

Шпаргалка по стандартным функциям SQL. Часть 2

АГРЕГАЦИЯ И ГРУППИРОВКА

- **COUNT**(expr) – количество значений для строк в группе
- **SUM**(expr) – сумма значений в группе
- **AVG**(expr) – среднее значение для строк в группе
- **MIN**(expr) – минимальное значение в группе
- **MAX**(expr) – максимальное значение в группе
-

Получить количество строк в таблице:

```
SELECT COUNT(*)
FROM city;
```

Получить количество значений, отличных от

```
NULL, в столбце rating:
SELECT COUNT(rating)
FROM city;
```

Получить количество уникальных значений в

```
столбце country_id:
SELECT COUNT(DISTINCT country_id)
FROM city;
```

CITY			
name	country_id		
Paris	1		
Marseille	1		
Lyon	1		
Berlin	2		
Hamburg	2		
Munich	2		
Warsaw	4		
Cracow	4		

→

CITY		
country_id	count	
1	3	
2	3	
4	2	

Пример выше — количество городов в каждой стране:

```
SELECT name,
COUNT(country_id)
FROM city
GROUP BY name;
```

Средний рейтинг по городу:

```
SELECT city_id, AVG(rating)
FROM ratings
GROUP BY city_id;
```

Частая ошибка: COUNT(*) вместе с LEFT JOIN

Когда вы соедините таблицы следующие таблицы: client LEFT JOIN project, и захотите посчитать количество клиентов, с которыми есть связанные проекты, то применив count (*) вы получите количество записей, включающих пустые значения у проектов (клиент не связан с проектом). Для получения количества клиентов, связанных с проектами, нужно использовать count (project_name).

ДАТА И ВРЕМЯ

3 основных типа, связанных со временем: **date**, **time**, и **timestamp**. Время представлено 24-часовым форматом. Время может быть указано в часах и минутах (напр., 15:30 – 3:30 p.m.) или с точностью до миллисекунд с часовым поясом (как показано ниже) :

```
2021-12-31 14:39:53.662522-05
┌───┴───┐ ┌───┴───┐
└── date ┘ └── time ┘
└──────────┘
timestamp
YYYY-mm-dd HH:MM:SS.ssssss±TZ
```

14:39:53.662522-05 почти 2:40 p.m. CDT (например в Чикаго; в UTC это будет 7:40 p.m.). Символы в приведенном примере означают

Часть даты:

- YYYY – 4-й значный год.
- mm – дополненный нулем месяц (с 01 – Январь до 12 – Декабрь).
- dd – дополненный нулем день.

Часть времени:

- HH – дополненный нулем час в 24-часовом формате.
- MM – минуты.
- SS – секунды. ssssss – доли секунды – их можно выразить 6 цифрами. *Omissible*.
- ±TZ – таймзона. Он должен начинаться с + или - и использовать две цифры относительно UTC.

Который сейчас час?

Чтобы ответить на этот вопрос в SQL, вы можете использовать:

- CURRENT_TIME – найти текущее время
- CURRENT_DATE – получить текущую дату. (GETDATE() в SQL Server.)
- CURRENT_TIMESTAMP – чтобы получить текущую метку времени.

Заполнение значений (даты/времени)

Чтобы создать дату, время или метку времени, просто запишите значение в виде строки и приведите его к нужному типу.

```
SELECT CAST('2021-12-31' AS date);
SELECT CAST('15:31' AS time);
SELECT CAST('2021-12-31 23:59:29+02' AS timestamp);
SELECT CAST('15:31.124769' AS time);
```

Будьте осторожны с последним примером — он будет интерпретирован как 15 минут 31 секунда и 124769 миллисекунд! Всегда полезно написать 00 для выделения часов: '00:15:31.124769'.

Вы можете пропустить преобразование в простых условиях – база данных поймет, что вы имеете в виду:

```
SELECT airline, flight_number, departure_time
FROM airport_schedule
WHERE departure_time < '12:00';
```

ИНТЕРВАЛЫ

Примечание: В SQL Server интервалы не реализованы — используйте функции DATEADD() and DATEDIFF().

Чтобы получить простейший интервал, вычитите одно значение времени из другого:

```
SELECT CAST('2021-12-31 23:59:59' AS timestamp) - CAST('2021-06-01 12:00:00' AS timestamp);
-- результат: 213 days 11:59:59
```

Определение интервала: **INTERVAL '1' DAY**

Этот синтаксис состоит из трех элементов: ключевого слова INTERVAL, значения в кавычках и ключевого слова части времени (в единственном числе).

используйте следующие части времени: YEAR, MONTH, WEEK, DAY, HOUR, MINUTE, SECOND. В MySQL опустите кавычки. Вы можете соединять различные ИНТЕРВАЛЫ, используя оператор + или -:

```
INTERVAL '1' YEAR + INTERVAL '3' MONTH
```

В некоторых базах данных есть более простой способ получить указанное выше значение. И он принимает формы множественного числа! **INTERVAL '1 year 3 months'**

В стандартном SQL есть еще два синтаксиса:

Синтаксис	Что это значит
INTERVAL 'x-y' YEAR TO MONTH	ИНТЕРВАЛ 'x лет у месяцев'
INTERVAL 'x-y' DAY TO SECOND	ИНТЕРВАЛ 'x дней у секунд'

В MySQL пишите year_month вместо YEAR TO MONTH и day_second вместо DAY TO SECOND.

Чтобы получить последний день месяца, добавьте один месяц и вычитите один день:

```
SELECT CAST('2021-02-01' AS date)
+ INTERVAL '1' MONTH
- INTERVAL '1' DAY;
```

Получить все события на следующие три месяца с сегодняшнего дня:

```
SELECT event_date, event_name
FROM calendar
WHERE event_date BETWEEN CURRENT_DATE AND CURRENT_DATE + INTERVAL '3' MONTH;
```

Получить часть даты:

```
SELECT EXTRACT(YEAR FROM birthday)
FROM artists;
```

Одно из возможных возвращаемых значений: 1946. В SQL Server, используйте функцию DATEPART(part, date).

ЧАСОВЫЕ ПОЯСА

В стандарте SQL тип даты не может иметь связанного часового пояса, но типы времени и метки времени могут. В реальном мире часовые пояса мало что значат без даты, поскольку смещение может меняться в течение года из-за перехода на летнее время. Итак, лучше всего работать со значениями временной метки.

При работе с типом timestamp с часовым поясом (сокращенно timestampz) вы можете ввести значение в вашем местном часовом поясе, и оно будет преобразовано в часовой пояс UTC при вставке в таблицу. Позже, когда вы выбираете из таблицы, он преобразуется обратно в ваш местный часовой пояс. Это невосприимчиво к изменениям часового пояса.

ЧАСОВОЙ ПОЯС

Для работы между разными часовыми поясами используйте ключевое слово AT TIME ZONE.

Если вы используете этот формат: {timestamp without time zone} AT TIME ZONE {time zone}, то база данных прочитает отметку времени в указанном часовом поясе и преобразует для отображения времени в местном часовом поясе.

Если вы используете этот формат: {timestamp with time zone} AT TIME ZONE {time zone}, то база данных конвертирует время из текущего часового пояса в целевой определенный AT TIME ZONE. Он вернет время в формате timestamp, приведенного к целевому (заданному) часовому поясу без его конкретного указания в значении. Вы можете определить часовой пояс с помощью популярных сокращений, таких как UTC, MST или GMT, или по континенту/городу, например:

America/New_York, Europe/London, и Asia/Tokyo.

Примеры

Установим таймзону 'America/New_York'.

```
SELECT TIMESTAMP '2021-07-16 21:00:00' AT TIME ZONE 'America/Los_Angeles';
-- result: 2021-07-17 00:00:00-04
```

Здесь БД возьмёт время без часового пояса и по указанию значения в at time zone (America/Los Angeles) определит, что время в часовом поясе Лос Анджелеса, которое в итоге будет сконвертировано в значение местного часового пояса (например Нью Йорк) для отображения.

```
SELECT TIMESTAMP WITH TIME ZONE '2021-06-20 19:30:00' AT TIME ZONE 'Australia/Sydney';
-- result: 2021-06-21 09:30:00
```

Здесь база данных получает timestamp указанный в местной таймзоне и преобразует его во время в Sydney (обратите внимание, что он не вернул таймзону).