

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Косенок Сергей Михайлович  
Должность: ректор  
Дата подписания: 11.06.2026 10:49:18  
Уникальный программный ключ:  
e3a68f3eaa1e62674b54f4998099d3d6bfdcf836

**Тестовое задание для диагностического тестирования по дисциплине:**

*Базы данных, 2 семестр*

Код, направление подготовки	27.03.04 Управление в технических системах
Направленность (профиль)	Инженерия автоматизированных, информационных и робототехнических систем
Форма обучения	очная
Кафедра-разработчик	Автоматики и компьютерных систем
Выпускающая кафедра	Автоматики и компьютерных систем

Проверяемая компетенция	Задание	Варианты ответов	Тип сложности вопроса
ПК-3.3, ПК-8.2, ПК-7.1, ПК-7.2	1. Что такое реляционная база данных?	<ol 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> </ol>	Низкий
ПК-3.3, ПК-8.2, ПК-7.1, ПК-7.2	2. Как выглядит запрос, для вывода ВСЕХ значений из таблицы Orders?	<ol 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> </ol>	Низкий
ПК-3.3, ПК-8.2, ПК-7.1, ПК-7.2	3. Какие данные будут получены в результате выполнения запроса? select id, date, customer_name from Orders;	<ol style="list-style-type: none"> <li>Неотсортированные номера и даты всех заказов с именами заказчиков</li> <li>Никакие, запрос составлен неверно</li> <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> </ol>	Низкий
ПК-3.3, ПК-8.2, ПК-7.1, ПК-7.2	4. Какие данные будут получены в результате выполнения запроса? select * from Orders where date between '2017-01-01' and '2017-12-31'	<ol 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> </ol>	Низкий
ПК-3.3, ПК-8.2, ПК-7.1, ПК-7.2	5. Какие данные будут получены в результате выполнения запроса? select DISTINCT seller_id order by seller_id from Orders;	<ol 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> </ol>	Низкий
ПК-3.3, ПК-8.2, ПК-7.1, ПК-7.2	6. Что делает спецсимвол '_' в паре с оператором LIKE: select * from Orders where customer_name like 'mik_';	<ol style="list-style-type: none"> <li>найдет все имена, которые начинаются на mik и состоят из 4 символов</li> <li>найдет все имена, которые начинаются на mik, вне зависимости от того, из какого количества символов они состоят</li> <li>найдет данные, где имя равно mik</li> <li>запрос составлен неверно, в паре с оператором LIKE не используются спецсимволы</li> </ol>	Средний
ПК-3.3, ПК-8.2, ПК-7.1, ПК-7.2	7. Какие данные будут получены в результате выполнения запроса? select concat('index', " ", `city`) AS delivery_address from Orders;	<ol style="list-style-type: none"> <li>ничего, запрос составлен неверно</li> <li>покажет уникальные значения индексов и адресов из таблицы Orders</li> <li>соединит поля с индексом и адресом из таблицы Orders и покажет их с псевдонимом delivery_address</li> </ol>	Средний

		4. соединит поля с индексом и адресом из таблицы Orders, но покажет их без псевдонима	
ПК-3.3, ПК-8.2, ПК-7.1, ПК-7.2	8. Для чего используется LIMIT: select * from Orders limit 10;	1. необходим, чтобы показать все заказы, содержащие цифру 10 2. необходим, чтобы показать первых 10 записей в запросе 3. необходим, чтобы показать рандомные 10 записей в запрос 4. не существует такого оператора	Средний
ПК-3.3, ПК-8.2, ПК-7.1, ПК-7.2	9. Выберите пример правильно составленного запроса с использованием агрегирующей функции SUM:	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;	Средний
ПК-3.3, ПК-8.2, ПК-7.1, ПК-7.2	10. Выберите корректно составленный запрос с функцией GROUP BY:	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;	Средний
ПК-3.3, ПК-8.2, ПК-7.1, ПК-7.2	11. Что покажет следующий запрос: select seller_id, count(*) from Orders GROUP BY seller_id HAVING seller_id IN (2,4,6);	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	Средний
ПК-3.3, ПК-8.2, ПК-7.1, ПК-7.2	12. Выберите пример корректно написанного запроса с использованием подзапроса, который выводит информацию о заказе с самой дорогой стоимостью:	1. select * from Orders where price = (select big(price) from Orders); 2. select * from Orders where price = max; 3. select count(*) from Orders; 4. select * from Orders where price = (select max(price) from Orders);	Средний
ПК-3.3, ПК-8.2, ПК-7.1, ПК-7.2	13. Выберите корректный пример составленного запроса с использованием JOIN. Данный запрос выведет нам данные ID заказа, имя заказчика и продавца:	1. select Orders.id, Orders.customer_name, Sellers.id from Orders LEFT JOIN ON Sellers AND Orders.seller_id = Sellers.id; 2. select id AND customer_name AND seller_id from Orders LEFT JOIN Sellers ON seller_id = id; 3. select Orders.id, Orders.customer_name, Sellers.id from Orders LEFT JOIN Sellers ON Orders.seller_id = Sellers.id; 4. select Orders.id, Orders.customer_name, Sellers.id from Orders JOIN Sellers WHEN Orders.seller_id = Sellers.id;	Средний
ПК-3.3, ПК-8.2, ПК-7.1, ПК-7.2	14. Как правильно добавить строку в таблицу? Какой запрос верный?	1. INSERT INTO `SimpleTable` (`some_text`) VALUES ("my text"); 2. INSERT INTO `SimpleTable` SET `some_text`="my text"; 3. SET INTO `SimpleTable` VALUE `some_text`="my text";	Средний

		4. UPDATE INTO `SimpleTable` SET `some_text`="my text";	
ПК-3.3, ПК-8.2, ПК-7.1, ПК-7.2	15. Какие поля из таблицы обязательно перечислять в INSERT для вставки данных?	<ol style="list-style-type: none"> <li>1. Конечно все,</li> <li>2. Только те, у которых нет DEFAULT значения,</li> <li>3. Те, у которых нет DEFAULT значения и которые не имеют атрибут auto_increment,</li> <li>4. Все поля имеют негласное DEFAULT значения, обязательных полей в SQL нет.</li> </ol>	Средний
ПК-3.3, ПК-8.2, ПК-7.1, ПК-7.2	16. В каких командах можно использовать LIMIT?	<ol style="list-style-type: none"> <li>1. Только Select,</li> <li>2. Select и Insert,</li> <li>3. Select, Update, Delete,</li> <li>4. Select, Insert, Delete, Update.</li> </ol>	Высокий
ПК-3.3, ПК-8.2, ПК-7.1, ПК-7.2	17. Как можно заранее узнать, какие записи будут удалены при выполнении DELETE?	<ol style="list-style-type: none"> <li>1. Зачем заранее, просто вызвать его и посмотреть какие записи пропали,</li> <li>2. Заменить DELETE на SELECT *, ведь в остальном синтаксис DELETE похож на синтаксис простого SELECT,</li> <li>3. Сделать DELETE с LIMIT 1, одну запись не жалко,</li> <li>4. SQL создан для хранения данных, их нельзя удалять.</li> </ol>	Высокий
ПК-3.3, ПК-8.2, ПК-7.1, ПК-7.2	18. Какой командой можно создать новую таблицу?	<ol style="list-style-type: none"> <li>1. CREATE TABLE,</li> <li>2. MAKE TABLE,</li> <li>3. SET TABLE,</li> <li>4. Создавать таблицы можно только через интерфейс СУБД, специальной SQL команды для этого нет.</li> </ol>	Высокий
ПК-3.3, ПК-8.2, ПК-7.1, ПК-7.2	19. Какого из перечисленных ниже видов JOIN на самом деле не существует:	<ol style="list-style-type: none"> <li>1. LEFT JOIN - который выведет все записи первой таблицы, а для ненайденных пар из правой таблицы проставит значение NULL,</li> <li>2. RIGHT JOIN - который выведет все записи второй таблицы, а на место недостающей информации из первой таблицы проставит NULL,</li> <li>3. INNER JOIN - который показывает только те записи, для которых нашлись пары,</li> <li>4. TRUE JOIN - который выведет все верные значения.</li> </ol>	Высокий
ПК-3.3, ПК-8.2, ПК-7.1, ПК-7.2	20. Можно ли поменять тип данных поля в уже существующей таблице?	<ol style="list-style-type: none"> <li>1. Да, при помощи команды ALTER,</li> <li>2. Да, достаточно сделать INSERT с новым типом данных,</li> <li>3. Нет, только пересоздать таблицу,</li> <li>4. Тип бывает только у таблицы, а не у поля таблицы.</li> </ol>	Высокий