Worksheet 2 - SQL clauses

Jo Hardin

2023-01-09

Your Name: _____

Names of people you worked with: _____

- Introduce yourself. What is your favorite type of bread? And, more importantly, have you ever made it yourself?
- How are you feeling about using GitHub for retrieving and submitting assignments? What is difficult about it? What is great about it?

Task:

The airlines SQL database contains information on 48 million flights from the Bureau of Transportation Statistics (you have worked with a small subset of this data in the nycflights13 package). Information from the database can be obtained through SQL queries. For example, the flights database contains the following tables:

```
SHOW TABLES;
```

Table 1: 4 records

Tables_in_airlines
airports
carriers
flights
planes

An example query

Below is an SQL query on the database, and the output of the query. Working with your neighbor, try to determine what each piece of the query is doing. What would be the equivalent dplyr code?[1]

```
SELECT
  name,
  SUM(1) AS N,
  SUM(arr_delay <= 15) / SUM(1) AS pct_ontime
FROM flights
JOIN carriers ON flights.carrier = carriers.carrier
WHERE year = 2016 AND month = 9
  AND dest = 'BOS'
GROUP BY name
HAVING N >= 100
ORDER BY pct_ontime DESC
LIMIT 0, 4;
```

Table 2: 4 records

name	N	pct_ontime
Virgin America	145	0.8069
Alaska Airlines Inc.	146	0.8014
Delta Air Lines Inc.	1257	0.7868
American Airlines Inc.	2277	0.7787

[1] Taken from: https://mdsr-book.github.io/mdsr3e/15-sqlI.html#sec-dplyr-sql

Solution:

The order of the **SQL** clauses is not necessarily the same as the order in **R** (or as you might say out loud).

- Only flights from September 2016 to 'BOS' are considered.
- `JOIN` combines the `flights` and `carriers` tables so that the name of the airline is connected to the actual flight. The function `JOIN` does an inner join (intersection).
- Looking at each airline separately, count the number of flights and the percent of flights that are on time.
- Keep only the airlines that have at least 100 flights into 'BOS'.
- Sort the values according to percent on time, with the highest percent on time listed first.
- Print the first 4 rows only.

```
flights <- dplyr::tbl(con_air, "flights")
carriers <- dplyr::tbl(con_air, "carriers")

flights |>
  filter( year == 2016 & month == 9 & dest == 'BOS') |>
  inner_join(carriers, by = "carrier") |>
  group_by(name) |>
  summarize(N = n(), pct_ontime = sum(arr_delay <= 15) / n()) |>
  filter(N >= 100) |>
  arrange(desc(pct_ontime)) |>
  head(4)
```

```
# Source:     SQL [4 x 3]
# Database:   mysql  [mdsr_public@mdsr.cdc7tgkkqd0n.us-east-1.rds.amazonaws.com:NA/airlines]
# Ordered by: desc(pct_ontime)
  name                          N pct_ontime
  <chr>                 <int64>      <dbl>
1 Virgin America            145      0.807
2 Alaska Airlines Inc.      146      0.801
3 Delta Air Lines Inc.     1257      0.787
4 American Airlines Inc.   2277      0.779
```