SQL For Data Analysis

A Language for Querying Structured Data

Anders Poirel

March 9, 2020

Data Science @ SC

SQL

- language for querying structured data
- structured data: data in row-column form where each row corresponds to a datapoint
- each database is composed of several tables with the above structure
- SQL is standardized but most implementations aren't compliant

SQLite

- A lightweight database using static files on a disk
- Good for prototyping a database before switching to something more complicated if needed
- In practice use SQL Alchemy to make code more portable between databases

Select queries

Structure of a SELECT query:

```
SELECT DISTINCT column, AGG_FUNC(column), ...
FROM mytable
    JOIN anothertable
    ON mytable.column = anothertable.column
WHERE expression
GROUP BY column
HAVING expression
ORDER BY column ASC/DESC
LIMIT count OFFSET count
```

Select

To select a particular column,

```
SELECT (DISTINCT) column
FROM mytable
```

To select all columns,

```
SELECT *
FROM mytable
```

Select Constraints (1)

To constrain the output of SELECT,

```
SELECT column1, ...
WHERE
    condition1
    AND/OR condition2 ...
```

Select Constraints (2)

Common operators in WHERE Expressions:

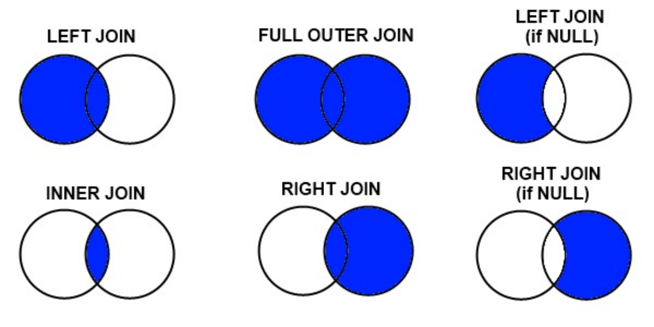
- =, !=, ¡, ¿, ¡=, ¿=
- BETWEEN ... AND ..
- NOT BETWEEN ... AND ..
- LIKE
- NOT LIKE
- %adcd%
- IN (...)
- NOT IN (...)

Joins (1)

To merge two tables based on a unique identifer,

```
SELECT column, ...
FROM mytable
INNER/LEFT/RIGHT/FULL JOIN anothertable
    ON mytable.id = anothertable.id
```

Joins (2)


LEFT JOIN
FULL OUTER JOIN
LEFT JOIN
(if NULL)
INNER JOIN
RIGHT JOIN
RIGHT JOIN
(if NULL)

Sorting

To sort based on a column

```
SELECT column, ...
FROM mytable
ORDER BY column ASC/DESC
```

Aggregates (1)

To aggregate by category,

```
SELECT AGG_FUNC(column)
WHERE expression
GROUP BY column
HAVING expression
```

Aggregates (2)

Common aggregates:

- AVG
- SUM
- MIN
- MAX

Select Queries Recap

```
SELECT DISTINCT column, AGG_FUNC(column), ...
FROM mytable
    JOIN anothertable
    ON mytable.column = anothertable.column
WHERE expression
GROUP BY column
HAVING expression
ORDER BY column ASC/DESC
LIMIT count OFFSET count
```