POPSQL

SQL Cheatsheet

Date & Time Functions

Get the date and time right now

POSTGRESQL | MYSQL | REDSHIFT | SNOWFLAKE | BIGQUERY

Date & Time

```
SELECT now();
-- timestamp
```

```
SELECT now();
```

```
SELECT now(); --string
SELECT sysdate; --timestamp
```

```
SELECT CURRENT_TIMESTAMP;
-- timestamp
```

```
SELECT CURRENT_DATETIME();
SELECT CURRENT_TIMESTAMP();
```

Date

```
SELECT current_date;
```

```
SELECT curdate;
```

```
SELECT current_date;
```

```
SELECT CURRENT_DATE;
```

```
SELECT CURRENT_DATE();
```

Time in timestamp format

```
SELECT now();
SELECT current_time;
```

```
SELECT curtime;
```

```
SELECT current_time;
```

```
SELECT CURRENT_TIME;
```

```
SELECT CURRENT_TIMESTAMP();
```

Find rows between two absolute timestamps

+5 POSTGRESQL, MYSQL, REDSHIFT, SNOWFLAKE & BIGQUERY

Assumes midnight of date specified

```
SELECT *
FROM table
WHERE time BETWEEN
'2019-01-01' AND '2019-01-31'
```

Specify in YYYY-MM-DD hh:mm:ss format to include time in result

```
SELECT *
FROM table
WHERE time BETWEEN
'2019-01-01 12:00:00'
AND '2019-01-31 23:30:00'
```

Find rows from last week

POSTGRESQL | MYSQL | REDSHIFT | SNOWFLAKE | BIGQUERY

```
SELECT *
FROM table
WHERE time > interval '1 week';
-- or '1 week'::interval, as you
like
```

```
SELECT *
FROM table
WHERE time > DATE_SUB(now(),
interval 1 week);
```

```
SELECT *
FROM table
WHERE time > DATEADD(week, -1,
sysdate);
```

```
SELECT *
FROM table
WHERE time > DATEADD(week, -1,
CURRENT_TIMESTAMP);
```

```
SELECT *
FROM table
WHERE time > TIMESTAMP_SUB(
    CURRENT_TIMESTAMP,
INTERVAL 168 HOUR);
```

Find rows between 1 and 2 weeks ago

POSTGRESQL | MYSQL | REDSHIFT | SNOWFLAKE | BIGQUERY

```
SELECT *
FROM table
WHERE time BETWEEN (now() -
interval '1 week') AND (now()
- interval '2 weeks');
```

```
SELECT *
FROM table
WHERE time BETWEEN
date_sub(now(), interval 2
week) AND date_add(now(),
interval 1 week);
```

```
SELECT *
FROM table
WHERE time BETWEEN
date_sub(now(), interval 2
week) AND date_add(now(),
interval 1 week);
```

```
SELECT *
FROM table
WHERE time BETWEEN
DATEADD(week, -2,
CURRENT_TIMESTAMP) AND
DATEADD(week, -1,
CURRENT_TIMESTAMP);
```

```
SELECT *
FROM table
WHERE time BETWEEN
TIMESTAMP_SUB(
CURRENT_TIMESTAMP,
INTERVAL 336 HOUR) AND
TIMESTAMP_SUB(
CURRENT_TIMESTAMP,
INTERVAL 168 HOUR);
```

Extract part of a timestamp

POSTGRESQL | MYSQL | REDSHIFT | SNOWFLAKE | BIGQUERY

```
SELECT date_part('year',
now());
-- or hour, day, month
```

```
SELECT year(now());
 -- or month(), day(), hour(),
minute(), second()
```

```
SELECT extract(minute from
sysdate); -- or hour, day,
month, year, century

SELECT date_part(minute,
sysdate); -- or hour, day,
month, year, century
```

```
SELECT EXTRACT(minute from
CURRENT_TIMESTAMP);
-- or hour, day, month, year
```

```
SELECT EXTRACT(MINUTE from
CURRENT_TIMESTAMP);
-- or hour, day, month, year
```

Get day of week from a timestamp

POSTGRESQL | MYSQL | REDSHIFT | SNOWFLAKE | BIGQUERY

Returns an integer corresponding to day of week

```
SELECT date_part('dow',now());
-- returns 0-6, where 0 is
Sunday and 6 is Saturday
```

```
SELECT
dayofweek('2019-12-12'); --
returns 1-7, where 1 is Sunday
and 7 is Saturday
```

```
SELECT extract(dow from
sysdate); -- returns 0-6, where
0 is Sunday and 6 is Saturday
```

```
SELECT extract(dow from
CURRENT_TIMESTAMP);
-- returns 0-6, where 0 is
Sunday and 6 is Saturday
```

```
SELECT extract(DAYOFWEEK from
CURRENT_TIMESTAMP);
-- returns 1-7, where 1 is
Sunday and 7 is Saturday
```

Returns a string like Monday or December

```
SELECT to_char(now(), 'day');
-- or month
```

```
SELECT dayname(now());
-- or monthname()
```

```
SELECT to_char(now(), 'day');
-- or month
```

```
SELECT DAYNAME(
CURRENT_TIMESTAMP);
 -- results abbreviated, e.g.
'Tue'. Use MONTHNAME() for
'Dec'.

SELECT TO_CHAR(
CURRENT_TIMESTAMP, 'DY'); --
gives full results, e.g.
'Tuesday'.For month, use 'MMMM'
```

```
SELECT FORMAT_DATE('%a',
CURRENT_DATE);
 -- results abbreviated, e.g.
'Tue'. Use '%A' for full
length. Use '%b' or '%B' for
abbreviated and full month,
respectively.
```

Convert a timestamp to a unix timestamp (integer seconds)

What is a Unix Timestamp?

The unix timestamp is a way to track time as a running total of seconds, ever since the Unix Epoch on January 1st, 1970 at UTC. This is very useful to computer systems for tracking and sorting dated information in dynamic and distributed applications both online and on the client side.

POSTGRESQL | MYSQL | REDSHIFT | SNOWFLAKE | BIGQUERY

Assumes midnight of date specified

```
SELECT date_part('epoch',
'2019-12-09');
```

```
SELECT date_part('epoch',
'2019-12-09');
```

```
SELECT DATE_PART(epoch,
TO_TIMESTAMP('2019-12-09'));
```

```
SELECT UNIX_SECONDS(
TIMESTAMP('2019-12-09'));
```

```
SELECT UNIX_SECONDS(
TIMESTAMP(
'2019-12-09 14:53:21'));
```

You can specify an exact timestamp to be converted down to the second

```
SELECT date_part('epoch',
'2019-12-09 14:53:21');
```

```
SELECT date_part('epoch',
'2019-12-09 14:53:21');
```

```
SELECT DATE_PART(epoch,
TO_TIMESTAMP('2019-12-09
14:53:21'));
```

```
SELECT unix_timestamp(
'2019-12-09 14:53:21');
```

```
SELECT unix_timestamp(
'2019-12-09');
```

Calculate the difference between two timestamps

POSTGRESQL | MYSQL | REDSHIFT | SNOWFLAKE | BIGQUERY

```
SELECT date_part('epoch',
'2019-12-31') -
date_part('epoch',
'2019-01-01'); -- or minute,
hour, week, day, etc


SELECT extract(epoch
from '2019-12-31')
-extract(epoch from
'2019-01-01');
-- alternatively done using
```

```
SELECT unix_timestamp(
'2019-12-31') -
unix_timestamp(
'2019-01-01');


SELECT
sec_to_time(unix_timestamp('20
19-12-31') -  unix_timestamp(
'2019-01-01'));
-- convert computed difference
to hh:mm:ss format
```

```
SELECT datediff(
    second,
    '2019-01-01',
    '2019-12-31'
);
 -- or hour, week, month, year
```

```
SELECT DATEDIFF(
    second,
    '2019-01-01',
    '2019-12-31'
);
 -- or hour, week, month, year
```

```
SELECT UNIX_SECONDS(TIMESTAMP(
'2019-12-31') -
UNIX_SECONDS(TIMESTAMP(
'2019-01-01');
 -- only need TIMESTAMP() if
data is not yet timestamp

SELECT TIMESTAMP_DIFF(
'2019-12-31 12:00:00',
'2019-01-01 12:00:00',SECOND);
-- alternatively you can use
TIMESTAMP_DIFF, or DATE_DIFF
depending on the datatype of
schema
```


Made by popsql.com