

Документ подписан простой электронной подписью
 Информация о владельце: Тестовое задание
 ФИО: Косенок Сергей Михайлович
 Должность: ректор
 Дата подписания: 23.07.2025 08:37:20
 Уникальный программный ключ:
 e3a68f3eaa1e02674b541f998699d306bdfcf836

для диагностического тестирования по дисциплине:

Базы данных

Код направления подготовки	38.03.05 Бизнес-информатика
Направленность (профиль)	Экономика предприятий управление бизнес-процессами
Форма обучения	Очная
Кафедра-разработчик	Менеджмента и бизнеса
Выпускающая кафедра	Менеджмента и бизнеса

Проверяемая компетенция	Задание	Варианты ответов	Тип сложности вопроса
ПК-7.1	1. Что такое реляционные базы данных:	1. <u>База данных, в которой информация хранится в виде двумерных таблиц, связанных между собой</u> 2. База данных, в которой одна ни с чем не связанная таблица 3. Любая база данных - реляционная 4. Совокупность данных, не связанных между собой	Низкий
ПК-7.1	2. Как выглядит запрос, для вывода ВСЕХ значений из таблицы Orders:	1. select ALL from Orders; 2. select % from Orders; 3. <u>select * from Orders;</u> 4. select *.Orders from Orders;	Низкий
ПК-7.1	3. Какие данные мы получим из этого запроса? select id, date, customer_name from Orders;	1. <u>Неотсортированные номера и даты всех заказов с именами заказчиков</u> 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
ПК-7.1	4. Какие операторы используются для объединения таблиц в SQL-запросах? Выберите два правильных варианта ответа.	1. <u>LEFT OUTER JOIN</u> 2. <u>JOIN</u> 3. WHERE 4. GROUP BY 5. ORDER BY 1. HAVING	Низкий

ПК-7.1	5. Какие из перечисленных типов данных являются числовыми в PostgreSQL? Выберите два правильных варианта ответа.	1. <u>INT</u> 2. VARCHAR 3. FLOAT 4. TEXT 5. <u>NUMERIC</u> 6. DATE	Низкий
ПК-7.1	6. Что делает спецсимвол '_' в паре с оператором LIKE: select * from Orders where customer_name like 'mik_';	1. <u>найдет все имена, которые начинаются на mik и состоят из 4 символов</u> 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	Средний
ПК-7.1	7. Что покажет следующий запрос: select DISTINCT seller_id order by seller_id from Orders;	1. Уникальные ID продавцов, отсортированные по возрастанию 2. Уникальные ID продавцов, отсортированные по убыванию 3. <u>Ничего, запрос составлен неверно, ORDER BY всегда ставится в конце запроса</u> 4. Неотсортированные никак уникальные ID продавцов	Средний
ПК-7.1	8. Что покажет следующий запрос: select * from Orders where date between '2017-01-01' and '2017-12-31'	1. Все данные по заказам, совершенным за 2017 год, за исключением 01 января 2017 года 2. Все данные по заказам, совершенным за 2017 год, за исключением 31 декабря 2017 года 3. <u>Все данные по заказам, совершенным за 2017 год</u> 4. Ничего, запрос составлен неверно	Средний
ПК-7.1	9. Выберите пример правильно составленного запроса с использованием агрегирующей функции SUM:	1. <u>select sum(price) from Orders;</u> 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-7.1	10. Выберите корректно составленный запрос с функцией GROUP BY:	1. <code>select count(*) from Orders GROUP seller_id;</code> 2. <code>select seller_id, count(*) from Orders GROUP seller_id;</code> 3. <u><code>select seller_id, count(*) from Orders GROUP BY seller_id;</code></u> 4. <code>select count(*) from Orders GROUP ON seller_id;</code>	Средний
ПК-7.1	11. Что покажет следующий запрос: <code>select seller_id, count(*) from Orders GROUP BY seller_id HAVING seller_id IN (2,4,6);</code>	1. <u>количество заказов сгруппированное по продавцам 2, 4 и 6</u> 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-7.1	12. Выберите пример корректно написанного запроса с использованием подзапроса, который выводит информацию о заказе с самой дорогой стоимостью:	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. <u><code>select * from Orders where price = (select max(price) from Orders)</code></u>	Средний
ПК-7.1	13. Выберите корректный пример составленного запроса с использованием JOIN. Данный запрос выведет нам данные ID заказа, имя заказчика и продавца:	1. <code>select Orders.id, Orders.customer_name, Sellers.id from Orders LEFT JOIN ON Sellers AND Orders.seller_id = Sellers.id;</code> 2. <code>select id AND customer_name AND seller_id from Orders LEFT JOIN Sellers ON seller_id = id;</code> 3. <u><code>select Orders.id, Orders.customer_name, Sellers.id from Orders LEFT JOIN Sellers ON Orders.seller_id = Sellers.id;</code></u> 4. <code>select Orders.id, Orders.customer_name, Sellers.id from Orders JOIN Sellers WHEN Orders.seller_id = Sellers.id;</code>	Средний
ПК-7.1	14. Как правильно добавить строку в таблицу? Какой запрос верный?	1. <u><code>INSERT INTO `SimpleTable` (`some_text`) VALUES ("my text");</code></u> 2. <code>INSERT INTO `SimpleTable` SET `some_text`="my text";</code> 3. <code>SET INTO `SimpleTable`</code>	Средний

		VALUE `some_text`="my text"; 4. UPDATE INTO `SimpleTable` SET `some_text`="my text";	
ПК-7.1	15. Какие поля из таблицы обязательно перечислять в INSERT для вставки данных?	1. Конечно все 2. Только те, у которых нет DEFAULT значения 3. <u>Те, у которых нет DEFAULT значения и которые не имеют атрибут auto_increment</u> 4. Все поля имеют негласное DEFAULT значения, обязательных полей в SQL нет	Средний
ПК-7.1	16. В каких командах можно использовать LIMIT?	1. Только Select 2. Select и Insert 3. <u>Select, Update, Delete</u> 4. Select, Insert, Delete, Update	Высокий
ПК-7.1	17. Как можно заранее узнать, какие записи будут удалены при выполнении DELETE?	1. Зачем заранее, просто вызвать его и посмотреть какие записи пропали 2. <u>Заменить DELETE на SELECT *, ведь в остальном синтаксис DELETE похож на синтаксис простого SELECT</u> 3. Сделать DELETE с LIMIT 1, одну запись не жалко 4. SQL создан для хранения данных, их нельзя удалять	Высокий
ПК-7.1	18. Какой командой можно создать новую таблицу?	1. <u>CREATE TABLE</u> 2. MAKE TABLE 3. SET TABLE 4. Создавать таблицы можно только через интерфейс СУБД, специальной SQL команды для этого нет	Высокий
ПК-7.1	19. Можно ли поменять тип данных поля в уже существующей таблице?	1. <u>Да, при помощи команды ALTER</u> 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-7.1	20. Какого из перечисленных ниже видов JOIN на самом деле не существует:	1. LEFT JOIN - который выведет все записи первой таблицы, а для ненайденных пар из правой таблицы проставит значение NULL 2. RIGHT JOIN - который выведет все записи второй таблицы, а на место недостающей информации из первой таблицы проставит NULL 3. INNER JOIN - который	Высокий

		показывает только те записи, для которых нашлись пары 4. <u>TRUE JOIN - который выведет все верные значения</u>	
--	--	---	--