

Документ подписан простой электронной подписью
 Информация о владельце:
 ФИО: Косенок Сергей Михайлович
 Должность: ректор
 Дата подписания: 15.06.2026 11:08:20
 Уникальный программный ключ:
 e3a68f38a71e62674b54f4998d99d13d6bfdcf836

Тестовое задание для диагностического тестирования по дисциплине:

Базы данных, 5 семестр

Код, направление подготовки	01.03.02 Прикладная математика и информатика
Направленность (профиль)	Технологии программирования и анализ данных
Форма обучения	очная
Кафедра-разработчик	Кафедра прикладной математики
Выпускающая кафедра	Кафедра прикладной математики

Проверяемая компетенция	Задание	Варианты ответов	Тип сложности вопроса
ПК-4.1, ПК-4.2, ПК-4.3, ПК-3.1, ПК-3.2, ПК-3.3	1. Отношение находится во второй нормальной форме, если оно находится в _____ нормальной форме и каждый не ключевой атрибут неприводимо зависит от _____ ключа.		Низкий
ПК-4.1, ПК-4.2, ПК-4.3, ПК-3.1, ПК-3.2, ПК-3.3	2. Как выглядит запрос, для вывода ВСЕХ значений из таблицы Orders:	1. select ALL from Orders; 2. select % from Orders; 3. select * from Orders; 4. select *.Orders from Orders;	Низкий
ПК-4.1, ПК-4.2, ПК-4.3, ПК-3.1, ПК-3.2, ПК-3.3	3. Какие данные мы получим из этого запроса? select id, date, customer_name from Orders;	1. Неотсортированные номера и даты всех заказов с именами заказчиков 2. Никакие, запрос составлен неверно 3. Номера и даты всех заказов с именами заказчиков, отсортированные по первой колонке 4. Номера и даты всех заказов с именами заказчиков, отсортированные по всем колонкам, содержащим слово Order	Низкий

<p>ПК-4.1, ПК-4.2, ПК-4.3, ПК-3.1, ПК-3.2, ПК-3.3</p>	<p>4. Нормальная форма – требование, предъявляемое к структуре таблиц в теории _____ баз данных, для устранения из базы данных _____ функциональных зависимостей между атрибутами.</p>		<p>Низкий</p>
<p>ПК-4.1, ПК-4.2, ПК-4.3, ПК-3.1, ПК-3.2, ПК-3.3</p>	<p>5. Что покажет следующий запрос: <code>select DISTINCT seller_id order by seller_id from Orders;</code></p>	<ol style="list-style-type: none"> 1. Уникальные ID продавцов, отсортированные по возрастанию 2. Уникальные ID продавцов, отсортированные по убыванию 3. Ничего, запрос составлен неверно, ORDER BY всегда ставится в конце запроса 4. Неотсортированные никак уникальные ID продавцов 	<p>Низкий</p>
<p>ПК-4.1, ПК-4.2, ПК-4.3, ПК-3.1, ПК-3.2, ПК-3.3</p>	<p>6. Что делает спецсимвол '_' в паре с оператором LIKE: <code>select * from Orders where customer_name like 'mik_';</code></p>	<ol style="list-style-type: none"> 1. Найдет все имена, которые начинаются на mik и состоят из 4 символов 2. Найдет все имена, которые начинаются на mik, вне зависимости от того, из какого количества символов они состоят 3. Найдет данные, где имя равно mik 4. Запрос составлен неверно, в паре с оператором like не используются спецсимволы 	<p>Средний</p>

<p>ПК-4.1, ПК-4.2, ПК-4.3, ПК-3.1, ПК-3.2, ПК-3.3</p>	<p>7. Что покажет следующий запрос: select concat(`index`,` `, `city`) AS delivery_address from Orders;</p>	<ol style="list-style-type: none"> 1. Ничего, запрос составлен неверно 2. Покажет уникальные значения индексов и адресов из таблицы Orders 3. Соединит поля с индексом и адресом из таблицы Orders и покажет их с псевдонимом delivery_address 4. Соединит поля с индексом и адресом из таблицы Orders, но покажет их без псевдонима 	<p>Средний</p>
<p>ПК-4.1, ПК-4.2, ПК-4.3, ПК-3.1, ПК-3.2, ПК-3.3</p>	<p>8. Для чего используется LIMIT: select * from Orders limit 10;</p>	<ol style="list-style-type: none"> 1. Необходимо, чтобы показать все заказы, содержащие цифру 10 2. Необходимо, чтобы показать первых 10 записей в запросе 3. Необходимо, чтобы показать случайные 10 записей в запросе 4. не существует такого оператора 	<p>Средний</p>
<p>ПК-4.1, ПК-4.2, ПК-4.3, ПК-3.1, ПК-3.2, ПК-3.3</p>	<p>9. Выберите пример правильно составленного запроса с использованием агрегирующей функции SUM:</p>	<ol style="list-style-type: none"> 1. select sum(price) from Orders; 2. select sum(price), customer_name from Orders; 3. select * from Orders where price=sum(); 4. select sum() from Orders group by price desc; 	<p>Средний</p>
<p>ПК-4.1, ПК-4.2, ПК-4.3, ПК-3.1, ПК-3.2, ПК-3.3</p>	<p>10. Выберите корректно составленный запрос с функцией GROUP BY:</p>	<ol style="list-style-type: none"> 1. select count(*) from Orders GROUP seller_id; 2. select seller_id, count(*) from Orders GROUP seller_id; 3. select seller_id, count(*) from Orders GROUP BY seller_id; 4. select count(*) from Orders GROUP ON seller_id; 	<p>Средний</p>

<p>ПК-4.1, ПК-4.2, ПК-4.3, ПК-3.1, ПК-3.2, ПК-3.3</p>	<p>11. Что покажет следующий запрос: <code>select seller_id, count(*) from Orders GROUP BY seller_id HAVING seller_id IN (2,4,6);</code></p>	<ol style="list-style-type: none"> 1. Количество заказов, сгруппированное по продавцам 2, 4 и 6 2. Количество продавцов, у которых 2, 4 или 6 товаров 3. Ничего, запрос составлен неверно, HAVING указывается до группировки 4. Ничего, запрос составлен неверно, для указания условия должно быть использовано WHERE 	<p>Средний</p>
<p>ПК-4.1, ПК-4.2, ПК-4.3, ПК-3.1, ПК-3.2, ПК-3.3</p>	<p>12. Выберите пример корректно написанного запроса с использованием подзапроса, который выводит информацию о заказе с самой дорогой стоимостью:</p>	<ol style="list-style-type: none"> 1. <code>select * from Orders where price = (select big(price) from Orders)</code> 2. <code>select * from Orders where price = max</code> 3. <code>select count(*) from Orders</code> 4. <code>select * from Orders where price = (select max(price) from Orders)</code> 	<p>Средний</p>
<p>ПК-4.1, ПК-4.2, ПК-4.3, ПК-3.1, ПК-3.2, ПК-3.3</p>	<p>13. Выберите операторы языка SQL</p>	<ol style="list-style-type: none"> 1. DECERT 2. SELECT 3. INSERT 4. DETECT 	<p>Средний</p>
<p>ПК-4.1, ПК-4.2, ПК-4.3, ПК-3.1, ПК-3.2, ПК-3.3</p>	<p>14. Установите соответствие между понятиями теории реляционных БД и их определениями</p>	<ol style="list-style-type: none"> 1. Строка таблицы 1. Кортеж 2. Заголовок столбца таблицы 2. Атрибут 3. Количество кортежей в отношении 3. Кардинальность 	<p>Средний</p>
<p>ПК-4.1, ПК-4.2, ПК-4.3, ПК-3.1, ПК-3.2, ПК-3.3</p>	<p>15. Какие поля из таблицы обязательно перечислять в INSERT для вставки данных?</p>	<ol style="list-style-type: none"> 1. Все 2. Только те, у которых нет DEFAULT значения 3. Те, у которых нет DEFAULT значения и которые не имеют атрибут <code>auto_increment</code> 	<p>Средний</p>

		4. Все поля имеют негласное DEFAULT значения, обязательных полей в SQL нет	
ПК-4.1, ПК-4.2, ПК-4.3, ПК-3.1, ПК-3.2, ПК-3.3	16. В каких командах можно использовать LIMIT?	1. Только Select 2. Select и Insert 3. Select, Update, Delete 4. Select, Insert, Delete, Update	Высокий
ПК-4.1, ПК-4.2, ПК-4.3, ПК-3.1, ПК-3.2, ПК-3.3	17. Как можно заранее узнать, какие записи будут удалены при выполнении DELETE?	1. Зачем заранее, просто вызвать его и посмотреть какие записи пропали 2. Заменить DELETE на SELECT *, ведь в остальном синтаксис DELETE похож на синтаксис простого SELECT 3. Сделать DELETE с LIMIT 1, одну запись не жалко 4. SQL создан для хранения данных, их нельзя удалять	Высокий
ПК-4.1, ПК-4.2, ПК-4.3, ПК-3.1, ПК-3.2, ПК-3.3	18. Какой командой можно создать новую таблицу?	1. CREATE TABLE 2. MAKE TABLE 3. SET TABLE 4. Создавать таблицы можно только через интерфейс СУБД, специальной SQL команды для этого нет	Высокий
ПК-4.1, ПК-4.2, ПК-4.3, ПК-3.1, ПК-3.2, ПК-3.3	19. Можно ли поменять тип данных поля в уже существующей таблице?	1. Да, при помощи команды ALTER 2. Да, достаточно сделать INSERT с новым типом данных 3. Нет, только пересоздать таблицу 4. Тип бывает только у таблицы, а не у поля таблицы	Высокий
ПК-4.1, ПК-4.2, ПК-4.3, ПК-3.1, ПК-3.2, ПК-3.3	20. Выберите элементы реляционной модели	1. Домен 2. Запрос 3. Отношение 4. Кардинальность	Высокий