Manipulating the granularity of time series data

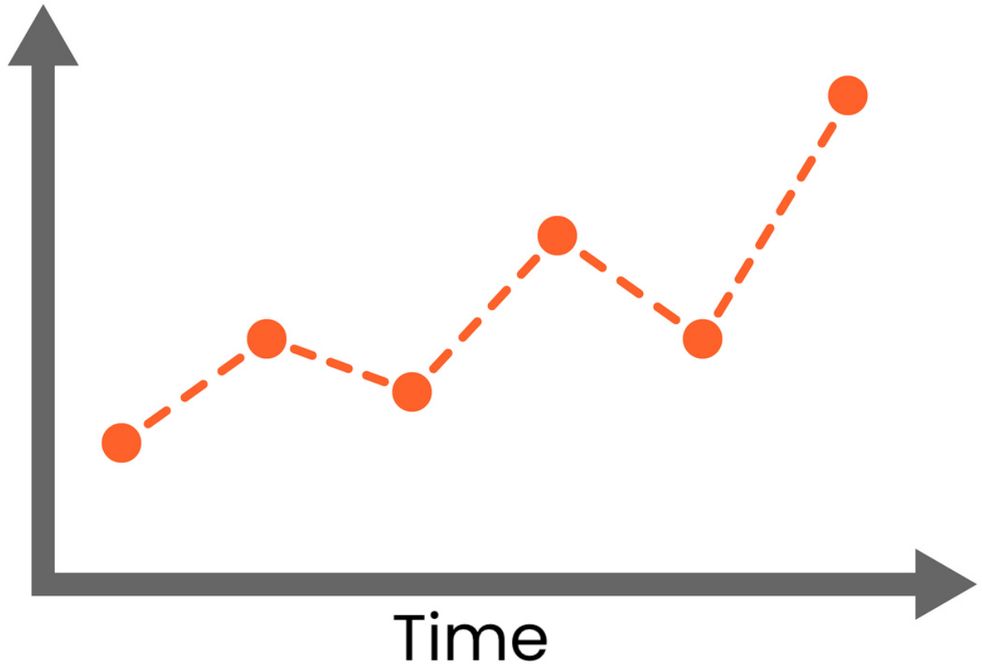
TIME SERIES ANALYSIS IN POSTGRESQL

Jasmin Ludolf
Content Developer, DataCamp

datacamp

Time series data

- Type of data and time data
- Ordered
- Collected over a period of time
- Common examples: stock prices, temperature



Time series data: train times

```
SELECT station, arrival_time
FROM train_schedule
WHERE train_id = 324
```

```
|station      |arrival_time|
|-------------|------------|
|San Francisco|    07:59:00|
|22nd Street  |    08:03:00|
|Millbrae     |    08:16:00|
|Hillsdale    |    08:24:00|
|Redwood City |    08:31:00|
|Palo Alto    |    08:37:00|
...
```

Multivariate time series

```
SELECT year_month, t_monthly_min, t_monthly_max, t_monthly_avg
FROM temperature_stations AS ts
JOIN temperatures_monthly AS tm USING(station_id)
WHERE station_id = 1
```

```
|year_month|t_monthly_min|t_monthly_max|t_monthly_avg|
|----------|-------------|-------------|-------------|
|2010-01-01|          6.7|         20.3|         13.6|
|2010-02-01|          8.3|         20.8|         14.8|
|2010-03-01|          9.1|         23.9|         17.0|
|2010-04-01|         11.3|         27.1|         20.0|
|2010-05-01|         15.0|         32.1|         24.8|
...
```

Time granularity

- More granularity = more precise measurements
- Common granularities: seconds, minutes, hours, days, weeks, months, quarters, and years

```
SELECT ts, views
FROM dc_news_fact
WHERE id = '12561'
ORDER BY ts;
```

```
|ts                   |views|
|---------------------|-----|
|2015-12-29 11:40:23| 4161|
|2015-12-29 12:00:23|  407|
|2015-12-29 12:20:23|    0|
|2015-12-29 12:40:23|  778|
|2015-12-29 13:00:23|  396|
...
```

Changing time granularity

- `DATE_TRUNC()` : alters granularity of time series data
  - Such as: century, decade, quarter, microseconds and more

- `DATE_TRUNC(field, source, time zone)`
  - field = granular value (eg. "hour")
  - source = the data
  - time zone = defaults to current setting

Changing time granularity

```
SELECT
    DATE_TRUNC('hour', ts) AS hour,
    SUM(VIEWS) AS views
FROM dc_news_fact
WHERE id = '12561'
GROUP BY hour
ORDER BY hour;
```

```
|hour                 |views|
|---------------------|-----|
|2015-12-29 09:00:00|    0|
|2015-12-29 10:00:00|    0|
|2015-12-29 11:00:00| 4161|
|2015-12-29 12:00:00| 1185|
|2015-12-29 13:00:00| 1146|
|2015-12-29 14:00:00|  697|
|2015-12-29 15:00:00| 1013|
|2015-12-29 16:00:00|  956|
|2015-12-29 17:00:00| 1307|
|2015-12-29 18:00:00|  700|
...
```

Extracting time granularity

- `DATE_PART()` : extracts specific data

- `DATE_PART(field, source)`
  - field = granular value (eg. "hour")
  - source = the data: a timestamp or interval

- `EXTRACT()` also extracts specific data
  - Recommended instead of `DATE_PART()`
  - `DATE_PART()` returns a result in double precision which is imprecise

Hour granularity

```
SELECT
    DATE_PART('hour', ts)
    AS hour_of_day,
    SUM(VIEWS) AS views
FROM dc_news_fact
WHERE id = '12561'
GROUP BY hour_of_day
ORDER BY hour_of_day;
```

```
|hour_of_day|views|
|-----------|-----|
|          0|  500|
|          1|  500|
|          2|  400|
|          3|  400|
|          4| 1200|
|          5|  100|
|          6|  100|
|          7| 2000|
|          8|  200|
|          9| 2500|
...
```

Day of week granularity

```
SELECT
    EXTRACT(dow FROM ts)
    AS day_of_week,
    SUM(VIEWS) AS views
FROM dc_news_fact
WHERE id = '12561'
GROUP BY day_of_week
ORDER BY day_of_week;
```

```
|day_of_week|views|
|-----------|-----|
|          2|15265|
|          3|13100|
|          4| 1200|
```

- Day of week: Sunday (0) to Saturday (6)

Let's practice!

TIME SERIES ANALYSIS IN POSTGRESQL

Adding and subtracting date and time data

TIME SERIES ANALYSIS IN POSTGRESQL

Jasmin Ludolf
Content Developer, DataCamp

datacamp

Time differences

- `AGE()` : subtracts the second argument from the first argument or the current date
- `AGE(timestamp, timestamp)` or `AGE(timestamp)`

```
SELECT
AGE('2018-01-02','2017-01-01') AS "2017",
AGE('2021-01-05','2020-01-01') AS "2020";
```

```
|2017        |2020          |
|------------|--------------|
|1 year 1 day|1 year 4 days|
```

Misinterpreting differences

```
SELECT
AGE('2017-12-31','2017-01-01') AS "2017",
AGE('2018-12-31','2018-01-01') AS "2018",
AGE('2019-12-31','2019-01-01') AS "2019",
AGE('2020-12-31','2020-01-01') AS "2020";
```

```
|2017           |2018           |2019           |2020           |
|---------------|---------------|---------------|---------------|
|11 mons 30 days|11 mons 30 days|11 mons 30 days|11 mons 30 days|
```

The subtract operator

```
SELECT
'2018-01-01'::DATE - '2017-01-01'::DATE AS "2017",
'2019-01-01'::DATE - '2018-01-01'::DATE AS "2018",
'2020-01-01'::DATE - '2019-01-01'::DATE AS "2019",
'2021-01-01'::DATE - '2020-01-01'::DATE AS "2020";
```

```
|2017|2018|2019|2020|
|----|----|----|----|
| 365| 365| 365| 366|
```

Using the subtract operator

- `-` : subtract operator
- Provides an `INTERVAL` data type
- `INTERVAL` allows us to store and manipulate a period of time

```
SELECT
    '2021-01-01 00:03:00'::TIMESTAMP - '2021-01-01 00:01:30'::TIMESTAMP
    AS interval;
```

```
|interval|
|--------|
|00:01:30|
```

Time intervals

```
WITH line_324 AS (
    SELECT station, arrival_time
    FROM train_schedule
    WHERE train_id=324 )
SELECT hillsdale.arrival_time - millbrae.arrival_time AS diff
FROM line_324 AS millbrae, line_324 AS hillsdale
WHERE millbrae.station='Millbrae'
AND hillsdale.station='Hillsdale';
```

- CTE : Common Table Expression, defines a temporary table for one query

[1] https://www.caltrain.com/schedules/weekdaytimetable.html

Time intervals

```
WITH line_324 AS (
    SELECT station, arrival_time
    FROM train_schedule
    WHERE train_id=324 )
SELECT hillsdale.arrival_time - millbrae.arrival_time AS diff
FROM line_324 AS millbrae, line_324 AS hillsdale
WHERE millbrae.station='Millbrae'
AND hillsdale.station='Hillsdale';
```

```
|diff    |
|--------|
|00:08:00|
```

[1] https://www.caltrain.com/schedules/weekdaytimetable.html

Subtracting an interval

```
SELECT
    '2020-02-01'::DATE - INTERVAL '1 month' AS "1 month sooner";
```

```
|1 month sooner     |
|-------------------|
|2020-01-01 00:00:00|
```

Converting an interval to a specified unit of time

- `EXTRACT(epoch FROM start_time - end_time)`

```
SELECT EXTRACT(epoch FROM
    ('2021-01-01 00:03:00'::TIMESTAMP - '2021-01-01 00:01:30'::TIMESTAMP))
    AS seconds;
```

```
|seconds|
|-------|
|   90.0|
```

Converting an interval to a specified unit of time

```
SELECT (EXTRACT(epoch FROM
    ('2021-01-01 00:01:00'::TIMESTAMP - '2020-12-31 23:59:30'::TIMESTAMP)))
    / 60 AS minutes;
```

```
|minutes|
|-------|
|    1.5|
```

Adding time

```
SELECT '2019-02-01'::DATE + INTERVAL '28 days' AS "28 days later";
```

```
|28 days later      |
|-------------------|
|2019-03-01 00:00:00|
```

```
SELECT '2020-02-01'::DATE + INTERVAL '28 days' AS "28 days later";
```

```
|28 days later      |
|-------------------|
|2020-02-29 00:00:00|
```

Adding a month to a date

```
SELECT '2019-02-01'::DATE + INTERVAL '1 month' AS "1 month later";
```

```
|1 month later       |
|--------------------|
|2019-03-01 00:00:00|
```

```
SELECT '2020-02-01'::DATE + INTERVAL '1 month' AS "1 month later";
```

```
|1 month later       |
|--------------------|
|2020-03-01 00:00:00|
```

Let's practice!

TIME SERIES ANALYSIS IN POSTGRESQL

Aggregating time series data

TIME SERIES ANALYSIS IN POSTGRESQL

Jasmin Ludolf
Content Developer, DataCamp

datacamp

Measuring the length of time series

- `dc_news_fact` : table with time series data for news articles
- `dc_news_dim` : table with the title of the articles

```
SELECT
    COUNT(*) AS length,
    title
FROM dc_news_fact
JOIN dc_news_dim USING(id)
GROUP BY title
ORDER BY length DESC;
```

Measuring the length of time series

```
|length|title                                                    |
|------|---------------------------------------------------------|
|   144|For the Wealthiest, a Private Tax System That Saves Them Billions|
|   144|These 5 charts prove that the economy does better under ...      |
|   144|Pet surrenders on rise as Fort McMurray's economy falls          |
|   144|How Is the Economy Doing? Politics May Decide Your Answer        |
|   144|Argentina's New President Moves Swiftly to Shake Up the Economy  |
```

Counting number of non-null entries in a time series

```
SELECT COUNT(views) AS nonnull, title
FROM dc_news_fact
JOIN dc_news_dim USING(id)
GROUP BY title
ORDER BY nonnull DESC;
```

```
|nonnull |title                          |
|--------|-------------------------------|
|     143|For the Wealthiest, a Pri...|
|     143|These 5 charts prove that...|
|     143|Pet surrenders on rise as...|
|     143|How Is the Economy Doing?...|
|     143|Argentina's New President...|
```

Counting number of non-zero entries in a time series

```
SELECT COUNT(views) AS nonzeros, title
FROM dc_news_fact
JOIN dc_news_dim USING(id)
WHERE views > 0
GROUP BY title
ORDER BY nonzeros DESC;
```

```
|nonzeros|title                        |
|--------|-----------------------------|
|      84|Pet surrenders on rise as...|
|      82|These 5 charts prove that...|
|      79|How Is the Economy Doing?...|
|      78|Argentina's New President...|
|      64|For the Wealthiest, a Pri...|
```

Calculating min and max over time series data

```
SELECT
    MIN(views) as min,
    MAX(views) as max,
    title
FROM dc_news_fact
JOIN dc_news_dim USING(id)
GROUP BY title
ORDER BY max DESC;
```

min	max	title
0	4161	For the Wealthiest, a Pri...
0	289	Argentina's New President...
0	141	Pet surrenders on rise as...
0	73	How Is the Economy Doing?...
0	53	These 5 charts prove that...

Summing time series data

```
SELECT SUM(views) as views, title
FROM dc_news_fact
JOIN dc_news_dim USING(id)
GROUP BY title
ORDER BY views DESC;
```

```
|views|title                              |
|-----|-----------------------------------|
|29565|For the Wealthiest, a Privat...|
| 1737|Pet surrenders on rise as Fo...|
| 1722|Argentina's New President Mo...|
| 1055|How Is the Economy Doing? Po...|
| 1043|These 5 charts prove that th...|
```

Adjusting time granularity

```
SELECT SUM(views) as views, DATE_TRUNC('day', ts) as date, title
FROM dc_news_fact
JOIN dc_news_dim USING(id)
GROUP BY title, date
ORDER BY title, date;
```

```
|views|date                 |title                                                |
|-----|---------------------|-----------------------------------------------------|
|   62|2015-12-27 00:00:00|Argentina's New President Moves Swiftly to Shake Up the ...|
|  414|2015-12-28 00:00:00|Argentina's New President Moves Swiftly to Shake Up the ...|
| 1246|2015-12-29 00:00:00|Argentina's New President Moves Swiftly to Shake Up the ...|
|15265|2015-12-29 00:00:00|For the Wealthiest, a Private Tax System That Saves Them...|
...
```

Adjusting time granularity

```
SELECT SUM(views) as views, ts::date as date, title
FROM dc_news_fact
JOIN dc_news_dim USING(id)
GROUP BY title, date
ORDER BY title, date;
```

```
|views|date      |title                                                    |
|-----|----------|---------------------------------------------------------|
|   62|2015-12-27|Argentina's New President Moves Swiftly to Shake Up the Economy |
|  414|2015-12-28|Argentina's New President Moves Swiftly to Shake Up the Economy |
| 1246|2015-12-29|Argentina's New President Moves Swiftly to Shake Up the Economy |
|15265|2015-12-29|For the Wealthiest, a Private Tax System That Saves Them Bill...|
...
```

Measuring the days

```
SELECT COUNT(DISTINCT ts::DATE) AS days, title
FROM dc_news_fact
JOIN dc_news_dim USING(id)
WHERE views > 0
GROUP BY title
ORDER BY days DESC, title;
```

```
|days|title                                                          |
|----|---------------------------------------------------------------|
|   3|Argentina's New President Moves Swiftly to Shake Up the Economy  |
|   3|For the Wealthiest, a Private Tax System That Saves Them Billions|
|   3|How Is the Economy Doing? Politics May Decide Your Answer        |
...
```

Let's practice!

TIME SERIES ANALYSIS IN POSTGRESQL

Applying statistical aggregates to time series data

TIME SERIES ANALYSIS IN POSTGRESQL

Jasmin Ludolf
Content Developer, DataCamp

datacamp

Statistical aggregates

- Aggregate functions: `MIN()`, `MAX()`, `SUM()`, `AVG()`, `COUNT()`
- Statistical aggregates: means and medians

Calculating the average

```
SELECT
  AVG(views)::INTEGER as avg_views
  title
FROM dc_news_fact
JOIN dc_news_dim USING(id)
GROUP BY title
ORDER BY avg_views DESC;
```

```
|avg_views|title                          |
|---------|-------------------------------|
|      207|For the Wealthiest, a Pr...|
|       12|Pet surrenders on rise a...|
|       12|Argentina's New Presiden...|
|        7|These 5 charts prove tha...|
|        7|How Is the Economy Doing...|
```

Average number of views per day

```
WITH day_views AS (
  SELECT id, ts::DATE AS date, SUM(views) AS views
  FROM dc_news_fact
  GROUP BY id, date
)
SELECT
  AVG(views)::INTEGER AS avg
  title
FROM day_views JOIN dc_news_dim USING(id)
GROUP BY title
ORDER BY avg DESC;
```

Average number of views per day

```
| avg|title                                                    |
|----|---------------------------------------------------------|
|9855|For the Wealthiest, a Private Tax System That Saves Them Billions|
| 579|Pet surrenders on rise as Fort McMurray's economy falls  |
| 574|Argentina's New President Moves Swiftly to Shake Up the Economy  |
| 352|How Is the Economy Doing? Politics May Decide Your Answer  |
| 348|These 5 charts prove that the economy does better under ...  |
```

Average number of views per day

```
SELECT
  (SUM(views)/COUNT(DISTINCT ts::DATE))::INTEGER as avg,
  title
FROM dc_news_fact JOIN dc_news_dim USING(id)
GROUP BY title
ORDER BY avg DESC;
```

```
|avg |title                                                    |
|----|---------------------------------------------------------|
|9855|For the Wealthiest, a Private Tax System That Saves Them Billions|
| 579|Pet surrenders on rise as Fort McMurray's economy falls  |
| 574|Argentina's New President Moves Swiftly to Shake Up the Economy  |
...
```

Discrete and continuous medians

- **Discrete median:** the first value closest to the middle value
- **Continuous median:** a value that cuts the dataset in half

Odd number of elements

- Series = (1, 2, 3, 4, 5)
- Discrete median = 3
- Continuous median = 3

Even number of elements

- Series = (1, 2, 3, 4)
- Discrete median = 2
- Continuous median = 2.5

Ordered-set aggregate functions

- `PERCENTILE_DISC()`
- `PERCENTILE_CONT()`
- Ordered-set aggregate functions :

```
PERCENTILE_DISC(fraction) WITHIN GROUP
(ORDER BY field)
```

```
SELECT
  PERCENTILE_CONT(0.5) WITHIN GROUP
  (ORDER BY value) AS median_cont,
  PERCENTILE_DISC(0.5) WITHIN GROUP
  (ORDER BY value) AS median_disc
FROM
(
  VALUES
  (1,1), (1,2), (1,3), (1,4), (1,5),
  (2,1), (2,5), (2,7), (2,11), (2,11)
) AS t (id, value)
GROUP BY id;
```

Ordered-set aggregate functions

```
|median_cont|median_disc|
|-----------|-----------|
|        3.0|          3|
|        7.0|          7|
```

Median, quantile, percentile, quartile

- Median is a type of percentile
- Percentile is a type of quantile

- **Quantile** : divides a sample into almost equal subsets
  - quartiles (four subsets)
  - deciles (ten subsets)

Calculating quartiles

```
SELECT
  PERCENTILE_DISC(0.25) WITHIN GROUP (ORDER BY value) AS ptile_25,
  PERCENTILE_DISC(0.5) WITHIN GROUP (ORDER BY value) AS ptile_50,
  PERCENTILE_DISC(0.75) WITHIN GROUP (ORDER BY value) AS ptile_75
FROM
(
  VALUES
  (1,1), (1,2), (1,3), (1,4), (1,5)
) AS t (id, value)
GROUP BY id;
```

```
|ptile_25|ptile_50|ptile_75|
|--------|--------|--------|
|       2|       3|       4|
```

Calculating an array of discrete quartiles

```
SELECT
  PERCENTILE_DISC(ARRAY[0.25, 0.5, 0.75])
  WITHIN GROUP (ORDER BY value) AS median_disc
FROM (
  VALUES
  (1,1), (1,2), (1,3), (1,4), (1,5) )
  AS t (id, value)
GROUP BY id;
```

```
|median_disc|
|-----------|
|{2,3,4}    |
```

Let's practice!

TIME SERIES ANALYSIS IN POSTGRESQL