Вопросы для уровня «Junior»

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

Как выглядит запрос для вывода ВСЕХ значений из таблицы Orders:

* SELECT ALL FROM Orders
* SELECT % FROM Orders
* **SELECT \* FROM Orders**

Есть ли ошибка в запросе: SELECT id, date, customer\_name FROM Orders WHERE customer\_name = Mike

* Запрос составлен правильно
* **Mike необходимо записать в кавычках ‘Mike’**
* Строчку с WHERE поменять местами с FROM

Порядок выполнения операторов AND и OR следующий:

* **Сначала выполняется AND, а затем OR**
* Сначала выполняется OR, а затем AND
* Порядок выполнения операторов AND и OR зависит от того, какой операторов стоит первым

Что покажет следующий запрос: SELECT id FROM Orders WHERE year (date) > 2018

* номера заказов, сделанных до 2018 года
* номера заказов, сделанных в 2018 году
* **номера заказов, сделанных после 2018 года**

Когда оператор AND и OR совместно используются после WHERE …

* Нужно обязательно использовать скобки для каждого условия
* OR имеет больший приоритет, нежели AND
* **AND имеет больший приоритет, нежели OR**

Что делает спецсимвол ‘\_’ в паре с оператором LIKE: SELECT \* FROM Orders WHERE customer\_name LIKE ‘mik\_’

* **найдет все имена, которые начинаются на mik и состоят из 4 символов**
* найдет все имена, которые начинаются на mik, вне зависимости от того, из какого количества символов они состоят
* найдет данные, где имя равно mik

Что такое JOIN:

* **операция объединения**
* операция группировки
* операция суммирования

Выберите корректно составленный запрос с функцией GROUP BY:

* 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;

Какого строкового типа данных нет в SQL:

* VARCHAR
* **STRING**
* TEXT

Какой командой можно создать новую таблицу?

* **CREATE TABLE**
* MAKE TABLE
* SET TABLE

Исторически первой была предложена … модель базы данных:

* Реляционная
* **Иерархическая**
* Сетевая

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";

С помощью какого запроса можно удалить все записи из таблицы А?

* DELETE A
* **DELETE FROM A**
* DELETE TABLE A

Для какого из ключевых слов можно добавить строку «TABLE employees», чтобы запрос выполнился для таблицы employees?

* INSERT INTO
* ALTER
* **DROP**

Что такое первичный ключ (primary key)? Укажите наиболее точное определение

* Одна колонка, которая однозначно идентифицирует запись в таблице и может быть описана как автоинкремент
* Колонка, в которую можно писать только уникальные значения
* **Одна или несколько колонок, которые однозначно идентифицируют запись в таблице**

Выберите пример правильно составленного запроса с использованием агрегирующей функции SUM:

* **SELECT SUM(price) FROM Orders;**
* SELECT SUM(price), customer\_name FROM Orders;
* SELECT SUM() FROM Orders GROUP BY price DESC;

Укажите корректную команду INSERT из перечисленных:

* **INSERT INTO students (id, first\_name, last\_name) VALUES (100, ‘Kim’, ‘Matheson’)**
* INSERT INTO students (id, first\_name, last\_name) (100, ‘Kim’, ‘Matheson’)
* INSERT INTO students (id, first\_name, last\_name) SELECT id, first\_name FROM new\_students WHERE last\_name IS NULL

Выберите правильный пример использования функции округления ROUND:

* SELECT id, price \* discount AS total price FROM Orders ROUND (2);
* SELECT id, price \* discount ROUND (2) AS total price FROM Orders;
* **SELECT id, ROUND (price \* discount, 2) AS total price FROM Orders;**

Можно ли поменять тип данных поля в уже существующей таблице?

* **Да, при помощи команды ALTER**
* Да, достаточно сделать INSERT с новым типом данных
* Нет, только пересоздать таблицу

Вопросы для уровня «Middle»

Какие данные мы получим из этого запроса: SELECT id, date, customer\_name FROM Orders

* **Неотсортированные номера и даты всех заказов с именами заказчиков**
* Номера и даты всех заказов с именами заказчиков, отсортированные по первой колонке
* Номера и даты всех заказов с именами заказчиков, отсортированные по всем колонкам, содержащим слово Order

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**

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
* **Сравнение с NULL можно проводить только с оператором IS**
* Сравнение с NULL можно проводить только с оператором ON

Что покажет следующий запрос: SELECT CONCAT(‘index’," ", ‘city’) AS delivery\_address FROM Orders

* ничего, запрос составлен неверно
* **соединит поля с индексом и адресом из таблицы Orders и покажет их с псевдонимом delivery\_address**
* соединит поля с индексом и адресом из таблицы Orders, но покажет их без псевдонима

Что такое агрегирующие функции:

* функции, которые фильтруют значения
* **функции, которые работают с набором данных, превращая их в одно итоговое значение**
* функции, которые суммируют все значения

Для чего используется LIMIT: SELECT \* FROM Orders LIMIT 10

* необходим, чтобы показать все заказы, содержащие цифру 10
* **необходим, чтобы показать первых 10 записей в запросе**
* необходим, чтобы показать рандомные 10 записей в запрос

Что покажет следующий запрос: SELECT seller\_id, COUNT(\*) FROM Orders GROUP BY seller\_id HAVING seller\_id IN (2,4,6)

* **количество заказов, сгруппированное по продавцам 2, 4 и 6**
* ничего, запрос составлен неверно, HAVING указывается до группировки
* ничего, запрос составлен неверно, для указания условия должно быть использовано WHERE

Как можно заранее узнать, какие записи будут удалены при выполнении DELETE?

* Зачем заранее, просто вызвать его и посмотреть какие записи пропали
* Заменить DELETE на SELECT \*, ведь в остальном синтаксис DELETE похож на синтаксис простого SELECT
* SQL создан для хранения данных, их нельзя удалять

Создана таблица persons с колонками id, name, age. Какой запрос найдет средний возраст всех людей с возрастом не менее 18-ти лет?

* SELECT AVG(age) FROM persons WHERE age >= 18 GROUP BY name
* SELECT AVG(age) FROM persons HAVING age >= 18
* **SELECT ANG(age) FROM persons WHERE age >= 18**

Какие поля из таблицы обязательно перечислять в INSERT для вставки данных?

* Конечно все
* Только те, у которых нет DEFAULT значения
* **Те, у которых нет DEFAULT значения и которые не имеют атрибут auto\_increment**

Для чего применяются индексы в БД?

* Для объединения таблиц
* **Для ускорения доступа к данным**
* Для успешного завершения транзакций

Что верно про запрос: SELECT firstName, lastName FROM Employee WHERE lastName BETWEEN ‘A%’ AND ‘D%’

* **Будут отображаться все сотрудники, имеющие фамилии начинающиеся с ‘A’ до ‘D’ по алфавиту, включая A и исключая D**
* Будут отображаться все сотрудники c фамилиями, начиная с «A» и заканчивая «D»
* Будут отображаться все сотрудники, имеющие фамилии начинающиеся с ‘A’ до ‘D’ по алфавиту, исключая A и D

Отдельные типы объектов в БД – это ..

* Атрибуты
* Поля
* **Сущности**

Что такое «планируемый запрос» (предопределенный запрос)?

* **Часто повторяющийся запрос, предусмотренный заранее**
* Запрос, который выполняется по таймеру
* Запрос без вложенных подзапросов

Синонимом какого понятия является понятие «кортеж»?

* Поле
* **Запись**
* Первичный ключ

В каком виде хранится словарь данных?

* **В виде реляционных таблиц**
* В виде иерархической структуры
* В виде графа

Возможно ли использование одновременно двух агрегирующих функций: select min(price), max(price) from Orders

* да, но данный запрос составлен неверно, надо так: SELECT \* FROM Orders WHERE price IN (min, max);
* **да, в результате мы получим минимальную и максимальную стоимости**
* нет, две функции использовать одновременно нельзя

Что такое транзакция в теории БД (выберите наиболее подходящий и полный ответ)?

* Любое атомарное действие над хранимыми данными в БД
* **Некоторое действие над базой данных, которое переводит БД из одного непротиворечивого состояния в другое**
* Некоторое действие над базой данных, которое можно откатить в случае возникновения ошибки

Что отвечает за безопасность данных в БД?

* **Система управления базой данных**
* Пользовательское приложение
* Непосредственно сама база данных

Как сделать несколько записей в таблицу за один запрос?

* Использовать MULTI INSERT INTO вместо INSERT INTO
* Использовать подзапрос
* **Перечислить через запятую все наборы значений после VALUES**

Вопросы для уровня «Senior»

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**

Выберите корректный пример использования функции CONCAT:

* SELECT CONCAT = index AND city FROM Orders
* SELECT CONCAT IN (‘index’, ‘city’) FROM Orders
* **SELECT CONCAT(‘index’," ", ‘city’) FROM Orders**

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 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

Какого из перечисленных ниже видов JOIN на самом деле не существует:

* LEFT JOIN - который выведет все записи первой таблицы, а для ненайденных пар из правой таблицы проставит значение NULL
* INNER JOIN - который показывает только те записи, для которых нашлись пары
* **TRUE JOIN - который выведет все верные значения**

Чем отличается CHAR и VARCHAR?

* Это одно и то же
* CHAR - это тип данных, а VARCHAR - подтип
* **CHAR дополняет строку пробелами до максимальной длины, а VARCHAR тратит лишнюю память на хранение значения длины строки**

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**

Выберите пример корректно написанного запроса с использованием подзапроса, который выводит информацию о заказе с самой дорогой стоимостью:

* SELECT \* FROM Orders WHERE price = (SELECT BIG(price) FROM Orders)
* SELECT \* FROM Orders WHERE price = max
* **SELECT \* FROM Orders WHERE price = (SELECT MAX(price) FROM Orders)**

Зачем существует команда UPDATE, если можно сначала удалить запись, а потом добавить новую, исправленную?

* Так меньше нагрузки на базу, ведь команда одна, а не две
* **Потому что в записи могут быть автоматически проставляемые поля, такие как auto\_increment или timestamp, которые собьются при внесении записи заново**
* Как раз удалять записи в SQL нельзя, вместо этого используется UPDATE с NULL-значениями для всех полей

В каких командах можно использовать LIMIT?

* Только SELECT
* **SELECT, UPDATE, DELETE**
* SELECT, INSERT, DELETE, UPDATE

Что вернет запрос «SELECT 1 FROM USERS»? Если таблица USERS существует

* Запрос выведет первую строчку таблицы USERS
* **Запрос выведет «1» столько раз, сколько строк в таблице USERS**
* Ошибка выполнения запроса

Какой тип оператора имеет больший приоритет в выражении SQL из представленных:

* Умножение(\*), деление (/)
* **Унарные ‘+’ и ‘-’**
* AND

Модель данных, в которой каждый элемент может быть связан с любым другим элементом, а также может быть и главным, и подчиненным по отношению к другим элементам?

* **Сетевая**
* Реляционная
* Иерархическая

Что такое аномалия в теории БД?

* **Противоречия между моделью предметной области и физической моделью данных**
* Противоречия между моделью предметной области и логической моделью данных
* Противоречия между моделью предметной области и реляционной моделью данных

Как получить значение текущего года в SQL?

* SELECT now()
* SELECT year()
* **SELECT year(now())**

Какая наименьшая единица хранения данных в БД?

* Хранимый файл
* Хранимая запись
* **Хранимое поле**

Что такое коррелированный запрос?

* Запрос называется коррелированным, если состоит из двух (или более) запросов, объединенных через UNION ALL
* Любой запрос можно назвать коррелированным
* **Коррелированный запрос это один из видов вложенных запросов**

В базах данных хранимая и логическая записи

* Полностью совпадают
* Полностью отличаются
* **Могут как совпадать, так и быть различными**

С помощью какого update-запроса можно обновить значения более чем одной колонки?

* UPDATE table\_name SET col1\_name = ‘col1val’ AND SET col2\_name = ‘col2val’
* **UPDATE table\_name SET col1\_name = ‘col1val’, col2\_name = ‘col2val’**
* UPDATE table\_name SET col1\_name = ‘col1val’ SET col2\_name = ‘col2val’

Что из перечисленного не обязательно для приведения отношений к нормальной форме Бойса-Кодда?

* Функционально полная зависимость всех неключевых атрибутов от ключа
* Нетранзитивная зависимость всех неключевых атрибутов от первичного ключа
* **Отсутствие многозначных зависимостей, не являющихся функциональными зависимостями**

Как называется набор хранимых записей одного типа?

* Представление базы данных
* Логическая таблица базы данных
* **Хранимый файл**