

Тестовое задание для диагностического тестирования по дисциплине:

**Базы данных**

Код, направление подготовки	09.03.04 Программная инженерия
Направленность (профиль)	Программное обеспечение компьютерных 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-3.1, ПК-3.2, ПК-5.1, ПК-7.2, ПК-7.3	1. Что такое реляционные базы 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-3.1, ПК-3.2, ПК-5.1, ПК-7.2, ПК-7.3	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-3.1, ПК-3.2, ПК-5.1, ПК-7.2, ПК-7.3	3. Какие данные мы получим из этого 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-3.1, ПК-3.2, ПК-5.1, ПК-7.2, ПК-7.3	4. Что покажет следующий запрос: 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-3.1, ПК-3.2, ПК-5.1, ПК-7.2, ПК-7.3	5. Что покажет следующий запрос: 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-3.1, ПК-3.2, ПК-5.1, ПК-7.2, ПК-7.3	6. Что делает спецсимвол '_' в паре с оператором LIKE: <code>select * from Orders where customer_name like 'mik_';</code>	<ol style="list-style-type: none"> <li>1. найдет все имена, которые начинаются на mik и состоят из 4 символов</li> <li>2. найдет все имена, которые начинаются на mik, вне зависимости от того, из какого количества символов они состоят</li> <li>3. найдет данные, где имя равно mik</li> <li>4. запрос составлен неверно, в паре с оператором like не используются спецсимволы</li> </ol>	Средний
ПК-3.1, ПК-3.2, ПК-5.1, ПК-7.2, ПК-7.3	7. Что покажет следующий запрос: <code>select concat(`index`,``, `city`) AS delivery_address from Orders;</code>	<ol style="list-style-type: none"> <li>1. ничего, запрос составлен неверно</li> <li>2. покажет уникальные значения индексов и адресов из таблицы Orders</li> <li>3. соединит поля с индексом и адресом из таблицы Orders и покажет их с псевдонимом delivery_address</li> <li>4. соединит поля с индексом и адресом из таблицы Orders, но покажет их без псевдонима</li> </ol>	Средний
ПК-3.1, ПК-3.2, ПК-5.1, ПК-7.2, ПК-7.3	8. Для чего используется LIMIT: <code>select * from Orders limit 10;</code>	<ol style="list-style-type: none"> <li>1. необходимо, чтобы показать все заказы, содержащие цифру 10</li> <li>2. необходимо, чтобы показать первых 10 записей в запросе</li> <li>3. необходимо, чтобы показать рандомные 10 записей в запросе</li> <li>4. не существует такого оператора</li> </ol>	Средний
ПК-3.1, ПК-3.2, ПК-5.1, ПК-7.2, ПК-7.3	9. Выберите пример правильно составленного запроса с использованием агрегирующей функции SUM:	<ol style="list-style-type: none"> <li>1. <code>select sum(price) from Orders;</code></li> <li>2. <code>select sum(price), customer_name from Orders;</code></li> <li>3. <code>select * from Orders where price=sum();</code></li> <li>4. <code>select sum() from Orders group by price desc;</code></li> </ol>	Средний
ПК-3.1, ПК-3.2, ПК-5.1, ПК-7.2, ПК-7.3	10. Выберите корректно составленный запрос с функцией GROUP BY:	<ol style="list-style-type: none"> <li>1. <code>select count(*) from Orders GROUP seller_id;</code></li> <li>2. <code>select seller_id, count(*) from Orders GROUP seller_id;</code></li> <li>3. <code>select seller_id, count(*) from Orders GROUP BY seller_id;</code></li> <li>4. <code>select count(*) from Orders GROUP ON seller_id;</code></li> </ol>	Средний
ПК-3.1, ПК-3.2, ПК-5.1, ПК-7.2, ПК-7.3	11. Что покажет следующий запрос: <code>select seller_id, count(*) from Orders GROUP BY seller_id HAVING seller_id IN (2,4,6);</code>	<ol style="list-style-type: none"> <li>1. количество заказов сгруппированное по продавцам 2, 4 и 6</li> <li>2. количество продавцов, у которых 2, 4 или 6 товаров</li> <li>3. ничего, запрос составлен неверно, HAVING указывается до группировки</li> <li>4. ничего, запрос составлен неверно, для указания условия должно быть использовано WHERE</li> </ol>	Средний
ПК-3.1, ПК-3.2, ПК-5.1, ПК-7.2, ПК-7.3	12. Выберите пример корректно написанного запроса с использованием подзапроса, который выводит информацию о заказе с самой дорогой стоимостью:	<ol style="list-style-type: none"> <li>1. <code>select * from Orders where price = (select big(price) from Orders)</code></li> <li>2. <code>select * from Orders where price = max</code></li> <li>3. <code>select count(*) from Orders</code></li> <li>4. <code>select * from Orders where price = (select max(price) from Orders)</code></li> </ol>	Средний
ПК-3.1, ПК-3.2, ПК-5.1, ПК-7.2, ПК-7.3	13. Выберите корректный пример составленного запроса с использованием JOIN. Данный	<ol style="list-style-type: none"> <li>1. <code>select Orders.id, Orders.customer_name, Sellers.id</code></li> </ol>	Средний

	запрос выведет нам данные ID заказа, имя заказчика и продавца:	<p>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-3.1, ПК-3.2, ПК-5.1, ПК-7.2, ПК-7.3	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-3.1, ПК-3.2, ПК-5.1, ПК-7.2, ПК-7.3	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-3.1, ПК-3.2, ПК-5.1, ПК-7.2, ПК-7.3	16. В каких командах можно использовать LIMIT?	<p>1. Только Select</p> <p>2. Select и Insert</p> <p>3. Select, Update, Delete</p> <p>4. Select, Insert, Delete, Update</p>	Высокий
ПК-3.1, ПК-3.2, ПК-5.1, ПК-7.2, ПК-7.3	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 DELETE похож на синтаксис простого SELECT</p> <p>3. Сделать DELETE с LIMIT 1, одну запись не жалко</p> <p>4. SQL создан для хранения данных, их нельзя удалять</p>	Высокий
ПК-3.1, ПК-3.2, ПК-5.1, ПК-7.2, ПК-7.3	18. Какой командой можно создать новую таблицу?	<p>1. CREATE TABLE</p> <p>2. MAKE TABLE</p> <p>3. SET TABLE</p> <p>4. Создавать таблицы можно только через интерфейс СУБД, специальной SQL команды для этого нет</p>	Высокий
ПК-3.1, ПК-3.2, ПК-5.1, ПК-7.2, ПК-7.3	19. Можно ли поменять тип данных поля в уже существующей таблице?	<p>1. Да, при помощи команды ALTER</p> <p>2. Да, достаточно сделать INSERT с новым типом данных</p> <p>3. Нет, только пересоздать таблицу</p> <p>4. Тип бывает только у таблицы, а не у поля таблицы</p>	Высокий
ПК-3.1, ПК-3.2, ПК-5.1, ПК-7.2, ПК-7.3	20. Какого из перечисленных ниже видов JOIN на самом деле не существует:	<p>1. LEFT JOIN - который выведет все записи первой таблицы, а для ненайденных пар из правой таблицы проставит значение NULL</p> <p>2. RIGHT JOIN - который выведет все записи второй таблицы, а на место недостающей информации из первой таблицы проставит NULL</p>	Высокий

		<ol style="list-style-type: none"><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>	
--	--	--	--