**DATEDIFF**

Для вычитания двух дат используется функция DATEDIFF(дата\_1, дата\_2), результатом которой является количество дней между дата\_1 и дата\_2. Например,

DATEDIFF('2020-04-01', '2020-03-28')=4

DATEDIFF('2020-05-09','2020-05-01')=8

! Я тут не уверен входит ли первый день в результат, иногда надо добавить самому, смотри пример ниже.

Пример: Вывести информацию о командировках во все города кроме Москвы и Санкт-Петербурга (фамилии и инициалы сотрудников, город , длительность командировки в днях, при этом первый и последний день относится к периоду командировки). Последний столбец назвать Длительность. Информацию вывести в упорядоченном по убыванию длительности поездки, а потом по убыванию названий городов (в обратном алфавитном порядке).

**SELECT name, city, DATEDIFF(date\_last, date\_first)+1 AS'Длительность'**

**FROM trip**

**WHERE city NOT LIKE 'Москва' AND city NOT LIKE 'Санкт-Петербург'**

**ORDER BY Длительность DESC**

Пример: Используйте вложенный запрос, чтобы найти длительность самой короткой командировки.

**SELECT name, city, date\_first, date\_last**

**FROM trip**

**WHERE DATEDIFF(date\_last, date\_first) = (SELECT MIN(DATEDIFF(date\_last, date\_first))**

**FROM trip**

**)**

Пример: Вывести фамилию с инициалами и общую сумму суточных, полученных за все командировки для тех сотрудников, которые были в командировках больше чем 3 раза, в отсортированном по убыванию сумм суточных виде. Последний столбец назвать Сумма.

**SELECT name, SUM(per\_diem \* (DATEDIFF(date\_last,date\_first)+1) )AS'Сумма'**

**FROM trip**

**GROUP BY name**

**HAVING COUNT(name)>3**

**ORDER BY Сумма DESC**

**MONTH**

Для того, чтобы выделить номер месяца из даты используется функция MONTH(дата).

Например, **MONTH('2020-04-12') = 4.**

Если определяется месяц для значений столбца date\_first, то используется запись MONTH(date\_first)

Пример: Вывести информацию о командировках, начало и конец которых относятся к одному месяцу (год может быть любой). В результат включить столбцы name, city, date\_first, date\_last. Строки отсортировать сначала в алфавитном порядке по названию города, а затем по фамилии сотрудника.

**SELECT name, city, date\_first, date\_last**

**FROM trip**

**WHERE MONTH(date\_first)=MONTH(date\_last)**

**ORDER BY city, name**

**MONTHNAME**

Для того, чтобы выделить название месяца из даты используется функция MONTHNAME(дата), которая возвращает название месяца на английском языке для указанной даты. Например, MONTHNAME('2020-04-12')='April'.

Пример: Вывести название месяца и количество командировок для каждого месяца. Считаем, что командировка относится к некоторому месяцу, если она началась в этом месяце. Информацию вывести сначала в отсортированном по убыванию количества, а потом в алфавитном порядке по названию месяца виде. Название столбцов – Месяц и Количество.

**SELECT DISTINCT MONTHNAME(date\_first)AS'Месяц', COUNT(MONTHNAME(date\_first))AS'Количество'**

**FROM trip**

**GROUP BY Месяц**

**ORDER BY Количество DESC, Месяц**

**DAY, MONTH,YEAR**

В SQL есть функции, которые позволяют выделить часть даты: день(DAY()), месяц (MONTH()), год(YEAR()) . Например:

DAY('2020-02-01') = 1 **Это позволяет найти что-то в определённой дате.**

MONTH('2020-02-01') = 2

YEAR('2020-02-01') = 2020

Пример: Вывести сумму суточных (произведение количества дней командировки и размера суточных) для командировок, первый день которых пришелся на февраль или март 2020 года. Значение суточных для каждой командировки занесено в столбец per\_diem. Вывести фамилию и инициалы сотрудника, город, первый день командировки и сумму суточных. Последний столбец назвать Сумма. Информацию отсортировать сначала в алфавитном порядке по фамилиям сотрудников, а затем по убыванию суммы суточных.

**SELECT name, city, date\_first, per\_diem \* (DATEDIFF(date\_last,date\_first)+1)AS'Сумма'**

**FROM trip**

**WHERE (MONTH(date\_first)=2 AND YEAR(date\_first)=2020) OR (MONTH(date\_first)=3 AND YEAR(date\_first)=2020)**

**ORDER BY name, Сумма DESC**

**Генерация случайных чисел и как это использовать в генерации случайной даты**

Генерации случайных чисел в интервале от 0 до 1 (не включительно) осуществляется с помощью функции RAND(). Если эту функцию умножить на 365, то она будет генерировать вещественные числа от 0 до 365 (не включительно), здесь 365 выбрано только для примера, это работает с любым числом. Осталось только отбросить дробную часть. Это можно сделать с помощью функции FLOOR(), которая возвращает наибольшее целое число, меньшее или равное указанному числовому значению. Таким образом, случайное число от 0 до 365 можно получить с помощью выражения:

**FLOOR(RAND() \* 365)**

Для сложения даты с числом используется функция:

DATE\_ADD(дата, INTERVAL число единица\_измерения),

где

единица\_измерения (использовать прописные буквы) – это день (DAY), месяц(MONTH), неделя(WEEK) и пр.,

число – целое число, которое хотим прибавить к дате.

дата – значение даты или даты и времени.

Функция к дате прибавляет указанное число, выраженное в днях, месяцах и пр. , в зависимости от заданного интервала, и возвращает новую дату.

Например:

1)-DATE\_ADD('2020-02-02', INTERVAL 45 DAY) возвращает 18 марта 2020 года

2)-DATE\_ADD('2020-02-02', INTERVAL 6 MONTH) возвращает 2 августа 2020 года

1-здесь мы добавляем к дате 45 дней

2-здесь мы добавляем к дате 6 месяцев

Как вариант вместо фиксированного числа можно сгенерить случайное число. Например

DATE\_ADD('2020-01-01', INTERVAL FLOOR(RAND() \* 365) DAY)

здесь мы добавляем к дате рандомно сгенериное число, такой код позволяет генерировать каждый раз новую/случайную дату.