**Вопросы по SQL**

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

*Вернёт все столбцы из таблицы.*

1. *Чем отличается оператор* ***union*** *от оператора* ***join****?*

*JOIN выполняет горизонтальное (обычно, следуя какой-то логиуке) соединение таблиц. UNION – вертикальное соеднинение.*

*Грубо говоря, JOIN соединяет столбы из разных таблиц, UNION - строки*

1. *Чем отличается* ***union*** *от* ***union all****?*

*UNION – В результат включаются только уникальные строки.*

*UNION ALL – в результат включаются все строки.*

1. *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** | **8** |
| *from* | *on* | *Left join* | *where* | *Group by* | *having* | *select* | *Order by* |

1. *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?*

*Перед полем date указывается название таблицы. Напр. Weather.date*

1. *Какая конструкция будет (в среднем) выполняться быстрее: В среднем быстрее будет выполняться вторая, тк SELECT-ы быстрые операции*

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'

1. *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* |

1. *Использовать DATE(date) – для извлечения даты*
2. *Использовать TIME(time) – для извлечения времени*
3. *Скомпановать результат*
4. *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?*

*JOIN быстрее*

1. *Чем отличается* ***raw\_number*** *от* ***rank****?*

*Row\_number – функция нумерации, возвращает номер строки*

*Rank – возвращает ранг каждой строки. Идёт анализ значений, и в случае нахождения одинаковых, функция возвращает одинаковый ранг с пропуском следующего.*

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

*SELECT \* FROM table WHERE Comment IS NOT NULL;*

*Пример таблицы*

|  |  |
| --- | --- |
| *Id* | *Comment* |
| *1* | *NULL* |
| *2* | *\_* |
| *3* | *Ada\_\_\_* |
| *4* | *Не известноыыю* |
| *5* | *Кто здесь??* |

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

*WHERE Comment LIKE “%[\_]%”*

1. *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

Ошибка при формировании таблицы была в том, что если проценты отсутствуют, то значения нужно заполнять нулями. При создании таблицы можно было поставить полю «Проценты» NOT NULL

Исправить можно: IF(Проценты IS NULL, 0, Проценты)

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

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

*SELECT Client, Action FROM tab*

*WHERE Date\_time IN (SELECT MAX(Date\_time) FROM tab*

*GROUP BY Client) /\* При условии, что даты Date\_time уникально*

*Универсальный запрос*

*SELECT t.client, t.act FROM tab t*

*INNER JOIN (SELECT client, MAX(date\_time) as last\_action FROM tab*

*GROUP BY client) m*

*ON t.client = m.client AND t.Date\_time = m.last\_action*

*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* |

1. *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*

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

*SELECT COUNT(ID\_CLIENT) FROM Tab*

*GROUP BY ID\_CLIENT /\* предполагая, что у каждого клиента 1 договор*

1. *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*

*INNER JOIN Clients ON Dogovor.ID\_CLIENT = Clients.ID*

*WHERE Clients.Age > 25*

1. 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 столбцыобеих 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:**

2

4

5 – В результате выполнения будет 5 строк

6

**Нарисуйте таблицу с результатом**

|  |  |  |
| --- | --- | --- |
| **Date** | **Первый документ** | **Второй документ** |
| 2019-01-01 | **1** | **2** |
| 2019-01-01 | **1** | **1** |
| 2019-01-01 | **2** | **2** |
| 2019-01-01 | **2** | **1** |
|  | **3** | **Null** |

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

Будет, в последней строке поля «Второй документ»

**Сколько будет строк, если left join заменить на inner join**

2

3

4 – Будет 4 строки

5

1. 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 row\_number() over(partition BY ID\_CLIENT ORDER BY ID\_CLIENT

1. Напишите конструцию (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 |

IF(DATEDIFF(Fact\_expiration, Plan\_expiration) < 0, 0, DATEDIFF(Fact\_expiration, Plan\_expiration))

1. 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)

MONTHNAME(**D\_DATEINPUT**) – выдаст имя месяца

MONTH(**D\_DATEINPUT**)– выдаст цифру

1. Преобразуйте запись в формате строки ’30-01-2020’ в дату

STR\_TO\_DATE(DATE\_, '%d-%m-%Y')

1. Какие существуют функции, позволяющиеся представить запись формата datetime (например, 2018-10-31 00:00:00)

* в строку нужного вида (например ’31 октября 2018’)

DATE\_FORMAT(date\_, '%d %M %Y') или

SELECT TO\_CHAR(SYSDATE, 'YYYY-MM-DD') FROM time;

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

DATE\_FORMAT(date\_, '%d %m %Y')

1. За что отвечает команда GO в SQL?

Почти то же самое, что и ";", но означает конец партии. Пример:

Существующая таблица не будет вызвана, поскольку после первого запроса мы получим ошибку.

SELECT \* FROM table-that-does-not-exist;

SELECT \* FROM exist-table;

Этот запрос обработается

SELECT \* FROM table-that-does-not-exist

GO

SELECT \* FROM good-table

GO

1. С помощью какой команды можно завести переменную?

DECLARE @название\_переменной тип\_данных

1. Какие бывают типы данных в SQL?

INT, DECIMAL, VARCHAR, DATETIME

1. Соотнесите команды SQL с их смысловой нагрузкой:

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



1. Как сделать так, чтобы команды в SQL-скрипте выполнялись пошагово, а не одновременно? (например при создании таблицы, добавлении полей и изменения их типов)
2. 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

**Вернёт ошибку. Значение переменной устанавливается через SET**