

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Гаскаев Сергей Валерьевич
Должность: Ректор
Дата подписания: 15.06.2026 12:16:04
Уникальный программный ключ:
04c19ed8bfb98f5b6cb77a486b9a8788b8327374



МИНОБРАЗОВАНИЯ И НАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)

Фонд оценочных средств для промежуточной аттестации по дисциплине «Технология баз данных» по направлению подготовки 01.03.02 «Прикладная математика и информатика» направленности «Прикладная математика и искусственный интеллект» ФГБОУ ВО «ЧелГУ»

стр. 1

Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю)
«Технология баз данных»

Направление подготовки (специальность)
01.03.02 «Прикладная математика и информатика»

Направленность (профиль)
«Прикладная математика и искусственный интеллект»

Присваиваемая квалификация
Бакалавр

Форма обучения
Очная

Год набора
2026

Челябинск, 2026 г.



Содержание

1. Паспорт фонда оценочных средств	3
2. Перечень формируемых компетенций	4
3. Содержание оценочных средств по дисциплине	5
3.1. Виды оценочных средств	5
3.2. Содержание оценочных средств	6
4. Порядок проведения и критерии оценивания промежуточной аттестации	44
4.1. Порядок проведения промежуточной аттестации	44
4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств	44
4.3. Результаты промежуточной аттестации и уровни сформированности компетенций.....	47



1. Паспорт фонда оценочных средств

Направление подготовки: 01.03.02 Прикладная математика и информатика.

Направленность: Прикладная математика и искусственный интеллект.

Дисциплина: Технология баз данных.

Семестры: 5, 6.

Форма промежуточной аттестации: зачёт в 5 семестре, экзамен в 6 семестре.

Для оценивания результатов обучения используется балльно-рейтинговая система.



2. Перечень формируемых компетенций

Изучение дисциплины «Технология баз данных» направлено на формирование компетенций, приведённых в Таблице 1.

Таблица 1. Результаты обучения по дисциплине.

Код и наименование компетенции согласно ФГОС	Индикаторы достижения компетенций согласно ОПОП ВО	Перечень планируемых результатов обучения по дисциплине
1	2	3
УК-3 Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде	УК-3.1. Демонстрирует понимание типологии и факторов формирования команд, лидерства и способов социального взаимодействия. УК-3.2. Осуществляет взаимодействие с другими членами команды, в т. ч. участвует в обмене информацией, знаниями и опытом. УК-3.3. Имеет опыт участия в командной работе.	Знать содержание работы членов команды на всех этапах проектирования и реализации проекта реляционной базы данных. Уметь взаимодействовать с другими членами команды при выполнении своей части работы по проектированию и реализации реляционной базы данных. Владеть опытом участия в командной работе по реализации проекта создания реляционной базы данных.
ОПК-2 Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач	ОПК-2.1. Имеет представление о существующих базовых математических методах и системах программирования, применяемых для разработки и реализации алгоритмов решения прикладных задач. ОПК-2.2. Демонстрирует умение применять математические методы и системы программирования для решения прикладных задач. ОПК-2.3. Имеет навыки разработки и применения алгоритмических и программных решений.	Знать о существующих математических методах и системах программирования. Уметь отбирать среди существующих математических методов, наиболее подходящих для решения конкретной прикладной задачи. Владеть навыками использования основных языков программирования, основных методов разработки программ, стандартов оформления программной документации.
ОПК-5: Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-5.1. Демонстрирует знание основ технологий программирования и базисных алгоритмов. ОПК-5.2. Демонстрирует умения разрабатывать алгоритмы и компьютерные программы. ОПК-5.3. Имеет практические навыки разработки компьютерных программ.	Знать основные алгоритмы в программировании. Уметь выбрать и обосновать выбор языка и среды программирования для разработки компьютерных программ. Владеть навыками разработки алгоритмов и компьютерных программ, пригодных для практического применения.



3. Содержание оценочных средств по дисциплине

3.1. Виды оценочных средств

Таблица 2. Виды оценочных средств.

Код, наименование компетенции согласно ФГОС	Перечень планируемых результатов обучения по дисциплине	Контролируемые темы/разделы (номер и название раздела из РПД п.2.2)	Семестр	Номер задания	Наименование оценочного средства
УК-3 Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде	Знать содержание работы членов команды на всех этапах проектирования и реализации проекта реляционной базы данных.	Теоретические основы технологии баз данных. Язык запросов SQL. Манипулирование данными. Проектирование баз данных. Управление базами данных.	5-6	1-10	Проектное задание
	Уметь взаимодействовать с другими членами команды при выполнении своей части работы по проектированию и реализации реляционной базы данных.				
	Владеть опытом участия в командной работе по реализации проекта создания реляционной базы данных.				
ОПК-2 Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач	Знать о существующих математических методах и системах программирования.			1-32	Лабораторная работа
	Уметь отбирать среди существующих математических методов, наиболее подходящих для решения конкретной прикладной задачи.				
	Владеть навыками использования основных языков программирования, основных методов разработки программ, стандартов оформления программной документации.				
ОПК-5: Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	Знать основные алгоритмы в программировании.			1-100	Тест
	Уметь выбрать и обосновать выбор языка и среды программирования для разработки			1-28	Вопросы к экзамену



	компьютерных программ.				
	Владеть навыками разработки алгоритмов и компьютерных программ, пригодных для практического применения.				

Типовые задания, критерии и показатели оценивания в рамках текущего контроля представлены в рабочей программе дисциплины (модуля). Полные комплекты оценочных средств и контрольно-измерительных материалов хранятся на кафедре.

3.2. Содержание оценочных средств

Промежуточная аттестация проводится в виде зачёта в 5 семестре, экзамена в 6 семестре.

Комплекты заданий лабораторных работ (1-18)

Лабораторная работа 1. Простой оператор Select (на основе одной таблицы).

Предикаты Or, And, Not. Получение итоговых значений. Псевдонимы. Упражнения №№ 1, 2, 3, 4, 5, 11, 12, 31, 33, 42. Всего 10 баллов.

Пример

Текст упражнения	Вариант запроса
№ 33 (1 балл) БД Корабли Укажите корабли, потопленные в сражениях в Северной Атлантике (North Atlantic). Вывод: ship.	<pre>select ship from Outcomes where result='sunk' and battle='North Atlantic'</pre>

Лабораторная работа 2. Традиционные операции над множествами и оператор SELECT. Вычисления в результирующем наборе запроса. Сортировка записей. Предикаты сравнения (between, like). Регулярные выражения.

Упражнения №№ 8, 35, 38, 44, 45. Всего 7 баллов.

Пример

Текст упражнения	Вариант запроса
№ 35 (2 балла) БД Компьютерная фирма В таблице Product найти модели, которые состоят только из цифр или только из латинских букв (A-Z, без учета регистра). Вывод: номер модели, тип модели.	<pre>SELECT model, typeFROM product WHERE model NOT LIKE '%[^0-9]%' OR model NOT LIKE '%[^a-z]%'</pre>

Лабораторная работа 3. Использование предикатов подзапросов с одним результирующим значением или с одним результирующим столбцом.

Упражнения №№ 7, 8, 9, 10, 13, 17, 23, 25, 26, 34, 36, 80, 144. Всего 21 баллов.

Пример

Текст упражнения	Вариант запроса
------------------	-----------------



№ 144 (1 балл) БД Компьютерная фирма. Найти производителей, которые производят PC как с самой низкой ценой, так и с самой высокой. Вывод: maker	<pre>select maker from product where model in (select model from pc where price =(select min(price) from pc))intersect select maker from product where model in (select model from pc where price=(select max(price) from pc))</pre>
--	--

Лабораторная работа 4. Групповые операции на основе одной таблицы. Упражнения №№ 15, 20, 22, 28, 40. Всего 8 баллов.

Пример

Текст упражнения	Вариант запроса
№ 40 (3 балла) БД Компьютерная фирма. Найти производителей, которые выпускают более одной модели, при этом все выпускаемые производителем модели являются продуктами одного типа. Вывести: maker, type.	<pre>SELECT maker, MIN(type) AS 'type' FROM Product GROUP BY maker HAVING MIN(type) = MAX(type) AND COUNT(model) > 1</pre>

Лабораторная работа 5. Использование предикатов, подзапросов, группировок и отбора групп по условию.

Упражнения №№ 37, 48, 49, 50, 63, 74, 85. Всего 12 баллов.

Пример

Текст упражнения	Вариант запроса
№ 63 (2 балла) БД Аэрофлот. Определить имена разных пассажиров, когда-либо летевших на одном и том же месте более одного раза.	<pre>SELECT name FROM Passenger WHERE ID_psg in (SELECT ID_psg FROM Pass_in_trip GROUP BY ID_psg, place HAVING count(*)>1)</pre>

Лабораторная работа 6. Явные операции внутреннего соединения таблиц. Упражнение №№ 111.

Повторение. Упражнения №№ 29, 30, 89, 92, 96, 121. Всего 18 баллов.

Пример

Текст упражнения	Вариант запроса
№ 111 (2 балла) БД Окраска. Найти НЕ белые и НЕ черные квадраты, которые окрашены разными цветами в пропорции 1:1:1. Вывод: имя квадрата, количество краски одного цвета	<pre>select Q_NAME, min(vol) volfrom utq join (select B_Q_ID, V_COLOR, sum(B_VOL) volfrom utB join utV on B_V_ID=V_ID group by B_Q_ID, V_COLOR) z on Q_ID =B_Q_ID group by q_id,Q_NAMEhaving count(*)=3 and sum(vol)<765 and sum(vol)>0 and min(vol)=max(vol)</pre>

Лабораторная работа 7. Использование нескольких дублий одного источника.



Явные операции внутреннего соединения таблиц.

Упражнения №№ 6, 14, 16, 18, 19, 21, 27, 73, 103, 108, 115, 124, 142. Всего 24 балла.

Пример

Текст упражнения	Вариант запроса
№ 6 (2 балла) БД Компьютерная фирма. Для каждого производителя, выпускающе- го ПК-блокноты с объёмом жёсткого дис- ка не менее 10 Гбайт, найти скорости та- ких ПК-блокнотов. Вывод: производитель, скорость.	SELECT DISTINCT maker, speed FROM Product INNER JOIN Laptop ON Laptop.model = Product.model and hd>=10

Лабораторная работа 8. Явные операции внешнего соединения таблиц. Упражнения №№ 46, 47, 56, 70, 71. Всего 12 баллов.

Пример

Текст упражнения	Вариант запроса
№ 71 (2 балла) БД Компьютерная фирма. Найти тех производителей ПК, все модели ПК которых имеются в таблице PC.	SELECT maker FROM product p LEFT JOIN pc ON pc.model = p.model WHERE type = 'PC' GROUP BY maker Having COUNT(p.model) = COUNT(pc.model)

Лабораторная работа 9. Вычисляемые таблицы, коррелированные подзапросы. Упражнения №№ 24, 39, 51, 61, 62, 67, 68, 71, 72, 77, 88, 89, 92, 114. Всего 23

Пример

Текст упражнения	Вариант запроса
№ 71 (2 балла) БД Компьютерная фирма. Найти тех производителей ПК, все модели ПК которых имеются в таблице PC.	SELECT distinct maker FROM product pr WHERE type= 'PC' and 0 = (SELECT COUNT(*) FROM (SELECT model FROM Product WHERE Pr.maker = Product.maker and type= 'PC' except SELECT model FROM pc) x)
№ 114 (2 балла) БД Аэрофлот Определить имена разных пассажиров, которым чаще других доводилось лететь на одном и том же месте. Вывод: имя и количество полетов на одном и том же месте.	WITH b AS (SELECT ID_psg, COUNT(*) as cnt FROM Pass_In_Trip GROUP BY ID_psg, place), b1 AS (SELECT DISTINCT ID_psg, cnt FROM b WHERE cnt =(SELECT MAX(cnt) FROM b)) SELECT name, cnt FROM b1 JOIN Passenger p ON b1.ID_psg = p.ID_psg

Лабораторная работа 10. Условные операторы и функции.

Упражнения №№ 29, 41, 52, 57, 59, 64, 83, 85, 112, 123, 128. Всего 22 баллов.

Пример

Текст упражнения	Вариант запроса
------------------	-----------------



№ 52 (2 балла) БД Корабли. Определить названия всех кораблей из таблицы Ships, которые могут быть линейным японским кораблем, имеющим число главных орудий не менее девяти, калибр орудий менее 19 дюймов и водоизмещение не более 65 тыс. тонн	<pre>select name from ships where class in (select class from classes where type='bb' and country='Japan' and isnull(numguns,9)>=9 and isnull(bore,18)<19 and isnull(displacement,64000)<=65000)</pre>
--	---

Лабораторная работа 11. Преобразование типов данных. Упражнения №№ 32, 53, 54, 58, 91, 146. Всего 16 баллов.

Пример

Текст упражнения	Вариант запроса
№ 53 (2 балла) БД Корабли. Определите среднее число орудий для классов линейных кораблей. Получить результат с точностью до 2-х десятичных знаков.	<pre>select cast(avg(numguns*1.0) as numeric(4,2)) as Avg_numGuns from classes where type='bb'</pre>

Лабораторная работа 12. Обработка данных типа Дата/Время. Преобразование типов данных.

Упражнения №№ 43, 60, 75, 84, 87, 110, 135, 141. Всего 18 баллов.

Упражнение №№ 76, 78, 79, 81, 93, 119, 120, 149. Всего 19 баллов.

Пример

Текст упражнения	Вариант запроса
№ 43 (2 балла) БД Корабли. Укажите сражения, которые произошли в годы, не совпадающие ни с одним из годов спуска кораблей на воду.	<pre>select name from battles where DATEPART(yy, date) not in (select coalesce(launched,0) from ships)</pre>

Лабораторная работа 13. Функции обработки строковых данных. Упражнения №№ 86, 95, 102, 127, 131, 136. Всего 18 баллов.

Пример

Текст упражнения	Вариант запроса
№ 43 (2 балла) БД Корабли. Укажите сражения, которые произошли в годы, не совпадающие ни с одним из годов спуска кораблей на воду.	<pre>select name from battles where DATEPART(yy, date) not in (select coalesce(launched,0) from ships)</pre>

Лабораторная работа 14. Генерация последовательностей. Упражнения №№ 66, 94, 98, 99, 104, 118, 140, 143. Всего 21 балл.

Пример

Текст упражнения	Вариант запроса
------------------	-----------------



<p>№ 140 (2 балла) БД Корабли. Определить, сколько битв произошло в течение каждого десятилетия, начиная с даты первого сражения в базе данных и до даты последнего. Вывод: десятилетие в формате "1940s", количество битв.</p>	<pre>WITH maxyear as (select max(year(date) - year(date)%10) m from battles), Series AS (SELECT min(year(date) - year(date)%10) a from battles UNION all SELECT a+10 FROM Series WHERE a+10 <=(select m from maxyear)) SELECT CAST(a AS varchar(5)) + 's', count(name) FROM Series left join battles on year(date) - year(date)%10=a group by a</pre>
---	--

Лабораторная работа 15. Оконные функции ранжирования.

Упражнения №№ 65, 82, 90, 100, 105, 116, 117, 125, 130, 137, 147. Всего 29 баллов.

Пример

Текст упражнения	Вариант запроса
<p>№ 105 (2 балла) БД Компьютерная фирма Статистики Алиса, Белла, Вика и Галина нумеруют строки у таблицы Product. Все четверо упорядочили строки таблицы по возрастанию названий производителей. Алиса присваивает новый номер каждой строке, строки одного производителя она упорядочивает по номеру модели. Трое остальных присваивают один и тот же номер всем строкам одного производителя. Белла присваивает номера начиная с единицы, каждый следующий производитель увеличивает номер на 1. У Вики каждый следующий производитель получает такой же номер, какой получила бы первая модель этого производителя у Алисы. Галина присваивает каждому следующему производителю тот же номер, который получила бы его последняя модель у Алисы. Вывести: maker, model, номера строк получившиеся у Алисы, Беллы, Вики и Галины соответственно.</p>	<pre>Select maker, model, row_number() over(order by maker,model) as num_Alis , dense_rank() over(order by maker) as num_Bella, RANK() over(order by maker) as num_Vika, count(*) over (order by maker) as num_Galin from Product</pre>

Лабораторная работа 16. Оконные функции агрегирования и сдвига. Упражнения №№ 69, 77,

88, 96, 105, 109, 126, 129. Всего 13 баллов.

Упражнения №№ 101, 106, 122, 132, 134. Всего 18 баллов.

Пример

Текст упражнения	Вариант запроса
------------------	-----------------



<p>№ 109 (2 балла) БД Окраска</p> <p>Вывести:</p> <ol style="list-style-type: none"> 1. Названия всех квадратов черного или белого цвета. 2. Общее количество белых квадратов. 3. Общее количество черных квадратов. 	<pre>SELECT Q_NAME, (sum(SUM(B_VOL)) over())/765 as Whites, count(*) over()- (sum(SUM(B_VOL)) over())/765 as Blacks FROM utQ LEFT JOIN utB ON Q_ID = B_Q_ID GROUP BY Q_ID, Q_NAME HAVING SUM(B_VOL) = 765 OR SUM(B_VOL) IS NULL</pre>
---	---

Лабораторная работа 17. Операторы PIVOT/ UNPIVOT. Упражнения №№ 86, 97, 113, 146.
Всего 5 баллов.

Пример

Текст упражнения	Вариант запроса
<p>№ 113 (2 балла) БД Окраска</p> <p>Сколько каждой краски понадобится чтобы докрасить все небелые квадраты до белого цвета.</p> <p>Вывод: количество каждой краски в порядке (R, G, B)</p>	<pre>SELECT sum(255-ISNULL ([R],0)) R , sum(255-isnull([G],0)) G, sum(255-isnull([B],0)) B FROM (/*merging all tables to find paint filling and color for all squares*/ select ISNULL(B_Q_ID, Q_ID) ID, V_COLOR, B_VOL Vol from utB RIGHT JOIN utQ on B_Q_ID=Q_ID LEFT JOIN utV on B_V_ID=V_ID) as SourceT PIVOT (/*rotating table and calculating each paint volume for each square*/ SUM(Vol) For V_COLOR IN ([R], [G], [B])) Pvt /*excluding white squares*/ where ISNULL ([R],0) + isnull([G],0) + isnull([B],0) <765</pre>

Лабораторная работа 18. Резерв.

Комплекты заданий лабораторных работ (19-32)

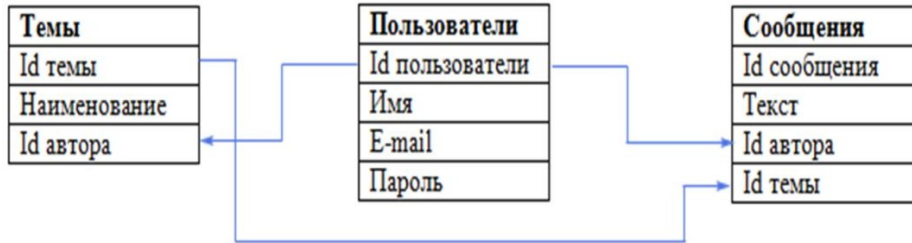
Лабораторная работа 19. Операторы модификации данных.
Упражнения №№ 1-20.

Пример

Текст упражнения	Вариант запроса
<p>Удалить из таблицы Laptop все портативные компьютеры с размером экрана менее 12 дюймов.</p>	<pre>DELETE FROM Laptop WHERE screen < 12</pre>

Лабораторная работа 20. Операторы DDL. Создание учебной базы данных в СУБД.
Задания:

1. Создать базу данных и таблицы в соответствии со схемой:



2. Введите в таблицы по 3-4 строки.
3. Выводите все записи таблицы topics.
4. Выведите все записи таблицы posts.
5. Выведите название темы, имя пользователя, создавшего сообщение, текст сообщения.
6. Выведите пользователя и количество созданных им тем.
7. Выведите пользователя и количество созданных им сообщений.
8. Выведите пользователя, количество созданных им тем и количество созданных им сообщений.

Лабораторная работа 21. Автоинкрементные поля, форматирование вывода данных, составной первичный ключ, создание БД Учебная часть.

Задания

Создайте БД_учебная часть, дайте ему название BD_UCH.

Откройте слайд 40 лекции 20 Проектирование баз данных.

Создайте четыре таблицы Должность, Стаж, Преподаватель, Преподаватель_ведет_Занятие. Начинать с тех, где нет внешних ключей. Названия для таблиц придумайте короткие, но отражающие содержание. Первичные поля естественные (автоинкрементируемость не используем). Тип и размер полей подберите в соответствии с предметной областью.

Введите данные в таблицы, используя слайд 39 лекции 17 Проектирование баз данных.

Создайте запросы для вывода данных из базы данных.

Напишите запрос: какую заработную плату получают сотрудники кафедры 25.

Лабораторная работа 22. Типы данных. Декларативное описание ограничений.

Создание БД Интернет-магазин.

Лабораторная работа 23. Именованные ограничения, задаваемые декларативно.

Изменения таблицы с использованием ALTER TABLE.

Лабораторная работа 24. Физическое проектирование БД Книжное издательство.

Описание составного внешнего ключа. Графическое изображение схемы БД.

Лабораторная работа 25. Создание представлений - именованных хранимых запросов.

Лабораторная работа 26. Правила внешних ключей.

Лабораторная работа 27-28. Триггеры обработки событий.

Лабораторная работа 29-32. Выполнение и представление результатов группового проектного задания по индивидуальной теме.



Проектное задание.

Порядок выполнения проектного задания

Распределение ведущей роли каждого члена команды в разработке проекта.

1.1. Проектирование концептуальной модели данных.

1.1.1. Проанализировать внешнюю модель предметной области своего варианта задания.

Выделить в ней локальные представления, соответствующие относительно независимым данным.

1.1.2. Выделить объекты, описывающие локальную предметную область проектируемой базы данных. Описать атрибуты, составляющие структуру каждого объекта.

1.1.3. Определить ключевые атрибуты.

1.1.4. Определить связи между объектами, удалить избыточные связи.

1.1.5. Проанализировать неключевые атрибуты и дополнить их, если необходимо.

1.1.6. Объединить локальные представления.

1.2. Проектирование логической модели данных.

Перевод концептуальной схемы данных в реляционную схему с целью последующей реализации в СУБД.

1.3. Физическое проектирование базы данных.

1.3.1. Создается пустая база данных.

1.3.2. Создаются таблицы базы данных.

1.3.3. Создается схема данных.

1.3.4. Таблицы заполняются данными.

1.3.5. Проектируются необходимые запросы.

Структура отчета по проектному заданию

1. Формулировка задания.
2. Описание роли в работе команды.
3. Концептуальная модель.
4. Логическая модель.
5. Скрипты создания базы и таблиц в ней.
6. Скрипты запросов, представлений, триггеров.
7. Результаты выполнения.
8. Скрипты запросов на наполнение таблиц данными.

Варианты заданий

Вариант 1. Прокат автомобилей

Внешняя модель: в пункте проката имеются автомобили различных марок и моделей, отличающиеся также категорией (грузовые, легковые, микроавтобусы и т.п.), типом двигателя (бензиновый, дизельный), типом кузова и набором технических параметров, состав которых различается для разных типов автомобилей. Возможно наличие нескольких экземпляров автомобилей одной марки и модели.

Каждый автомобиль имеет уникальный номер государственной регистрации.

Клиент выбирает автомобиль из имеющихся в наличии в данный момент и вносит в кассу залоговый платеж, предъявляет документ для регистрации и пользуется автомобилем в течение установленного времени.

При возврате автомобиля отмечается время, вычисляется стоимость проката, исходя из ставки почасовой оплаты и производится окончательный расчет с клиентом. Ставка почасовой оплаты зависит от категории автомобиля и мощности двигателя. Сумма залогового платежа определяется маркой и моделью автомобиля.



Пользователи: 1. Менеджер редактирует справочники (категории, марки, модели автомобилей, тарифы почасовой оплаты, суммы залоговых платежей); проводит финансовый анализ. 2. Кассир регистрирует операции проката, прием платежей от клиентов. 3. Клиент может просматривать характеристики автомобилей, имеющихся в пункте проката.

Компоненты пользовательского интерфейса: 1. Формы для просмотра и редактирования справочников, регистрации клиентов, регистрации проката главная кнопочная форма. 2. Запросы: выдача на экран клиентов, взявших в прокат автомобиль, список свободных автомобилей по маркам и моделям. 3. Отчеты: прайс-лист, выручка за период времени по категориям автомобилей.

Вариант 2. Оптовый товарный склад

Внешняя модель: На складе хранятся товары, предназначенные для продажи. Каждый товар принадлежит к определенной группе товаров и характеризуется наименованием, производителем, количеством, единицей измерения и базовой ценой реализации. Товары поступают на склад партиями – одна партия от одного поставщика. В одной партии могут быть поставлены товары одного или нескольких наименований. При поступлении партии фиксируется дата поставки, поставщик, количество единиц товара, единицы измерения, оптовая цена единицы товара.

Все товары со склада продаются по единой цене, независимо от времени их поставки и цены поставщика. Отпуск товаров покупателям производится партиями. При этом фиксируется дата отпуска, состав партии с учетом количества и фактической цены реализации каждого товара.

Пользователи:

1. Менеджер регистрирует поставщиков товаров, редактирует справочники (группы товаров, фактическую цену товара), проводит финансовый анализ.
2. Кладовщик регистрирует поставки и отпуск товаров.
3. Покупатель просматривает прейскурант товаров (по группам).

Компоненты пользовательского интерфейса:

1. Формы для просмотра и редактирования групп товаров, поставщиков, покупателей и отпускных цен на товары; для регистрации партии поставки и отпуска товара, главная кнопочная форма.
2. Запросы. Складской запас товаров по группам, количество товаров в каждой группе, стоимость складского запаса по группам товаров.
3. Отчеты. Складской запас товаров по группам, торговая выручка за период времени по группам проданных товаров.

Вариант 3. Спортивно-оздоровительный комплекс.

Внешняя модель: В состав комплекса входит несколько специализированных спортивных залов, в которых занимаются соответствующие секции (плавание, спортивная гимнастика, тяжелая атлетика, шахматы, спортивная борьба, шейпинг и т.п.). Список секций может изменяться администратором.

Один зал может иметь одну или несколько специализаций. Для занятия в секциях формируются группы клиентов. В одной секции может быть несколько групп.

Клиенты занимаются в секциях на платной основе. В каждой секции устанавливаются единые ставки почасовой оплаты. Ведется персональный почасовой учет посещаемости занятий и учет внесения платежей. Для проведения занятий резервируется время под каждое занятие в соответствующем зале. В штате работают тренеры, каждый из которых может проводить занятия в одной или нескольких секциях. С одной группой может заниматься только один тренер.

Пользователи: 1. Администратор ведет учет спортивных залов (по специализациям), учет тренерского состава (по секциям и группам), планирование работы секций (по группам и залам); финансовый учет (платежи клиентов). 2. Тренер осуществляет оперативный учет посещения занятий, ведет набор клиентов в группы. 3. Клиент просматривает список секций, расписание



занятий в секциях и группах.

Компоненты пользовательского интерфейса: 1. Формы для просмотра и редактирования списков залов, секций и тренеров; просмотра и редактирования списков групп клиентов; регистрации посещения занятий, регистрация внесения платежей, главная кнопочная форма. 2. Запросы: расписание проведения занятий в группах, списки клиентов группы, список групп секций. 3. Отчеты: финансовый результат работы секции за период времени (квартал, год), финансовый результат работы спортивного зала за месяц.

Вариант 4. Учет результатов вступительных экзаменов.

Внешняя модель: учебное заведение объявляет набор абитуриентов на ряд специальностей, распределенных по факультетам. Для каждой специальности определен план набора студентов. Для каждой специальности определен единый перечень дисциплин, по которым проводятся вступительные экзамены. Экзамены проводятся в традиционной форме (результат каждого экзамена оценивается по 4-х балльной системе – неудовлетворительно, удовлетворительно, хорошо, отлично. Абитуриент, получивший оценку «неудовлетворительно» до следующего экзамена не допускается). Абитуриент представляет в приемную комиссию документ об образовании и подает заявление, в котором указывает специальность, на которую поступает. После этого он зачисляется в группу для сдачи экзаменов.

Формируется расписание сдачи экзаменов для каждой группы и вывешивается в фойе на доску объявлений. После каждого экзамена публикуются результаты и корректируются списки групп. После сдачи всех экзаменов публикуются списки абитуриентов по специальностям, зачисленных на первый курс.

Пользователи: 1. Секретарь приемной комиссии формирует списки групп абитуриентов по факультетам и специальностям, регистрирует результаты экзаменов. 2. Председатель комиссии проводит анализ результатов вступительных экзаменов по факультетам и специальностям.

Компоненты пользовательского интерфейса: 1. Формы формирования перечня дисциплин по факультетам, формирование и редактирование списков групп абитуриентов, заполнение экзаменационных ведомостей, главная кнопочная форма. 2. запросы: планы набора по специальностям, список абитуриентов, не сдавших экзамен, рейтинговые списки абитуриентов после сдачи очередного экзамена. 3. Отчеты: списки абитуриентов, зачисленных на первый курс (по факультетам и специальностям).

Вариант 5. Учет контингента и успеваемости студентов.

Внешняя модель: В ВУЗе несколько факультетов, на каждом обучаются студенты на нескольких специальностях. Студент одновременно может обучаться на одной специальности факультета.

Студенты распределены по курсам и группам, на каждой специальности может быть несколько групп одного курса.

Для каждой специальности определен учебный план: перечень дисциплин в каждом семестре (семестры пронумерованы с 1 по 9). В учебном плане специальности в каждом семестре для каждой дисциплины определены: количество контрольных работ (от 1 до 4), зачет (есть/нет), экзамен (есть/нет). Результаты экзаменов оцениваются по 4-х балльной системе (2-5), результаты контрольных работ и зачетов – по двухбалльной (зачтено/незачтено). Оценка 0 ставится студенту, не явившемуся на экзамен (зачет) или не сдавшему контрольную работу. Допускаются многократные пересдачи.

Пользователи: 1. Секретарь подготавливает зачетно-экзаменационные ведомости для групп, ведет учет выполнения студентами контрольных работ, сдачи зачетов и экзаменов. 2. Декан проводит анализ текущей успеваемости студентов в течение семестра, анализ результатов экзаменационных сессий, выбирает кандидатов на поощрение и отчисление.



Компоненты пользовательского интерфейса: 1. Формы для формирования групп студентов, учебных планов по специальностям и курсам, обработка результатов экзаменационной сессии, главная кнопочная форма. 2. Запросы: успеваемость студентов в семестре, списки студентов, имеющих задолженности по контрольной работе, списки студентов, не сдавших зачеты/экзамены, списки отличников. 3. Отчеты: зачетно-экзаменационная ведомость для группы по дисциплине. Результаты экзаменационной сессии.

Вариант 6. Учет работы компьютерных классов.

Внешняя модель: В учебном заведении имеется несколько компьютерных классов. В каждом классе имеются рабочие места для студентов, оборудованные персональными компьютерами определенной марки, модели и комплектации. На каждом ПК установлено программное обеспечение, необходимое для проведения учебных занятий по соответствующим дисциплинам. За каждым классом закреплен оператор. В классе занимаются студенты, изучающие разные дисциплины. В начале семестра каждый студент проходит процедуру регистрации в качестве пользователя и закрепляется за определенным рабочим местом соответствующего компьютерного класса.

Студенты распределены по группам. Групповые занятия проводятся в соответствии с расписанием, формируемым на каждый семестр для каждого компьютерного класса. Единица расписания – учебное занятие длиной два академических часа (90 мин.), т.н. «пара». Все «пары» пронумерованы с 1 по 8, время начала и окончания занятий фиксировано. Расписание составляется на 6 рабочих дней недели и одинаково для всего семестра. Каждое занятие проводит преподаватель.

В системе ведется учет изменений программного обеспечения и комплектации компьютеров, учет сеансов работы пользователей. Студент имеет право пользоваться своим рабочим местом вне расписания занятий группы при условии, что оно не занято другой группой по расписанию.

Пользователи: 1. Администратор регистрирует изменения в оборудовании компьютерных классов и программном обеспечении ПК, формирование расписания, анализ загрузки компьютерных классов занятиями по расписанию. 2. Оператор регистрирует пользователей-студентов, ведет учет посещения учебных занятий по расписанию группы и дополнительных индивидуальных занятий.

Компоненты пользовательского интерфейса: 1. Формы для ввода данных, редактирования списков групп студентов, формирования расписания. 2. Запросы: свободное время компьютерных классов, посещаемость занятий студентами. 3. Отчеты: расписание класса на семестр, состав оборудования и программного обеспечения компьютерных классов (по классам).

Вариант 7. Учет продажи запчастей к автомобилям.

Внешняя модель: На складе хранятся товары, предназначенные к продаже: запасные части к автомобилям. Каждый товар принадлежит к определенной группе по типу запчастей и по маркам автомобилей, для которых он предназначен.

Товары поступают на склад партиями. В одной партии могут быть поставлены товары нескольких наименований, принадлежащие к различным группам. При поступлении партии товара фиксируется дата поставки, поставщик и состав партии товара – наименование товара, технические характеристики, группы товара (по категории и марке автомобиля), производитель, количество единиц товара, оптовая цена единицы товара.

Все одноименные товары реализуются по одной цене, независимо от времени их поставки и цены поставщика.

Товар отпускается со склада поштучно. При этом фиксируется дата отпуска, количество единиц и розничная цена каждого товара с учетом скидки. Торговая скидка (в

% от розничной цены) устанавливается в размере 5% при наличии у покупателя дисконтной карты.



Пользователи: 1. Менеджер регистрирует поставщиков, редактирует справочники (группы товаров, розничные цены, торговые скидки), проводит финансовый анализ. 2. Кладовщик/продавец регистрирует поставку и продажу товаров. 3. Покупатель: поиск товаров и просмотр прайс-листа.

Компоненты пользовательского интерфейса: 1. Формы для просмотра и редактирования справочников, регистрации партии поставки товара, регистрации отпуска товара со склада, главная кнопочная форма. 2. Запросы: складской запас по группам товаров, стоимость складского запаса по группам товаров, поиск товаров, имеющихся на складе (по первой букве, по категории, по наименованию). 3. Отчеты: прайс-лист, торговая выручка за период времени (месяц, квартал) по группам товаров.

Вариант 8. Читательский абонемент библиотеки.

Внешняя модель: При поступлении книги в библиотечный фонд в базе данных фиксируется ее название, автор (или группа авторов), издательство, год издания, библиотечный и авторский коды книги, ее аннотация, а также жанр (детектив, фантастика, учебник и т. п.). Возможно наличие нескольких экземпляров одинаковых книг. При поступлении в фонд каждый экземпляр книги получает уникальный номер.

При регистрации читателя в библиотеке фиксируется его фамилия, имя, домашний адрес, телефон и место работы. Экземпляры книг выдаются читателям на определенный срок (фиксируется дата выдачи, дата возврата, дата фактического возврата). За систематическое нарушение сроков возврата книг (более 7-ми раз) читатель лишается права пользования библиотечным фондом.

Пользователи: 1. Библиотекарь регистрирует читателя, регистрирует выдачу и возврат книг, поиск читателей должников, ведет «черный» список читателей. 2. Читатель ведет поиск книг по жанрам, названиям и авторам с просмотром аннотаций к книгам.

Компоненты пользовательского интерфейса: 1. Формы для регистрации читателя, поступления книги в фонд библиотеки, выдача/возврат книг, главная кнопочная форма. 2. Запросы: количество книг в библиотеке по жанрам, авторам; список книг одного автора, книг по жанру, книг у читателя, список должников (с указанием срока задолженности в днях). 3. Отчеты: рейтинг популярности книг по жанрам, по авторам.

Вариант 9. Прокат видеопродукции.

Внешняя модель: В пункте проката имеются фильмы на различных носителях (кассеты и диски различных форматов). Возможно наличие нескольких экземпляров одинаковых видеофильмов, в том числе и на разных носителях. Видеофильм характеризуется жанром, компанией-производителем, годом выпуска, списком авторов и исполнителей (продюсеров, режиссеров, операторов, актеров первого и второго планов). Видеофильмы выдаются напрокат клиентам на определенный срок на условиях посуточной оплаты с залоговым платежом. Сумма залогового платежа и размер тарифа посуточной оплаты определяется типом носителя.

Ведется учет выдачи/возврата видеопродукции клиентам и персональный учет денежных средств на лицевых счетах клиентов. Клиент вносит на свой лицевой счет денежную сумму, равную или большую сумме залогового платежа. Если клиент не возвратил взятый на прокат видеопродукт в течение установленного срока, соответствующий экземпляр списывается из фонда, а с лицевого счета клиента списывается сумма залогового платежа за невозвращенный видеопродукт. При возврате видеопродукта клиентом в течение установленного срока, с его лицевого счета списывается денежная сумма за срок проката по установленному тарифу. Оставшейся на лицевом счете суммой клиент может распорядиться по своему усмотрению: забрать или использовать в качестве залога за новую видеопродукцию.

Пользователи: 1. Менеджер редактирует справочники (носители, жанры, авторы и др.), проводит финансовый анализ. 2. Кассир регистрирует поступление видеопродукции, регистрирует



операции выдачи/возврата видеопродукции. Принимает платежи клиентов. 3. Клиент может вести поиск ведеопродукта по жанрам, авторам, исполнителям, просмотр прайс-листа.

Компоненты пользовательского интерфейса: 1.Формы для просмотра и редактирования справочников, регистрации поступления видеопродукции, регистрации операций выдачи/возврата. 2. Запросы: поиск по жанрам, исполнителям, авторам, названию; суммы остатков на лицевых счетах клиентов. 3. Отчеты: прайс-лист, торговая выручка за период времени по жанрам и типам носителей.

Вариант 10. Читальный зал периодических изданий.

Внешняя модель: В фонде библиотеки хранятся периодические издания (журналы, газеты) различных категорий (технические, научные, развлекательные, рекламно-информационные, педагогические и др.). Каждое периодическое издание характеризуется названием, годом выпуска и номером. Возможно наличие в фонде нескольких экземпляров одного и того же периодического издания. При поступлении в фонд каждый экземпляр получает уникальный номер.

Периодические издания содержат статьи, каждая из которых имеет название. Автора (или группу авторов). Аннотации статей хранятся в базе данных. При регистрации читателя в библиотеке фиксируется его фамилия, имя, отчество, домашний адрес, телефон и место работы. Регистрация действует 1 год. В январе каждого года читатель должен пройти перерегистрацию. Читатель, не прошедший регистрацию/перерегистрацию не допускается в читальный зал. Читатель имеет право пользоваться фондом читального зала, но не имеет прав уносить издания домой.

Пользователи: 1. Библиотекарь регистрирует читателя, выдает и принимает периодические издания. 2. Читатель ведет поиск статей по названию, автору, поиск изданий по наименованию журнала (газеты). Компоненты пользовательского интерфейса: 1.Формы для регистрации читателя, издания, выдачи/возврата экземпляров, главная кнопочная форма. 2. Запросы: список статей запрашиваемого автора, опубликованных за период времени, список журналов, имеющихся в фонде по названиям, по годам выпуска, поиск статьи по автору, названию статьи. 3.Отчеты: рейтинг популярности периодических изданий (по категориям).

Теоретические вопросы к экзамену

1. Основные понятия технологий баз данных: база данных, свойства базы данных, система баз данных, основные функции системы баз данных, язык баз данных, стандарты языка баз данных SQL.

2. Основные понятия технологий баз данных: СУБД, типология, компоненты и функции СУБД.

3. Архитектура системы баз данных: ANSI/SPARC-архитектура описания БД с тремя уровнями абстракции.

4. Архитектура системы баз данных: сетевая архитектура системы баз данных.

5. Модели и типы данных: иерархическая модель, сетевая модель, реляционная модель (с примерами).

6. Реляционная модель: состав, характеристика структурной части реляционной модели данных.

7. Реляционная модель: состав, характеристика целостной части реляционной модели данных.

8. Реляционная модель: состав, характеристика манипуляционной части реляционной модели данных.

9. Отношения, совместимые по типу, переименование атрибутов и теоретико-множественные операции с отношениями.

10. Специальные реляционные операции с отношениями.



11. Зависимые реляционные операторы и запросы, невыразимые средствами реляционной алгебры.
12. Язык запросов SQL: общие характеристики, составные части SQL, основные операторы DML.
13. Язык запросов SQL: общие характеристики, составные части SQL, основные операторы DDL.
14. Этапы разработки базы данных.
15. Метод сущность-связь: понятие семантического моделирования; элементы ER-модели; модель данных "сущность-связь"; диаграммы "сущность-связь"
16. Критерии оценки качества логической модели данных.
17. Функциональная зависимость атрибутов отношений.
18. Вторая нормальная форма (2НФ) отношений и приведение к ней.
19. Третья нормальная форма (3НФ) отношений и приведение к ней.
20. Нормальная форма Бойса-Кодда и приведение к ней.
21. Сравнение нормализованных и ненормализованных моделей.
22. Целостность баз данных: понятие целостности данных; средства контроля целостности данных. Декларативные ограничения целостности.
23. Целостность баз данных: понятие целостности данных; средства контроля целостности данных. Процедурные ограничения целостности (триггеры).
24. Безопасность баз данных: понятие безопасности базы данных; модели защиты и разграничения доступа.
25. Транзакции: понятие транзакции; АСИД-свойства транзакций; управление параллельными транзакциями (с примерами).
26. Восстановление баз данных и резервное копирование: журнализация транзакций; восстановление базы данных; резервное копирование базы данных (с примерами).
27. Физические модели БД. Системы управления файлами. Хеширование.
28. Индексирование. Индексный файл.

База практических заданий к экзамену

1. Задача на составление запроса с использованием традиционных множественных операторов.

Текст задания	Примерный вариант выполнения
(26) БД Компьютерная фирма Найдите среднюю цену ПК и ПК-блокнотов, выпущенных производителем А (латинская буква). Вывести: одна общая средняя цена.	<pre>select avg (price) as Avg_price from (SELECT price FROM pc where model in(select model from product WHERE mak-er='A') UNION ALL SELECT price FROM laptop where model in(select model from product WHERE mak-er='A')) all price</pre>
(23) БД Аэрофлот Найдите производителей, которые производили бы как ПК со скоростью не менее 750 МГц, так и ПК-блокноты со скоростью не менее 750 МГц. Вывести: Maker	<pre>SELECT maker FROM Product where model in (select model from PC where speed>=750) INTERSECT SELECT maker FROM Product where model in (select model from laptop where speed>=750)</pre>



2. Задача на составление запроса с условием и группировкой без соединения таблиц.

Текст задания	Примерный вариант выполнения
(85) БД Компьютерная фирма Найти производителей, которые выпускают только принтеры или только РС. При этом искомые производители РС должны выпускать не менее 3 моделей.	<pre>select maker from product group by maker having count(distinct type) = 1 and (min(type) = 'printer' or (min(type) = 'pc' and count(model) > 2))</pre>
(92) БД Окраска Выбрать все белые квадраты, которые окрашивались только из баллончиков, пустых к настоящему времени. Вывести имя квадрата.	<pre>with empty as (select B_V_ID from utB group by B_V_ID having SUM(B_VOL) = 255), white as (select B_Q_ID from utB group by B_Q_ID having SUM(B_VOL) = 255*3) select Q_NAME from utQ where Q_ID IN(select * from white) AND Q_ID NOT IN(select B_Q_ID from utB where B_V_ID NOT IN(select * from empty))</pre>

3. Задача на составление запроса с подзапросами в условии и группировкой на основе соединения таблиц.

Текст задания	Примерный вариант выполнения
(142) БД Аэрофлот Среди пассажиров, летавших на самолетах только одного типа, определить тех, кто прилетал в один и тот же город не менее 2-х раз. Вывести имена пассажиров.	<pre>select name from passenger where id_psg in(select id_psg from pass_in_trip join trip on pass_in_trip.trip_no=trip.trip_no group by id_psg having max(plane)=min(plane) and count(town_to)>count(distinct town_to))</pre>
(124) БД Аэрофлот Среди пассажиров, которые пользовались услугами не менее двух авиакомпаний, найти тех, кто совершил одинаковое количество полётов самолетами каждой из этих авиакомпаний. Вывести имена таких пассажиров.	<pre>select (select name from passenger where id_psg=p.id_psg) as name from (select id_psg, id_comp, count(*) as flite_on_comp from pass_in_trip pit join trip ton pit.trip_no = t.trip_no group by id_psg, id_comp) p group by id_psg having count(*) > 1 and (max(flite_on_comp) = min(flite_on_comp))</pre>

4. Задача на составление запроса с использованием вычисляемой таблицы.

Текст задания	Примерный вариант выполнения
(72) БД Аэрофлот Среди тех, кто пользуется услугами только какой-нибудь одной компании, определить имена разных пассажиров, летавших чаще других.	<pre>with a as (Select id_psg, count(distinct id_comp) as comp, count(date) as col from pass_in_trip as p join trip on trip.trip_no=p.trip_no group by id_psg</pre>



Вывести: имя пассажира и число полетов.	having count(distinct id_comp)=1 select name, col from a join passenger as p on p.id_psg=a.id_psg where col=(select max(col) from a)
(24) БД Компьютерная фирма Перечислите номера моделей любых типов, имеющих самую высокую цену по всей имеющейся в базе данных продукции.	With a as(select model, price from pc union select model, price from Laptop union select model, price from printer) select model from a where price = (select max(price) from a)

5. Задача на составление запроса с использованием условных операторов и функций.

Текст задания	Примерный вариант выполнения
(41) БД Компьютерная фирма Для каждого производителя, у которого присутствуют модели хотя бы в одной из таблиц PC, Laptop или Printer, определить максимальную цену на его продукцию. Вывод: имя производителя, если среди цен на продукцию данного производителя присутствует NULL, то выводить для этого производителя NULL, иначе максимальную цену.	With tablica AS (SELECT model, price, iif(price is null,1,0) i from laptop UNION select model, price, iif(price is null,1,0) FROM printer UNION select model, price, iif(price is null,1,0) FROM pc) SELECT maker, CASE WHEN sum(I)>0 then NULL else MAX(price) end max_price from product join tablica ON tablica.model = product.model GROUP BY maker
(64) БД Вторсырье Используя таблицы Income и Outcome, для каждого пункта приема определить дни, когда был приход, но не было расхода и наоборот. Вывод: пункт, дата, тип операции (inc/out), денежная сумма за день.	select point, date, iif(Sum(inc)=0,'out','inc') as oper, Sum(inc) + Sum(out) as out from (Select point, date, inc, 0 as out from Income union all Select point, date, 0 as inc, out from Outcome) as d group by point, date having Sum(inc)=0 or Sum(out)=0

6. Задача на составление запроса с преобразованием типа данных.

Текст задания	Примерный вариант выполнения
(146) БД Компьютерная фирма Для ПК с максимальным кодом из таблицы PC вывести все его характеристики (кроме кода) в два столбца: - название характеристики (имя	With t as(select * from PC where code >= all(select code from pc)) select 'speed' as chr, CAST(speed as varchar) as value from t union



соответствующего столбца в таблице PC); - значение характеристики	<pre>select 'model' , model from tunion select 'ram' , CAST(ram as varchar) from tunion select 'hd' , CAST(hd as varchar) from tunion select 'cd' , cd a from tunion select 'price' , CAST(price as varchar) as valuefrom t</pre>
(58) БД Компьютерная фирма Для каждого типа продукции и каждого производителя из таблицы Product с точностью до двух десятичных знаков найти процентное отношение числа моделей данного типа данного производителя к общему числу моделей этого производи-теля. Вывод: maker, type, процентное отношение числа моделей данного типа к общему числу моделей производителя	<pre>select maker,type,cast(1.0* (select count(*) from Product wheremaker=q1.maker and type=q2.type) / (select count(*) from Product where mak- er=q1.maker)*100 as numeric(10,2)) from (select distinct maker from Product) as q1,(select distinct type from Product) as q2</pre>

7. Задача на составление запроса с обработкой данных типа Дата/Время.

Текст задания	Примерный вариант выполнения
(142) БД Аэрофлот Среди пассажиров, летавших на самолетах только одного типа, определить тех, кто прилетал в один и тот же город не менее 2-х раз. Вывести имена пассажиров.	
(124) БД Аэрофлот Среди пассажиров, которые пользовались услугами не менее двух авиакомпаний, найти тех, кто совершил одинаковое количество полётов самолетами каждой из этих авиакомпаний. Вывести имена таких пассажиров.	

8. Задача на составление запроса с обработкой строковых данных.

Текст задания	Примерный вариант выполнения
(79) БД Аэрофлот Определить пассажиров, которые больше 0 других времени провели в полетах. Вывод: имя пассажира, общее время в минутах, проведенное в полетах	<pre>with t as(select name, sum(case when DATEDIFF(mi, t.time_out,t.time_in)> 0 then DATEDIFF(mi, t.time_out,t.time_in) else DATEDIFF(mi, time_out, time_in)+1440end) vr from pass_in_trip pt join trip t on pt.trip_no=t.trip_no join passenger c on pt.id_psg=c.id_psg group by pt.id_psg, name) select name, vrfrom t where vr=(select max(vr) from t)</pre>



(135) БД Окраска В пределах каждого часа, в течение которого выполнялись окраски, найти максимальное время окраски (B_DATETIME).	<pre>select max(B_DATETIME)from utb group by DATEPART(yy, B_DATETIME),DATEPART(mm, B_DATETIME), DATEPART(dd, B_DATETIME),DATEPART(hh, B_DATETIME)</pre>
--	---

9. Задача на составление запроса с генерацией последовательностей.

Текст задания	Примерный вариант выполнения
(104) БД Корабли Для каждого класса крейсеров, число орудий которого известно, пронумеровать (последовательно от единицы) всеорудия. Вывод: имя класса, номер орудия в формате 'bc-N'.	<pre>With Num as (SELECT class, 1 n from classes where type ='bc' and numguns is not null union all SELECT class,n+1 from num Where n+1<=(select numGuns from classes c1where c1.class=num.class)) SELECT class,'bc-'+CAST(n AS CHAR(2)) n FROM num</pre>
(66) БД Аэрофлот Для всех дней в интервале с 01/04/2003 по 07/04/2003 определить число рейсов из Rostov. Вывод: дата, количество рейсов.	<pre>WITH Calend AS (SELECT CONVERT(datetime,'20030401') AS tripdate UNION ALL SELECT tripdate+1 FROM Calend WHERE tripdate<'20030407') SELECT Calend.tripdate, coalesce(tripnum, 0)Qty FROM Calend LEFT JOIN (select date, COUNT(DISTINCT trip_no) AStripnum from Pass_in_trip AS PiT where PiT.trip_no IN (SELECT DISTINCT trip_no FROM Trip WHERE town_from='Rostov') group by date) y ON Calend.tripdate=y.date</pre>

10. Задача на составление запроса с использованием оконных функций ранжирования.

Текст задания	Примерный вариант выполнения
(137) БД Компьютерная фирма Для каждой пятой модели (в порядке возрастания номеров моделей) из таблицы Product определить тип продукции и среднюю цену модели.	<pre>select type,avg(price)avg_price from(select model,type,row_number()over(order by model)i from product)a left join(select model,price from pc union all select model,price from laptop union all select model,price from printer</pre>



	<pre>)b on a.model=b.model where i%5=0 group by a.model,type,i</pre>
(65) БД Компьютерная фирма Пронумеровать уникальные пары {maker, type} из Product, упорядочив их следующим образом: - имя производителя (maker) по возрастанию; - тип продукта (type) в порядке PC, Laptop, Printer. Если некий производитель выпускает несколько типов продукции, то выводить его имя только в первой строке; остальные строки для ЭТОГО производителя должны содержать пустую строку символов (").	<pre>with a as (select row_number() over(order by maker, len(type)) no, row_number() over(partition by maker order by maker, len(type)) temp, maker, type from (select distinct maker, type from product) temp) select no,case</pre>
	<pre>when temp = 1 then maker else "end as maker,type from a</pre>

11. Задача на составление запроса с использованием оконных функций ранжирования.

Текст задания	Примерный вариант выполнения
(137) БД Компьютерная фирма Для каждой пятой модели (в порядке возрастания номеров моделей) из таблицы Product определить тип продукции и среднюю цену модели.	<pre>select type,avg(price)avg_price from(select model,type,row_number()over(order bymodel)i from product)a left join(select model,price from pcunion all select model,price from laptopunion all select model,price from printer)b on a.model=b.modelwhere i%5=0 group by a.model,type,i</pre>
(65) БД Компьютерная фирма Пронумеровать уникальные пары {maker,type} из Product, упорядочив их следующим образом: - имя производителя (maker) по возрастанию; - тип продукта (type) в порядке PC, Laptop, Printer. Если некий производитель выпускает несколько типов продукции, то выводить его имя только в первой строке; остальные строки для ЭТОГО производителя должны содержать пустую строку символов (").	<pre>with a as (select row_number() over(order by maker,len(type)) no, row_number() over(partition by maker order bymaker, len(type)) temp, maker, type from (select distinct maker, type from product)temp) select no,case</pre>
	<pre>when temp = 1 then maker else "end as maker,type from a</pre>

12. Задача на составление запроса с использованием оконных функций агрегирования.

Текст задания	Примерный вариант выполнения
---------------	------------------------------



(69) БД Вторсырье По таблицам Income и Outcome для каждого пункта приема найти остатки де- нежных средств на конец каждого дня, в который выполнялись операции по приходу и/или расходу на данном пункте. Учтеть при этом, что деньги не изымаются, а) X остатки/задолженность переходят на следующий день. Вывод: пункт приема, день в формате "dd/mm/yyyy", остатки/задолженность на конец этого дня.	select point, convert(char(10),date,103) day, SUM(sum(inc)) OVER (PARTITION BY point ORDER BY date) rem from (Select point, date, inc from Income UNION ALL Select point, date, -out from Outcome UNION ALL Select point, date, 0 from X) group by point, date
---	--

(109) БД Окраска 1. Названия всех квадратов черного или белого цвета. 2. Общее количество белых квадратов. 3. Общее количество черных квадратов.	SELECT Q_NAME, (sum(SUM(B_VOL)) over()) / 765 as Whites, count(*) over() (sum(SUM(B_VOL)) over()) / 765 as Blacks FROM utQ LEFT JOIN utB ON Q_ID = B_Q_ID GROUP BY Q_ID, Q_NAME HAVING SUM(B_VOL) = 765 OR SUM(B_VOL) IS NULL
--	---

13. Задача на составление запроса с использованием оконных функций сдвига.

Текст задания	Примерный вариант выполнения
(126) БД Аэрофлот Для последовательности пассажиров, упорядоченных по id_psg, определить того, кто совершил наибольшее число полетов, а также тех, кто находится в последовательности непосредственно перед и после него. Для первого пассажира в последовательности предыдущим будет последний, а для последнего пассажира последующим будет первый. Для каждого пассажира, отвечающего условию, вывести: имя, имя предыдущего пассажира, имя следующего пассажира.	with pass as (select id_psg, count(*) kol_flite, max(count(*)) over() max_flite from pass_in_trip group by id_psg), y as (select id_psg, kol_flite from pass where kol_flite = max_flite), f as (select name from passenger where ID_psg = (select max(ID_psg) from passenger)), l as (select name from passenger where ID_psg = (select min(ID_psg) from passenger)) select name, prev, next from (select ID_psg, name, LAG(name, 1, (select name from f)) OVER (ORDER BY ID_psg) prev, LEAD(name, 1, (select name from l)) next from passenger) l join y on l.id_psg = y.id_psg
(129) БД Окраска Предполагая, что среди идентификаторов квадратов имеются пропуски, найти минимальный и максимальный "свободный" идентификатор в диапазоне между име-	select min(Q_id)+1, max(Q)-1 from (select Q_id, lead(Q_id) over (order by Q_id) Q from utq) t where Q - Q_id > 1



Ющимися максимальным и минимальным идентификаторами.
Например, для последовательности идентификаторов квадратов 1,2,5,7 результат должен быть 3 и 6.
Если пропусков нет, вместо каждого искомого значения выводить NULL.

14. Задача на составление запроса на модификацию данных (INSERT, DELETE, UP-DATE).

Текст задания	Примерный вариант выполнения
Вставить в таблицу Product_D все строки из таблицы Product, относящиеся к моделям персональных компьютеров (type = 'PC').	INSERT INTO Product_D SELECT * FROM Product WHERE type = 'PC'
Указать «No PC» (нет ПК) в столбце type для тех моделей ПК из таблицы Product, для которых нет соответствующих строк в таблице PC.	UPDATE Product SET type = 'No PC' WHERE type = 'pc' AND model NOT IN (SELECT model FROM PC)
Удалить те модели ПК из таблицы Product, для которых нет соответствующих строк в таблице PC.	DELETE FROM Product WHERE type = 'pc' AND model NOT IN (SELECT model FROM PC)

15. Задача на создание реляционных таблиц.

15.1 Схема отношения СОТРУДНИКИ (Employees)

Содержание поля	Имя поля	Тип, длина	Примечания
Табельный номер	E_TAB	N(4)	первичный ключ
Фамилия	E_FNAME	C(20)	обязательное поле
Имя, отчество	E_LNAME	C(30)	обязательное поле
Дата рождения	E_BORN	D	
Пол	E_SEX	C(1)	обязательное поле
Номер комнаты	E_ROOM	N(3)	составной внешний ключ(к Rooms)
Номер телефона	E_TEL	C(10)	
ИНН	E_INN	C(12)	обязательное поле

Схема отношения КОМНАТЫ (Rooms)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер комнаты	R_NO	N(3)	обязательное поле
Номер телефона	R_TEL	C(10)	пара полей Номер комнаты и Номер телефона уникальная

1. Создайте таблицы:

2. Для всех ограничений целостности создайте правила.

3. Учтите ограничения целостности, которые не указаны в таблице:

- значения всех числовых атрибутов – больше 0 (или null, если атрибут необязателен).
- область значений атрибута E_Sex отношения EMPLOYEES – символы 'м' и 'ж'.
- отношение ROOMS не имеет первичного ключа, но комбинация значений (R_no, Tel)

уникальна.

Выполнение задания



```
create table rooms (  
r_no numeric(3) not null,r_tel varchar(10),  
unique ( r_no, r_tel));  
create table employees (  
e_tab numeric(4) primary key,e_fname varchar(20) not null, e_lname varchar(30) not null, e_born  
date,  
e_sex char(1) not null check(e_sex in ('ж','м')),e_room numeric(3),  
e_tel varchar(10), e_inn char(12) not null,  
foreign key(e_room,e_tel) references rooms(r_no,r_tel));
```

15.2 Схема отношения КНИГИ (Books)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер контракта	B CONTRACT	N(6)	первичный ключ
Дата подписания контракта	B DATE	D	обязательное поле
Менеджер	B MAN	N(4)	внешний ключ (к Employees)
Название книги	B TITLE	N(40)	обязательное поле
Цена	B PRICE	N(6,2)	цена экземпляра книги
Ответственный редактор	B EDIT	N(4)	внешний ключ (к Employees)

Схема отношения СОТРУДНИКИ (Employees)

Содержание поля	Имя поля	Тип, длина	Примечания
Табельный номер	E_TAB	N(4)	первичный ключ
Фамилия	E_FNAME	C(20)	обязательное поле
Имя, отчество	E_LNAME	C(30)	обязательное поле
Дата рождения	E_BORN	D	необязательное поле
Пол	E_SEX	C(1)	обязательное поле
Номер телефона	E_TEL	C(10)	необязательное поле
ИНН	E_INN	C(12)	обязательное поле

1. Создайте таблицы:
2. Для всех ограничений целостности создайте правила.
3. Учтите ограничения целостности, которые не указаны в таблице:
 - значения всех числовых атрибутов – больше 0 (или null, если атрибут необязателен).
 - область значений атрибута E_Sex отношения EMPLOYEES – символы 'м' и 'ж'.

16. Задача на создание триггера обработки события.

16.1 Существует таблица ITEMS (заказы) в БД КНИГОИЗДАТЕЛЬСТВО

Содержание поля	Имя поля	Тип, длина	Примечания
Номер заказа	ID	N(6)	внешний ключ (к Orders)
Код книги (№ контракта)	contract	N(6)	внешний ключ (к Books)
Количество	COUNT	N(4)	обязательное поле

• Создайте таблицу А для отслеживания операций вставки, удаления, изменения строк в таблице ITEMS с атрибутами:

A_ID_old – номер удаленного/ измененного заказа; A_ID_new – номер нового /измененного заказа; A_contr_old – код удаленной/ замененной книги; A_contr_new – код заказанной / измененной книги; A_count_old – количество удаленной/ замененной книги. A_count_new – количество



заказанной / измененной книги; A_date – дата операции;

operation – операция (значения атрибута из списка 'delete', 'insert', 'update');

- Создайте триггер, который будет после удаления строки из таблицы items вставлять в таблицу A строку (номер удаленного/ измененного заказа; null, код удаленной/ замененной книги; null, количество удаленной/ замененной книги, null, datetime('now'), 'delete').

- Создайте триггер, который будет после изменения строки в таблице items вставлять в таблицу A строку (номер измененного заказа; номер нового заказа; код замененной книги; код новой книги; количество замененной книги, количество новой книги, datetime('now'), 'update').

- Создайте триггер, который будет после вставки строки в таблицу items вставлять в таблицу A строку (null, номер вставленного заказа; null, код вставленной книги; null, количество вставленной книги, datetime('now'), 'insert').

Выполнение задания

```
CREATE TABLE IF NOT EXISTS a (A_Id_old INTEGER, A_ID_new INTEGER, a_contr_old
NUMERIC(6), a_contr_new NUMERIC(6), a_count_old NUMERIC(6), a_count_new NUMERIC(6),
A_date TEXT NOT NULL,
operation TEXT NOT NULL
);
```

```
CREATE TRIGGER IF NOT EXISTS chk_delete
after delete on itemsfor each row
begin
INSERT INTO A (A_id_old, A_id_new, a_contr_old, a_contr_new, a_count_old, a_count_new,
A_date, operation)
VALUES (old.i_id, null, old.i_contract, null, old.i_count, null, datetime('now'),'delete');
end;
```

```
CREATE TRIGGER IF NOT EXISTS chk_update
after update on itemsfor each row
begin
INSERT INTO A (A_id_old, A_id_new, a_contr_old, a_contr_new, a_count_old, a_count_new,
A_date, operation)
VALUES (old.i_id, new.i_id, old.i_contract, new.i_contract, old.i_count, new.i_count,
datetime('now'), 'update');
end;
```

```
CREATE TRIGGER IF NOT EXISTS chk_insert
after insert on itemsfor each row
begin
INSERT INTO A (A_id_old, A_id_new, a_contr_old, a_contr_new, a_count_old, a_count_new,
A_date, operation)
VALUES (null, new.i_id, null, new.i_contract, null, new.i_count, datetime('now'), 'in-
sert');
end;
```

Существует таблица BOOKS (книги).

Содержание поля	Имя поля	Тип, длина	Примечания
-----------------	----------	------------	------------



Номер контракта	B CONTRACT	N(6)	первичный ключ
Дата подписания контракта	B DATE	D	обязательное поле
Менеджер	B MAN	N(4)	внешний ключ (к Employees)
Название книги	B TITLE	N(40)	обязательное поле
Цена	B PRICE	N(6,2)	цена экземпляра книги
Ответственный редактор	B EDIT	N(4)	внешний ключ (к Employees)
Тираж			
Остаток тиража	B REST		Производное поле

Существует таблица ITEMS (заказы) в БД КНИГОИЗДАТЕЛЬСТВО

Содержание поля	Имя поля	Тип, длина	Примечания
Номер заказа	I ID	N(6)	внешний ключ (к Orders)
Код книги (№ контракта)	I contract	N(6)	внешний ключ (к Books)
Количество	I COUNT	N(4)	обязательное поле

- Создайте триггер, который изменяет поле `b_rest` **после вставки** строки в таблицу `items` (строки заказа) с количеством книг (учесть проверку достаточности остаткатиража).
- Триггер, который изменяет поле `b_rest` **после изменения** в строке таблицы `items` (строки заказа) количества книг (учесть проверку достаточности остаткатиража).
- Триггер, который изменяет поле `b_rest` **после удаления** строки из таблицы `items` (строки заказа).

Выполнение задания:

```
CREATE TRIGGER IF NOT EXISTS chk_insert
before insert on itemsfor each row
when ((select b_rest from books where new.i_contract = b_contract) >= new.i_count)begin
update books set b_rest = b_rest - NEW.i_count where books.b_contract =NEW.i_contract;
end;
```

```
CREATE TRIGGER IF NOT EXISTS chk_update
after update on itemsfor each row
when ((select b_rest from books where new.i_contract = b_contract) >= new.i_count)begin
update books set b_rest = b_rest - NEW.i_count + old.i_count where books.b_contract =
NEW.i_contract;
end;
```

```
CREATE TRIGGER IF NOT EXISTS chk_delete
after delete on itemsfor each row
begin
update books set b_rest = b_rest + old.i_count where books.b_contract =NEW.i_contract;
end;
```

Задача на изменение реляционной таблицы

Имеется таблица

```
CREATE TABLE Customers ( Id INT IDENTITY,
Age INT CONSTRAINT DF_Customer_Age DEFAULT 18, FirstName NVARCHAR(20) NOT
NULL,
```



LastName NVARCHAR(20) NOT NULL, Email VARCHAR(30),
Phone VARCHAR(20),
CONSTRAINT PK_Customer_Id PRIMARY KEY (Id),
CONSTRAINT CK_Customer_Age CHECK(Age >0 AND Age < 100), CONSTRAINT
UQ_Customer_Email UNIQUE (Email), CONSTRAINT UQ_Customer_Phone UNIQUE (Phone)
)

Задания:

- Переименуйте таблицу.
- Добавьте новый столбец Sex_of_Customer, определите его тип, размер, ограничения.
- Удалите столбец Email.
- Измените ограничение CONSTRAINT CK_Customer_Age, указав диапазон возраста от 16 до 75.
- Измените столбец Phone, установив обязательность ввода значения.

17. Задача на установление уровня доступа в базе данных.

- Назначьте системную привилегию на создание, удаление, изменение таблиц пользователю petrov.
- Назначьте объектную привилегию на вставку, удаление, изменение строк в таблицах S и P пользователю Ivanov с правом ее передачи.
- Отмените системную привилегию, выданную на создание, удаление, изменение таблиц пользователю Petrov.
- Отмените права передачи привилегии без отмены самой объектной привилегии на вставку, удаление, изменение строк в таблицах S и P пользователю Ivanov.

Выполнение задания:

- grant create, drop, alter table to petrov;
- grant insert, update, delete on S, P to Ivanov with grant option;
- revoke create, drop, alter table from petrov;
- revoke grant option for insert, update, delete on S, P from ivanov.

18. Задача на нахождение значения высказывания в трехзначной логике.

- Чему равно значение выражения A or (B and C) в трехзначной логике, если A=T, B=T, C=U?
- Чему равно значение выражения A and (B and C) в трехзначной логике, если A=T, B=T, C=U?
- Чему равно значение выражения A and (not B and C) в трехзначной логике, если A=T, B=T, C=U?
- Чему равно значение выражения not (B or not C) в трехзначной логике, если A=T, B=T, C=U?

Выполнение задания:

T or (T and U)=T or U=T. Ответ: T.

T and (T and U)=T and U=U. Ответ: U.

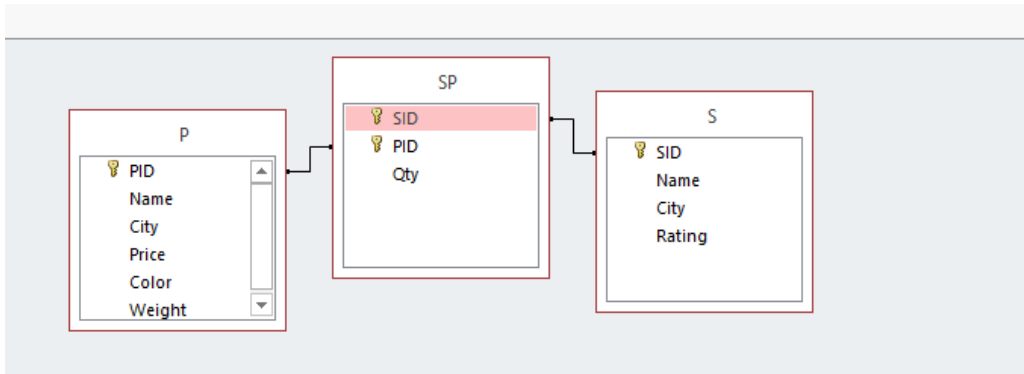
T and (not T and U)=T and (F and U)=T and U=U. Ответ: U. not (T or not U)= not (T or U)=not T=F. Ответ: F.

19. Задача на создание реляционного выражения. БД включает три таблицы:

S – ПОСТАВЩИКИ;

P – ДЕТАЛИ;

SP – ПОСТАВКИ



Составить реляционное выражение

1) Получить имена поставщиков, которые не поставляют модель с кодом P2.

2) Решение:

$R1 := \pi_{SID} (S)$

$R2 := \sigma_{PID=P2}(SP)$

$R3 := \pi_{SID} (R2)$

$R4 := R1 - R3$

$R5 := R4 \bowtie S$

$Result := \pi_{Name}(R5)$

$\pi_{Name}((\pi_{SID} (S) - \pi_{SID} \sigma_{PID=P2} (SP))) \bowtie S)$

3) Получить имена поставщиков, которые поставляют детали красного цвета. Решение:

$R1 := \sigma_{Color='красный'} (P)$

$R2 := \pi_{SID} (R1)$

$R3 := R2 \bowtie SP$

$R4 := R3 \bowtie S$

$Result := \pi_{Name}(R4)$

$\pi_{Name}((\pi_{SID} (\sigma_{Color='красный'} (P)) \bowtie SP) \bowtie S)$

17. Задача на нормализацию отношений в базе данных.

18. Задача на создание ER-диаграммы.

19. Задача на создание схемы реляционных отношений на основе ER-модели.

База тестовых вопросов

Теоретические основы технологии баз данных.

1. Совокупность специальным образом организованных данных, хранимых в памяти ЭВМ и отображающих состояние объектов и их взаимосвязи называется:

- A) информационной системой
- B) системой управления базами данных
- C) **базой данных +**
- D) банком данных

2. Комплекс языковых и программных средств для создания, ведения и совместного использования БД многими пользователями называется:

- A) **системой управления базами данных+**
- B) прикладной программой
- C) сервером баз данных



- D) приложением
3. Лицо или группа лиц, на которых возложены максимальные обязанности по работе и обслуживанию базы данных - это:
- A) разработчик базы данных
 - B) администратор базы данных +**
 - C) пользователь базы данных
 - D) обслуживающий персонал
4. Отношение в реляционной модели данных состоит из:
- A) тела отношения
 - B) заголовка отношения
 - C) тела и заголовка отношения+**
 - D) множества кортежей
5. Схема отношения в реляционной модели данных определяется:
- A) строкой таблицы
 - B) таблицей
 - C) заголовком столбца таблицы
 - D) строкой заголовков таблицы+**
6. Строка таблицы в реляционной модели данных называется:
- A) отношение
 - B) сущность
 - C) атрибут
 - D) кортеж+**
7. Множество допустимых значений атрибута в реляционной модели данных называется:
- A) отношение
 - B) сущность
 - C) домен**
 - D) атрибут
8. Описание свойств объекта в реляционной модели данных называется:
- A) кортеж
 - B) домен
 - C) сущность+**
 - D) атрибут
9. Какая модель данных не обеспечивает представления любой предметной области:
- A) иерархическая+**
 - B) сетевая
 - C) реляционная
 - D) объектно-ориентированная
10. Какая модель данных предназначена для аналитической обработки данных:
- A) многомерная+**
 - B) постреляционная
 - C) реляционная
 - D) объектно-ориентированная
11. Какая модель применяет механизмы инкапсуляции, наследования и полиморфизма:
- A) многомерная
 - B) сетевая



- С) реляционная
D) объектно-ориентированная+
12. Поле, указывающее на запись в другой таблице, называется:
А) первичным ключом
B) внешним ключом+
С) связью
D) схемой базы данных
13. Группа связанных таблиц называется:
А) составным первичным ключом
B) внешним ключом
С) связью
D) схемой базы данных+
14. Информация о таблицах, полях таблиц (имена, типы данных, формат), первичных и внешних ключах и других объектах базы данных называется:
А) макетом таблиц
B) метаданными+
С) связью
D) схемой базы данных
15. Какое определение наиболее точно определяет свойства первичного ключа:
А) однозначная идентификация записи и атомарность
B) избыточность и атомарность
C) однозначная идентификация записи и отсутствие избыточности+
D) отсутствие избыточности и целостность
16. Значение, используемое для однозначного определения записи, называется:
A) первичный ключ
B) внешний ключ
С) кортеж
D) отношение
17. Реляционная модель данных состоит из трех частей:
A. структурной, целостной, манипуляционной+
B. структурной, сетевой, манипуляционной.
C. структурной, целостной, иерархической
D. сетевой, целостной, манипуляционной
18. Структурная часть реляционной модели данных описывает
A. какие объекты рассматриваются реляционной моделью+
B. способы манипулирования реляционными данными – реляционную алгебру и реляционное исчисление
C. ограничения целостности сущностей и целостности внешних ключей
D. структуру порядка или алгебры в любых реляционных базах данных
19. Домен – это
A. семантическое понятие
B. синтаксическое понятие
C. логическое понятие
D. статистическое понятие
20. Домен
A. несет определенную смысловую нагрузку+



- В.** определен на некотором ссылочном типе данных или на другом домене.
С. должен иметь некоторое логическое условие, позволяющее описать свойства атрибута.
Д. имеет составное имя (тип данных.имя атрибута).
21. Атрибут отношения есть
А. пара вида <Имя_атрибута : Имя_домена>+
В. пара вида <Имя_базы : Имя_поля>.
С. пара вида <Имя_поля : Тип_данных>.
Д. пара вида <Имя_базы: Имя_домена>.
22. Какие операторы реляционной алгебры не являются примитивными, т. е. их можно выразить через другие реляционные операторы?
А. Пересечение+
В. Разность
С. Объединение
Д. Декартово произведение
Е. Выборка
Ф. Проекция
Г. Соединение+
Н. Деление+
23. К простым типам данных относятся следующие типы:
А. логический, строковый, численный+
В. вещественный, структурированный, ссылочный.
С. вещественный, структурированный, перечислимый.
Д. вещественный, интервальный, ссылочный.
24. Чему равно значение выражения $A \text{ and } (B \text{ or } C)$ в трехзначной логике, если $A=T$, $B=T$, $C=U$?
А. T+
В. F
С. U
Д. значение не определено.
25. К теоретико-множественным операторам реляционной алгебры относятся:
А. Соединение
В. Декартово произведение+
С. Пересечение+
Д. Разность+

Проектирование баз данных.

26. Как называется связь, если каждому значению первичного ключа главной таблицы соответствует одна или ни одной записи в подчиненной таблице:
А) Многие-к-Одному
В) Один-ко-Многим
С) Один-к-Одному+
Д) Многие-ко-Многим
27. Как называется связь, если каждому значению первичного ключа главной таблицы соответствует одна, несколько или ни одной записи в подчиненной таблице:
А) Один-к-Двум
В) Один-ко-Многим+



- C) Один-к-Одному
D) Многие-ко-Многим
28. Реорганизация данных путем ликвидации избыточности и иных аномалий называется:
A) верификация данных
B) целостность данных
C) **нормализация+**
D) инкапсуляция
29. Какое определение относится к аномалии данных:
A) верификация
B) целостность
C) наследование
D) **избыточность+**
30. Концептуальный уровень представления данных:
A) определяет форматы размещения данных
B) описывает взаимосвязи между логическими записями
C) **определяет структуру базы данных и отношения между объектами+**
D) определяет значения данных
31. Форматы размещения данных определяет:
A) **физический уровень+**
B) логический уровень
C) концептуальный уровень
D) модель представления данных
32. Какая нормальная форма не имеет аномалий модификации данных:
A) **третья+**
B) вторая
C) первая
D) первая, вторая и третья
33. Какое из перечисленных свойств не является свойством реляционной базы?
A. Порядок следования строк в таблице произвольный
B. Каждый столбец имеет уникальное имя
C. **Несколько узлов уровня связаны с узлом одного уровня+**
D. Для каждой таблицы можно определить первичный ключ
34. Какая наименьшая единица хранения данных в БД?
A. хранимая запись
B. **хранимое поле**
C. хранимый байт
D. ничего из вышеперечисленного
E. хранимый файл
35. Какая связь между сущностями всегда требует подключения дополнительной реляционной таблицы?
A. **много-ко-многим**
B. один-ко-многим
C. много-к-одному
36. Для нормализованной базы данных характерно: Выберите один или несколько ответов:
A. **В одном поле таблицы не может храниться несколько значений+**
B. **Функциональная зависимость не ключевого атрибута от ключевого+**
C. Избыточность базы данных



D. Все неключевые атрибуты функционально независимы друг от друга+

37. Первый этап проектирования называется

A. Даталогическое моделирование

B. Онтологическое моделирование+

C. Инфологическое моделирование

D. Физическое моделирование

38. Второй этап проектирования называется

A. Даталогическое моделирование

B. Онтологическое моделирование

C. Инфологическое моделирование+

D. Физическое моделирование

39. Третий этап проектирования называется

A. Даталогическое моделирование+

B. Онтологическое моделирование

C. Инфологическое моделирование

D. Физическое моделирование

40. Четвертый этап проектирования называется

A. Даталогическое моделирование

B. Онтологическое моделирование

C. Инфологическое моделирование

D. Физическое моделирование+

41. Содержанием этапа инфологического проектирования является:

A. построение компьютерной модели предметной области

B. построение реляционной модели

C. построение информационной модели предметной области+

42. Верно ли утверждение «Цель инфологического моделирования обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных».

A. Да+

B. Нет

43. В состав инфологической модели входят:

A. определения сущностей+

B. уникальные идентификаторы сущностей+

C. отношения между сущностями+

D. диаграммы «сущность-связь» (Entity — Relation- ship Diagrams+

E. определения атрибутов сущностей+

F. супертипы и подтипы+

44. Какая связь между логической моделью базы данных и СУБД?

A. это не связанные понятия

B. логическая модель базы данных использует спецификации СУБД

C. СУБД отображает логическую модель базы данных в структуру хранения+

D. логическая модель базы данных описывает структуру хранения данных системой управления базами данных

45. Какой порядок действий при построении концептуальной модели?

A. определение сущностей, определение атрибутов, установление связей+

B. определение атрибутов, определение сущностей, установление связей

C. выбор связей, определение сущностей, определение атрибутов



- D.** выбор экземпляров сущностей, установление связей между экземплярами
46. Какие этапы проектирования базы данных не входят в первую стадию концептуального проектирования?
- A.** проектирование обобщенного концептуального представления+
- B.** выбор СУБД
- C.** проектирование концептуального представления, специфицированного к модели данных СУБД (логической модели)
- D.** проектирование представления данных в памяти компьютера (структур хранения)
47. Какой основной вопрос решает логическое проектирование базы данных?
- A.** Что хранить+
- B.** Где хранить
- C.** Как хранить
- D.** Как долго хранить
48. Сколько типов связей существует между сущностями?
- A.** 1
- B.** 2
- C.** 3+
- D.** 4
49. Укажите признак (признаки) второй нормальной формы
- A.** Таблица не должна иметь повторяющихся записей+
- B.** Любое не ключевое поле должно однозначно идентифицироваться полным набором ключевых полей+
- C.** Ни одно из не ключевых полей таблицы не должно идентифицироваться с помощью другого не ключевого поля
50. Как описывается сущность?
- A.** совокупностью атрибутов+
- B.** набором экземпляров
- C.** совокупностью объектов
- D.** записью об объекте

Язык запросов SQL. Манипулирование данными.

51. Какое ключевое слово используется для исключения дубликатов в выходном наборе:
- A)** **DISTINCT**+
- B)** DROP
- C)** DELETE
- D)** NULL
52. Какая инструкция применяется для изменения данных:
- A)** **UPDATE**+
- B)** SELECT
- C)** INSERT
- D)** DELETE
53. Какая инструкция применяется для удаления данных из таблицы:
- A)** **DELETE**+
- B)** NULL
- C)** DROP
- D)** UPDATE
54. Инструкция DELETE применяется для:
- A)** **удаления данных из таблицы**+



- В) добавления данных в таблицу
С) изменения данных в таблице
D) удаления таблицы
55. Какая инструкция не содержит ошибок:
A) **SELECT Фио, Группа FROM Студенты WHERE Группа like 'АС*';+**
B) DELETE Фио, Группа FROM Студенты WHERE Группа like 'АС*';
C) INSERT INTO (Фио, Группа) VALUES ('Петров', 'АС-1');
D) UPDATE Студенты SET Фио='Петров' WHERE Группа like 'АС*'
56. Какая инструкция содержит ошибки в синтаксисе:
A) SELECT Фио, Группа FROM Студенты WHERE Группа like 'АС*';
B) DELETE FROM Студенты WHERE Группа='АС-1';
C) **INSERT INTO Студенты(Фио, Группа) VALUE ('Петров', 'АС-1');+**
D) UPDATE Студенты SET Фио='Петров' WHERE Группа='АС-1';
57. DDL (Data Definition Language) означает:
A) **язык определения данных+**
B) язык управления (манипулирования) данными
C) язык управления пользовательскими привилегиями
D) операторы для определения курсора
58. Укажите неверное утверждение:
A) предложение GROUP BY группирует строки по определенному признаку
B) **предложение WHERE следует сразу за SELECT+**
C) предложение ORDER BY сортирует результаты запроса
D) в предложении FROM указывается список таблиц
59. В предложении SELECT можно указать:
A) имя таблицы
B) **имя столбца+**
C) имя пользователя
D) имя базы данных
60. Список таблиц, из которых извлекаются данные, указываются в предложении:
A) WHERE
B) ORDER
C) SELECT
D) **FROM+**
61. Какое условие содержит ошибки в синтаксисе:
A) WHERE Группа is not null;
B) **WHERE (Группа='АС*') and (ФИО='Петров');+**
C) WHERE ФИО in('Петров', 'Сидоров');
D) WHERE Группа not like 'АС*';
62. В предложении ORDER BY ФИО DESC, Группа указывается сортировка:
A) сначала по возрастанию по полю ФИО, затем по возрастанию по полю Группа
B) **сначала по убыванию по полю ФИО, затем по возрастанию по полю Группа+**
C) сначала по убыванию по полю Группа, затем по возрастанию по полю ФИО
D) сначала по убыванию по полю ФИО, затем по убыванию по полю Группа
63. Инструкция INSERT INTO:
A) удаляет данные из таблицы
B) добавляет данные в таблицу
C) **изменяет данные в таблице+**



- D) добавляет таблицу
64. Какое условие принадлежности диапазону неправильное:
- A) WHERE Цена between 100 and 200
B) WHERE (Оценка \geq 2) and (Оценка \leq 5)
C) WHERE (Тираж $<$ 3000) and (Тираж $>$ 2000)
D) WHERE (Дата $>$ 01.01.1982) and (Дата $<$ 31.12.1982)+
65. Агрегатная функция AVG в запросе вычисляет:
- A) **среднее значение+**
B) сумму значений
C) число записей
D) минимальное значение
66. За предложением SELECT обязательно следует:
- A) WHERE
B) ORDER BY
C) **FROM+**
D) GROUP BY
67. В предложении HAVING указывается:
- A) список таблиц
B) список полей
C) путь к базе данных
D) **критерий отбора+**
68. Где выполняется сортировка по убыванию по двум полям:
- A) **ORDER BY Группа DESC, Фио DESC+**
B) ORDER BY Фио ASC, Группа ASC
C) ORDER BY Фио ASC, Группа DESC
D) ORDER BY Фио DESC, Группа ASC
69. Как выбрать все записи из таблицы "Persons", для которых значение колонки "FirstName" начинается с "a"?
- A. SELECT * FROM Persons WHERE FirstName LIKE '%a'
B. SELECT * FROM Persons WHERE FirstName STARTSWITH 'a'
C. SELECT * FROM Persons WHERE FirstName = '%a%'
D. **SELECT * FROM Persons WHERE FirstName LIKE 'a%'+**
E. SELECT * FROM Persons WHERE FirstName = 'a'
70. С помощью какого запроса можно поместить значение 'Perez' в поле LastName таблицы Persons?
- A. INSERT INTO Persons VALUE ('Perez')
B. INSERT INTO Persons ('Perez') INTO LastName
C. INSERT ('Perez') INTO Persons (LastName)
D. **INSERT INTO Persons (LastName) VALUES ('Perez'+**
71. Какими запросами можно получить все поля и записи таблицы Employers Выберите один или несколько ответов:
- A. SELECT *.Employers
B. SELECT [all] FROM Employers
C. **SELECT * FROM Employers+**
D. SELECT Employers
72. Стандартные строковые функции SQL включают:
- A. **SUBSTRING+**



- B. MIDDLE**
- C. EXISTS**
- D. UPPER+**
- E. LOWER+**

73. Какие из запросов вернут все записи, не содержащие значение NULL в колонке value?

- A. SELECT * FROM table WHERE value != NULL
- B. SELECT * FROM table WHERE value <> NULL
- C. SELECT * FROM table WHERE value IS NOT NULL+**
- D. SELECT * FROM table WHERE value LIKE NOT NULL

74. С помощью каких операторов в выборке можно убрать дубликаты?

- A. DISTINCT+
- B. GROUP BY+
- C. ORDER BY
- D. DELETE

75. Какое ключевое слово используется для фильтрации значений, полученных в результате применения агрегирующих функций в результатах запроса с использованием GROUP BY?

- A. WHERE
- B. HAVING+**
- C. WHERE и HAVING
- D. Ни одно из перечисленных

Управление базами данных.

76. Обеспечение безопасности в СУБД достигается:

- A) шифрованием данных и заданием паролей+**
- B) заданием условий для значений хранимых данных
- C) ведением журнала изменений в БД
- D) использованием механизма транзакций

77. Какое свойство характерно для транзакции:

- A) атомарность+**
- B) целостность
- C) безопасность
- D) избыточность

78. Для ускорения обмена между внешней и оперативной памятью используются:

- A) транзакции
- B) буферы+**
- C) серверы
- D) языки манипулирования данными

79. Неделимая последовательность операций над данными называется:

- A) сущностью
- B) транзакцией+**
- C) целостностью
- D) атомарностью

80. Ведение журнала изменений в базе данных обеспечивает:

- A) целостность данных
- B) безопасность данных
- C) дополнительный контроль надежности хранения данных+**
- D) многопользовательский доступ

81. Инструкция CREATE TABLE применяется для:



- A) **создания таблицы+**
B) добавления данных в таблицу
C) изменения данных в таблице
D) удаления таблицы
82. Какая инструкция применяется для удаления таблицы:
A) **DROP TABLE+**
B) DELETE TABLE
C) DROP INDEX
D) DROP VIEW
83. Какая инструкция применяется для удаления представления:
A) **DROP VIEW+**
B) CREATE VIEW
C) DELETE VIEW
D) ALTER VIEW
84. Какая инструкция не относится к языку определения данных:
A) CREATE VIEW
B) DROP SCHEMA
C) **INSERT INTO+**
D) ALTER TABLE
85. Какой тип не применяется для описания целых чисел:
A) INT
B) TINYINT
C) **BIT+**
D) SMALLINT
86. Тип DECIMAL используется для описания:
A) **чисел с плавающей запятой без потери точности+**
B) вещественных чисел
C) целых чисел
D) двоичной информации
87. Для описания графических данных используется тип:
A) CHAR
B) FLOAT
C) REAL
D) **IMAGE+**
88. Какая инструкция применяется для управления привилегиями пользователей:
A) INSERT
B) **GRANT+**
C) COMMIT
D) DELETE
89. Какой тип данных используется для представления даты:
A) **DATE+**
B) CHAR
C) FLOAT
D) IMAGE
90. Какая часть инструкции описывает создание внешнего ключа:
A) CONSTRAINT PK1 UNIQUE
B) **CONSTRAINT PK1 PRIMARY KEY(ПолеA)**



C) **CONSTRAINT PK1 FOREIGN KEY(ПолеB) REFERENCES**таблица1(ПолеA)+

D) **CONSTRAINT PK1 NOT NULL** (ПолеA, ПолеB)

91. Какая инструкция создания первичного ключа неправильная:

A) ПолеA **INT CONSTRAINT PK1 PRIMARY KEY**

B) **CONSTRAINT PK1 PRIMARY KEY(ПолеA)**

C) **CONSTRAINT PRIMARY KEY(ПолеA, ПолеB)+**

D) **CONSTRAINT PK1 PRIMARY KEY(ПолеA, ПолеB)**

92. Какое ключевое слово не относится к созданию внешнего ключа:

A) **CONSTRAINT**

B) **FOREIGN KEY**

C) **PRIMARY KEY+**

D) **REFERENCES**

93. Обязательное поле в инструкции **CREATE TABLE** описывается как:

A) **NOT NULL+**

B) **NULL**

C) **DEFAULT NULL**

D) **IS NOT NULL**

94. Обязательное поле **ГРУППА** длиной не более 15 символов в инструкции создания таблицы

описывается как:

A) **ГРУППА text(15) not null+**

B) **ГРУППА char(15) is not null**

C) **ГРУППА text(15) null**

D) **ГРУППА text(15) is not null**

95. Какая запись описывает символьное поле:

A) **CHAR(n)+**

B) **DECIMAL(n)**

C) **NUMERIC(n)**

D) **FLOAT(n)**

96. **FLOAT(15)** – это описание:

A) **символьного поля длиной не более 15 символов**

B) **числового поля со значением не более 15**

C) **вещественного числа с точностью 15 знаков+**

D) **двоичного файла длиной 15 байт**

97. Какая инструкция относится к **DDL**:

A) **DROP+**

B) **INSERT**

C) **DELETE**

D) **SELECT**

98. Предположим, что Вам нужно удалить таблицу **SOME_TABLE**, имеющую несколько индексов, относящихся к ней. Что из перечисленного удалит все индексы вместе с таблицей?

A. **DROP INDEX FROM SOME_TABLE**

B. **DROP TABLE SOME_TABLE+**

C. **ALTER TABLE SOME_TABLE DROP CONSTRAINT**

D. **ALTER TABLE SOME_TABLE DROP PRIMARY KEY CASCADE**

99. Для чего нужны представления?

A. **Для упрощения логики программы с помощью создания наиболее часто используемых запросов как таблиц+**



- В.** Для более быстрого доступа к данным
 - С.** Поддержки целостности данных на декларативном уровне
 - Д.** Поддержки целостности данных на процедурном уровне
100. Какой тип данных может принимать только два значения TRUE и FALSE:
- А) логический+**
 - В)** текстовый
 - С)** тип даты и времени
 - Д)** тип OLE



4. Порядок проведения и критерии оценивания промежуточной аттестации

4.1. Порядок проведения промежуточной аттестации

Промежуточная аттестация проводится в форме зачета в 5 семестре и в форме экзамена в 6 семестре.

Зачет выставляется на основе балльно-рейтинговой системы по итогам текущего контроля.

Экзамен проводится в три этапа. На первом этапе студент выполняет тест закрытого вида. Продолжительность – 30 минут. На втором этапе студент отвечает на теоретический вопрос билета. Продолжительность – 10 минут. На третьем этапе студент выполняет открытое практическое задание.

4.2. Критерии оценивания промежуточной аттестации по видам оценочных средств

Критерии оценивания теоретического вопроса на экзамене

Максимальный балл за ответ на теоретический вопрос – 40 баллов.

Отлично/зачтено 31-40 баллов	Хорошо/зачтено 21-30 баллов	Удовлетворительно/ зачтено 11-20 баллов	Неудовлетворительно/ не зачтено 0-10 баллов
Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе материал различной литературы, правильно обосновывает принятое нестандартное решение, владеет разносторонними навыками и приемами выполнения практических задач	Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения, а также имеет достаточно полное представление о значимости знаний по дисциплине.	Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает сложности при выполнении практических работ и затрудняется связать теорию вопроса с практикой.	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, неуверенно отвечает, допускает серьезные ошибки, не имеет представлений по методике выполнения практической работы. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по данной дисциплине.



Критерии оценивания теста

Тест формируется в системе электронного обучения MOODLE. Максимальный балл за тест — 20 баллов.

Оценка	Отлично/ зачтено	Хорошо/ зачтено	Удовлетворительно/ зачтено	Неудовлетворительно/ не зачтено
Баллы	19-20 баллов	15-18 баллов	11-14 балл	0-10аллов

Критерии оценивания практического задания на экзамене

Максимальный балл за выполнение практического задания — 40 баллов.

Отлично/ зачтено/ 31-40 баллов	Хорошо/зачтено/ 21-30 баллов	Удовлетворительно/ зачтено/ 11-20 баллов	Неудовлетворительно/ не зачтено/ 0-10 баллов
Обучающийся способен на отлично использовать современные технологические методы и средства по проектированию, реализации и управлению реляционными базами данных, применять современные приложения для разработки, реализации и управления базами данных, выполняет практическое задание без ошибок, предлагает различные способы выполнения.	Обучающийся хорошо умеет использовать современные технологические методы и средства по проектированию, реализации и управлению реляционными базами данных, применять современные приложения для разработки, реализации и управления базами данных, выполняет практическое с незначительными ошибками и недочетами.	Обучающийся допускает фактические ошибки при выполнении практического задания, способен их исправить, выполняет задания типового уровня по образцу.	Обучающийся не справляется с выполнением практического задания.

Критерии оценивания комплекса лабораторных работ (1-18).

Выполнение упражнений, включенных в комплект лабораторных работ (1-18), оценивается автоматически на сайте sql-ex.ru.

Максимальное количество баллов, полученных при выполнении упражнений, 300. Переводим в 50-балльную шкалу.

Оценка	Отлично/ зачтено	Хорошо/ зачтено	Удовлетворительно /зачтено	Неудовлетворительно/ не зачтено
Баллы	300-270 баллов 50-45	269-210 баллов 44-35	209-150 баллов 34-25	149-0 баллов 24-0



Критерии оценивания проектного задания

Отлично/зачтено/ 50-45 баллов	Хорошо/ зачтено/ 44-35 баллов	Удовлетворительно/ зачтено/ 34-25 баллов	Неудовлетворительно/ незачтено/ 24-0 балла
<p>Обучающийся представил без ошибок и в полном объеме результаты следующих этапов проектного задания: формализованное описание предметной области (2); ER-диаграмму; логическую схему отношений (10); физическое проектирование БД (10); заданы ограничения декларативным и процедурным методами (3); запросы на ввод данных в таблицы (5); триггеры для обработки событий (5); запросы на вывод информации (5). Обучающийся продемонстрировал умение взаимодействовать с другими членами команды при проектировании и реализации реляционной базы данных (10)</p>	<p>Обучающийся представил, допустив ошибки и неточности, в полном объеме результаты следующих этапов проектного задания: формализованное описание предметной области (2); ER-диаграмму; логическую схему отношений (10); физическое проектирование БД (10); заданы ограничения декларативным и процедурным методами (3); запросы на ввод данных в таблицы (5); триггеры для обработки событий (5); запросы на вывод информации (5). Обучающийся продемонстрировал умение взаимодействовать с другими членами команды при проектировании и реализации реляционной базы данных (10)</p>	<p>Обучающийся представил, допустив ошибки и неточности, в неполном объеме результаты следующих этапов проектного задания: формализованное описание предметной области (2); ER-диаграмму; логическую схему отношений (10); физическое проектирование БД (10); заданы ограничения декларативным и процедурным методами (3); запросы на ввод данных в таблицы (5); триггеры для обработки событий (5); запросы на вывод информации (5). Обучающийся затруднялся взаимодействовать с другими членами команды при проектировании и реализации реляционной базы данных (10)</p>	<p>Обучающийся не знает основных положений вопроса, не ориентируется в основных понятиях и этапах проектирования БД, представляет запросы с ошибками. Обучающийся не продемонстрировал умения взаимодействовать с другими членами команды при проектировании и реализации реляционной базы данных.</p>



4.3. Результаты промежуточной аттестации и уровни сформированности компетенций

Зачет в 5 семестре выставляется на основе балльно-рейтинговой системы по итогам текущего контроля выполнения комплекса лабораторных работ (1-18).

0-75 баллов – не зачтено;

76-150 баллов – зачтено.

Схема подведения итогов промежуточной аттестации в форме экзамена.

Баллы, полученные за отдельные задания экзамена (тестирование, теоретический вопрос, практическое задание) суммируются.

Виды заданий	Баллы
Тест	0-20
Теоретический вопрос	0-40
Практическое задание	0-40

0-49 баллов – неудовлетворительно;

50-69 баллов - удовлетворительно;

70-89 баллов - хорошо;

90-100 баллов - отлично.

При подведении итогов промежуточной аттестации учитываются результаты текущей аттестации.

Виды заданий	Баллы
Лабораторный практикум (1-18)	0-50
Проектное задание	0-50

0-49 баллов - неудовлетворительно (2);

50-69 баллов - удовлетворительно (3);

70-89 баллов - хорошо (4);

90-100 баллов - отлично (5).

Полученные за текущую аттестацию баллы могут либо повысить, либо понизить отметку за экзамен за счет вычисления среднего балла за экзамен и текущую аттестацию.

0-49 баллов - неудовлетворительно (2);

50-69 баллов - удовлетворительно (3);

70-89 баллов - хорошо (4);

90-100 баллов - отлично (5).

Особенности проведения процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья обозначены в рабочей программе дисциплины (модуля).

Уровни сформированности компетенций определяется следующим образом:

1. Продвинутый уровень сформированности компетенций соответствует оценке «зачтено»/«отлично»:

Обучающийся владеет знаниями предмета в полном объеме учебной программы, достаточно глубоко осмысливает дисциплину; самостоятельно, в логической последовательности и исчерпывающе отвечает на все вопросы, подчеркивает при этом самое существенное, умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать



изученный материал, выделять в нем главное: устанавливать причинно-следственные связи; четко формирует ответы.

2. Базовый уровень соответствует оценке «зачтено»/«хорошо»:

Обучающийся владеет знаниями дисциплины почти в полном объеме программы (имеются пробелы знаний только в некоторых, особенно сложных разделах); самостоятельно и отчасти при наводящих вопросах дает полноценные ответы на вопросы; не всегда выделяет наиболее существенное, не допускает вместе с тем серьезных ошибок в ответах.

3. Пороговый уровень соответствует оценке «зачтено»/«удовлетворительно»:

Обучающийся владеет основным объемом знаний по дисциплине; проявляет затруднения в самостоятельных ответах, оперирует неточными формулировками; в процессе ответов допускает ошибки по существу вопросов.

4. Низкий уровень соответствует оценке «не зачтено»/«неудовлетворительно»:

Обучающийся не освоил обязательного минимума знаний предмета, не способен ответить на вопросы даже при дополнительных наводящих вопросах экзаменатора.

