениях чаще всего используется архитектура**
- |                               |                |
|-------------------------------|----------------|
| A). Файл-сервер               | <i>неверно</i> |
| B). Клиент-сервер             | <i>неверно</i> |
| C). Трехуровневая архитектура | <i>верно</i>   |
- 30) Архитектура, в соответствии с которой на сервере хранятся только файлы базы данных называется:**
- |                   |                |
|-------------------|----------------|
| A). Файл-сервер   | <i>верно</i>   |
| B). Клиент-сервер | <i>неверно</i> |
- 31) Архитектура, в соответствии с которой на сервере хранятся файлы базы данных и СУБД называется:**
- |                   |                |
|-------------------|----------------|
| A). Файл-сервер   | <i>неверно</i> |
| B). Клиент-сервер | <i>верно</i>   |
- 32) Внешний уровень архитектуры ANSI-SPARS отражает...**
- |  |                |
|--|----------------|
| A). представления данных, необходимых для работы конкретных приложений             | <i>верно</i>   |
| B). представления данных, необходимых для единого администрирования баз данных     | <i>неверно</i> |
| C). представления данных, необходимых для представления данных в памяти компьютера | <i>неверно</i> |
- 33) Концептуальный уровень архитектуры ANSI-SPARS отражает...**
- |  |                |
|--|----------------|
| A). представления данных, необходимых для работы конкретных приложений             | <i>неверно</i> |
| B). представления данных, необходимых для единого администрирования баз данных     | <i>верно</i>   |
| C). представления данных, необходимых для представления данных в памяти компьютера | <i>неверно</i> |
- 32) Внутренний уровень архитектуры ANSI-SPARS отражает...**
- |  |                |
|--|----------------|
| A). представления данных, необходимых для работы конкретных приложений             | <i>неверно</i> |
| B). представления данных, необходимых для единого администрирования баз данных     | <i>неверно</i> |
| C). представления данных, необходимых для представления данных в памяти компьютера | <i>верно</i>   |

**33) Связь между таблицами Поставщик и Товары будет организована по полям:**

Таб\_1. Поставщик

Код поставщика	Наименование	Юр. адрес	Директор	Телефон

Таб\_2. Товары

Код товара	Наименование	Цена

- А). Код\_поставщика – Код\_товара *неверно*  
 В). Код\_поставщика – Наименование товара *неверно*  
 С). Код\_товара – Наименование поставщика *неверно*  
 D). Через таблицу Поставки *верно*

**34) Тип связи между таблицами Поставщик и Поставки:**

Таб\_1. Поставщик

Код поставщика	Наименование	Юр. адрес	Директор	Телефон

Таб\_2. Поставки

Код товара	Код поставщика	Дата	Количество

- А). 1 : 1 *неверно*  
 В). 1 : Многие *верно*  
 С). Многие: 1 *неверно*  
 D). Многие: Многие *неверно*

**35) Тип связи между таблицами Кафедра и Информация по полю Зав. кафедр:**

Код	Название	Заведующий	Количество сотрудников	Из них остепененных

Таб\_1. Кафедра

Фамилия	Имя	Отчество	Код кафедры	Степень	Звание	Адрес

Таб\_2. Информация по зав. кафедр

- A). 1: 1 *верно*  
 B). 1: Многие *неверно*  
 C). Многие: 1 *неверно*  
 D). Многие: Многие *неверно*

**36) Что такое ER-диаграмма?**

- A). Диаграмма сущностей и связей *верно*  
 B). Диаграмма ошибок *неверно*  
 C). Диаграмма производительности БД *неверно*  
 D). Диаграмма потоков данных *неверно*

**37) Анализ предметной области и построение ее концептуального представления является задачей**

- A). инфологического проектирования *верно*  
 B). даталогического проектирования *неверно*  
 C). физического проектирования *неверно*

**38) Представление модели в терминах выбранной СУБД является задачей**

- A). инфологического проектирования *неверно*  
 B). даталогического проектирования *верно*  
 C). физического проектирования *неверно*

**39) Определение структуры хранения данных является задачей**

- A). инфологического проектирования *неверно*  
 B). даталогического проектирования *неверно*  
 C). физического проектирования *верно*

**40) Любой различимый объект, информацию о котором необходимо хранить в базе данных, - это...**

- A). связь *неверно*  
 B). сущность *верно*  
 C). домен *неверно*  
 D). кортеж *неверно*  
 E). атрибут *неверно*  
 F). ключ *неверно*

**41) Поименованная характеристика сущности- это...**

- A). связь *неверно*  
 B). сущность *неверно*  
 C). домен *неверно*  
 D). кортеж *неверно*  
 E). атрибут *верно*  
 F). ключ *неверно*

42) Минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности, - это...

- A). связь *неверно*
- B). сущность *неверно*
- C). домен *неверно*
- D). кортеж *неверно*
- E). атрибут *неверно*
- F). ключ *верно*

43) Ассоциирование двух или более сущностей - это...

- A). связь *верно*
- B). сущность *неверно*
- C). домен *неверно*
- D). кортеж *неверно*
- E). атрибут *неверно*
- F). ключ *неверно*

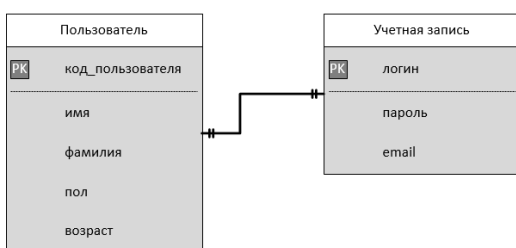
44) Связь между двумя сущностями называется..

- A). унарной *неверно*
- B). бинарной *верно*
- C). простой *неверно*
- D). обязательной *неверно*
- E). необязательной *неверно*

45) Связь сущности самой с собой называется..

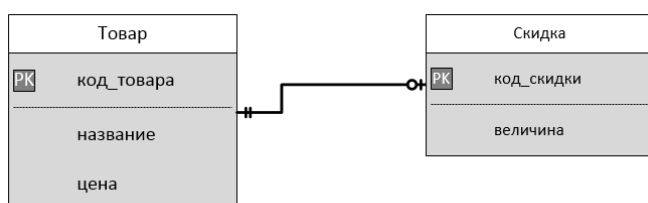
- A). унарной *верно*
- B). бинарной *неверно*
- C). простой *неверно*
- D). обязательной *неверно*
- E). необязательной *неверно*

46) На рисунке изображена связь



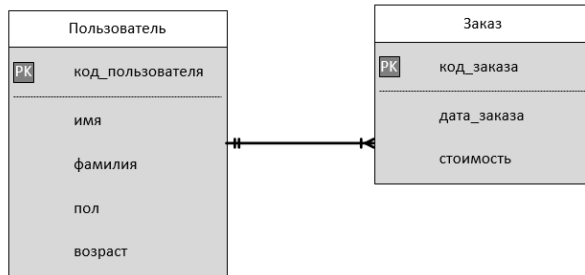
- A). один-к-одному *верно*
- B). один-ко-многим *неверно*
- C). многие-ко-многим *неверно*

47) На рисунке изображена связь



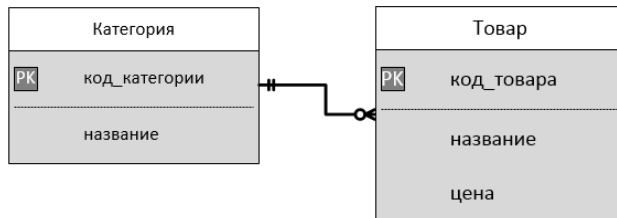
- А). один-к-одному *верно*  
 В). один-ко-многим *неверно*  
 С). многие-ко-многим *неверно*

**48) На рисунке изображена связь**



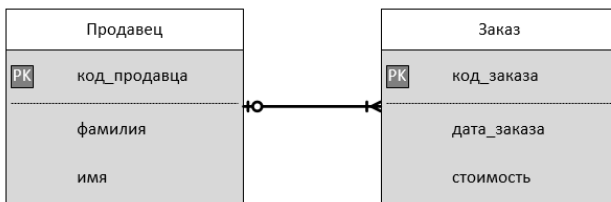
- А). один-к-одному *неверно*  
 В). один-ко-многим *верно*  
 С). многие-ко-многим *неверно*

**49) На рисунке изображена связь**



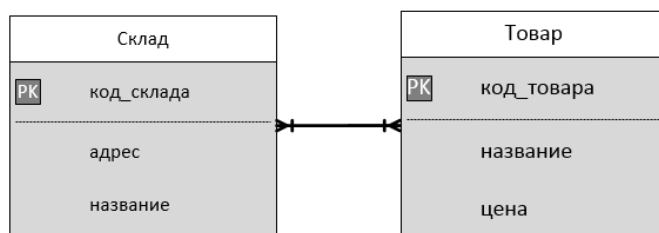
- А). один-к-одному *неверно*  
 В). один-ко-многим *верно*  
 С). многие-ко-многим *неверно*

**50) На рисунке изображена связь**



- А). один-к-одному *неверно*  
 В). один-ко-многим *верно*  
 С). многие-ко-многим *неверно*

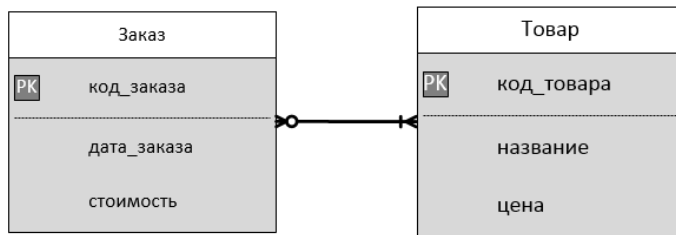
**51) На рисунке изображена связь**



- А). один-к-одному *неверно*

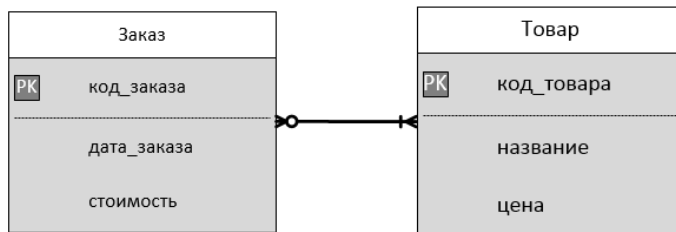
- В). один-ко-многим *неверно*  
 С). многие-ко-многим *верно*

52) На рисунке изображена связь



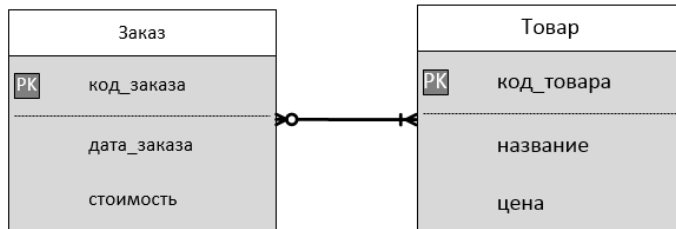
- А). один-к-одному *неверно*  
 В). один-ко-многим *неверно*  
 С). многие-ко-многим *верно*

53) На рисунке у сущности Заказ класс принадлежности является



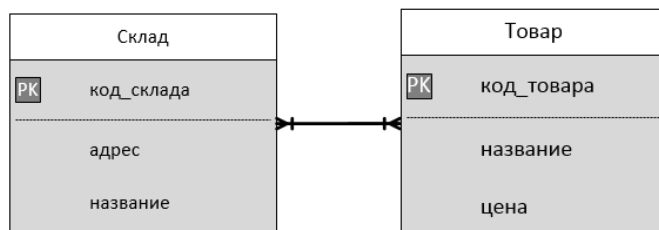
- А). обязательной *неверно*  
 В). необязательный *верно*

54) На рисунке у сущности Товар класс принадлежности является



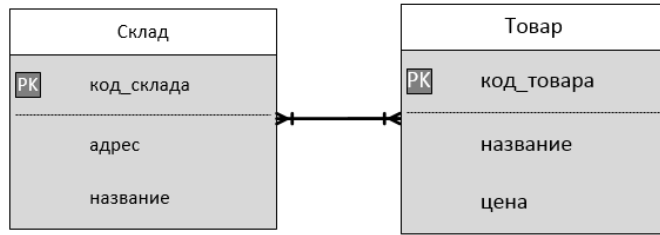
- А). обязательной *верно*  
 В). необязательный *неверно*

55) На рисунке у сущности Товар класс принадлежности является



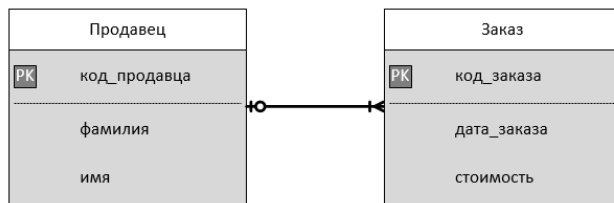
- А). обязательной *верно*  
 В). необязательный *неверно*

56) На рисунке у сущности Склад класс принадлежности является



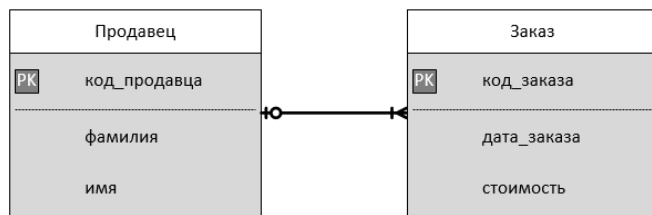
- А). обязательной *верно*  
 В). необязательный *неверно*

57) На рисунке у сущности Заказ класс принадлежности является



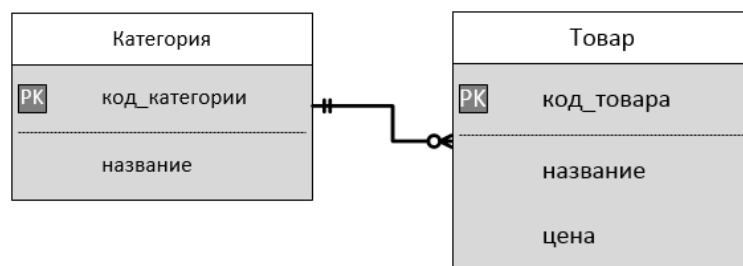
- А). обязательной *верно*  
 В). необязательный *неверно*

58) На рисунке у сущности Продавец класс принадлежности является



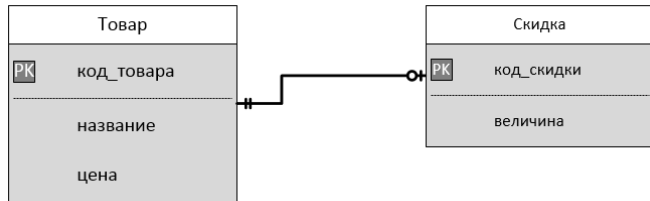
- А). обязательной *неверно*  
 В). необязательный *верно*

59) На рисунке у сущности Товар класс принадлежности является



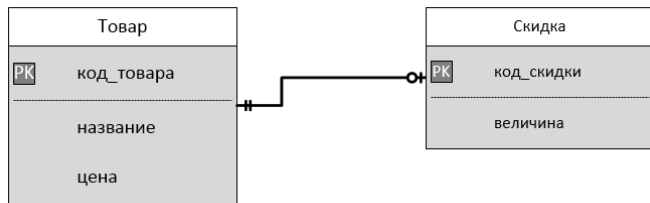
- А). обязательной *неверно*  
 В). необязательный *верно*

60) На рисунке у сущности Товар класс принадлежности является



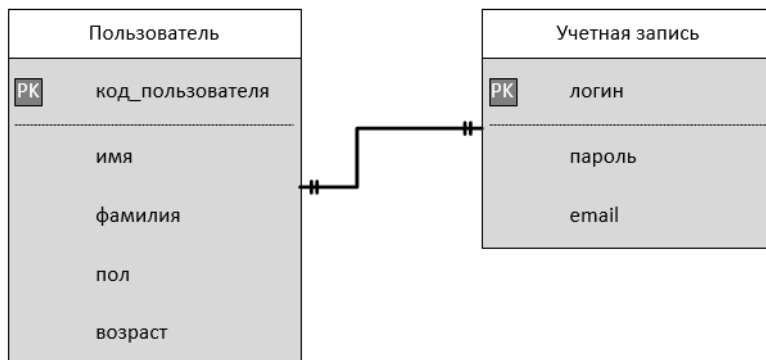
- А). обязательной *верно*  
 В). необязательный *неверно*

61) На рисунке у сущности Скидка класс принадлежности является



- А). обязательной *неверно*  
 В). необязательный *верно*

62) На рисунке у сущности Учетная запись класс принадлежности является



- А). обязательной *верно*  
 В). необязательный *неверно*

63) Ограничения, накладываемые предметной областью, являются...

- А). внешними *верно*  
 В). внутренними *неверно*

64) Ограничения, накладываемые организацией данных в СУБД, являются...

- А). внешними *неверно*  
 В). внутренними *верно*

65) Определение первичных ключей относится к...

- А). соответствию по типам *неверно*  
 В). ссылочной целостности *неверно*  
 С). целостности отношений *верно*



**66) Определение внешних ключей относится к...**

- A). соответствию по типам *неверно*  
 B). ссылочной целостности *верно*  
 C). целостности отношений *неверно*

**67) НФБК представляет собой расширенную**

- A). первую нормальную форму; *неверно*  
 B). вторую нормальную форму; *неверно*  
 C). третью нормальную форму; *верно*  
 D). четвертую нормальную форму; *неверно*  
 E). пятую нормальную форму. *неверно*

**68) Сколько необходимо отношений для связи 1:1**

- A). Одно *неверно*  
 B). Два *неверно*  
 C). два либо три *неверно*  
 D). Три *неверно*  
 E). Одно, два или три *верно*

**69) Сколько необходимо отношений для связи 1:M**

- A). одно *неверно*  
 B). Два *неверно*  
 C). два либо три *верно*  
 D). Три *неверно*

**70) Сколько необходимо отношений для связи M:N**

- A). Одно *неверно*  
 B). Два *неверно*  
 C). два либо три *неверно*  
 D). Три *верно*

**71) Что из перечисленного является СУБД**

- A). FireBird *верно*  
 B). MS SQL Server *верно*  
 C). MS Visuo *неверно*  
 D). MS Excel *неверно*  
 E). ORACLE *верно*

**72) На ER-диаграмме сущность изображается в виде**

- A). Прямоугольника *верно*  
 B). Линии *неверно*  
 C). Ромба *неверно*  
 D). Овала *неверно*  
 E). Не изображается *неверно*

**73) На ER-диаграмме кортеж изображается**

- A). Прямоугольника *неверно*  
 B). Линии *неверно*  
 C). Ромба *неверно*  
 D). Овала *неверно*  
 E). Не изображается *верно*

**74) Реальный или представляемый объект, информация о котором должна сохраниться в базе данных**

- A). Атрибут *неверно*
- B). Сущность *верно*
- C). Связь *неверно*
- D). Домен *неверно*
- E). Кортеж *неверно*

**75) Разбиение одной таблицы на две или более других, обладающих лучшими свойствами при вставке, изменении и удалении данных**

- A). Моделирование *неверно*
- B). Декомпозиция *верно*
- C). Целостность *неверно*
- D). Администрирование *неверно*
- E). Стандартизация *неверно*

**76) Существующие стратегии разработки баз данных:**

- A). Сверху вниз *верно*
- B). Слева направо *неверно*
- C). Справа налево *неверно*
- D). Снизу вверх *верно*

**77) Самый распространенный тип связи**

- A). Один к одному *неверно*
- B). Все к одному *неверно*
- C). Один ко многим *верно*
- D). Многие ко многим *неверно*
- E). Рекурсивная *неверно*

**78) Нормальная форма, в которой находится отношение, если оно находится в 1НФ и все его неключевые атрибуты зависят от всего ключа:**

- A). 1НФ *неверно*
- B). 2 НФ *верно*
- C). 3 НФ *неверно*
- D). НФ Бойса-Кода *неверно*
- E). Нет правильных ответов *неверно*

**79) В какой нормальной форме находится таблица, удовлетворяющая определению отношения**

- A). 1НФ *верно*
- B). 2 НФ *неверно*
- C). 3 НФ *неверно*
- D). НФ Бойса-Кода *неверно*
- E). Нет правильных ответов *неверно*

**80) Нормальная форма, в которой находится отношение, если оно находится во 2 НФ и не содержит транзитивных зависимостей**

- A). 1НФ *неверно*
- B). 2 НФ *неверно*
- C). 3 НФ *верно*
- D). НФ Бойса-Кода *неверно*

Е). Нет правильных ответов *неверно*

**81) БД содержит информацию о собаках из клуба собаководства: кличка, порода, дата рождения, пол, количество медалей за участие в выставках. Какие типы должны иметь поля?**

- А). текстовое, текстовое, числовое, текстовое, числовое; *неверно*  
 В). текстовое, текстовое, дата, текстовое, числовое; *верно*  
 С). текстовое, текстовое, дата, числовое, числовое; *неверно*  
 D). текстовое, текстовое, числовое, логическое, числовое; *неверно*

**82) Неполная функциональная зависимость – это зависимость**

- А). от части внешних ключей *неверно*  
 В). от части альтернативного ключа *неверно*  
 С). от части первичного ключа *верно*  
 D). от части внешнего ключа *неверно*

**83) Нормальные формы вложены в порядке**

- А). 1НФ, 2НФ, НФБК, 3НФ *неверно*  
 В). НФБК, 1НФ, 2НФ, 3НФ *неверно*  
 С). 1НФ, НФБК, 2НФ, 3НФ *неверно*  
 D). 1НФ, 2НФ, 3НФ, НФБК *верно*

**84) Процесс превращения объектов инфологической модели (сущности и связи) в логические таблицы базы данных называется**

- А). определением целостности данных *неверно*  
 В). нормализацией отношений *верно*  
 С). типизацией отношений *неверно*

85) В какой нормальной форме находится это отношение

<u>Код товара</u>	Название	Цена	<u>Склад</u>	Адрес	Количество
1	Телевизор	20000	1	Ленина,13	34
1	Телевизор	20000	2	Гагарина,85	33
1	Телевизор	20000	3	Советская, 44	23
2	Пылесос	7000	1	Ленина,13	1
2	Пылесос	7000	3	Советская, 44	12
3	Ноутбук	30000	2	Гагарина,85	24
3	Ноутбук	30000	3	Советская, 44	22

- А). 1НФ *верно*  
 В). 2НФ *неверно*  
 С). 3НФ *неверно*  
 D). НФБК *неверно*

86) В какой нормальной форме находится это отношение

Код товара	Название	Цена	Категория	Наценка
1	Телевизор	20000	Телевизоры	40
2	3D телевизор	30000	Телевизоры	40
3	Миксер	4000	Техника для дома	20
4	Блендер	3000	Техника для дома	20
5	Фотоаппарат	18000	Фото и видео	50
6	Видеокамера	25000	Фото и видео	50
7	Фотопринтер	10000	Фото и видео	50

- A). 1НФ            *неверно*  
 B). 2НФ            *верно*  
 C). 3НФ            *неверно*  
 D). НФБК         *неверно*

87) Если степень бинарной связи 1:1 и класс принадлежности обеих сущностей является обязательным, то требуется построение

- A). 1 таблицу         *верно*  
 B). 2 таблицы         *неверно*  
 C). 3 таблицы         *неверно*  
 D). 4 таблицы         *неверно*

88) Если степень бинарной связи 1:1 и класс принадлежности одной сущности является обязательным, а другой сущности - не обязательным, то требуется построение

- A). 1 таблицу         *неверно*  
 B). 2 таблицы         *верно*  
 C). 3 таблицы         *неверно*  
 D). 4 таблицы         *неверно*

89) Если степень бинарной связи 1:1 и класс принадлежности обеих сущностей не является обязательным, то требуется построение

- A). 1 таблицу         *неверно*  
 B). 2 таблицы         *неверно*  
 C). 3 таблицы         *верно*  
 D). 4 таблицы         *неверно*

90) Если степень *бинарной связи* 1:N, и класс принадлежности *n-связной сущности* является *обязательным*, то достаточно построить

- A). 1 таблицу *неверно*
- B). 2 таблицы *верно*
- C). 3 таблицы *неверно*
- D). 4 таблицы *неверно*

91) Если степень *бинарной связи* 1:N и класс принадлежности *n-связной сущности* не является *обязательным*, то необходимо построить

- A). 1 таблицу *неверно*
- B). 2 таблицы *неверно*
- C). 3 таблицы *верно*
- D). 4 таблицы *неверно*

92) Если степень *бинарной связи* M:N, то необходимо построить

- A). 1 таблицу *неверно*
- B). 2 таблицы *неверно*
- C). 3 таблицы *верно*
- D). 4 таблицы *неверно*

93) Если связь является *трехсторонней*, необходимо построить

- A). 1 таблицу *неверно*
- B). 2 таблицы *неверно*
- C). 3 таблицы *неверно*
- D). 4 таблицы *верно*

#### Тестовые задания по разделу «Язык SQL. Управление данными»

1) Что означает SQL? (Г)

- A) Supreme Query Language
- Б) Structured Questionnaire Level
- B) Server Question Level
- Г) Structured Query Language
- Д) Super Quick Language

2) Предположим, что создана таблица `persons` с колонками `id`, `name`, `age`. Какой запрос найдет средний возраст всех людей с возрастом не менее 18-ти лет? (А)

- A) `select avg(age) from persons where age >= 18`
- Б) `select avg(age) from persons having age >= 18`
- B) `select avg(age) from persons where age >= 18 group by name`
- Г) `select avg(age) from persons having age >= 18 group by name`

3) Назовите оператор языка SQL для создания запросов на выбор данных (А)

- A). Select
- B). Distinct
- C). Where
- D). Having
- E). Create

**4) Назовите оператор команды Select, который обеспечивает возможность устранения избыточных значений. (B)**

- A). Order by
- B). Distinct
- C). Where
- D). Having
- E). Create

**5) Назовите предложение команды Select, которая позволяет производить выборку данных, в зависимости от истинности поставленного условия. (C)**

- A). Order by
- B). Distinct
- C). Where
- D). Having
- E). Create

**5) Назовите команду, которая позволяет группировать данные. (E)**

- A). Order by
- B). Distinct
- C). Where
- D). Having
- E). Group by

**6) Назовите предложение команды Select, которое позволяет устанавливать условия для агрегатных функций (D)**

- A). Order by
- B). Distinct
- C). Where
- D). Having
- E). Group by

**7) Назовите предложение команды Select, которое используется для сортировки результата запроса. (A)**

- A). Order by
- B). Distinct
- C). Where
- D). Having
- E). Group by

**8) Операторы AND, OR, NOT относятся к (B)**

- A). Реляционным операторам
- B). Логическим операторам
- C). Специальным операторам
- D). Агрегатным функциям
- E). Нет правильного варианта

**9) Среди предложенных названий выберите то, которое является названием агрегатной функции (F)**

- A). COUNT
- B). SUM
- C). AVG

- D). MAX
- E). MIN
- F). Все варианты верные

**10) Выберите правильную структуру запроса SELECT (A)**

- A). SELECT ... FROM...WHERE...GROUP BY... HAVING...ORDER BY...
- B). SELECT ... WHERE...FROM... GROUP BY... HAVING...ORDER BY...
- C). SELECT ... FROM...WHERE... ORDER BY GROUP BY... HAVING...
- D). SELECT ... WHERE...GROUP BY... HAVING...ORDER BY... FROM...

**11) Функция id\_category in(1,3,5) аналогична следующему предикату (B)**

- A). id\_category=1 and id\_category=3 and id\_category=5
- B). id\_category=1 or id\_category=3 or id\_category=5
- C). id\_category>1 and id\_category<5
- D). id\_category =1 or 3 or 5

**12) Какой из запросов вернет отсортированный по убыванию результат (D)**

- A). SELECT name FROM product ORDER BY name
- B). SELECT name FROM product GROUP BY name
- C). SELECT name FROM product ORDER BY name ASC
- D). SELECT name FROM product ORDER BY name DESC
- E). SELECT name FROM product GROUP BY name DESC

**13) Какие запросы вернет отсортированный по возрастанию результат (A,C)**

- A). SELECT name FROM product ORDER BY name
- B). SELECT name FROM product GROUP BY name
- C). SELECT name FROM product ORDER BY name ASC
- D). SELECT name FROM product ORDER BY name DESC
- E). SELECT name FROM product GROUP BY name DESC

**14) Сколько строк вернет запрос**

**SELECT AVG(price) FROM product (B)**

- A). 0
- B). 1
- C). 10
- D). Зависит от содержимого таблицы product

**15) Сколько строк вернет запрос**

**SELECT AVG(price), SUM(price) FROM product (B)**

- A). 0
- B). 1
- C). 2
- D). Зависит от содержимого таблицы product

**16) Сколько строк вернет запрос**

**SELECT price FROM product (D)**

- A). 0
- B). 1
- C). 10
- D). Зависит от содержимого таблицы product

**17) Количество товаров в таблице product вернет запрос (B)**

- A). SELECT SUM(name) FROM product
- B). SELECT COUNT(\*) FROM product
- C). SELECT MAX(name) FROM product
- D). SELECT SUM(\*) FROM product
- E). SELECT MAX(\*) FROM product

**18) Дана таблица product(id, name, price) Какие запросы вернет все данные этой таблицы (A,B)**

- A). SELECT \* FROM product
- B). SELECT id, name, price FROM product
- C). SELECT \*
- D). SELECT id, name FROM product

**19) Какой запрос вернет все товары, у которых не задана цена (C)**

- A). SELECT \* FROM product WHERE price<>0
- B). SELECT \* FROM product WHERE price=0
- C). SELECT \* FROM product WHERE price IS NULL
- D). SELECT \* FROM product WHERE price IS NOT

**20) В таблице product 5 строк, в таблице order 10 строк. Сколько строк вернет следующий запрос**

**SELECT \* FROM product,order (C)**

- A). 15
- B). 10
- C). 30
- D). 2

**21) Какие из запросов содержат внутреннее соединение (C,D)**

- A). SELECT name, address, discount FROM user LEFT JOIN user\_buyer ON user.id=user\_buyer.id
- B). SELECT name, address, discount FROM user RIGHT JOIN user\_buyer ON user.id=user\_buyer.id
- C). SELECT name, address, discount FROM user JOIN user\_buyer ON user.id=user\_buyer.id
- D). SELECT name, address, discount FROM user INNER JOIN user\_buyer ON user.id=user\_buyer.id

**22) Какие из запросов содержат внешнее соединение (A,B)**

- A). SELECT name, address, discount FROM user LEFT JOIN user\_buyer ON user.id=user\_buyer.id
- B). SELECT name, address, discount FROM user RIGHT JOIN user\_buyer ON user.id=user\_buyer.id
- C). SELECT name, address, discount FROM user JOIN user\_buyer ON user.id=user\_buyer.id
- D). SELECT name, address, discount FROM user INNER JOIN user\_buyer ON user.id=user\_buyer.id

**23) Операция объединения нескольких запросов (A)**

- A). UNION
- B). JOIN
- C). INTERSECT



D). EXCEPT

**24) Какой предикат возвращает true, если подзапрос содержит одну или более строк, иначе возвращает false (A)**

- A). EXISTS
- B). ALL
- C). ANY
- D). SOME
- E). IN

**25) Выражение price > ANY(SELECT cast FROM price\_list) вернет true, если (B)**

- A). Все поля cast из таблицы price\_list меньше цены price
- B). Хотя бы одно поле cast из таблицы price\_list меньше цены price
- C). Все поля cast из таблицы price\_list больше цены price
- D). Сумма всех полей cast из таблицы price\_list меньше цены price

**26) Выражение price > SOME (SELECT cast FROM price\_list) вернет true, если (B)**

- A). Все поля cast из таблицы price\_list меньше цены price
- B). Хотя бы одно поле cast из таблицы price\_list меньше цены price
- C). Все поля cast из таблицы price\_list больше цены price
- D). Сумма всех полей cast из таблицы price\_list меньше цены price

**27) Выражение price > ALL (SELECT cast FROM price\_list) вернет true, если (A)**

- A). Все поля cast из таблицы price\_list меньше цены price
- B). Хотя бы одно поле cast из таблицы price\_list меньше цены price
- C). Все поля cast из таблицы price\_list больше цены price
- D). Сумма всех полей cast из таблицы price\_list меньше цены price

**28) Какова структура соединения таблиц (A)**

- A). SELECT...FROM...JOIN... ON...
- B). SELECT... JOIN... ON...FROM...
- C). SELECT... JOIN...FROM... ON...
- D). SELECT...FROM... ON...

**29) По умолчанию при объединении двух запросов дублирование строк (A)**

- A). Исключается
- B). Не исключается

**30) В запросе**

**SELECT ...**

**FROM ...**

**LEFT JOIN ....**

**ON ...**

**WHERE....**

**Выражение, заданное после ON применяется (A)**

- A). К правой части объединения
- B). К левой части объединения
- C). Ко всем данным запроса

31) В запросе

**SELECT ...**

**FROM ...**

**LEFT JOIN ....**

**ON ...**

**WHERE....**

**Выражение, заданное после WHERE применяется (C)**

- A). К правой части объединения
- B). К левой части объединения
- C). Ко всем данным запроса

32) **Функция CAST (B)**

- A). Округляет число с заданной точностью
- B). Осуществляет преобразование типов данных
- C). Осуществляет конкатинацию строк
- D). Переводит строку в нижний регистр

33) **Функция CONCAT (C)**

- A). Округляет число с заданной точностью
- B). Осуществляет преобразование типов данных
- C). Осуществляет конкатинацию строк
- D). Переводит строку в нижний регистр

34) **Синтаксис функции CAST (B)**

- A). CAST(выражение)
- B). CAST(выражение AS тип данных)
- C). CAST(выражение) AS тип данных
- D). CAST(тип данных AS выражение)

35) **В функции IF(предикат1, предикат2, предикат3) предикат1 (A)**

- A). Задает условие
- B). Задает возвращаемое значение, в случаи истина
- C). Задает возвращаемое значение, в случаи ложь
- D). Задает подзапрос

36) **В функции IF(предикат1, предикат2, предикат3) предикат2 (B)**

- A). Задает условие
- B). Задает возвращаемое значение, в случаи истина
- C). Задает возвращаемое значение, в случаи ложь
- D). Задает подзапрос

37) **В функции IF(предикат1, предикат2, предикат3) предикат3 (C)**

- A). Задает условие
- B). Задает возвращаемое значение, в случаи истина
- C). Задает возвращаемое значение, в случаи ложь
- D). Задает подзапрос

**38) Может ли запрос SELECT нарушить целостность базы данных (A)**

- A). Нет
- B). Да

**Все перечисленные ключевые слова относятся к DML (Data Manipulation Language): INSERT, CREATE, DELETE, UPDATE верно- b**

- a. да
- b. нет

**Все перечисленные ключевые слова относятся к DDL (Data Definition Language): CREATE, ALTER, DROP верно- a**

- c. да
- d. нет

**40) Все перечисленные ключевые слова относятся к DML (Data Manipulation Language): INSERT, DELETE, UPDATE верно- a**

- a. Да
- b. Нет

**Все перечисленные ключевые слова относятся к DDL (Data Definition Language): INSERT, ALTER, DROP верно- b**

- c. да
- d. нет

**41) Выберите ключевые слова, которые относятся к DML (Data Manipulation Language) верно- a,b,f**

- a. INSERT
- b. DELETE
- c. ALTER
- d. DROP
- e. CREATE
- f. UPDATE

**42) Выберите ключевые слова, которые относятся к DDL (Data Definition Language) верно- c,d,e**

- a. INSERT
- b. DELETE
- c. ALTER
- d. DROP
- e. CREATE
- f. UPDATE

**43) Укажите корректные команды INSERT из перечисленных (таблица students состоит из трех столбцов: id, first\_name, last\_name) верно- e**

- a. INSERT INTO students(id, first\_name, last\_name) SELECT id,first\_name FROM new\_students WHERE last\_name IS NULL
- b. INSERT INTO students SELECT VALUES(95,'Иван','Иванов')
- c. INSERT INTO students(id, first\_name, last\_name)
- d. INSERT INTO students(id, first\_name, last\_name) (95,'Иван','Иванов')
- e. INSERT INTO students(id, first\_name, last\_name) VALUES(95,'Иван','Иванов')

44) Даны таблицы **верно- b,e**

```

DOGS
Name  Age
-----
Snoopy 4
Benny  2

CATS
Name  Age
-----
Kleo  3
Linda 6

ANIMALS
Name  Age
-----

```

Name имеет тип char(10), а Age – int(10)

Выберите корректные insert запросы **верно- a,c,d**

- a. INSERT INTO animals(age, name) VALUES('Pam',1);
- b. INSERT INTO animals(11,'Kay');
- c. INSERT INTO animals(Age,Name) SELECT Age, Name FROM cats;
- d. INSERT INTO animals SELECT \* FROM cats;
- e. INSERT INTO animals VALUES SELECT Name, Age FROM dogs;

45) Выберите верный синтаксис DELETE выражение **верно- b**

- a. DELETE FROM HAVING
- b. DELETE FROM WHERE
- c. DELETE WHERE
- d. DELETE FROM ORDER BY

46) Может ли первичный ключ (primary key) быть одновременно и внешним ключом (foreign key)? **верно- а**

- a. Да
- b. Нет

47) Команда CREATE позволяет **верно- а**

- a. Создать таблицу
- b. Такой команды не существует
- c. Добавить записи

48) Каким запросом можно удалить все записи из таблицы my\_table (но не саму таблицу) **верно- а**

- a. DELETE FROM my\_table
- b. DELETE my\_table
- c. DELETE TABLE my\_table
- d. DROP FROM my\_table

49) Дана таблица Friends со следующими записями

ID	NAME	AGE
01	Frank	18
02	Sharon	NULL
03	John	20
04	Jennifer	NULL

Сколько записей будет удалено в результате выполнения выражения **DELETE FROM Friends WHERE age=NULL** верно- **с**

- a. 2
- b. 1
- c. 0
- d. 4

50) Какое ключевое слово используется для удаления таблицы верно- **с**

- a. DELETE
- b. DESTROY
- c. DROP
- d. UPDATE

51) Какой вариант кода создает таблицу с одной колонкой строкового типа? верно- **д**

- a. CREATE TABLE test(test\_id TINYINT(10) IS NULL PRIMARY KEY)
- b. CREATE TABLE test(test\_id INT PRIMARY KEY)
- c. CREATE TABLE test(test\_id UNSIGNED PRIMARY KEY)
- d. CREATE TABLE test(test\_id VARCHAR(10) NOT NULL)

52) Можно ли создать таблицу без колонок? верно- **б**

- a. да
- b. нет

53) С помощью какого ключевого слова можно удалить записи таблицы? верно- **а**

- a. DELETE
- b. DESTROY
- c. DROP
- d. UPDATE

54) С помощью какого запроса можно поместить значение 'Perez' в поле LastName таблицы Persons? верно- **с**

- a. INSERT INTO Persons ('Perez') INTO LastName
- b. INSERT ('Perez') INTO Persons (LastName)
- c. INSERT INTO Persons (LastName) VALUES ('Perez')
- d. INSERT INTO Persons VALUE ('Perez')

55) Дана таблица Friends со следующими записями верно- **с**

ID	NAME	AGE
01	Frank	18
02	Sharon	NULL
03	John	20
04	Jennifer	NULL

Сколько записей будет создано в таблице `new_friends` в результате запроса

```
CREATE TABLE new_friends
SELECT id,name,age
FROM Friends
WHERE age<20
```

- a. 0
- b. 2
- c. 1
- d. 3
- e. 4
- f. Запрос выдаст ошибку

56) Дана таблица `Friends` со следующими записями **верно- b**

ID	NAME	AGE
01	Frank	18
02	Sharon	NULL
03	John	20
04	Jennifer	NULL

Сколько столбцов будет создано в таблице `new_friends` в результате запроса

```
CREATE TABLE new_friends
SELECT id, name
FROM Friends
WHERE age<20
```

- a. 0
- b. 2
- c. 1
- d. 3
- e. 4
- f. Запрос выдаст ошибку

57) Дана таблица `Friends` со следующими записями **верно- a**

ID	NAME	AGE
01	Frank	18
02	Sharon	NULL
03	John	20
04	Jennifer	NULL

Сколько строк будет создано в таблице `new_friends` в результате запроса

```
CREATE TABLE new_friends
SELECT id,name
FROM Friends
WHERE age=NULL
```

- a. 0
- b. 2
- c. 1
- d. 3
- e. 4
- f. Запрос выдаст ошибку

**58) В результате выполнения запроса верно- b**  
**CREATE TABLE new\_ptoducts LIKE products**

Будет...

- a. Создана таблица new\_ptoducts со структурой аналогичной таблице products, содержащая все записи из таблицы products
- b. Создана пустая таблица new\_ptoducts со структурой аналогичной таблице products
- c. Создана таблица products со структурой аналогичной таблице new\_ptoducts, содержащая все записи из таблицы new\_ptoducts

**59) Выберите команды, позволяющие изменить имя таблицы верно- а,с**

- a. ALTER TABLE catalog.order RENAME catalog.orders
- b. RENAME TABLE FROM catalog.order TO catalog.orders
- c. RENAME TABLE catalog.order TO catalog.orders
- d. ALTER TABLE catalog.order TO catalog.orders

**60) Команда позволяющая добавить столбец в таблицу имеет вид верно- b**

- a. INSERT INTO имя\_столбца VARCHAR(20)
- b. ALTER TABLE имя\_таблицы ADD имя\_столбца VARCHAR(20)
- c. ALTER TABLE имя\_таблицы INSERT имя\_столбца VARCHAR(20)
- d. ALTER TABLE имя\_таблицы DROP имя\_столбца VARCHAR(20)

**61) Команда позволяющая удалить столбец из таблицы имеет вид верно- с**

- a. DELETE FROM имя\_таблицы(имя\_столбца)
- b. ALTER TABLE имя\_таблицы DELETE имя\_столбца
- c. ALTER TABLE имя\_таблицы DROP имя\_столбца
- d. ALTER TABLE имя\_таблицы TO имя\_столбца

**62) Выберите команды, изменяющие определение столбца таблицы верно- а,b,с**

- a. ALTER TABLE имя\_таблицы MODIFY имя\_столбца varchar(30)
- b. ALTER TABLE имя\_таблицы ALTER имя\_столбца SET DEFAULT 0
- c. ALTER TABLE имя\_таблицы ALTER имя\_столбца DROP DEFAULT
- d. ALTER TABLE имя\_таблицы ADD имя\_столбца varchar(30)

**63) Ограничение ссылочной целостности задается с помощью верно- b**

- a. PRIMARY KEY
- b. FOREIGN KEY
- c. CHECK
- d. NOT NULL
- e. UNIQUE

**64) Ограничения первичного ключа задается с помощью верно- а**

- a. PRIMARY KEY
- b. FOREIGN KEY
- c. CHECK
- d. NOT NULL
- e. UNIQUE

**65) Ограничение уникальности задается с помощью верно- е**

- a. PRIMARY KEY
- b. FOREIGN KEY
- c. CHECK
- d. NOT NULL

e. UNIQUE

**66) Ограничение на отсутствие значения задается с помощью верно- d**

- a. PRIMARY KEY
- b. FOREIGN KEY
- c. CHECK
- d. NOT NULL
- e. UNIQUE

**67) Запретить удалять строку из таблицы, если на нее есть ссылка из дочерней таблицы, позволяет команда верно- c**

- a. ON DELETE CASCADE
- b. ON UPDATE CASCADE
- c. ON DELETE RESTRICT
- d. ON UPDATE RESTRICT
- e. ON DELETE SET NULL
- f. ON UPDATE SET NULL

**68) Запретить изменять строку из таблицы, если на нее есть ссылка из дочерней таблицы, позволяет команда поддержки целостности данных верно- d**

- a. ON DELETE CASCADE
- b. ON UPDATE CASCADE
- c. ON DELETE RESTRICT
- d. ON UPDATE RESTRICT
- e. ON DELETE SET NULL
- f. ON UPDATE SET NULL

**69) Удаление строки родительской таблицы приводит к удалению всех связанных с ней строк в дочерней таблице, если установлена поддержка целостности верно- a**

- a. ON DELETE CASCADE
- b. ON UPDATE CASCADE
- c. ON DELETE RESTRICT
- d. ON UPDATE RESTRICT
- e. ON DELETE SET NULL
- f. ON UPDATE SET NULL

**70) Даны команды для таблицы, в которой первый столбец является первичным ключом верно- b**

**INSERT INTO order\_line VALUES (6,8,1,79);**

**INSERT INTO order\_line VALUES (6,8,1,80)**

**ON DUPLICATE KEY UPDATE count\_p=80;**



**После выполнения этих команд в таблице**

- a. Будут добавлены две строки:

6	8	1	79
6	8	1	80

- b. Будут добавлена одна строка:

6	8	1	80
---	---	---	----

- c. Будут добавлена одна строка:

6	8	1	79
---	---	---	----

- d. Будут добавлены две строки:

6	8	1	79
7	8	1	80

- 71) Дана пустая таблица, созданная с помощью SQL-выражения: **верно- а**  
**CREATE TABLE people (**

**Id INT PRIMARY KEY AUTO\_INCREMENT,**  
**Name VARCHAR(45) NULL**

**)**

**Можно ли с помощью команды**

**INSERT INTO people SET id=1, name='Иван'**

**вставить новую строку в эту таблицу**

- a. да  
b. нет

- 72) Дана пустая таблица, созданная с помощью SQL-выражения: **верно- b**

**CREATE TABLE people (**

**Id INT PRIMARY KEY,**  
**Name VARCHAR(45) NULL**

**)**

**Можно ли с помощью команды**

**INSERT INTO people SET name='Иван'**

**вставить новую строку в эту таблицу**

- a. да  
b. нет

- 73) Дана пустая таблица, созданная с помощью SQL-выражения: **верно- а**  
**CREATE TABLE people (**

**Id INT PRIMARY KEY AUTO\_INCREMENT,**  
**Name VARCHAR(45) NULL**

**)**

**Можно ли с помощью команды**

**INSERT INTO people SET name='Иван'**

**вставить новую строку в эту таблицу**

- a. да  
b. нет

74) Дана таблицы **верно- b**

```

DOGS
Name  Age
-----
Snoopy 4
Benny  2

CATS
Name  Age
-----
Kleo  3
Linda 6

ANIMALS
Name  Age
-----

```

После команды

**UPDATE dogs SET age=(SELECT max(age) FROM cats);**

Таблица dogs будет иметь вид

a.

Name	Age
Snoopy	4
Benny	2

b.

Name	Age
Snoopy	6
Benny	6

c.

Name	Age
Snoopy	10
Benny	6

75) Дана таблицы **верно- c**

```

DOGS
Name  Age
-----
Snoopy 4
Benny  2

CATS
Name  Age
-----
Kleo  3
Linda 6

ANIMALS
Name  Age
-----

```

После команды

**UPDATE dogs SET age=(SELECT max(age) FROM cats) WHERE age>2;**

Таблица dogs будет иметь вид

a.

Name	Age
Snoopy	4
Benny	2

b.

Name	Age
Snoopy	6
Benny	6

c.

Name	Age
Snoopy	6
Benny	2

76) Дана таблицы **верно- с**

```

DOGS
Name  Age
-----
Snoopy 4
Benny  2

CATS
Name  Age
-----
Kleo  3
Linda 6

ANIMALS
Name  Age
-----

```

После команды

**UPDATE cats SET age=(SELECT max(age) FROM cats WHERE age<4);**

Таблица cats будет иметь вид

a.

Name	Age
Kleo	4
Linda	6

b.

Name	Age
Kleo	4
Linda	3

c.

Name	Age
Kleo	3
Linda	3

77) Дана таблица Friends со следующими записями **верно- b**

ID	NAME	AGE
01	Frank	18
02	Sharon	NULL
03	John	20
04	Jennifer	NULL

78) Какая из перечисленных команд заменит NULL значения в столбце age на 0?

- UPDATE friends SET age=0 WHERE age=NULL
- UPDATE friends SET age=0 WHERE age IS NULL
- UPDATE friends SET age=0 WHERE age IS NOT NULL
- UPDATE friends SET age=0

79) Можно ли с помощью команды UPDATE изменить записи в нескольких таблицах сразу **верно- a**

- Да
- Нет

80) Можно ли с помощью команды INSERT вставить записи в несколько таблиц сразу **верно- b**

- Да
- Нет

#### 4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

##### 1 ЭТАП – ЗНАТЬ

##### Критерии оценки устных ответов студентов

Оценка	Правильность (ошибочность) выполнения задания
«отлично»	полно и аргументировано отвечает по содержанию задания; понимает материал, обосновывает свои суждения, применяет знания на практике, иллюстрирует ответ примерами не только по предложенной литературе; излагает материал последовательно и правильно.
«хорошо»	полно и аргументировано отвечает по содержанию задания; понимает материал, обосновывает свои суждения, применяет знания на практике, иллюстрирует ответ примерами не только по предложенной литературе; излагает материал последовательно и правильно, но допускает 1-2 ошибки, которые исправляет сам.
«удовлетворительно»	знает и понимает основные положения по содержанию задания; излагает материал неполно, но допускает неточности в определении понятий или формулировке правил; не достаточно глубоко и доказательно обосновать свои суждения иллюстрирует ответ примерами только по предложенной литературе; излагает материал непоследовательно и допускает 3-4 ошибки.
«неудовлетворительно»	допускает существенные ошибки в формулировке определений и правил, искажающие их смысл;

	излагает материал непоследовательно, неуверенно и не по существу задания; допускает существенные ошибки, не позволяющие раскрыть смысл задания, являющиеся серьезным препятствием к успешному овладению следующим материалом.
--	--

## 2 ЭТАП – УМЕТЬ

### Методические рекомендации по выполнению индивидуальных заданий

Работа над выполнением индивидуальных заданий реализуется в разделах дисциплины «Проектирование баз данных» и «Язык SQL. Управление данными»

#### Индивидуальное задание № 1

Правила выполнения задания:

1. Выделить сущности, представленной предметной области.
2. Определить атрибуты каждой из сущностей и выделить среди них ключевые атрибуты.
3. Установить связи между сущностями. Для каждой связи определить класс принадлежности и степень связи.

Результат работы представить в виде ER-диаграммы.

#### Критерии оценивания выполнения индивидуального задания №1

Оценка	Правильность (ошибочность) выполнения задания
«отлично»	представлена ER-диаграмма правильной структуры: -выделены все сущности; -определены все атрибуты и правильно определены ключевые атрибуты; -определены связи между сущностями, для каждой связи заданы классы принадлежности и степени связи. при защите работы студент показывает глубокое знание вопросов темы
«хорошо»	представлена ER-диаграмма с несущественными ошибками в структуре: -выделены все сущности; -определены не все атрибуты и правильно определены ключевые атрибуты; -определены связи между сущностями, для каждой связи заданы классы принадлежности и степени связи, возможно с некоторыми ошибками. при защите работы студент без затруднений отвечает на вопросы
«удовлетворительно»	представлена ER-диаграмма с существенными ошибками в структуре: -выделены все сущности; -выделены не все атрибуты и не везде правильно определены ключевые атрибуты; -определены связи между сущностями, но не заданы степени связи и классы принадлежности. при защите показывает слабое знание вопросов темы, не всегда дает исчерпывающие и обоснованные вопросы на заданные вопросы, допускает существенные ошибки
«неудовлетворительно»	представлена ER-диаграмма с грубыми ошибками в структуре: -выделены не все сущности;

	-выделены не все атрибуты и неправильно определены ключевые атрибуты; -не верно определены связи между сущностями, но не заданы степени связи и классы принадлежности. при защите работы студент не может ответить на вопросы
--	---

### Индивидуальные задания №2

Правила выполнения заданий:

1. Преобразовать полученную в индивидуальном задании №1 ER-диаграмму ко второй нормальной форме.
2. Преобразовать диаграмму из п.1. к третьей нормальной форме.
3. Преобразовать полученную в индивидуальном задании №1 ER-диаграмму к нормальному виду в соответствии с правилами нормальной формы Бойса-Кода.

Результат работы представить в виде ER-диаграмм всех этапов преобразования.

### Критерии оценивания выполнения индивидуального задания №2

Оценка	Правильность (ошибочность) выполнения задания
«отлично»	правильно представлена все этапы преобразования ER-диаграмм к нормальной форме обеими способами: и через 2 и 3 НФ и через НФБК; при защите работы студент показывает глубокое знание вопросов темы
«хорошо»	правильно представлена все этапы преобразования ER-диаграмм к нормальной форме одним из способов: через 2 и 3 НФ или через НФБК; при защите работы студент без затруднений отвечает на вопросы
«удовлетворительно»	представлена некоторые этапы преобразования ER-диаграмм к нормальной форме одним способами: через 2 и 3 НФ или через НФБК, в процессе преобразования допущены ошибки; при защите показывает слабое знание вопросов темы, не всегда дает исчерпывающие и обоснованные вопросы на заданные вопросы, допускает существенные ошибки
«неудовлетворительно»	представлена некоторые этапы преобразования ER-диаграмм с грубыми ошибками, схема не нормализована; при защите работы студент не может ответить на вопросы

### Индивидуальные задания №3

Правила выполнения заданий:

1. В соответствии с описанием предметной области и формулировкой запросов разработать схему базы данных:

- таблицы;
- связи между таблицами;
- индексы.

Результат задания: представить скрин схемы базы данных. Для phpMyAdmin: на уровне базы данных выбрать пункт меню Дизайнер.

2. Заполнить таблицы данными так, чтобы можно было проверить корректность работы запросов.

3. Весь перечень запросов представить в виде хранимых процедур (возможно с входными и выходными параметрами)

4. Определить ограничения прав различных категорий пользователей с помощью задания прав пользователя и представлений.

Результат работы представить в виде базы данных, скрина схемы базы данных, текста всех запросов.

### Критерии оценивания выполнения индивидуального задания №3

Оценка	Правильность (ошибочность) выполнения задания
«отлично»	представлена заполненная базы данных правильной структуры, выполнено более 90% всех запросов; определены ограничения для всех категорий пользователей; при защите работы студент показывает глубокое знание вопросов темы
«хорошо»	представлена заполненная базы данных, в структуре базы данных могут быть незначительные ошибки; выполнено 75-90% всех запросов; определены ограничения для всех категорий пользователей; при защите работы студент без затруднений отвечает на вопросы
«удовлетворительно»	представлена заполненная базы данных, в структуре базы данных могут быть грубые ошибки; выполнено 60-75% всех запросов; при защите показывает слабое знание вопросов темы, не всегда дает исчерпывающие и обоснованные вопросы на заданные вопросы, допускает существенные ошибки
«неудовлетворительно»	представлена базы данных не соответствующей структуры, выполнено менее 60% всех запросов; при защите работы студент не может ответить на вопросы

### Методические рекомендации по решению кейс-задач

Задачи решения кейс-задач:

1. Научить студентов выполнять основные этапы по разработке сайтов в соответствии с требованиями заказчика.

2. Развить навыки организации работы в команде из 2-3 человек в соответствии с регламентом работы.

3. Развить компетенции студентов в ходе внесения предложений по определению функционала сайта.

4. Сформировать интерес к процессу обучения за счет решения практических общественно-значимых задач в рамках создания web-сайтов для некоммерческих организаций и объединений.

Количество участников: 20 человек+1 преподаватель

## План работы:

Этапы работы	Время
Организационный момент	10 мин
Деление на пары	5 мин
Постановка задачи	5 мин
Решение задачи	30 мин
Подведение итогов первого этапа, удаление решенных задач	5 мин
Перерыв	5 мин
Деление на пары	5 мин
Добавление ограничений на способы решения задачи	5 мин
Решение задачи	30 мин
Подведение итогов второго этапа, удаление решенных задач	5 мин
Перерыв	5 мин
Деление на пары	5 мин
Добавление ограничений на способы решения задачи	5 мин
Решение задачи	30 мин
Подведение итогов третьего этапа, удаление решенных задач	5 мин
Подведение итогов игры	5 мин
Итого:	160 мин

## Критерии оценивания выполнения кейс-задач

Оценка	Правильность (ошибочность) выполнения задания
«отлично»	правильно сформированы SQL-запросы ко всем трем поставленным задачам; выполняет ведущую роль в работе;
«хорошо»	правильно сформированы SQL-запросы к двум поставленным задачам; участвуют в работе, проявляет активность;
«удовлетворительно»	правильно сформированы SQL-запросы к одной поставленной задаче; участвуют в работе, не проявляет инициативы;
«неудовлетворительно»	не правильно сформированы SQL-запросы к поставленным задачам, мало участвует в решении задач, не проявляет инициативы.

## 3 ЭТАП – ВЛАДЕТЬ

## Критерии оценки по выполнению проектов

Оценка «отлично» выставляется студенту, если он выполнил работу на 90%, при защите работы студент показывает глубокое знание вопросов темы.

Оценка «хорошо» выставляется студенту, если он выполнил работу на 75%, при защите работы студент без затруднений отвечает на вопросы.

Оценка «удовлетворительно» выставляется студенту, если он выполнил работу на 60% с незначительными ошибками; при защите показывает слабое знание вопросов темы, не всегда дает исчерпывающие и обоснованные вопросы на заданные вопросы, допускает существенные ошибки.

Оценка «неудовлетворительно» - работа не выполнена.

## Критерии оценивания знаний на экзамене

Оценка «ОТЛИЧНО»:

1. Глубокое и прочное усвоение программного материала.
2. Знание пакетов прикладных программ



3. Знание основных принципов построения пакетов прикладных программ.
4. Знание основных задач прикладных программ.
5. Свободное владение пакетами прикладных программ.
6. Точность и обоснованность выводов.
7. Безошибочное выполнение практического задания.
8. Точные, полные и логичные ответы на дополнительные вопросы.

Оценка «ХОРОШО»:

1. Хорошее знание программного материала.
2. Недостаточно полное изложение теоретического вопроса экзаменационного билета.

3. Наличие незначительных неточностей в употреблении терминов, классификаций.

4. Знание основных пакетов прикладных программ
5. Неполнота представленного иллюстративного материала.
6. Точность и обоснованность выводов.
7. Логичное изложение вопроса, соответствие изложения научному стилю.
8. Негрубая ошибка при выполнении практического задания.
9. Правильные ответы на дополнительные вопросы.

Оценка «УДОВЛЕТВОРИТЕЛЬНО»:

1. Поверхностное усвоение программного материала.
2. Недостаточно полное изложение теоретического вопроса экзаменационного билета.

3. Затруднение в приведении примеров, подтверждающих теоретические положения.

4. Наличие неточностей в употреблении терминов, классификаций.
5. Неумение четко сформулировать выводы.
6. Отсутствие навыков научного стиля изложения.
7. Грубая ошибка в практическом задании.
8. Неточные ответы на дополнительные вопросы.

Оценка «НЕУДОВЛЕТВОРИТЕЛЬНО»:

1. Незнание значительной части программного материала.
2. Неспособность привести примеры пакетов прикладных программ
3. Неумение выделить главное, сделать выводы и обобщения.
4. Грубые ошибки при выполнении практического задания.
5. Неправильные ответы на дополнительные вопросы.

### Критерии оценивания курсовой работы

Оценка	Критерии
«отлично»	<ul style="list-style-type: none"> <li>- создана и заполнена матрица дашборда;</li> <li>- разработана структура дашборда;</li> <li>- разработанный дашборд полностью отображает разделы задания;</li> <li>- дано всестороннее и глубокое освещение избранной темы в тесной взаимосвязи с практикой, а её автор показал умение работать с различными видами источников, систематизировать, классифицировать, обобщать материал, формулируя выводы, соответствующие поставленным целям.</li> </ul>
«хорошо»	<ul style="list-style-type: none"> <li>- создана и заполнена матрица дашборда;</li> <li>- разработана структура дашборда;</li> <li>- разработанное приложение соответствует техническому заданию не в полном объёме;</li> <li>- имеются недоработки и ошибки;</li> </ul>

	<ul style="list-style-type: none"> <li>- работа, отвечающая основным, предъявляемым к ней требованиям.</li> <li>- студент показывает глубокие знания по предмету и владеет навыками научного исследования, но при этом имеются незначительные замечания по содержанию работы, по процедуре защиты (студент не может дать аргументировано ответы на вопросы).</li> </ul>
«удовлетворительно»	<ul style="list-style-type: none"> <li>- разработана архитектура классов;</li> <li>- создана и заполнена матрица дашборда;</li> <li>- разработана структура дашборда;</li> <li>- разработанное приложение не соответствует техническому заданию</li> <li>- работа не выполнена в полном объёме</li> <li>- в основном, соблюдены общие требования, но неполно раскрыты разделы плана, работа носит реферативный характер, отсутствуют аргументированные выводы. Автор курсовой работы посредственно владеет материалом, поверхностно отвечает на вопросы, в процессе защиты курсовой работы.</li> </ul>
«неудовлетворительно»	установлен акт несамостоятельного выполнения, работы, имеются принципиальные замечания по многим параметрам, содержание не соответствует теме, допущены грубые теоретические ошибки

#### **Методические рекомендации по решению тестовых заданий**

1. Отвечая на вопрос правильного ответа, щелкните на радиокнопке рядом с правильным ответом.
2. Отвечая на вопрос с несколькими правильными вариантами ответа, щелкните на чекбоксах рядом со всеми правильными ответами.
3. Отвечая на вопросы на правильную последовательность, впишите порядковый номер в поле ввода рядом с ответом.
4. Вопросов в тесте – 30.
5. Время на выполнение теста – 40 минут.

#### **Критерии оценивания решения тестовых заданий**

Оценка	Правильность (ошибочность) выполнения задания
«отлично»	91-100% правильных ответов
«хорошо»	74-90% правильных ответов
«удовлетворительно»	59-73% правильных ответов
«неудовлетворительно»	58% и менее правильных ответов