

Use the `DATE_TRUNC()` function to round off a timestamp. By aggregating data using rounded timestamps you can find time-based trends like daily purchases or messages sent per second. In the example below, the first column shows the full timestamp; the second and third columns show rounded timestamps.

This function requires two components, the unit of time and the timestamp column name. `DATE_TRUNC('interval', time_column):`

- | | | | |
|---|-------------|----|------------|
| 1 | microsecond | 8 | month |
| 2 | millisecond | 9 | quarter |
| 3 | second | 10 | year |
| 4 | minute | 11 | decade |
| 5 | hour | 12 | century |
| 6 | day | 13 | millennium |
| 7 | week | | |

SQL QUERY

```
1 SELECT created_at,
2       DATE_TRUNC('day', created_at) AS day,
3       DATE_TRUNC('minute', created_at) AS minute
4 FROM accounts
```

RESULT

	created_at	day	minute
1	2015-06-17 17:32:58	2015-06-17 00:00:00	2015-06-17 17:32:00
2	2015-05-02 22:30:42	2015-05-02 00:00:00	2015-05-02 22:30:00
3	2015-05-02 09:10:09	2015-05-02 00:00:00	2015-05-02 09:10:00

#DATAPOINTERS 003

Mastering the `DATE_TRUNC()` Function



modeanalytics.com

`DATE_TRUNC('interval', time_column)`

- microsecond
- millisecond
- second
- minute
- hour
- day
- week
- month
- quarter
- year
- decade
- century
- millenium

`DATE_TRUNC()` is particularly useful when you want to aggregate information over an interval of time.

```
select date_trunc('day',occurred_at),count(user_id)
FROM benn.fake_fact_events
where event_name='complete_signup'
and occurred_at >='2014-03-10'
and occurred_at <='2014-05-26'
group by 1
order by 1 desc
```

