

Вопросы по SQL

1. За что отвечает символ звёздочка (*) в SQL-запросе select?

Звездочка (*) в SQL указывает на выбор всех столбцов из таблицы.

2. Чем отличается оператор union от оператора join?

JOIN объединяет данные из таблиц на основе какого-то условия между ними, UNION объединяет несколько SELECT запросов.

JOIN объединяет данные в новые столбцы, UNION - в новые строки, причем для JOIN количество столбцов из каждой таблицы может быть различно и иметь различный тип данных, а для UNION количество столбцов и тип данных должны быть идентичны.

При JOIN запросе могут быть дубликаты, а UNION автоматически удаляет дубликаты.

3. Чем отличается union от union all?

UNION удаляет дубликаты, UNION ALL - оставляет их, поэтому он может работать быстрее.

4. Расставьте в хронологическом порядке операции над БД выполняемые в следующем запросе, написав номер операции над ней в таблице ниже:

```
select top 10 *
from weather w (nolock) left join holidays h (nolock)
    on w.date = h.date and h.event != 'Christmas_holidays'
where h.type = 'Weekend'
group by month(w.date)
having datepart(year, h.date) % 4 = 0
order by h.weekday
```

1	2	3	4	5	6	7
FROM	LEFT JOIN	WHERE	GROUP BY	HAVING	ORDER BY	SELECT

5. Если имеются две таблицы с совпадающими названиями полей (например таблицы Weather с полем date и таблица Holidays с полем date), объединенный join-ом по некоторому полю != date , а нужно вывести результат только одной таблицы. Каким образом это указывается в запросе?

Нужно использовать префикс таблицы:

```
SELECT W.*
FROM Weather AS W
JOIN Holidays AS H ON W.some_field = H.some_field;
```

6. Какая конструкция будет (в среднем) выполняться быстрее:

```
select
    Клиент,
    case when Источник_клиента = 'интернет-заявка'
        then '1'
    when Источник_клиента = 'оффлайн'
        then '2'
    end "Категория клиента"
from Clients (nolock)
```

=====

```

select
    Клиент,
    1 as Категория клиента
from Clients (nolock)
where Источник_клиента = 'интернет-заявка'
union all
select
    Клиент,
    2 as Категория клиента
from Clients (nolock)
where Источник_клиента = 'оффлайн'

```

UNION ALL будет быстрее, т.к. он не удаляет дубликаты и не требует проверки условий для каждой строки (как в CASE).

7. Имеются 2 поля в формате datetime, но из-за неудачного проектирования запись идет таким образом, что в одном поле дата, но без времени, а в другом поле – время, но без даты, нужно получить одно поле в котором будет и дата и время:

date (datetime, но время не пишется – оно всегда статично)	Time (пишется время, но день всегда статичен)	Нужное поле – содержит и время, и дату
2020-01-14 00:00:00:000	1970-01-01 01:38:00:000	2020-01-14 01:38:00:0000
2020-01-15 00:00:00:000	1970-01-01 12:56:00:000	2020-01-15 12:56:00:0000
2020-01-16 00:00:00:000	1970-01-01 11:58:04:000	2020-01-16 11:58:04:0000
2020-01-17 00:00:00:000	1970-01-01 08:38:07:000	2020-01-17 08:38:07:0000

```

SELECT
    CONCAT(CAST(date_field AS DATE), ' ', CAST(time_field AS TIME)) AS correct_datetime
FROM Tab;

```

8. Что является более быстрым при прочих равных при select-запросе из двух объединенных таблиц – указание условий в блоке join или указание условий в блоке where?

Производительность будет примерно идентична, т.к. план выполнения запросов будет эквивалентен.

9. Чем отличается row_number от rank?

ROW_NUMBER() возвращает уникальный номер строки, начиная с 1.

RANK() возвращает уникальный номер строки, начиная с 1. Если есть строки с одинаковым значением, то им присваивается одинаковый ранг. После группы таких строк, следующий ранг увеличивается на количество строк в такой группе.

10. Какой командой можно вернуть все строки таблицы, где поле Comment содержит хотя бы пробел или пустое выражение (но не NULL):

Пример таблицы

Id	Comment
1	NULL
2	_
3	Ada__
4	Не известныыю
5	Кто здесь??

```
SELECT * FROM Tab WHERE Comment LIKE '% %' or Comment LIKE '';
```

11. Напишите условие, которое вернет все строки, содержащие нижнее подчеркивание в поле Comment.

```
SELECT * FROM Tab WHERE Comment LIKE '%\_%' ESCAPE '\';
```

12. Перед вами таблица и запрос, формирующий, поле «Сумма», которое должно содержать сумму значений полей «Проценты» и «Основной долг». При этом поля «Проценты» и «Основной долг» не заполнялись значениями, если оплат не поступало. Какая была допущена ошибка при формировании поля Сумма, на ваш взгляд и как её исправить?

```
Select 'Проценты' + 'Основной долг' as 'Сумма' from Tab
```

Проценты	Основной долг	Сумма
10	8 000	8 010
12	9 000	9 012
NULL	11 000	NULL
NULL	80 000	NULL

Если в строке есть хотя бы одно значение NULL, то результат выражения будет NULL. Можно исправить с помощью COALESCE:

```
SELECT COALESCE('Проценты', 0) + COALESCE('Основной долг', 0) AS 'Сумма' FROM Tab;
```

13. Напишите запрос, возвращающий последнее действие клиента в хронологическом порядке на веб-странице в виде таблицы «Клиент такой-то» - «Переход на вкладку Займы»

Tab

Client	Date_time	Action
312	2020-01-01 14:22:01	Авторизация
312	2020-01-01 14:24:01	Переход на вкладку Займы
312	2020-01-01 14:25:01	Переход на вкладку Займы
312	2020-01-01 14:28:01	Досрочное гашение тела
312	2020-01-01 14:29:01	Выход из личного кабинета
5967	2020-01-01 14:10:01	Авторизация
5967	2020-01-01 14:12:01	Переход на вкладку Займы
5967	2020-01-01 14:15:01	Досрочное гашение тела
899	2020-01-01 14:40:01	Авторизация

```

WITH RankedActions AS (
    SELECT
        Client,
        Action,
        ROW_NUMBER() OVER (PARTITION BY Client ORDER BY Date_time DESC) AS RowNum
    FROM Tab
);
SELECT
    Client,
    Action
FROM RankedActions
WHERE RowNum = 1;

```

14. Имеется таблица

Tab

ID_CLIENT	ID_DOGOVOR	Interest_rate (Процентная ставка по договору)
1	1	1
2	2	1.2
1	3	1.2
1	4	1.5
2	5	2
3	6	2

Напишите запрос, который выведет список клиентов (без повторений), у которых процентная ставка больше 1.3%

```

SELECT DISTINCT ID_CLIENT
FROM Tab
WHERE Interest_rate > 1.3;

```

15. Напишите запрос, подсчитывающий по таблице из предыдущего задания, сколько у каждого из клиентов договоров

```

SELECT ID_CLIENT, COUNT(*) AS DOGOVOR_Count
FROM Tab
GROUP BY ID_CLIENT;

```

16. Имеются две таблицы: Dogovor (список договоров клиентов, уникальный ключ ID – номер договора) и Clients (содержит персональные данные клиента, уникальный ключ – ID- идентификатор клиента):

Dogovor

ID	ID_CLIENT	Interest_rate (Процентная ставка по договору)
1	1	1
2	2	1.2
3	1	1.2
4	1	1.5
5	2	2
6	3	2

Clients

ID	Second_name (фамилия)	Age (возраст)
1	Иванов	23
2	Кукина	42
3	Костицин	30

Напишите запрос, который вернет номера **договоров** тех клиентов, которым больше 25 лет

```
SELECT Dogovor.ID
FROM Dogovor
JOIN Clients ON Dogovor.ID_CLIENT = Clients.ID
WHERE Clients.Age > 25;
```

17. Имеются две таблицы, смысл таблиц не имеет значения

Tab1

Document_number	Date	Expiration_days
1	2019-01-01	7
2	2019-01-01	8
3	2020-03-03	9

Tab2

Document_number	Date
1	2019-01-01
2	2019-01-01

Имеется запрос по **left join** таблицы Tab2 к таблице Tab1 по полю **Date** (есть в обеих таблицах), благодаря которому возвращаются столбцы обеих таблиц (все):

Запрос написан в виде:

```
Select
    Tab1.Date,
    Tab1.Document_number as 'Первый документ',
    Tab2.Document_number as 'Второй документ'
From Tab1
Left join Tab2
    On Tab1.Date = Tab2.Date
```

Подумайте, сколько строк будет в результате его выполнения: 5

Нарисуйте таблицу с результатом

date	Первый документ	Второй документ
2019-01-01	1	1
2019-01-01	1	2
2019-01-01	1	1
2019-01-01	2	2
2020-03-03	3	NULL

Есть ли тут значения (NULL) или в каком-либо поле?

будут ли пустые значения (NULL) хоть в каком-нибудь поле?

Да, NULL будет в поле 'Второй документ'

Сколько будет строк, если left join заменить на inner join: 4

18. Имеется таблица по клиентам и договорам, заключенным с ними, напишите команду (часть запроса), которая бы проставляла порядковый номер договора по каждому клиенту

Пример таблицы и поля, в котором проставляется порядковый номер договора по клиенту

ID_CLIENT	ID_DOGOVOR	Порядковый номер договора клиента
1	1	1
2	2	1
1	3	2
1	4	3
2	5	2
3	6	1

```
SELECT
    ID_CLIENT,
    ID_DOGOVOR,
    ROW_NUMBER() OVER (PARTITION BY ID_CLIENT ORDER BY ID_DOGOVOR) AS 'Порядковый номер договора клиента'
FROM Tab;
```

19. Напишите конструкцию (часть запроса), которая считает число дней просрочки клиента по следующей таблице (а если клиент погасил ранее, чем дата по договору, должен проставляться нолик – нет просрочки)

Договор	Fact_expiration (фактическая дата закрытия договора)	Plan_expiration (Планируемая дата закрытия договора)	Формируемое поле по числу дней просрочки
1	2018-07-11	2018-08-10	0
2	2019-02-15	2019-01-25	20
3	2019-07-15	2019-07-14	1

```
SELECT
    ID,
    Fact_expiration,
    Plan_expiration,
    CASE WHEN Fact_expiration < Plan_expiration THEN 0
         ELSE DATEDIFF(DAY, Plan_expiration, Fact_expiration)
    END AS 'Формируемое поле по числу дней просрочки'
FROM Tab;
```

20. Имеется таблица по списку договоров клиентов, в ней есть поле D_DATEINPUT с датой заключения договора в формате "2019-05-01 18:12:15" типа datetime

Напишите команду для отделения от даты месяца (в каком виде будет возвращаться месяц – May или «05» или «5» - не имеет значения)

```
SELECT MONTH(D_DATEINPUT) AS Month FROM Tab;
```

21. Преобразуйте запись в формате строки '30-01-2020' в дату

```
SELECT CONVERT(DATE, '30-01-2020', 105);
```

22. Какие существуют функции, позволяющие представить запись формата datetime (например, 20181031

00:00:00:012)

В строку нужного вида (например '31 октября 2018'):

FORMAT:

```
SELECT FORMAT(CAST('20181031 00:00:00:012' As DATETIME), 'dd MMM yyyy', 'ru-RU');
```

В запись того же формата datetime, но в нужном виде (например 31.10.18):

FORMAT:

```
SELECT FORMAT(CAST('20181031 00:00:00:012' As DATETIME), 'dd.MM.yy');
```

CONVERT:

```
SELECT CONVERT(VARCHAR, CAST('20181031 00:00:00:012' As DATETIME), 4);
```

23. За что отвечает команда GO в SQL?

Команда GO в SQL необходима для разделения кода на батчи. Если в скрипте использовать только ;, то при возникновении хотя бы одной ошибки ничего не выполнится, а если после запросов использовать GO, то не будет обработан только запрос с ошибкой.

24. С помощью какой команды можно завести переменную?

```
DECLARE @Variable DataType;
```

```
SET @Variable = VALUE;
```

```
SELECT @Variable = Field FROM Tab WHERE (CONDITION);
```

25. Какие бывают типы данных в SQL?

Основные общие типы данных - это целочисленные типы, строковые типы и типы данных даты и время. Каждая категория также делится на конкретные подтипы. К тому же, в каждой СУБД реализованы дополнительные типы данных.

26. Соотнесите команды SQL с их смысловой нагрузкой:

Команда	Описание
Drop (if exists)	Удалить таблицу из базы данных (полностью, как объект, а не просто стереть))
Update	Изменить строки в таблицы в соответствии с некоторым условием
Insert	Добавить записи к уже существующей таблице
Select ... into... from ...	Создать новую таблицу с одновременным помещением туда записей
Create table ...	Создать пустую таблицу, дать наименования полям и назначить типы данных для полей
Alter table ... add...	Изменить типы данных полей или добавить новые столбцы
Alter table ... alter column ...	Добавить столбцы к существующей таблице
Truncate ...	Очистить таблицу, но не удалять её

27. Как сделать так, чтобы команды в SQL-скрипте выполнялись пошагово, а не одновременно? (например при создании таблицы, добавлении полей и изменения их типов)

Использовать команду GO для разделения на батчи.

28. Вернет ли ошибку нижеприведенный запрос или отработает без ошибок:

```
declare @name varchar = 'Nikolay'  
select * from Clients_offline where Name = @name  
GO  
select * from Clients_online where Name = @name
```

Запрос вернет ошибку, т.к. команда GO завершит батч, и переменная станет недоступной.