1. Как найти дубликат записи? Опишите процесс для дублирования записей с одним и несколькими полями.

Дублирование записей с одним полем:

SELECT name, COUNT(email)

FROM users

GROUP BY email

HAVING COUNT(email) > 1

Дублирование записей с несколькими полями:

SELECT name, email, COUNT(\*)

FROM users

GROUP BY name, email

HAVING COUNT(\*) > 1

2. Как, используя CTE, найти пятый по величине оклад в таблице?

Declare @N int

set @N = 5;

WITH CTE AS

(

SELECT Name, Salary, EmpID, RN = ROW\_NUMBER()

OVER (ORDER BY Salary DESC)

FROM Employee

)

SELECT Name, Salary, EmpID

FROM CTE

WHERE RN = @N

3. Есть две таблицы, созданные и заполненные следующим образом:

CREATE TABLE dbo.envelope(id int, user\_id int);

CREATE TABLE dbo.docs(idnum int, pageseq int, doctext varchar(100));

INSERT INTO dbo.envelope VALUES

(1,1),

(2,2),

(3,3);

INSERT INTO dbo.docs(idnum,pageseq) VALUES

(1,5),

(2,6),

(null,0);

Какой будет результат выполнения запроса:

UPDATE docs SET doctext=pageseq FROM docs INNER JOIN envelope ON envelope.id=docs.idnum

WHERE EXISTS (

SELECT 1 FROM dbo.docs

WHERE id=envelope.id

);

Объясните свой ответ.

Результат будет выглядеть следующим образом:

idnum pageseq doctext

1 5 5

2 6 6

NULL 0 NULL

Оператор **EXISTS** – это отвлекающий маневр. Тут всегда будет true, т. к. **ID** не является членом **dbo.docs**. Таким образом, **ID** будет ссылаться на таблицу **envelope**, сравнивая себя с самим собой.

4. Что не так с этим запросом? Исправьте его, если нужно.

SELECT Id, YEAR(BillingDate) AS BillingYear

FROM Invoices

WHERE BillingYear >= 2010;

Неверное выражение BillingYear в условии **WHERE**. Несмотря на то, что он определен как алиас в выборке **SELECT** перед **WHERE**, логический порядок обработки условий отличается. Программисты привыкли к тому, что операторы выполняются сверху вниз или слева направо, но в [**T-SQL**](https://ru.wikipedia.org/wiki/Transact-SQL) это происходит в другом порядке.

Правильный запрос будет выглядеть так:

SELECT Id, YEAR(BillingDate) AS BillingYear

FROM Invoices

WHERE YEAR(BillingDate) >= 2010;

5. Учитывая данные из таблицы A:

x

------

2

-2

4

-4

-3

0

2

Напишите один запрос для вычисления суммы всех положительных и отрицательных значений **x**.

select sum(case when x>0 then x else 0 end)

sum\_pos,sum(case when x<0 then x else 0 end)

sum\_neg from a;

6. Как скопировать данные из одной таблицы в другую?

INSERT INTO table2 (column1, column2, column3, ...)

SELECT column1, column2, column3, ...

FROM table1

WHERE condition;

7. Дана таблица tbl и поля nmbr со следующими значениями:

1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1

Написать запрос, чтобы установить **2** вместо **0** и установить **3** вместо **1**.

update TBL set Nmbr = case when Nmbr = 0 then 2 else 3 end;

8. Предположим, есть две таблицы Emp (ID, name, DeptId), Dept(ID, name).

В таблице **Emp** 10 записей, а в **Dept** – 5. Сколько строк будет отображаться в результате следующего SQL-запроса:

Select \* From Emp, Dept

Объясните свой ответ.

Запрос выведет 50 строк, т. к. тут перекрестное объединение, являющееся значением по умолчанию, когда условие **WHERE**, отсутствует.

9. Напишите SQL-запрос, с применением UNION ALL (не UNION), использующий WHERE для устранения дубликатов.

SELECT \* FROM mytable WHERE a=X UNION ALL SELECT \* FROM mytable WHERE b=Y AND a!=X

Ключевое место – **AND** **a!=X**. Это дает вам все плюшки **UNION DISTINCT**, избегая при этом просадки производительности.

10. Чем отличается VARCHAR от NVARCHAR?

**NVARCHAR** способен хранить значения в формате Unicode – каждый символ занимает 2 байта, а кодировка занимает 65 536 символов.

**VARCHAR** хранит значения в формате ASCII – каждый символ занимает байт, а кодировка занимает 256 символов.

11. Что такое план запросов? Когда бы вы его использовали? Как посмотреть план?

План запросов – это "карта", отображающая графически или в виде текста способы получения данных, последовательность операций, сортировки, группировки и т. д. Эти планы очень полезны разработчику для понимания и анализа характеристик производительности запроса или хранимой процедуры.

Во многих системах SQL план выполнения можно получить с помощью ключевого слова **EXPLAIN**. В Microsoft SQL Server Query Analyzer имеет в раскрывающемся меню параметр “Show Execution Plan”. Если его активировать, то при выполнении запроса план будет отображаться в отдельном окне.

12. Как из таблицы выбрать все записи c четными ID? А с нечетными?

Это один из распространенных вопросов по SQL на собеседованиях.

Выбор четных записей:

Select \* from table where id % 2 = 0

Выбор нечетных записей:

Select \* from table where id % 2 != 0

13. Что делают функции NVL и NVL2? Чем они отличаются?

Как **NVL**(exp1, exp2), так и **NVL2**(exp1, exp2, exp3) исследуют значение exp1 на предмет NULL. Функция **NVL** заменяет значение, если встречается NULL. Функция **NVL2** проверяет значение exp1. Если оно не NULL, то возвращается exp2, иначе – exp3.

14. Важен ли в составном индексе порядок столбцов?

Важен. Когда строится составной индекс, сначала индексируется первый столбец, а потом строятся индексы других столбцов, на основании главного (первого).

15. В чем разница между однорядными и многорядными функциями? Для чего используется GROUP BY?

Однорядные функции одновременно работают только с одной строкой, а многорядные функции – с данными из нескольких строк.

Команда **GROUP BY** объединяет все записи, имеющие одинаковые значения в определенном поле или любой группе полей.

16. В чем разница между условиями WHERE и HAVING?

Если в запросе не используется **GROUP BY**, то условия **WHERE** и **HAVING** эквивалентны.

Но при использовании **GROUP BY**:

* Условие **HAVING** используется для фильтрации значений из группы (т. е. для проверки условий после выполнения агрегации в группы).
* Условие **WHERE** используется для фильтрации записей из результата до срабатывания какой-либо группировки.

17. Что выведет приведенный запрос, если в таблице Employee 10 записей?

BEGIN TRAN

TRUNCATE TABLE Employees

ROLLBACK

SELECT \* FROM Employees

Этот запрос вернет 10 записей, т. к. **TRUNCATE** выполняется в транзакции.

18. Можно ли явным образом задать значение поля AUTO\_INCREMENT?

Да, можно. Например, так:

SET IDENTITY\_INSERT TABLE1 ON

INSERT INTO TABLE1 (ID,NAME)

SELECT ID,NAME FROM TEMPTB1

SET IDENTITY\_INSERT OFF

19. Как получить последний id без использования функции max?

В MySQL:

select id from table order by id desc limit 1

В SQL Server:

select top 1 id from table order by id desc

20. Чем отличается IN от EXISTS?

Это еще один из популярных вопросов по SQL.

**IN**:

* Работает результирующим набором.
* Не применяется во вложенных запросах.
* Сравнивает все значения в списке результатов
* Имеет сравнительно низкую производительность при работе с большими результатами и подзапросами.

**EXISTS**:

* Работает с виртуальными таблицами.
* Используется со связанными запросами.
* Выводит результаты сравнения, если **true**.
* Обладает высокой производительностью, что облегчает обработку больших подзапросов.