**Вопрос 1**

Что такое реляционные базы данных?

База данных, в которой информация хранится в виде двумерных таблиц, связанных между собой+

База данных, в которой одна ни с чем не связанная таблица

Любая база данных - реляционная

Совокупность данных, не связанных между собой

**Вопрос 2**

Что покажет следующий запрос?  
select \* from Orders where date between '2017-01-01' and '2017-12-31'

Все данные по заказам, совершенным за 2017 год, за исключением 01 января 2017 года

Все данные по заказам, совершенным за 2017 год, за исключением 31 декабря 2017 года

Все данные по заказам, совершенным за 2017 год+

Ничего, запрос составлен неверно

**Вопрос 3**

Что не так с этим запросом?  
select id, date from Orders where seller\_id = NULL

Все верно, запрос покажет все заказы, продавцы которых не проставлены

NULL нужно взять в кавычки

Сравнение с NULL можно проводить только с оператором IS+

Сравнение с NULL можно проводить только с оператором ON

**Вопрос 4**

Что покажет следующий запрос: select DISTINCT seller\_id order by seller\_id from Orders;

Уникальные ID продавцов, отсортированные по возрастанию+

Уникальные ID продавцов, отсортированные по убыванию

Ничего, запрос составлен неверно, ORDER BY всегда ставится в конце запроса

Неотсортированные никак уникальные ID продавцов

**Вопрос 5**

Что такое агрегирующие функции?

функции, которые фильтруют значения

функции, которые сортируют значения

функции, которые работают с набором данных, превращая их в одно итоговое значение+

функции, которые суммируют все значения

**Вопрос 6**

Выберите корректно составленный запрос с функцией GROUP BY:

select count(\*) from Orders GROUP seller\_id;

select seller\_id, count(\*) from Orders GROUP seller\_id;

select seller\_id, count(\*) from Orders GROUP BY seller\_id;+

select count(\*) from Orders GROUP ON seller\_id;

**Вопрос 7**

Подумайте, как можно легко заполнить таблицу 10 миллионов тестовых записей?

Можно будет использовать какую нибудь программу которая сгенерирует рандомные данные в таблицу

**Вопрос 8**

Выберите корректный пример составленного запроса с использованием JOIN. Данный запрос выведет нам данные ID заказа, имя заказчика и продавца:

select Orders.id, Orders.customer\_name, Sellers.id from Orders LEFT JOIN ON Sellers AND Orders.seller\_id = Sellers.id;

select id AND customer\_name AND seller\_id from Orders LEFT JOIN Sellers ON seller\_id = id;

select Orders.id, Orders.customer\_name, Sellers.id from Orders LEFT JOIN Sellers ON Orders.seller\_id = Sellers.id;+

select Orders.id, Orders.customer\_name, Sellers.id from Orders JOIN Sellers WHEN Orders.seller\_id = Sellers.id;

**Вопрос 9**

Выберите правильный пример запроса с использованием UNION:

select id, city from Orders order by id union select id, city from Sellers order by city;

select id, city, seller\_id from Orders and select city, id from Sellers order by id;

select id, city from Orders union select id, city from Sellers order by id;+

Все запросы верные

**Вопрос 10**

Чем отличается CHAR и VARCHAR?

Это одно и то же

VARCHAR не существует

CHAR - это тип данных, а VARCHAR - подтип

CHAR дополняет строку пробелами до максимальной длины, а VARCHAR тратит лишнюю память на хранение значения длины строки+

**Вопрос 11**

Как правильно добавить строку в таблицу? Какой запрос верный?

INSERT INTO `SimpleTable` (`some\_text`) VALUES ("my text");+

INSERT INTO `SimpleTable` SET `some\_text`="my text";

SET INTO `SimpleTable` VALUE `some\_text`="my text";

UPDATE INTO `SimpleTable` SET `some\_text`="my text";

**Вопрос 12**

Напишите запрос, возвращающий все значения из таблицы «Countries», за исключением страны с ID=8, имеет вид:

SELECT \* FROM Countries WHERE ID <> 8

**Вопрос 13**

Напишите запрос для выборки данных из таблицы «Customers», где условием является проживание заказчика в городе Москва

SELECT \* FROM Customers WHERE city = 'Москва'

**Вопрос 14**

Для создания новой виртуальной таблицы, которая базируется на результатах сделанного ранее SQL запроса, какая команда используется?

CREATE VIEW

**Вопрос 15**

В таблице «Animals» базы данных зоопарка содержится информация обо всех обитающих там животных, в том числе о лисах: red fox, grey fox, little fox. Напишите запрос, возвращающий информацию о возрасте лис.

SELECT age FROM Animals WHERE type = 'fox';

**Вопрос 16**

Какая команда используется для объединения результатов запроса без удаления дубликатов?

UNION ALL