

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Ефанов Алексей Валерьевич

Должность: Директор Невинномысского технологического института (филиал) СКФУ

Дата подписания: 10.10.2022 15:36:52

Уникальный программный ключ:

49214306dd433e7a1b0f8632f645f9d53c99e3d0

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение  
высшего образования**

**«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**УТВЕРЖДАЮ**

Директор НТИ (филиал) СКФУ

Ефанов А.В.

Ф.И.О.

«\_\_\_\_\_» \_\_\_\_\_ 2022 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

для проведения текущего контроля успеваемости и промежуточной аттестации по  
дисциплине

Управление данными

Направление подготовки	09.03.02 Информационные системы и технологии
Направленность (профиль)	Информационные системы и технологии в бизнесе
Форма обучения	Очная
Год начала обучения	2022
Реализуется в 7 семестре	

## Введение

1. Назначение: обеспечение методической основы для организации и проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине «Управление данными». Текущий контроль успеваемости и промежуточная аттестация по дисциплине – вид систематической проверки знаний, умений, навыков студентов. Задачами текущего контроля успеваемости и промежуточной аттестации являются получение первичной информацию о ходе и качестве освоения компетенций, а также стимулирование регулярной целенаправленной работы студентов. Для формирования определенного уровня компетенций.

2. ФОС является приложением к программе дисциплины «Управление данными» и в соответствии с образовательной программой высшего образования по направлению подготовки 09.03.02 Информационные системы и технологии.

3. Разработчик: Кочеров Ю. Н., доцент базовой кафедры Регионального индустриального парка, кандидат технических наук

4. Проведена экспертиза ФОС.

Члены экспертной группы:

Председатель:

Мельникова Е.Н. – председатель УМК НТИ (филиал) СКФУ

Члены комиссии:

А.И. Колдаев, и.о. зав. кафедрой информационных систем, электропривода и автоматике  
Э.Е. Тихонов, доцент базовой кафедры территории опережающего социально-экономического развития

Представитель организации-работодателя:

Горшков М. Г., директор ООО «Арнест-информационные технологии»

Экспертное заключение: фонд оценочных средств соответствует ОП ВО по направлению подготовки 09.03.02 Информационные системы и технологии и рекомендуется для оценивания уровня сформированности компетенций при проведении текущего контроля успеваемости и промежуточной аттестации студентов по дисциплине «Управление данными».

05 марта 2022 г.

5. Срок действия ФОС определяется сроком реализации образовательной программы.

## Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

Код оцениваемой компетенции, индикатора (ов)	Этап формирования компетенции (№ темы) (в соответствии с рабочей программой дисциплины)	Средства и технологии оценки	Вид контроля, аттестация (текущий/промежуточный)	Тип контроля (устный, письменный или использованном технических средств)	Наименование оценочного средства
ИД-1 ПК-5 ИД-2 ПК-5 ИД-3 ПК-5	1-11	Собеседование	Текущий	Устный	Вопросы для собеседования
ИД-1 ПК-5 ИД-2 ПК-5 ИД-3 ПК-5	1-11	Тестирование	Текущий	Устный	Паспорт фонда тестовых заданий

### 1. Описание показателей и критериев оценивания на различных этапах их формирования, описание шкал оценивания

Уровни сформированности компетенции(ий), индикатора (ов)	Дескрипторы			
	Минимальный уровень не достигнут (Неудовлетворительно) 2 балла	Минимальный уровень (удовлетворительно) 3 балла	Средний уровень (хорошо) 4 балла	Высокий уровень (отлично) 5 баллов
ПК-5 Способен выполнить проектирование и дизайн ИС:				
Результаты обучения по дисциплине (модулю): Индикатор: ИД-1 ПК-5 ИД-2 ПК-5	Не удовлетворительно понимает формирование системного базового представления, первичных знаний, умений и навыков обучающихся по основам построения систем управления базами данных (БД); Не удовлетворительно осуществляет формирование базы данных и модели	Слабо понимает формирование системного базового представления, первичных знаний, умений и навыков обучающихся по основам построения систем управления базами данных (БД); Слабо осуществляет	Понимает формирование системного базового представления, первичных знаний, умений и навыков обучающихся по основам построения систем управления базами	

	<p>данных; введение в реляционную модель данных;  Не удовлетворительно применяет реляционную алгебру;  инфологическое моделирование;</p>	<p>формирование базы данных и модели данных;  введение в реляционную модель данных;  Слабо применяет реляционную алгебру;  инфологическое моделирование;</p>	<p>данных (БД);  Осуществляет формирование базы данных и модели данных;  введение в реляционную модель данных;  Применяет реляционную алгебру;  инфологическое моделирование;</p>	
<p>Результаты обучения по дисциплине (модулю):  <i>Индикатор:</i>  ИД-3 ПК-5</p>				<p>Понимает получение представления о роли и месте баз данных в автоматизированных системах, о назначении и основных характеристиках различных систем управления базами данных, их функциональных возможностях;  получение базового уровня по программированию в современной СУБД;  Применяет нормализацию данных; язык SQL;  Осуществляет физическую организацию баз данных;  алгоритмы выполнения запросов;  ограничение прав доступа;</p>

				язык SQL. Язык хранимых модулей;
--	--	--	--	----------------------------------

### Описание шкалы оценивания

В рамках рейтинговой системы успеваемость студентов по каждой дисциплине оценивается в ходе текущего контроля и промежуточной аттестации.

### Текущий контроль

Рейтинговая оценка знаний студента (в случаях, предусмотренных нормативными актами СКФУ).

№ п/п	Вид деятельности студентов	Сроки выполнения	Количество баллов
5 семестр			
1	Собеседование по темам 1-5, Защита практических работ	8	25
2	Собеседование по теме 6-10, Защита практических работ	16	30
	Итого за 5 семестр:		55
	Итого:		55

Максимально возможный балл за весь текущий контроль устанавливается равным **55**. Текущее контрольное мероприятие считается сданным, если студент получил за него не менее 60% от установленного для этого контроля максимального балла. Рейтинговый балл, выставляемый студенту за текущее контрольное мероприятие, сданное студентом в установленные графиком контрольных мероприятий сроки, определяется следующим образом:

<i>Уровень выполнения контрольного задания</i>	<i>Рейтинговый балл (в % от максимального балла за контрольное задание)</i>
<i>Отличный</i>	<i>100</i>
<i>Хороший</i>	<i>80</i>
<i>Удовлетворительный</i>	<i>60</i>
<i>Неудовлетворительный</i>	<i>0</i>

### Промежуточная аттестация

Промежуточная аттестация в форме **зачета или зачета с оценкой**

Процедура зачета (зачета с оценкой) как отдельное контрольное мероприятие не проводится, оценивание знаний обучающегося происходит по результатам текущего контроля.

Зачет выставляется по результатам работы в семестре, при сдаче всех контрольных точек, предусмотренных текущим контролем успеваемости. Если по итогам семестра обучающийся имеет от 33 до 60 баллов, ему ставится отметка «зачтено». Обучающемуся, имеющему по итогам семестра менее 33 баллов, ставится отметка «не зачтено».

Количество баллов за зачет ( $S_{зач}$ ) при различных рейтинговых баллах

по дисциплине по результатам работы в семестре

Рейтинговый балл по дисциплине по результатам работы в семестре ( $R_{сем}$ )	Количество баллов за зачет ( $S_{зач}$ )
$50 \leq R_{сем} \leq 60$	40
$39 \leq R_{сем} < 50$	35
$33 \leq R_{сем} < 39$	27
$R_{сем} < 33$	0

При зачете с оценкой используется шкала пересчета рейтингового балла по дисциплине в оценку по 5-балльной системе

*Шкала пересчета рейтингового балла по дисциплине в оценку по 5-балльной системе*

Рейтинговый балл по дисциплине	Оценка по 5-балльной системе
88 – 100	Отлично
72 – 87	Хорошо
53 – 71	Удовлетворительно
< 53	Неудовлетворительно

## 2. Типовые контрольные задания и иные материалы, характеризующие этапы формирования компетенций

### Вопросы для собеседования

#### Пороговый уровень

##### Базы данных и модели данных

1. Иерархическая и сетевая модель базы данных
2. Объектно-ориентированные СУБД

##### Введение в реляционную модель данных

1. Реляционная модель данных

##### Реляционная алгебра

1. Понятие реляционной алгебры
2. Замкнутость в реляционной алгебре

##### Инфологическое моделирование

1. Модель "сущность—связь"

##### Нормализация данных

1. Декомпозиция отношений
2. Проблема дублирования данных
3. Висячие записи
4. Проблемы, возникающие из-за неудачной структуры данных
5. Пятая нормальная форма (5НФ)
6. Функциональная зависимость
7. Связь между функциональной зависимостью и полной декомпозицией отношения
8. Первая нормальная форма (1НФ)

##### Язык SQL

1. История SQL
2. Структура SQL
3. Язык запросов
4. Простые запросы на выборку данных
5. Агрегатные (групповые) функции

##### Физическая организация баз данных

1. Организация размещения данных

2. Представление данных

#### **Алгоритмы выполнения запросов**

1. Физические операторы доступа
2. Компиляция запросов

#### **Транзакции**

1. Свойства транзакций
2. Параллельное выполнение транзакций
3. Уровни изолированности

#### **Язык SQL. Ограничение прав доступа**

1. Управление пользователями базы данных
2. Управление пользователями базы данных

#### **Язык SQL. Язык хранимых модулей**

1. Преимущества хранимых процедур

### **Повышенный уровень**

#### **Базы данных и модели данных**

1. Объектно-реляционные субд

#### **Введение в реляционную модель данных**

2. Целостность сущности и ссылок

#### **Реляционная алгебра**

1. Традиционные операции над множествами
2. Свойства основных операций реляционной алгебры

#### **Инфологическое моделирование**

1. Переход к реляционной модели данных

#### **Нормализация данных**

1. Вторая нормальная форма (2НФ)
2. Третья нормальная форма (3НФ)
3. Нормальная форма Бойса-Кодда (НФБК)
4. Многозначная зависимость
5. Четвертая нормальная форма (4НФ)
6. Пример проектирования на принципах нормализации
7. Нормализация и инфологическая модель

#### **Язык SQL**

1. Вложенные запросы
2. Внешние соединения
3. Изменение данных
4. Операции реляционной алгебры и SQL
5. Представления
6. Определение схемы БД и ограничений целостности

#### **Физическая организация баз данных**

1. Организация индексов

#### **Алгоритмы выполнения запросов**

1. Примеры из практики

#### **Транзакции**

1. Тупики
2. Надежное хранение данных

#### **Язык SQL. Ограничение прав доступа**

1. Режимы аутентификации
2. Администрирование системы безопасности

#### **Язык SQL. Язык хранимых модулей**

1. Типы хранимых процедур

## Компетентностно-ориентированные задания

1. Используя MS SQL Server создать файл базы данных, с помощью SQL-команды
2. Показать, как создаются таблицы в MS SQL Server
3. Показать, как создаются поля в MS SQL Server
4. Показать, как создаются ограничения внешнего ключа в MS SQL Server
5. Показать, как создается диаграмма баз данных в MS SQL Server
6. Показать как создаются запросы на выборку

### 1. Критерии оценивания компетенций\*

Оценка «отлично» выставляется студенту, если он

Понимает получение представления о роли и месте баз данных в автоматизированных системах, о назначении и основных характеристиках различных систем управления базами данных, их функциональных возможностях; получение базового уровня по программированию в современной СУБД;

Применяет нормализацию данных; язык SQL;

Осуществляет физическую организацию баз данных; алгоритмы выполнения запросов; ограничение прав доступа; язык SQL. Язык хранимых модулей

Оценка «хорошо» выставляется студенту, если он

Понимает формирование системного базового представления, первичных знаний, умений и навыков обучающихся по основам построения систем управления базами данных (БД);

Осуществляет формирование базы данных и модели данных; введение в реляционную модель данных;

Применяет реляционную алгебру; инфологическое моделирование

Оценка «удовлетворительно» выставляется студенту, если он

Слабо понимает формирование системного базового представления, первичных знаний, умений и навыков обучающихся по основам построения систем управления базами данных (БД);

Слабо осуществляет формирование базы данных и модели данных; введение в реляционную модель данных;

Слабо применяет реляционную алгебру; инфологическое моделирование;

Оценка «неудовлетворительно» выставляется студенту, если он

Не удовлетворительно понимает формирование системного базового представления, первичных знаний, умений и навыков обучающихся по основам построения систем управления базами данных (БД);

Не удовлетворительно осуществляет формирование базы данных и модели данных; введение в реляционную модель данных;

Не удовлетворительно применяет реляционную алгебру; инфологическое моделирование

\* в соответствии с результатами освоения дисциплины

### 2. Описание шкалы оценивания

Максимально возможный балл за весь текущий контроль устанавливается равным 55. Текущее контрольное мероприятие считается сданным, если студент получил за него не менее 60% от установленного для этого контроля максимального балла. Рейтинговый балл, выставляемый студенту за текущее контрольное мероприятие, сданное студентом в установленные графиком контрольных мероприятий сроки, определяется следующим образом:

Уровень выполнения контрольного задания	Рейтинговый балл (в % от максимального балла за контрольное задание)
Отличный	100
Хороший	80
Удовлетворительный	60
Неудовлетворительный	0

### 3. Методические материалы, определяющие процедуры оценивания знаний,

**умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

Пример:

Процедура проведения данного оценочного мероприятия включает в себя: защиту практических занятий

Предлагаемые студенту вопросы позволяют проверить ИД-1 ПК-5, ИД-2 ПК-5, ИД-3 ПК-5 компетенции

Для подготовки к данному оценочному мероприятию необходимо 10 минут.

При подготовке к ответу студенту предоставляется право пользования отчетами о выполненные практические занятия.

При проверке задания, оцениваются последовательность и логика ответа

**Оценочный лист**

№ п/п	Фамилия, имя студента	Вид работы						Итог
		Соответствие ответа заданию	Раскрытие проблемы, темы	Ясность, четкость, логичность, научность изложения	Обоснованность излагаемой позиции, ответа	Самостоятельность в формулировке позиции	Четкость, обоснованность, научность выводов	

**Паспорт фонда тестовых заданий по дисциплине управление данными**

№ п/п	Тест	Ключ
1.	<p>Что такое реляционные базы данных:</p> <ul style="list-style-type: none"> <li>База данных, в которой информация хранится в виде двумерных таблиц, связанных между собой</li> <li>База данных, в которой одна ни с чем не связанная таблица</li> <li>Любая база данных - реляционная</li> <li>Совокупность данных, не связанных между собой</li> </ul>	База данных, в которой информация хранится в виде двумерных таблиц, связанных между собой
2.	<p>Как выглядит запрос, для вывода ВСЕХ значений из таблицы Orders:</p> <ul style="list-style-type: none"> <li>select ALL from Orders;</li> <li>select % from Orders;</li> <li>select * from Orders;</li> <li>select *.Orders from Orders;</li> </ul>	select * from Orders;
3.	<p>Какие данные мы получим из этого запроса?</p> <ul style="list-style-type: none"> <li>select id, date, customer_name from Orders;</li> <li>Неотсортированные номера и даты всех заказов с именами заказчиков</li> <li>Никакие, запрос составлен неверно</li> </ul>	Неотсортированные номера и даты всех заказов с именами заказчиков

	<ul style="list-style-type: none"> <li>• Номера и даты всех заказов с именами заказчиков, отсортированные по первой колонке</li> <li>• Номера и даты всех заказов с именами заказчиков, отсортированные по всем колонкам, содержащим слово Order</li> </ul>	
4.	<p>Есть ли ошибка в запросе? <code>select id, date, customer_name from Orders where customer_name = Mike;</code></p> <ul style="list-style-type: none"> <li>• Запрос составлен правильно</li> <li>• Mike необходимо записать в кавычках 'Mike'</li> <li>• Нужно убрать лишние поля из запроса</li> <li>• Строчку с where поменять местами с from</li> </ul>	Mike необходимо записать в кавычках 'Mike'
5.	<p>Что покажет следующий запрос: <code>select * from Orders where date between '2017-01-01' and '2017-12-31'</code></p> <ul style="list-style-type: none"> <li>• Все данные по заказам, совершенным за 2017 год, за исключением 01 января 2017 года</li> <li>• Все данные по заказам, совершенным за 2017 год, за исключением 31 декабря 2017 года</li> <li>• Все данные по заказам, совершенным за 2017 год</li> <li>• Ничего, запрос составлен неверно</li> </ul>	Все данные по заказам, совершенным за 2017 год
6.	<p>Что не так с этим запросом <code>select id, date from Orders where seller_id = NULL;</code></p> <ul style="list-style-type: none"> <li>• Все верно, запрос покажет все заказы, продавцы которых не проставлены</li> <li>• NULL нужно взять в кавычки</li> <li>• Сравнение с NULL можно проводить только с оператором IS</li> <li>• Сравнение с NULL можно проводить только с оператором ON</li> </ul>	Сравнение с NULL можно проводить только с оператором IS
7.	<p>Порядок выполнения операторов AND и OR следующий:</p> <p>Сначала выполняется AND, а затем OR</p> <p>Сначала выполняется OR, а затем AND</p> <p>Порядок выполнения операторов AND и OR зависит от того, какой операторов стоит первым</p> <p>Операторы AND и OR выполняются одновременно</p>	Сначала выполняется AND, а затем OR
8.	<p>Что покажет следующий запрос: <code>select DISTINCT seller_id order by seller_id from Orders;</code></p> <ul style="list-style-type: none"> <li>• Уникальные ID продавцов, отсортированные по возрастанию</li> <li>• Уникальные ID продавцов, отсортированные по убыванию</li> <li>• Ничего, запрос составлен неверно, ORDER BY всегда ставится в конце запроса</li> <li>• Неотсортированные никак уникальные ID продавцов</li> </ul>	Ничего, запрос составлен неверно, ORDER BY всегда ставится в конце запроса
9.	<p>Что делает спецсимвол '_' в паре с оператором LIKE:</p> <p><code>select * from Orders where customer_name like 'mik_';</code></p>	найдет все имена, которые начинаются на mik и состоят из 4 символов

	<ul style="list-style-type: none"> <li>• найдет все имена, которые начинаются на <code>miк</code> и состоят из 4 символов</li> <li>• найдет все имена, которые начинаются на <code>miк</code>, вне зависимости от того, из какого количества символов они состоят</li> <li>• найдет данные, где имя равно <code>miк</code></li> <li>• запрос составлен неверно, в паре с оператором <code>like</code> не используются спецсимволы</li> </ul>	
10.	<p>Выберите корректный пример использования функции <code>CONCAT</code>:</p> <ul style="list-style-type: none"> <li>• <code>select concat = index and city from Orders;</code></li> <li>• <code>select concat IN (`index`, `city`) from Orders;</code></li> <li>• <code>select concat(`index`, " ", `city`) from Orders;</code></li> <li>• нет правильного примера</li> </ul>	<pre>select concat(`index`, " ", `city`) from Orders;</pre>
11.	<p>Что покажет следующий запрос: <code>select concat(`index`,` `), `city`) AS delivery_address from Orders;</code></p> <ul style="list-style-type: none"> <li>• ничего, запрос составлен неверно</li> <li>• покажет уникальные значения индексов и адресов из таблицы <code>Orders</code></li> <li>• соединит поля с индексом и адресом из таблицы <code>Orders</code> и покажет их с псевдонимом <code>delivery_address</code></li> <li>• соединит поля с индексом и адресом из таблицы <code>Orders</code>, но покажет их без псевдонима</li> </ul>	соединит поля с индексом и адресом из таблицы <code>Orders</code> и покажет их с псевдонимом <code>delivery_address</code>
12.	<p>Выберите правильный пример использования функции округления <code>ROUND</code></p> <ul style="list-style-type: none"> <li>• <code>select id, price * discount AS total price from Orders ROUND (2);</code></li> <li>• <code>select id, price * discount ROUND (2) AS total price from Orders;</code></li> <li>• <code>select id, ROUND (price * discount, 2) AS total price from Orders;</code></li> <li>• нет правильного примера</li> </ul>	<pre>select id, ROUND (price * discount, 2) AS total price from Orders;</pre>
13.	<p>Что покажет следующий запрос: <code>select id from Orders where year (date) &gt; 2018;</code></p> <p>номера заказов, сделанных до 2018 года  номера заказов, сделанных в 2018 году  уникальные номера заказов  номера заказов, сделанных после 2018 года</p>	номера заказов, сделанных после 2018 года
14.	<p>Для чего используется <code>LIMIT</code>: <code>select * from Orders limit 10;</code></p> <ul style="list-style-type: none"> <li>• необходимо, чтобы показать все заказы, содержащие цифру 10</li> <li>• необходимо, чтобы показать первых 10 записей в запросе</li> <li>• необходимо, чтобы показать рандомные 10 записей в запросе</li> <li>• не существует такого оператора</li> </ul>	необходимо, чтобы показать первых 10 записей в запросе
15.	Что такое агрегирующие функции:	

	<ul style="list-style-type: none"> <li>• функции, которые фильтруют значения</li> <li>• функции, которые сортируют значения</li> <li>• функции, которые работают с набором данных, превращая их в одно итоговое значение</li> <li>• функции, которые суммируют все значения</li> </ul>	функции, которые работают с набором данных, превращая их в одно итоговое значение
16.	<p>Выберите пример правильно составленного запроса с использованием агрегирующей функции SUM:</p> <ul style="list-style-type: none"> <li>• <code>select sum(price) from Orders;</code></li> <li>• <code>select sum(price), customer_name from Orders;</code></li> <li>• <code>select * from Orders where price=sum();</code></li> <li>• <code>select sum() from Orders group by price desc;</code></li> </ul>	<code>select sum(price) from Orders;</code>
17.	<p>Возможно ли использование одновременно двух агрегирующих функций: <code>select min(price), max(price) from Orders;</code></p> <ul style="list-style-type: none"> <li>• да, но данный запрос составлен неверно, надо так: <code>select * from Orders where price IN (min, max);</code></li> <li>• да, в результате мы получим минимальную и максимальную стоимости</li> <li>• да, в результате мы получим стоимости, отсортированные от минимальной к максимальной</li> <li>• нет, две функции использовать одновременно нельзя</li> </ul>	да, в результате мы получим минимальную и максимальную стоимости
18.	<p>Выберите корректно составленный запрос с функцией GROUP BY:</p> <ul style="list-style-type: none"> <li>• <code>select count(*) from Orders GROUP seller_id;</code></li> <li>• <code>select seller_id, count(*) from Orders GROUP seller_id;</code></li> <li>• <code>select seller_id, count(*) from Orders GROUP BY seller_id;</code></li> <li>• <code>select count(*) from Orders GROUP ON seller_id;</code></li> </ul>	<code>select seller_id, count(*) from Orders GROUP BY seller_id;</code>
19.	<p>Что покажет следующий запрос: <code>select seller_id, count(*) from Orders GROUP BY seller_id HAVING seller_id IN (2,4,6);</code></p> <ul style="list-style-type: none"> <li>• количество заказов сгруппированное по продавцам 2, 4 и 6</li> <li>• количество продавцов, у которых 2, 4 или 6 товаров</li> <li>• ничего, запрос составлен неверно, HAVING указывается до группировки</li> <li>• ничего, запрос составлен неверно, для указания условия должно быть использовано WHERE</li> </ul>	количество заказов сгруппированное по продавцам 2, 4 и 6
20.	<p>Выберите пример корректно написанного запроса с использованием подзапроса, который выводит информацию о заказе с самой дорогой стоимостью:</p> <ul style="list-style-type: none"> <li>• <code>select * from Orders where price = (select big(price) from Orders)</code></li> <li>• <code>select * from Orders where price = max</code></li> <li>• <code>select count(*) from Orders</code></li> </ul>	<code>select * from Orders where price = (select max(price) from Orders)</code>

	<ul style="list-style-type: none"> <li>• <code>select * from Orders where price = (select max(price) from Orders)</code></li> </ul>	
21.	<p>Что такое JOIN:</p> <ul style="list-style-type: none"> <li>• операция объединения</li> <li>• операция группировки</li> <li>• операция суммирования</li> <li>• операция создания</li> </ul>	операция объединения
22.	<p>Какого из перечисленных ниже видов JOIN на самом деле не существует:</p> <ul style="list-style-type: none"> <li>• LEFT JOIN - который выведет все записи первой таблицы, а для ненайденных пар из правой таблицы проставит значение NULL</li> <li>• RIGHT JOIN - который выведет все записи второй таблицы, а на место недостающей информации из первой таблицы проставит NULL</li> <li>• INNER JOIN - который показывает только те записи, для которых нашлись пары</li> <li>• TRUE JOIN - который выведет все верные значения</li> </ul>	TRUE JOIN - который выведет все верные значения
23.	<p>Выберите корректный пример составленного запроса с использованием JOIN. Данный запрос выведет нам данные ID заказа, имя заказчика и продавца:</p> <ul style="list-style-type: none"> <li>• <code>select Orders.id, Orders.customer_name, Sellers.id from Orders LEFT JOIN ON Sellers AND Orders.seller_id = Sellers.id;</code></li> <li>• <code>select id AND customer_name AND seller_id from Orders LEFT JOIN Sellers ON seller_id = id;</code></li> <li>• <code>select Orders.id, Orders.customer_name, Sellers.id from Orders LEFT JOIN Sellers ON Orders.seller_id = Sellers.id;</code></li> <li>• <code>select Orders.id, Orders.customer_name, Sellers.id from Orders JOIN Sellers WHEN Orders.seller_id = Sellers.id;</code></li> </ul>	<pre>select Orders.id, Orders.customer_name, Sellers.id from Orders LEFT JOIN Sellers ON Orders.seller_id = Sellers.id;</pre>
24.	<p>Выберите правильный пример запроса с использованием UNION:</p> <ul style="list-style-type: none"> <li>• <code>select id, city from Orders order by id union select id, city from Sellers order by city;</code></li> <li>• <code>select id, city, seller_id from Orders and select city, id from Sellers order by id;</code></li> <li>• <code>select id, city from Orders union select id, city from Sellers order by id;</code></li> <li>• Все запросы верные</li> </ul>	<pre>select id, city from Orders union select id, city from Sellers order by id;</pre>
25.	<p>Какого строкового типа данных нет в SQL:</p> <ul style="list-style-type: none"> <li>• VARCHAR</li> <li>• STRING</li> <li>• CHAR</li> <li>• TEXT</li> </ul>	STRING
26.	<p>Чем отличается CHAR и VARCHAR?</p> <ul style="list-style-type: none"> <li>• Это одно и то же</li> <li>• VARCHAR не существует</li> </ul>	CHAR дополняет строку пробелами до максимальной длины, а VARCHAR тратит

	<ul style="list-style-type: none"> <li>• CHAR - это тип данных, а VARCHAR - подтип</li> <li>• CHAR дополняет строку пробелами до максимальной длины, а VARCHAR тратит лишнюю память на хранение значения длины строки</li> </ul>	лишнюю память на хранение значения длины строки
27.	<p>Как получить значение текущего года в SQL?</p> <ul style="list-style-type: none"> <li>• select now();</li> <li>• select year();</li> <li>• select year(now());</li> <li>• select year from Date;</li> </ul>	select year(now());
28.	<p>Как правильно добавить строку в таблицу? Какой запрос верный?</p> <ul style="list-style-type: none"> <li>• INSERT INTO `SimpleTable` (`some_text`) VALUES ("my text");</li> <li>• INSERT INTO `SimpleTable` SET `some_text`="my text";</li> <li>• SET INTO `SimpleTable` VALUE `some_text`="my text";</li> <li>• UPDATE INTO `SimpleTable` SET `some_text`="my text";</li> </ul>	INSERT INTO `SimpleTable` (`some_text`) VALUES ("my text");
29.	<p>Какие поля из таблицы обязательно перечислять в INSERT для вставки данных?</p> <p>Конечно все</p> <p>Только те, у которых нет DEFAULT значения</p> <p>Те, у которых нет DEFAULT значения и которые не имеют атрибут auto_increment</p> <p>Все поля имеют негласное DEFAULT значения, обязательных полей в SQL нет</p>	Те, у которых нет DEFAULT значения и которые не имеют атрибут auto_increment
30.	<p>Как сделать несколько записей в таблицу за один запрос?</p> <ul style="list-style-type: none"> <li>• Использовать MULTI INSERT INTO вместо INSERT INTO</li> <li>• Использовать подзапрос</li> <li>• Перечислить через запятую все наборы значений после VALUES</li> <li>• Никак, расхордимся по домам</li> </ul>	Перечислить через запятую все наборы значений после VALUES
31.	<p>Зачем существует команда UPDATE, если можно сначала удалить запись, а потом добавить новую, исправленную.</p> <ul style="list-style-type: none"> <li>• Именно так и делаю, UPDATE не использую</li> <li>• Так меньше нагрузки на базу, ведь команда одна, а не две</li> <li>• Потому что в записи могут быть автоматически предоставляемые поля, такие как auto_increment или timestamp, которые собьются при внесении записи заново</li> <li>• Как раз удалять записи в SQL нельзя, вместо этого используется UPDATE с NULL-значениями для всех полей</li> </ul>	Потому что в записи могут быть автоматически предоставляемые поля, такие как auto_increment или timestamp, которые собьются при внесении записи заново
32.	<p>В каких командах можно использовать LIMIT?</p> <ul style="list-style-type: none"> <li>• Только Select</li> <li>• Select и Insert</li> </ul>	Select, Update, Delete

	<ul style="list-style-type: none"> <li>• Select, Update, Delete</li> <li>• Select, Insert, Delete, Update</li> </ul>	
33.	<p>Как можно заранее узнать, какие записи будут удалены при выполнении DELETE?</p> <ul style="list-style-type: none"> <li>• Зачем заранее, просто вызвать его и посмотреть какие записи пропали</li> <li>• Заменить DELETE на SELECT *, ведь в остальном синтаксис DELETE похож на синтаксис простого SELECT</li> <li>• Сделать DELETE с LIMIT 1, одну запись не жалко</li> <li>• SQL создан для хранения данных, их нельзя удалять</li> </ul>	Заменить DELETE на SELECT *, ведь в остальном синтаксис DELETE похож на синтаксис простого SELECT
34.	<p>Какой командой можно создать новую таблицу?</p> <ul style="list-style-type: none"> <li>• CREATE TABLE</li> <li>• MAKE TABLE</li> <li>• SET TABLE</li> <li>• Создавать таблицы можно только через интерфейс СУБД, специальной SQL команды для этого нет</li> </ul>	CREATE TABLE
35.	<p>Можно ли поменять тип данных поля в уже существующей таблице?</p> <ul style="list-style-type: none"> <li>• Да, при помощи команды ALTER</li> <li>• Да, достаточно сделать INSERT с новым типом данных</li> <li>• Нет, только пересоздать таблицу</li> <li>• Тип бывает только у таблицы, а не у поля таблицы</li> </ul>	Да, при помощи команды ALTER

### 1. Критерии оценивания компетенций\*

Оценка «отлично» выставляется студенту, если он

Понимает получение представления о роли и месте баз данных в автоматизированных системах, о назначении и основных характеристиках различных систем управления базами данных, их функциональных возможностях; получение базового уровня по программированию в современной СУБД;

Применяет нормализацию данных; язык SQL;

Осуществляет физическую организацию баз данных; алгоритмы выполнения запросов; ограничение прав доступа; язык SQL. Язык хранимых модулей

Оценка «хорошо» выставляется студенту, если он

Понимает формирование системного базового представления, первичных знаний, умений и навыков обучающихся по основам построения систем управления базами данных (БД);

Осуществляет формирование базы данных и модели данных; введение в реляционную модель данных;

Применяет реляционную алгебру; инфологическое моделирование

Оценка «удовлетворительно» выставляется студенту, если он

Слабо понимает формирование системного базового представления, первичных знаний, умений и навыков обучающихся по основам построения систем управления базами данных (БД);

Слабо осуществляет формирование базы данных и модели данных; введение в реляционную модель данных;

Слабо применяет реляционную алгебру; инфологическое моделирование;

Оценка «неудовлетворительно» выставляется студенту, если он

Не удовлетворительно понимает формирование системного базового представления, первичных знаний, умений и навыков обучающихся по основам построения систем управления

базами данных (БД);

Не удовлетворительно осуществляет формирование базы данных и модели данных;  
введение в реляционную модель данных;

Не удовлетворительно применяет реляционную алгебру; инфологическое моделирование