Lecture 18: Intro to SQL

Data stored in multiple tables

The `nycflights13` package contains information on flights from NYC airports in 2013. The data is stored across several data frames:

- `airlines`: information on each airline
- `airports`: information on each airport
- `flights`: information on each flight
- `planes`: information on each plane
- `weather`: hourly weather data

Limitations

```
1 nycflights13::flights |>
2   object.size() |>
3   print(units = "Mb")
```

```
38.8 Mb
```

- R stores objects in memory (RAM), which can be easily accessed
- The amount of RAM on your computer is a limit on the possible size of objects
- Objects larger than a few Gb are generally too big to load

Full airlines data

The `nycflights13` package contains a small subset of a database on 48 million flights. The `airlines` database includes the following tables:

- `airports`
- `carriers`
- `flights`
- `planes`

This data is too big to store locally, but can be on servers which we can access remotely.

Connecting to an SQL server

```
1  library(tidyverse)
2  library(mdsr)
3  library(DBI)
4
5  db <- dbConnect_scidb("airlines")
6
7  query <- "
8    SHOW TABLES;
9  "
10 dbGetQuery(db, query)
```

```
  Tables_in_airlines
1           airports
2           carriers
3            flights
4             planes
```

airlines database

connecting to remote server

represent query in R as a string

query that we want

name of the connection

need ; at the end of each line

An example query

```
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 = 'JFK'
GROUP BY name
HAVING N >= 100
ORDER BY pct_ontime DESC
LIMIT 0,4;
```

Warm-up

https://sta279-f23.github.io/class_activities/ca_lecture_18_warmup.html

Warm-up

AS : naming (column, table, etc.)

What do you think each part of this query is doing?

```
1  SELECT
2    name,
3    SUM(1) AS N,
4    SUM(arr_delay <= 15) / SUM(1) AS pct_ontime
5  FROM flights
6  JOIN carriers ON flights.carrier = carriers.carrier
7  WHERE year = 2016 AND month = 9
8    AND dest = 'JFK'
9  GROUP BY name
10 HAVING N >= 100
11 ORDER BY pct_ontime DESC
12 LIMIT 0,4;
```

counting # rows (= # flights)

choosing columns to display

fraction of "on time" flights

← inner join (linking "carrier" in flights and carriers)

like filter in dplyr (keep rows in "flights" where year = 2016, month = 9, dest = 'JFK')

second filter

only display first 4 rows

	name	N	pct_ontime
1	Delta Air Lines Inc.	2396	0.8689
2	Virgin America	347	0.8329
3	JetBlue Airways	3463	0.8169
4	American Airlines Inc.	1397	0.7817

calculate pct_ontime for each airline

ordered by pct_ontime in descending order

LIMIT 0, 4

rows to skip

rows to read in

General structure of an SQL query

```
1 SELECT ...
2 FROM ...
3 JOIN ...
4 WHERE ...
5 GROUP BY ...
6 HAVING ...
7 ORDER BY ...
8 LIMIT ...
```

- The SELECT and FROM clauses are *required*
- Clauses must be written in this order

SELECT ... FROM

take the first 10 rows

take all the columns

table to get data from

```
1 SELECT * FROM carriers LIMIT 0, 10;
```

	carrier	name
1	02Q	Titan Airways
2	04Q	Tradewind Aviation
3	05Q	Comlux Aviation, AG
4	06Q	Master Top Linhas Aereas Ltd.
5	07Q	Flair Airlines Ltd.
6	09Q	Swift Air, LLC
7	0BQ	DCA
8	0CQ	ACM AIR CHARTER GmbH
9	0GQ	Inter Island Airways, d/b/a Inter Island Air
10	0HQ	Polar Airlines de Mexico d/b/a Nova Air

- `SELECT`: the columns to be retrieved
- `FROM`: the table containing the data
- `LIMIT`: limit the rows to return

SELECT ... FROM

```
1 SELECT ... FROM ... LIMIT 0, 10;
```

What if I want the `year`, `origin`, `dest`, `dep_delay`, and `arr_delay` columns from the `flights` table?

SELECT ... FROM

What if I want the `year`, `origin`, `dest`, `dep_delay`, and `arr_delay` columns from the `flights` table?

```
1 SELECT
2   year, origin, dest,
3   dep_delay, arr_delay
4 FROM flights
5 LIMIT 0, 5;
```

```
  year origin dest dep_delay arr_delay
1 2010    EWR  OMA       181       159
2 2010    FLL  SWF       281       256
3 2010    JFK  SJU         8         5
4 2010    IAD  BNA       125       112
5 2010    LAX  FAT        82        77
```

SELECT ... FROM

```sql
SELECT
  year, origin, dest,
  dep_delay, arr_delay
FROM flights
LIMIT 0, 5;
```

What if I also want to calculate the difference between arrival delay and departure delay?

arr_delay - dep_delay AS delay_diff

SELECT ... FROM

What if I also want to calculate the difference between arrival delay and departure delay?

mutating, summarizing, and choosing

```
1 SELECT
2   year, origin, dest, dep_delay, arr_delay,
3   arr_delay - dep_delay AS delay_diff
4 FROM flights
5 LIMIT 0, 3;
```

	year	origin	dest	dep_delay	arr_delay	delay_diff
1	2010	EWR	OMA	181	159	-22
2	2010	FLL	SWF	281	256	-25
3	2010	JFK	SJU	8	5	-3

What are the equivalent `dplyr` functions?

- mutate to create the new column
- select to choose a subset of columns

Converting from R to SQL

```r
flights <- tbl(db, "flights")

flights |>
  select(year, origin, dest, dep_delay, arr_delay) |>
  mutate(delay_diff = arr_delay - dep_delay) |>
  head() |>
  show_query()
```

```
<SQL>
SELECT
  `year`,
  `origin`,
  `dest`,
  `dep_delay`,
  `arr_delay`,
  `arr_delay` - `dep_delay` AS `delay_diff`
FROM `flights`
LIMIT 6
```

Calculating summary statistics

Back to our original SQL query:

```
1  SELECT
2    SUM(1) AS N,
3    SUM(arr_delay <= 15) / SUM(1) AS pct_ontime
4  FROM flights
5  LIMIT 0, 10;
```

```
         N pct_ontime
1 47932811     0.8222
```

Calculating summary statistics

SELECT can also be used to calculate summary statistics. For example, if we want the average departure delay:

```
1  SELECT
2    AVG(dep_delay) AS mean_dep_delay
3  FROM flights
4  LIMIT 0, 10;
```

```
  mean_dep_delay
1         8.9586
```

WHERE

Now suppose that I only want the mean departure delay for flights from EWR in 2013:

notice that the ordering is different from dplyr

```
1  SELECT
2    AVG(dep_delay) AS mean_dep_delay
3  FROM flights
4  WHERE year = 2013 AND origin = 'EWR'
5  LIMIT 0, 10;
```

```
  mean_dep_delay
1         14.703
```

What do you think should I do if I want the mean delay for each airport in November 2013?

WHERE month = 11 AND year = 2013

GROUP BY origin

GROUP BY

```
1  SELECT
2    AVG(dep_delay) AS mean_dep_delay
3  FROM flights
4  WHERE year = 2013 AND month = 9 11
5  GROUP BY origin
6  LIMIT 0, 10;
```

	mean_dep_delay
1	6.3220
2	2.2489
3	6.7138
4	-4.7167
5	1.6506
6	7.0526
7	2.3741
8	21.8136
9	-12.7778
10	-2.9286

Do you notice anything about this output?

Don't know which origin corresponds to each statistic

Add origin to SELECT

GROUP BY

```sql
SELECT
  origin,
  AVG(dep_delay) AS mean_dep_delay
FROM flights
WHERE year = 2013 AND month = 9
GROUP BY origin
LIMIT 0, 10;
```

	origin	mean_dep_delay
1	ABE	6.3220
2	ABI	2.2489
3	ABQ	6.7138
4	ABR	-4.7167
5	ABY	1.6506
6	ACK	7.0526
7	ACT	2.3741
8	ACV	21.8136
9	ADK	-12.7778
10	ADQ	-2.9286

Class activity

https://sta279-f23.github.io/class_activities/ca_lecture_18.html