

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Ефанов Алексей Валерьевич

Должность: Директор Невиномысского технологического института (филиал) СКФУ

Дата подписания: 12.10.2021 09:19:00


Уникальный программный идентификатор:
49214306dd433e7a1b0f8632f645f9d53c99e3d0

Министерство науки и высшего образования российской федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

И. о. зав. кафедрой ИСЭиА

 Колдаев А.И.
«15» 03 2021 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

для проведения текущего контроля успеваемости и промежуточной
аттестации по дисциплине:

Управление данными

(ЭЛЕКТРОННЫЙ ДОКУМЕНТ)

Направление подготовки 09.03.02 Информационные системы и технологии

Профиль Информационные системы и технологии в бизнесе

Квалификация выпускника бакалавр

Форма обучения заочная

Год начала обучения 2021

Изучается на 4 курсе зимняя сессия

Предисловие

1. Назначение: для проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине «Управление данными» для студентов направления подготовки 09.03.02 Информационные системы и технологии.
2. Фонд оценочных средств текущего контроля успеваемости и промежуточной аттестации на основе рабочей программы дисциплины «Управление данными» в соответствии с образовательной программой по направлению подготовки 09.03.02 Информационные системы и технологии, утвержденной на заседании Учёного совета НТИ (филиал) СКФУ.
3. Разработчик: Кочеров Ю. Н. канд., техн., наук, доцент базовой кафедры регионального индустриального парка
4. ФОС рассмотрен и утвержден на заседании базовой кафедры регионального индустриального парка.
5. ФОС согласован с выпускающей кафедрой информационных систем, электропривода и автоматики.
6. Проведена экспертиза ФОС. Члены экспертной группы, проводившие внутреннюю экспертизу:

Председатель: Кузьменко В.В., и.о. директора НТИ (филиал) СКФУ, профессор кафедры гуманитарных и математических дисциплин

Члены экспертной группы:

Должикова М.В. – заместитель директора по учебно-воспитательной работе НТИ (филиал) СКФУ;

Колдаев А.И. – доцент кафедры информационных систем, электропривода и автоматики.

Эксперт, проводивший внешнюю экспертизу:

Остапенко Н.А., – кандидат технических наук, ведущий инженер-конструктор КБ модернизации ООО КИЭП «Энергомера» филиал АО «Электротехнические заводы «Энергомера»

7. Экспертное заключение: фонд оценочных средств отвечает основным требованиям федерального государственного образовательного стандарта и способствует формированию требуемых компетенций.

Срок действия ФОС: на срок реализации образовательной программы.

Паспорт фонда оценочных средств
для проведения текущего контроля успеваемости и промежуточной
аттестации по дисциплине:
Управление данными

Направление подготовки 09.03.02 Информационные системы и технологии
Профиль Информационные системы и технологии в бизнесе
Квалификация выпускника бакалавр
Форма обучения заочная
Год начала обучения 2021
Изучается на 4 курсе зимняя сессия

Код оцениваемой компетенции	Этап формирования компетенции (№ темы)	Средства и технологии оценки	Вид контроля, аттестация	Тип контроля	Наименование оценочного средства
ПК-3	Тема 1-11	Собеседование	Устный	Текущий	Вопросы для собеседования

Составитель Кочеров Ю.Н.

Министерство науки и высшего образования российской федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

**Вопросы для собеседования
по дисциплине Управление данными
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

Министерство науки и высшего образования российской федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

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

ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий

№ п/п	Тест	Ключ
1.	<p>Что такое реляционные базы данных:</p> <ul style="list-style-type: none"> • База данных, в которой информация хранится в виде двумерных таблиц, связанных между собой • База данных, в которой одна ни с чем не связанная таблица • Любая база данных - реляционная • Совокупность данных, не связанных между собой 	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"> • select ALL from Orders; • select % from Orders; • select * from Orders; • select *.Orders from Orders; 	select * from Orders;
3.	<p>Какие данные мы получим из этого запроса?</p> <ul style="list-style-type: none"> • select id, date, customer_name from Orders; • Неотсортированные номера и даты всех заказов с именами заказчиков • Никакие, запрос составлен неверно • Номера и даты всех заказов с именами заказчиков, отсортированные по первой колонке • Номера и даты всех заказов с именами заказчиков, отсортированные по всем колонкам, содержащим слово Order 	Неотсортированные номера и даты всех заказов с именами заказчиков
4.	<p>Есть ли ошибка в запросе? select id, date, customer_name from Orders where customer_name = Mike;</p> <ul style="list-style-type: none"> • Запрос составлен правильно • Mike необходимо записать в кавычках 'Mike' • Нужно убрать лишние поля из запроса • Строчку с where поменять местами с from 	Mike необходимо записать в кавычках 'Mike'
5.	<p>Что покажет следующий запрос: select * from Orders where date between '2017-01-01' and '2017-12-31'</p> <ul style="list-style-type: none"> • Все данные по заказам, совершенным за 2017 год, за исключением 01 января 2017 года 	Все данные по заказам, совершенным за 2017 год

	<ul style="list-style-type: none"> • Все данные по заказам, совершенным за 2017 год, за исключением 31 декабря 2017 года • Все данные по заказам, совершенным за 2017 год • Ничего, запрос составлен неверно 	
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"> • Все верно, запрос покажет все заказы, продавцы которых не проставлены • NULL нужно взять в кавычки • Сравнение с NULL можно проводить только с оператором IS • Сравнение с NULL можно проводить только с оператором ON 	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 оператор 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"> • Уникальные ID продавцов, отсортированные по возрастанию • Уникальные ID продавцов, отсортированные по убыванию • Ничего, запрос составлен неверно, ORDER BY всегда ставится в конце запроса • Неотсортированные никак уникальные ID продавцов 	Ничего, запрос составлен неверно, ORDER BY всегда ставится в конце запроса
9.	<p>Что делает спецсимвол '_' в паре с оператором LIKE: <code>select * from Orders where customer_name like 'mik_';</code></p> <ul style="list-style-type: none"> • найдет все имена, которые начинаются на mik и состоят из 4 символов • найдет все имена, которые начинаются на mik, вне зависимости от того, из какого количества символов они состоят • найдет данные, где имя равно mik • запрос составлен неверно, в паре с оператором like не используются спецсимволы 	найдет все имена, которые начинаются на mik и состоят из 4 символов
10.	<p>Выберите корректный пример использования функции CONCAT:</p> <ul style="list-style-type: none"> • <code>select concat = index and city from Orders;</code> • <code>select concat IN ('index', 'city') from Orders;</code> • <code>select concat('index', " ", 'city') from Orders;</code> • нет правильного примера 	<code>select concat('index', " ", 'city') from Orders;</code>
11.	<p>Что покажет следующий запрос: <code>select concat('index', " ", 'city') AS delivery_address from Orders;</code></p>	соединит поля с индексом и адресом из таблицы Orders и покажет их с псевдонимом

	<ul style="list-style-type: none"> • ничего, запрос составлен неверно • покажет уникальные значения индексов и адресов из таблицы Orders • соединит поля с индексом и адресом из таблицы Orders и покажет их с псевдонимом delivery_address • соединит поля с индексом и адресом из таблицы Orders, но покажет их без псевдонима 	delivery_address
12.	<p>Выберите правильный пример использования функции округления ROUND</p> <ul style="list-style-type: none"> • select id, price * discount AS total price from Orders ROUND (2); • select id, price * discount ROUND (2) AS total price from Orders; • select id, ROUND (price * discount, 2) AS total price from Orders; • нет правильного примера 	select id, ROUND (price * discount, 2) AS total price from Orders;
13.	<p>Что покажет следующий запрос: select id from Orders where year (date) > 2018;</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 LIMIT: select * from Orders limit 10;</p> <ul style="list-style-type: none"> • необходим, чтобы показать все заказы, содержащие цифру 10 • необходим, чтобы показать первых 10 записей в запросе • необходим, чтобы показать рандомные 10 записей в запросе • не существует такого оператора 	необходим, чтобы показать первых 10 записей в запросе
15.	<p>Что такое агрегирующие функции:</p> <ul style="list-style-type: none"> • функции, которые фильтруют значения • функции, которые сортируют значения • функции, которые работают с набором данных, превращая их в одно итоговое значение • функции, которые суммируют все значения 	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"> • select sum(price) from Orders; • select sum(price), customer_name from Orders; • select * from Orders where price=sum(); • select sum() from Orders group by price desc; 	select sum(price) from Orders;
17.	<p>Возможно ли использование одновременно двух агрегирующих функций: select min(price), max(price) from Orders;</p> <ul style="list-style-type: none"> • да, но данный запрос составлен неверно, надо так: select * from Orders where price IN (min, max); • да, в результате мы получим минимальную и 	да, в результате мы получим минимальную и максимальную стоимости

	<p>максимальную стоимости</p> <ul style="list-style-type: none"> да, в результате мы получим стоимости, отсортированные от минимальной к максимальной нет, две функции использовать одновременно нельзя 	
18.	<p>Выберите корректно составленный запрос с функцией GROUP BY:</p> <ul style="list-style-type: none"> select count(*) from Orders GROUP seller_id; select seller_id, count(*) from Orders GROUP seller_id; select seller_id, count(*) from Orders GROUP BY seller_id; select count(*) from Orders GROUP ON seller_id; 	select seller_id, count(*) from Orders GROUP BY seller_id;
19.	<p>Что покажет следующий запрос: select seller_id, count(*) from Orders GROUP BY seller_id HAVING seller_id IN (2,4,6);</p> <ul style="list-style-type: none"> количество заказов сгруппированное по продавцам 2, 4 и 6 количество продавцов, у которых 2, 4 или 6 товаров ничего, запрос составлен неверно, HAVING указывается до группировки ничего, запрос составлен неверно, для указания условия должно быть использовано WHERE 	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"> select * from Orders where price = (select big(price) from Orders) select * from Orders where price = max select count(*) from Orders select * from Orders where price = (select max(price) from Orders) 	select * from Orders where price = (select max(price) from Orders)
21.	<p>Что такое JOIN:</p> <ul style="list-style-type: none"> операция объединения операция группировки операция суммирования операция создания 	операция объединения
22.	<p>Какого из перечисленных ниже видов JOIN на самом деле не существует:</p> <ul style="list-style-type: none"> LEFT JOIN - который выведет все записи первой таблицы, а для ненайденных пар из правой таблицы проставит значение NULL RIGHT JOIN - который выведет все записи второй таблицы, а на место недостающей информации из первой таблицы проставит NULL INNER JOIN - который показывает только те записи, для которых нашлись пары TRUE JOIN - который выведет все верные 	TRUE JOIN - который выведет все верные значения

	значени	
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"> • <code>select Orders.id, Orders.customer_name, Sellers.id from Orders LEFT JOIN ON Sellers AND Orders.seller_id = Sellers.id;</code> • <code>select id AND customer_name AND seller_id from Orders LEFT JOIN Sellers ON seller_id = id;</code> • <code>select Orders.id, Orders.customer_name, Sellers.id from Orders LEFT JOIN Sellers ON Orders.seller_id = Sellers.id;</code> • <code>select Orders.id, Orders.customer_name, Sellers.id from Orders JOIN Sellers WHEN Orders.seller_id = Sellers.id;</code> 	<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"> • <code>select id, city from Orders order by id union select id, city from Sellers order by city;</code> • <code>select id, city, seller_id from Orders and select city, id from Sellers order by id;</code> • <code>select id, city from Orders union select id, city from Sellers order by id;</code> • Все запросы верные 	<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"> • VARCHAR • STRING • CHAR • TEXT 	STRING
26.	<p>Чем отличается CHAR и VARCHAR?</p> <ul style="list-style-type: none"> • Это одно и то же • VARCHAR не существует • CHAR - это тип данных, а VARCHAR - подтип • CHAR дополняет строку пробелами до максимальной длины, а VARCHAR тратит лишнюю память на хранение значения длины строки 	<p>CHAR дополняет строку пробелами до максимальной длины, а VARCHAR тратит лишнюю память на хранение значения длины строки</p>
27.	<p>Как получить значение текущего года в SQL?</p> <ul style="list-style-type: none"> • <code>select now();</code> • <code>select year();</code> • <code>select year(now());</code> • <code>select year from Date;</code> 	<pre>select year(now());</pre>
28.	<p>Как правильно добавить строку в таблицу? Какой запрос верный?</p> <ul style="list-style-type: none"> • <code>INSERT INTO `SimpleTable` (`some_text`) VALUES ("my text");</code> • <code>INSERT INTO `SimpleTable` SET `some_text`="my text";</code> • <code>SET INTO `SimpleTable` VALUE `some_text`="my text";</code> • <code>UPDATE INTO `SimpleTable` SET</code> 	<pre>INSERT INTO `SimpleTable` (`some_text`) VALUES ("my text");</pre>

	<code>`some text`="my text";</code>	
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"> • Использовать MULTI INSERT INTO вместо INSERT INTO • Использовать подзапрос • Перечислить через запятую все наборы значений после VALUES • Никак, расходимся по домам 	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"> • Именно так и делаю, UPDATE не использую • Так меньше нагрузки на базу, ведь команда одна, а не две • Потому что в записи могут быть автоматически проставляемые поля, такие как auto_increment или timestamp, которые сойдутся при внесении записи заново • Как раз удалять записи в SQL нельзя, вместо этого используется UPDATE с NULL-значениями для всех полей 	Потому что в записи могут быть автоматически проставляемые поля, такие как auto_increment или timestamp, которые сойдутся при внесении записи заново
32.	<p>В каких командах можно использовать LIMIT?</p> <ul style="list-style-type: none"> • Только Select • Select и Insert • Select, Update, Delete • Select, Insert, Delete, Update 	Select, Update, Delete
33.	<p>Как можно заранее узнать, какие записи будут удалены при выполнении DELETE?</p> <ul style="list-style-type: none"> • Зачем заранее, просто вызвать его и посмотреть какие записи пропали • Заменить DELETE на SELECT *, ведь в остальном синтаксис DELETE похож на синтаксис простого SELECT • Сделать DELETE с LIMIT 1, одну запись не жалко • SQL создан для хранения данных, их нельзя удалять 	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"> • CREATE TABLE • MAKE TABLE • SET TABLE • Создавать таблицы можно только через 	CREATE TABLE

	интерфейс СУБД, специальной SQL команды для этого нет	
35.	<p>Можно ли поменять тип данных поля в уже существующей таблице?</p> <ul style="list-style-type: none"> • Да, при помощи команды ALTER • Да, достаточно сделать INSERT с новым типом данных • Нет, только пересоздать таблицу • Тип бывает только у таблицы, а не у поля таблицы 	Да, при помощи команды ALTER

Составитель Кочеров Ю.Н.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

**КРИТЕРИИ И ШКАЛЫ ОЦЕНИВАНИЯ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО
КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ**

«5» (отлично): студент демонстрирует системные теоретические знания, владеет терминологией, делает аргументированные выводы и обобщения, приводит примеры, показывает свободное владение монологической речью и способность быстро реагировать на уточняющие вопросы.

«4» (хорошо): студент демонстрирует прочные теоретические знания, владеет терминологией, делает аргументированные выводы и обобщения, приводит примеры, показывает свободное владение монологической речью, но при этом делает несущественные ошибки, которые быстро исправляет самостоятельно или при незначительной коррекции преподавателем.

«3» (удовлетворительно): студент демонстрирует неглубокие теоретические знания, проявляет слабо сформированные навыки анализа явлений и процессов, недостаточное умение делать аргументированные выводы и приводить примеры, показывает недостаточно свободное владение монологической речью, терминологией, логичностью и последовательностью изложения, делает ошибки, которые может исправить только при коррекции преподавателем.

«2» (неудовлетворительно): студент демонстрирует незнание теоретических основ предмета, не умеет делать аргументированные выводы и приводить примеры, показывает слабое владение монологической речью, не владеет терминологией, проявляет отсутствие логичности и последовательностью изложения, делает ошибки, которые не может исправить даже при коррекции преподавателем, отказывается отвечать на занятии.