

Документ подписан простой электронной подписью

Информация о владельце: Тестовое задание для диагностического тестирования по дисциплине:

ФИО: Косенок Сергей Михайлович

Должность: ректор

Дата подписания: 20.06.2025 07:53:01

Уникальный программный ключ:

e3a68f3eaa1e02674b541e998099d3d6bfdcf836

Базы данных

Код направления подготовки	ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ
Направленность (профиль)	Безопасность информационных систем и технологий
Форма обучения	Очная
Кафедра-разработчик	Информатики и вычислительной техники
Выпускающая кафедра	Информатики и вычислительной техники

Проверяемая компетенция	Задание	Варианты ответов	Тип сложности вопроса
ПК-4	1. Что такое реляционные базы данных:	1. База данных, в которой информация хранится в виде двумерных таблиц, связанных между собой 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-4	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	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
ПК-4	4. Что покажет следующий запрос: select * from Orders where	1. Все данные по заказам, совершенным за 2017 год,	Низкий

	date between '2017-01-01' and '2017-12-31'	<p>за исключением 01 января 2017 года</p> <p>2. Все данные по заказам, совершенным за 2017 год, за исключением 31 декабря 2017 года</p> <p>3. Все данные по заказам, совершенным за 2017 год</p> <p>4. Ничего, запрос составлен неверно</p>	
ПК-4	5. Что покажет следующий запрос: select DISTINCT seller_id order by seller_id from Orders;	<p>1. Уникальные ID продавцов, отсортированные по возрастанию</p> <p>2. Уникальные ID продавцов, отсортированные по убыванию</p> <p>3. Ничего, запрос составлен неверно, ORDER BY всегда ставится в конце запроса</p> <p>4. Неотсортированные никак уникальные ID продавцов</p>	Низкий
ПК-4	6. Что делает спецсимвол '_' в паре с оператором LIKE: select * from Orders where customer_name like 'mik_';	<p>1. найдет все имена, которые начинаются на mik и состоят из 4 символов</p> <p>2. найдет все имена, которые начинаются на mik, вне зависимости от того, из какого количества символов они состоят</p> <p>3. найдет данные, где имя равно mik</p> <p>4. запрос составлен неверно, в паре с оператором like не используются спецсимволы</p>	Средний
ПК-4	7. Что покажет следующий запрос: select concat(`index`,` `, `city`) AS delivery_address from Orders;	<p>1. ничего, запрос составлен неверно</p> <p>2. покажет уникальные значения индексов и адресов из таблицы Orders</p> <p>3. соединит поля с индексом и адресом из таблицы Orders и покажет их с псевдонимом delivery_address</p> <p>4. соединит поля с индексом и адресом из таблицы Orders, но покажет их без псевдонима</p>	Средний

	8. Для чего используется LIMIT: select * from Orders limit 10;	<ol style="list-style-type: none"> 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-4	9. Выберите пример правильно составленного запроса с использованием агрегирующей функции SUM:	<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; 	Средний
ПК-4	10. Выберите корректно составленный запрос с функцией GROUP BY:	<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; 	Средний
ПК-4	11. Что покажет следующий запрос: select seller_id, count(*) from Orders GROUP BY seller_id HAVING seller_id IN (2,4,6);	<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 	Средний
ПК-4	12. Выберите пример корректно написанного запроса с использованием подзапроса, который выводит информацию о заказе с самой дорогой стоимостью:	<ol style="list-style-type: none"> 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-4	13. Выберите корректный пример составленного	<ol style="list-style-type: none"> 1. select Orders.id, Orders.customer_name, 	Средний

	запроса с использованием JOIN. Данный запрос выведет нам данные ID заказа, имя заказчика и продавца:	<p>Sellers.id from Orders LEFT JOIN ON Sellers AND Orders.seller_id = Sellers.id;</p> <p>2. select id AND customer_name AND seller_id from Orders LEFT JOIN Sellers ON seller_id = id;</p> <p>3. select Orders.id, Orders.customer_name, Sellers.id from Orders LEFT JOIN Sellers ON Orders.seller_id = Sellers.id;</p> <p>4. select Orders.id, Orders.customer_name, Sellers.id from Orders JOIN Sellers WHEN Orders.seller_id = Sellers.id;</p>	
ПК-4	14. Как правильно добавить строку в таблицу? Какой запрос верный?	<p>1. INSERT INTO `SimpleTable` (`some_text`) VALUES ("my text");</p> <p>2. INSERT INTO `SimpleTable` SET `some_text`="my text";</p> <p>3. SET INTO `SimpleTable` VALUE `some_text`="my text";</p> <p>4. UPDATE INTO `SimpleTable` SET `some_text`="my text";</p>	Средний
ПК-4	15. Какие поля из таблицы обязательно перечислять в INSERT для вставки данных?	<p>1. Конечно все</p> <p>2. Только те, у которых нет DEFAULT значения</p> <p>3. Те, у которых нет DEFAULT значения и которые не имеют атрибут auto_increment</p> <p>4. Все поля имеют негласное DEFAULT значения, обязательных полей в SQL нет</p>	Средний
ПК-4	16. В каких командах можно использовать LIMIT?	<p>1. <u>Только Select</u></p> <p>2. Select и Insert</p> <p>3. Select, Update, Delete</p> <p>4. Select, Insert, Delete, Update</p>	Высокий
ПК-4	17. Как можно заранее узнать, какие записи будут удалены при выполнении DELETE?	<p>1. Зачем заранее, просто вызвать его и посмотреть какие записи пропали</p> <p>2. Заменить DELETE на SELECT *, ведь в остальном синтаксис</p>	Высокий

		<p>DELETE похож на синтаксис простого SELECT</p> <ol style="list-style-type: none"> 3. Сделать DELETE с LIMIT 1, одну запись не жалко 4. SQL создан для хранения данных, их нельзя удалять 	
ПК-4	18. Какой командой можно создать новую таблицу?	<ol style="list-style-type: none"> 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	19. Можно ли поменять тип данных поля в уже существующей таблице?	<ol style="list-style-type: none"> 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	20. Какого из перечисленных ниже видов JOIN на самом деле не существует:	<ol style="list-style-type: none"> 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 показывает только те записи, для которых нашлись пары 4. TRUE JOIN - который выведет все верные значения 	Высокий