CORNELL SPORTS ANALYTICS

CORNELLSPORTSANALYTICS.COM

CSA SQL WORKSHOP

Katherine Faiola & Max Griner

WHAT is SQL?

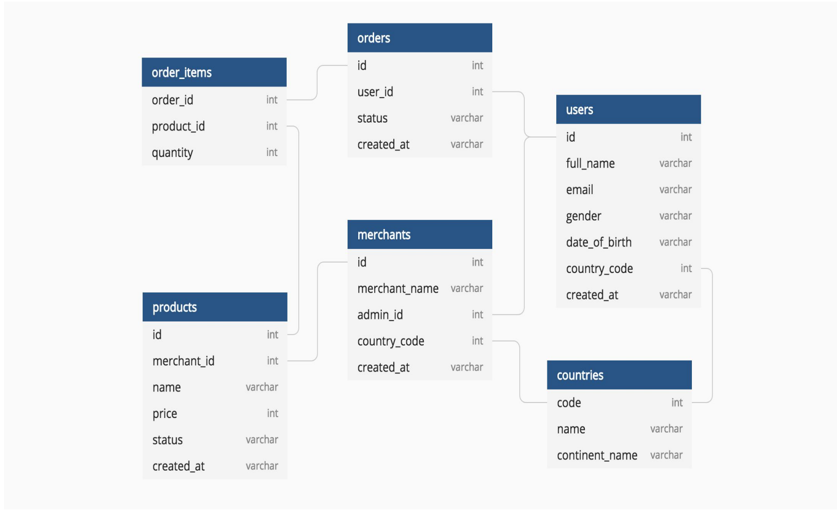
Structured Query Language

Language allowing us to manipulate and explore data

WHY SQL...?

- Data is important!
- Breaks down each step of the data retrieval and updating process into small chunks called **queries**
    - Help with obtaining stored data
- Allow us to **clean data**, an integral step in any data science project
    - Visualize messy data in manageable pieces to identify flaws

DATABASES



- A collection of linked tables each storing a different subset of data
    - Each table serves its own purpose
- Useful to combine tables
    - **SQL** helps us with this
- Necessary to support most major websites

DATA SOURCES

- Kaggle
  - Source of the NBA data featured in this workshop
    - https://www.kaggle.com/nathanlauga/nba-games?select=games_details.csv
      - Overview of NBA games from the 2019 Season to the 2003 Season
      - Overall game stats, player stats within games, and final standings for each season
      - Unique ID columns allow for easy organization and connections between these tables

CORNELL SPORTS ANALYTICS

CORNELLSPORTSANALYTICS.COM

KEYWORDS (the BASIS of SQL)

SELECT*

- Everything--literally *everything*--in SQL revolves around SELECT statements!

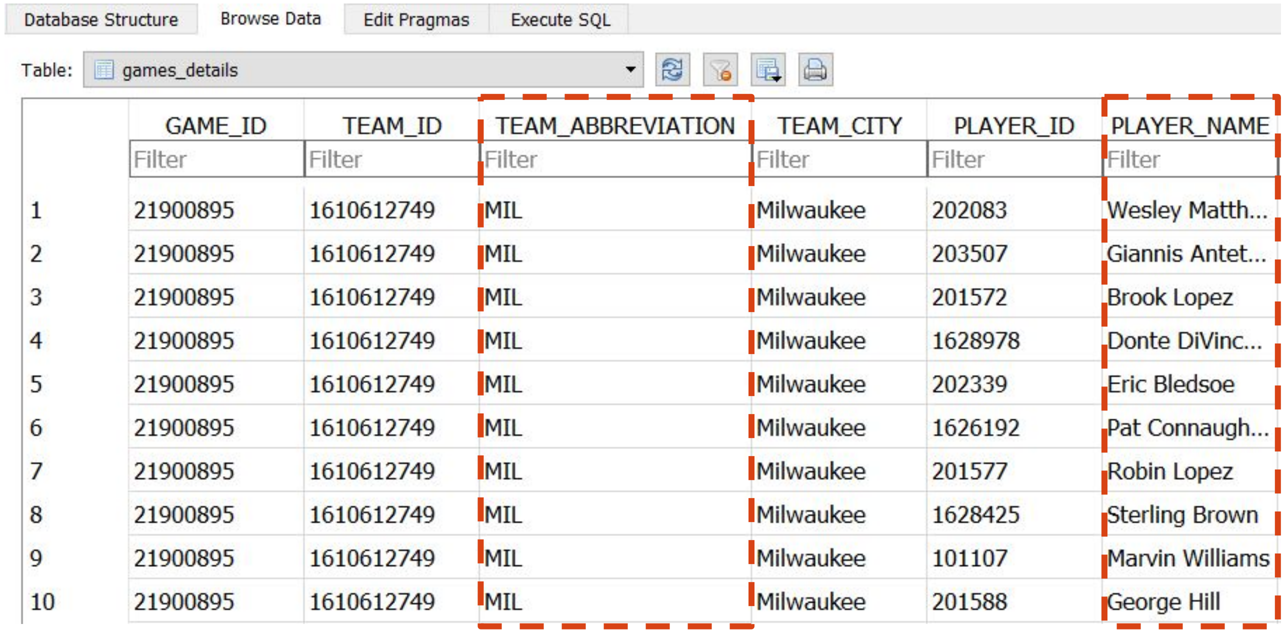
- **SELECT** <column(s)> **FROM** <table name> **;**
    - This is a select statement that grabs the specified columns from the given table
    - To select ***all*** columns from a table use ***** in place of <column(s)>

- The **semicolon (;)** marks the end of the statement

Note: SQL keywords like 'select' and 'from' are case-INsensitive, but typically all-caps can be a conventional way of writing them that makes it easy to read

SQLite DEMO

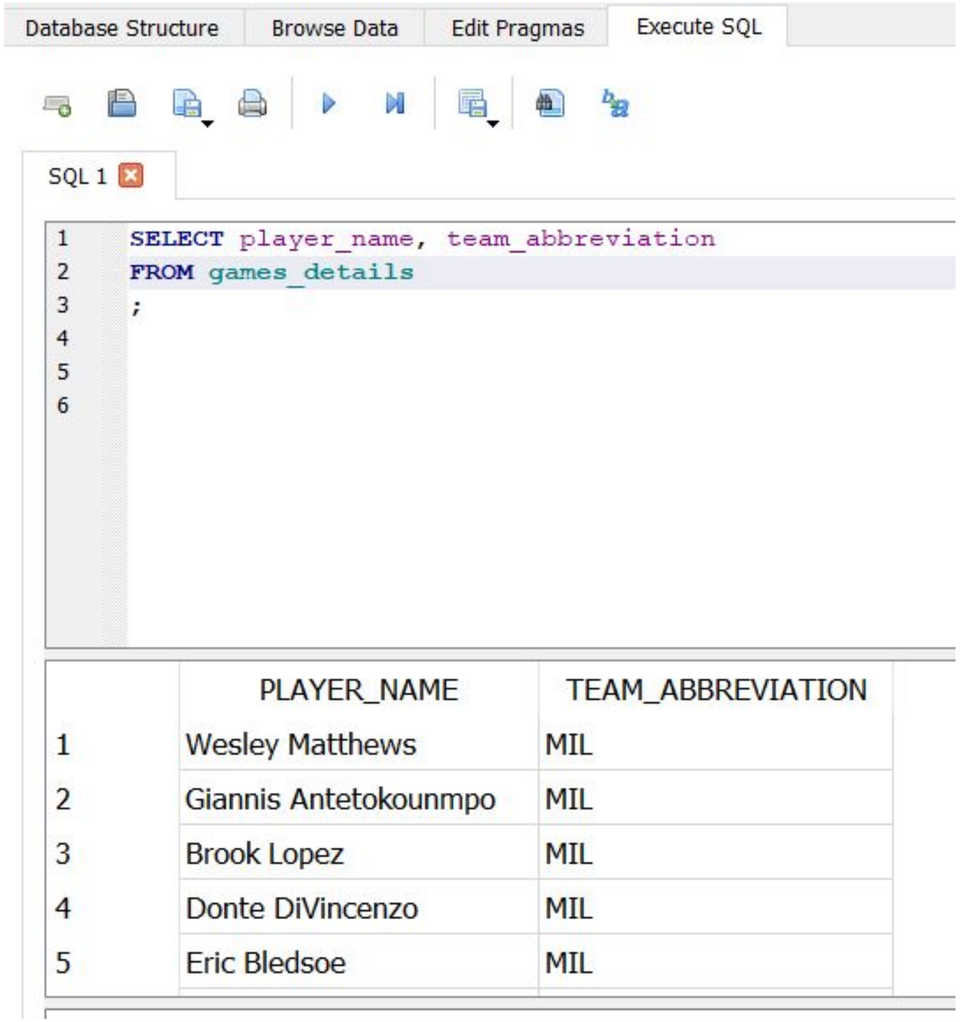
- Structure of the NBA dataset in DB Browser for SQLite:
  - Game_details table:



	GAME_ID	TEAM_ID	TEAM_ABBREVIATION	TEAM_CITY	PLAYER_ID	PLAYER_NAME
	Filter	Filter	Filter	Filter	Filter	Filter
1	21900895	1610612749	MIL	Milwaukee	202083	Wesley Matth...
2	21900895	1610612749	MIL	Milwaukee	203507	Giannis Antet...
3	21900895	1610612749	MIL	Milwaukee	201572	Brook Lopez
4	21900895	1610612749	MIL	Milwaukee	1628978	Donte DiVinc...
5	21900895	1610612749	MIL	Milwaukee	202339	Eric Bledsoe
6	21900895	1610612749	MIL	Milwaukee	1626192	Pat Connaugh...
7	21900895	1610612749	MIL	Milwaukee	201577	Robin Lopez
8	21900895	1610612749	MIL	Milwaukee	1628425	Sterling Brown
9	21900895	1610612749	MIL	Milwaukee	101107	Marvin Williams
10	21900895	1610612749	MIL	Milwaukee	201588	George Hill

SQLite DEMO

- DEMO:
    - In the table games_details:
        - How do you grab all the players and teams?
            - What columns contain this data?
            - What keywords are needed?



```sql
SELECT player_name, team_abbreviation
FROM games_details
;
```

	PLAYER_NAME	TEAM_ABBREVIATION
1	Wesley Matthews	MIL
2	Giannis Antetokounmpo	MIL
3	Brook Lopez	MIL
4	Donte DiVincenzo	MIL
5	Eric Bledsoe	MIL

WHERE

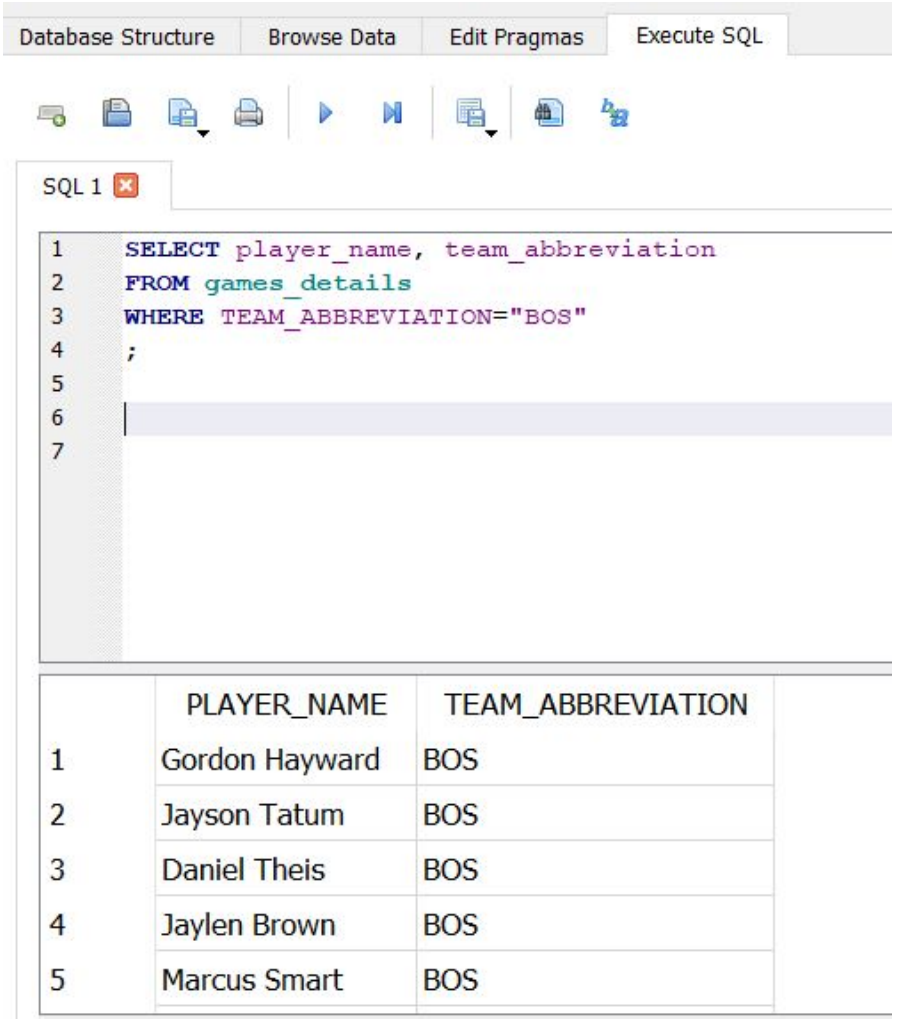
- The **WHERE** keyword gets only certain rows where the values meet the specifications

- **SELECT** <column(s)> **FROM** <table name> **WHERE** <condition(s)>**;**

- Conditions can be...
    - if a certain column is greater than some value
    - which data points occur on a certain date
    - if strings have a character ‘e’
        - use the **LIKE** keyword to see if a string fits a certain pattern

SQLite DEMO

- DEMO:
  - In the table games_details:
    - How do you grab all the players and teams from the Boston Celtics?
      - What columns contain this data?
      - What is the Team Abbreviation for the Celtics?
      - What keywords are needed?



```
SELECT player_name, team_abbreviation
FROM games_details
WHERE TEAM_ABBREVIATION="BOS"
;
```

	PLAYER_NAME	TEAM_ABBREVIATION
1	Gordon Hayward	BOS
2	Jayson Tatum	BOS
3	Daniel Theis	BOS
4	Jaylen Brown	BOS
5	Marcus Smart	BOS

ORDER BY & LIMIT

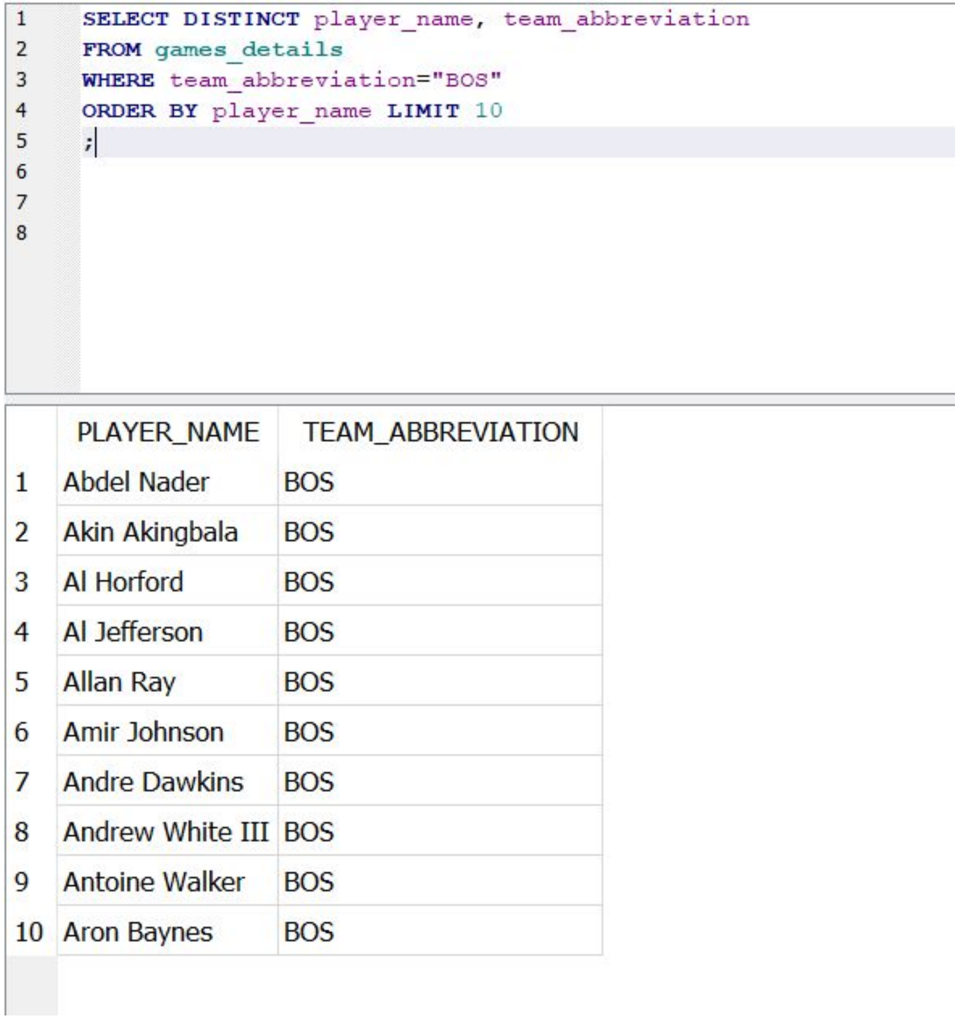
- **ORDER BY <column name> [ASC | DESC]**
  - Order the output by the column(s) value(s)
  - ASC (ascending order) is the *default*

- **LIMIT <number>**
  - Limit the number of column value(s) returned in the output
  - LIMIT 10; will make the query output 10 rows

- A query using: … ORDER BY <column(s) name(s)> DESC … LIMIT 50; is essentially a top-50 ranking of the column(s) values(s)

SQLite DEMO

- DEMO:
    - In the table games_details:
        - How do you grab all the players and teams from the Boston Celtics who are first 10 players sorted alphabetically by first name?
            - What columns contain this data?
            - What keywords are needed?



```
SELECT DISTINCT player_name, team_abbreviation
FROM games_details
WHERE team_abbreviation="BOS"
ORDER BY player_name LIMIT 10
;
```

	PLAYER_NAME	TEAM_ABBREVIATION
1	Abdel Nader	BOS
2	Akin Akingbala	BOS
3	Al Horford	BOS
4	Al Jefferson	BOS
5	Allan Ray	BOS
6	Amir Johnson	BOS
7	Andre Dawkins	BOS
8	Andrew White III	BOS
9	Antoine Walker	BOS
10	Aron Baynes	BOS

ALIASES

- a ***temporary*** *name* for a table or column in a statement
    - Helpful when dealing with long column and table names (shorter aliases mean less typing)
      ⇒ however, don't sacrifice clarity for brevity
    - Simple way to display a column's temporary name in the query's output
- SELECT <column name> **AS** <***alias***> FROM <table name> **AS** <***another alias***>;

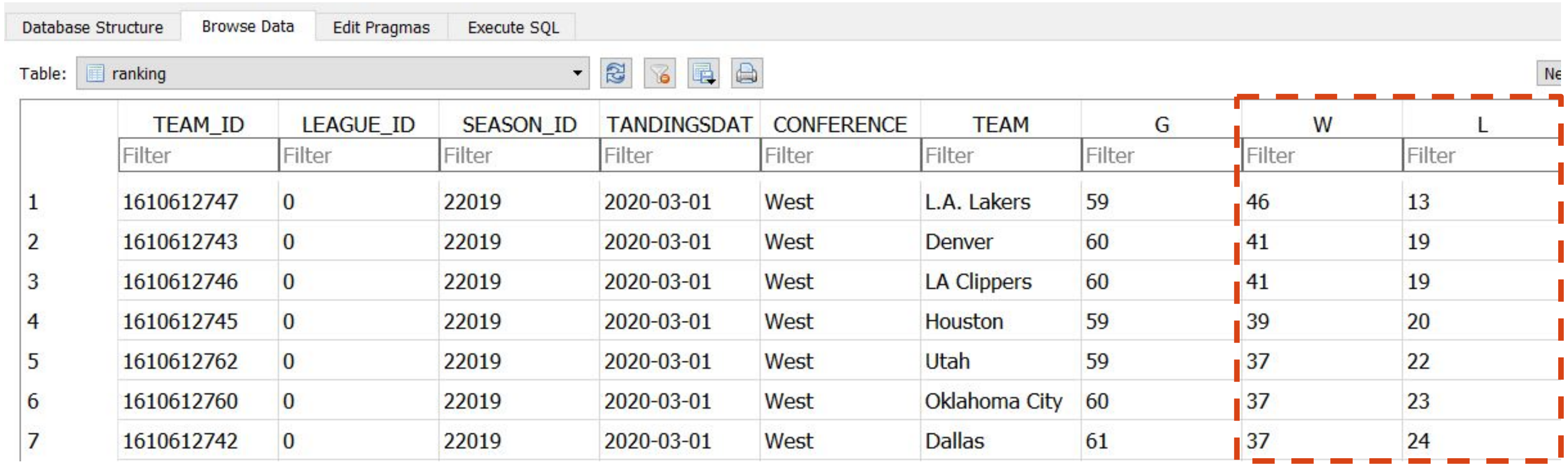
AGGREGATES

- **AVG(**<column>**);**
- **MIN(**<column>**);**
- **MAX(**<column>**);**
- **SUM(**<column>**);**
- **COUNT(**<column>**);**
- **MEAN(**<column>**);**
- **MEDIAN(**<column>**);**
- **MODE(**<column>**);**

REMEMBER: these aggregate functions must be inside a SELECT statement!

SQLite DEMO

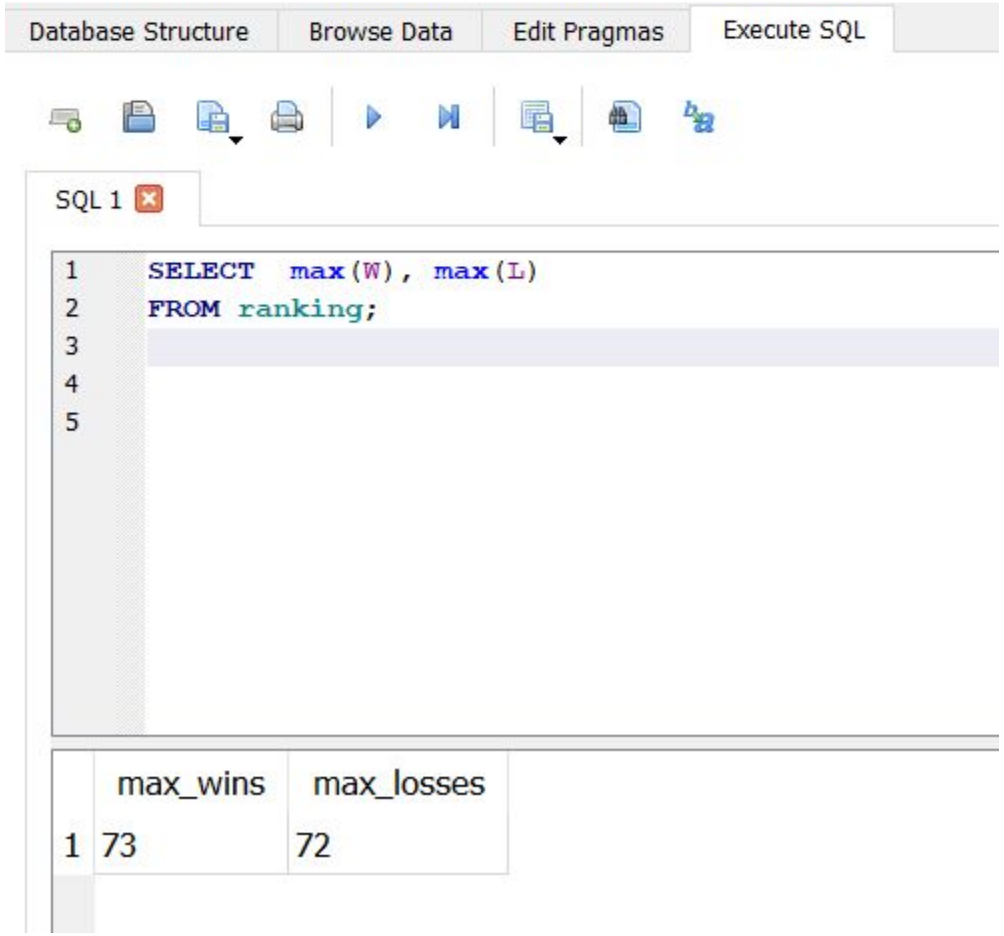
- Structure of the NBA dataset in DB Browser for SQLite:
  - Rankings table:



	TEAM_ID	LEAGUE_ID	SEASON_ID	TANDINGSDAT	CONFERENCE	TEAM	G	W	L
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1610612747	0	22019	2020-03-01	West	L.A. Lakers	59	46	13
2	1610612743	0	22019	2020-03-01	West	Denver	60	41	19
3	1610612746	0	22019	2020-03-01	West	LA Clippers	60	41	19
4	1610612745	0	22019	2020-03-01	West	Houston	59	39	20
5	1610612762	0	22019	2020-03-01	West	Utah	59	37	22
6	1610612760	0	22019	2020-03-01	West	Oklahoma City	60	37	23
7	1610612742	0	22019	2020-03-01	West	Dallas	61	37	24

SQLite DEMO

- DEMO:
  - In the table ranking:
    - How do you grab the maximum number of wins and maximum number of losses out of all the teams?
      - What columns contain this data?
      - What keywords are needed?



```
SELECT  max(W), max(L)
FROM ranking;
```

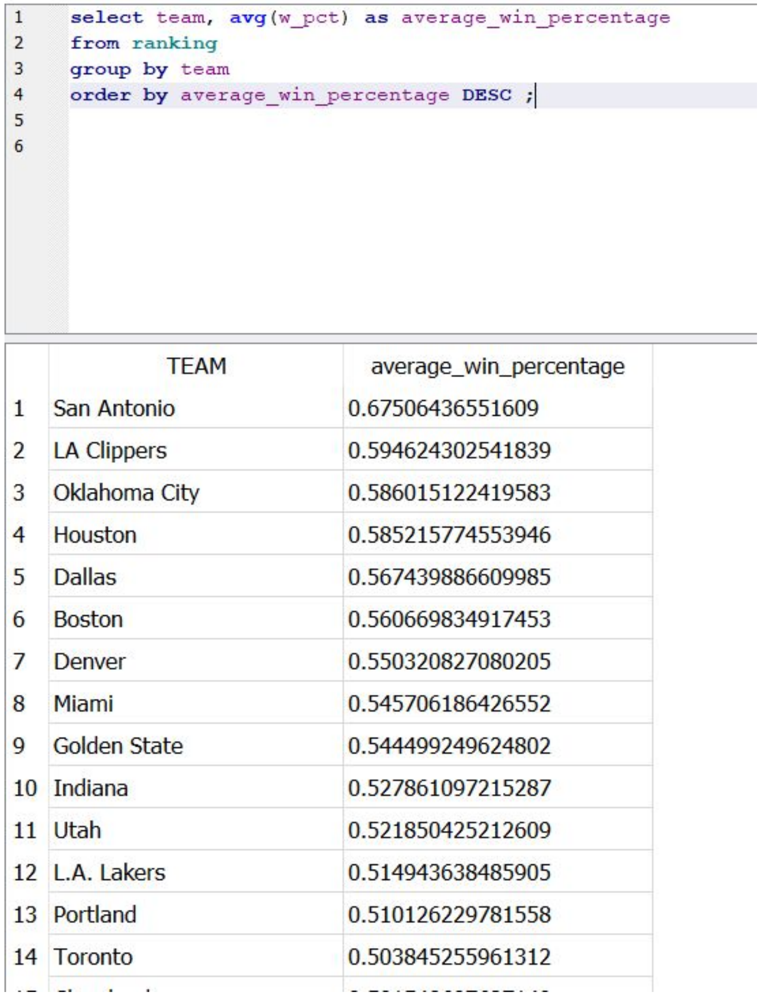
	max_wins	max_losses
1	73	72

GROUP BY

- SELECT … FROM … **GROUP BY** (<column(s)>);
- Reduces all rows which have the same values in the <column(s)> specified into “summary rows” that can give us summary statistics
- Usually used in conjunction with the aggregates described above
- Can group by multiple columns

SQLite DEMO

- DEMO:
  - In the rankings table:
    - How do you find the average win percentage for each team?
      - What column(s) contain this data?
      - What keywords are needed?



```sql
select team, avg(w_pct) as average_win_percentage
from ranking
group by team
order by average_win_percentage DESC ;
```

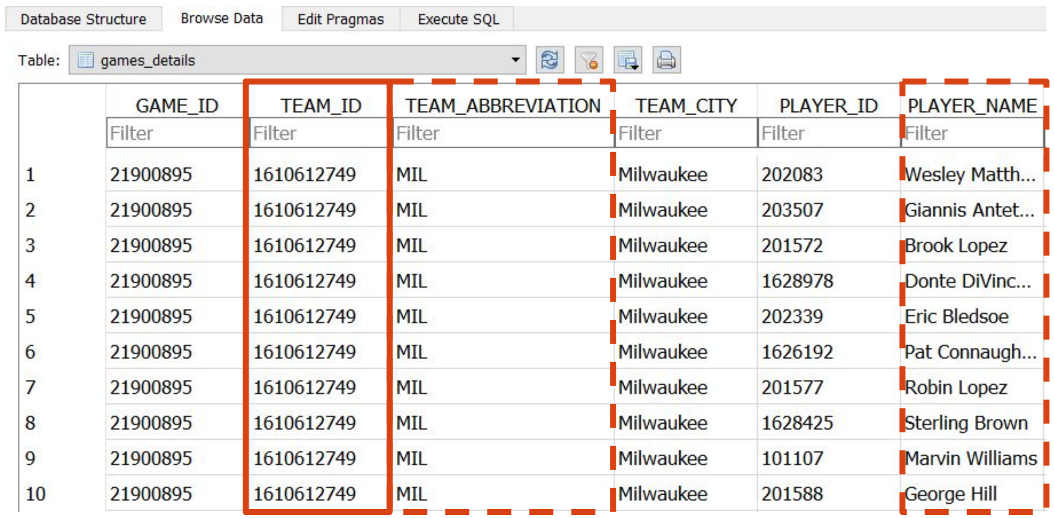
	TEAM	average_win_percentage
1	San Antonio	0.67506436551609
2	LA Clippers	0.594624302541839
3	Oklahoma City	0.586015122419583
4	Houston	0.585215774553946
5	Dallas	0.567439886609985
6	Boston	0.560669834917453
7	Denver	0.550320827080205
8	Miami	0.545706186426552
9	Golden State	0.544499249624802
10	Indiana	0.527861097215287
11	Utah	0.521850425212609
12	L.A. Lakers	0.514943638485905
13	Portland	0.510126229781558
14	Toronto	0.503845255961312

PROBLEM...

- What happens if we want to grab data that is located in two separate tables?

- Example:
    - We want to grab head coaches, players and teams
        - coaches are stored in the teams table *but...*
        - players are in the games_details table

CORNELL SPORTS ANALYTICS



Database Structure | Browse Data | Edit Pragmas | Execute SQL

Table: games_details

	GAME_ID	TEAM_ID	TEAM_ABBREVIATION	TEAM_CITY	PLAYER_ID	PLAYER_NAME
	Filter	Filter	Filter	Filter	Filter	Filter
1	21900895	1610612749	MIL	Milwaukee	202083	Wesley Matth...
2	21900895	1610612749	MIL	Milwaukee	203507	Giannis Antet...
3	21900895	1610612749	MIL	Milwaukee	201572	Brook Lopez
4	21900895	1610612749	MIL	Milwaukee	1628978	Donte DiVinc...
5	21900895	1610612749	MIL	Milwaukee	202339	Eric Bledsoe
6	21900895	1610612749	MIL	Milwaukee	1626192	Pat Connaugh...
7	21900895	1610612749	MIL	Milwaukee	201577	Robin Lopez
8	21900895	1610612749	MIL	Milwaukee	1628425	Sterling Brown
9	21900895	1610612749	MIL	Milwaukee	101107	Marvin Williams
10	21900895	1610612749	MIL	Milwaukee	201588	George Hill

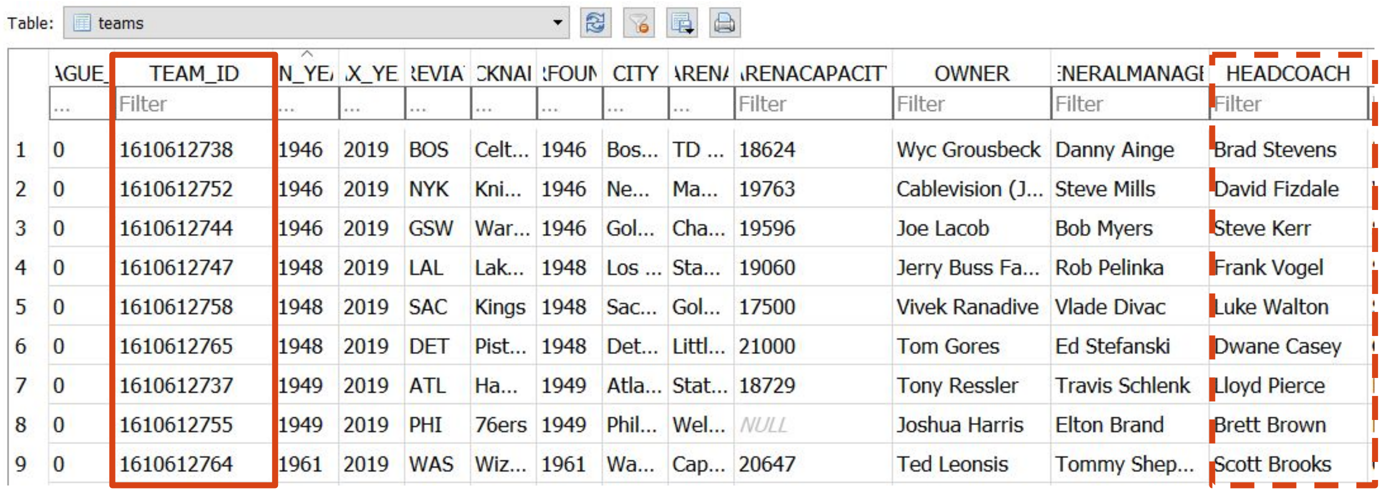


Table: teams

	AGUE_	TEAM_ID	N_YEA	X_YE	REVIA	CKNAI	FOUN	CITY	ARENA	ARENACAPACIT	OWNER	ENERALMANAGE	HEADCOACH
	...	Filter	...	...	...	...	...	...	...	Filter	Filter	Filter	Filter
1	0	1610612738	1946	2019	BOS	Celt...	1946	Bos...	TD ...	18624	Wyc Grousbeck	Danny Ainge	Brad Stevens
2	0	1610612752	1946	2019	NYK	Kni...	1946	Ne...	Ma...	19763	Cablevision (J...	Steve Mills	David Fizdale
3	0	1610612744	1946	2019	GSW	War...	1946	Gol...	Cha...	19596	Joe Lacob	Bob Myers	Steve Kerr
4	0	1610612747	1948	2019	LAL	Lak...	1948	Los ...	Sta...	19060	Jerry Buss Fa...	Rob Pelinka	Frank Vogel
5	0	1610612758	1948	2019	SAC	Kings	1948	Sac...	Gol...	17500	Vivek Ranadive	Vlade Divac	Luke Walton
6	0	1610612765	1948	2019	DET	Pist...	1948	Det...	Littl...	21000	Tom Gores	Ed Stefanski	Dwane Casey
7	0	1610612737	1949	2019	ATL	Ha...	1949	Atla...	Stat...	18729	Tony Ressler	Travis Schlenk	Lloyd Pierce
8	0	1610612755	1949	2019	PHI	76ers	1949	Phil...	Wel...	NULL	Joshua Harris	Elton Brand	Brett Brown
9	0	1610612764	1961	2019	WAS	Wiz...	1961	Wa...	Cap...	20647	Ted Leonsis	Tommy Shep...	Scott Brooks

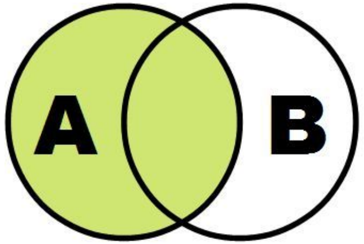
SOLUTION? JOINS!

- JOIN queries combine columns from multiple tables into one output
    - Puts data that's originally separated into one place

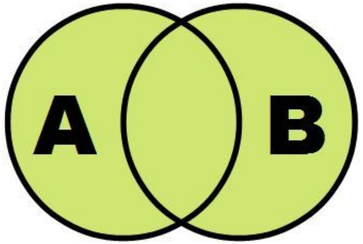
- **SELECT <column(s)>**
  FROM <table 1>
  JOIN <table 2> ON table1.<join column>=table2.<join column>

- *NOTE:*
    - The **ON** keyword tells us what columns match in the 2 tables
    - Does not add rows, only adds columns

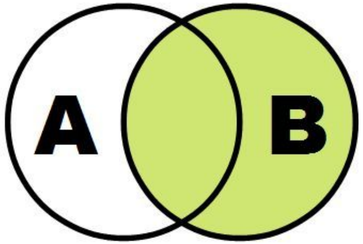
JOINS



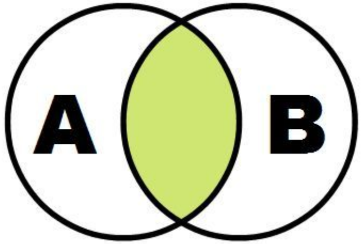
```
SELECT *
FROM A
LEFT JOIN B
ON A.id = B.id
```



```
SELECT *
FROM A
FULL OUTER JOIN B
ON A.id = B.id
```



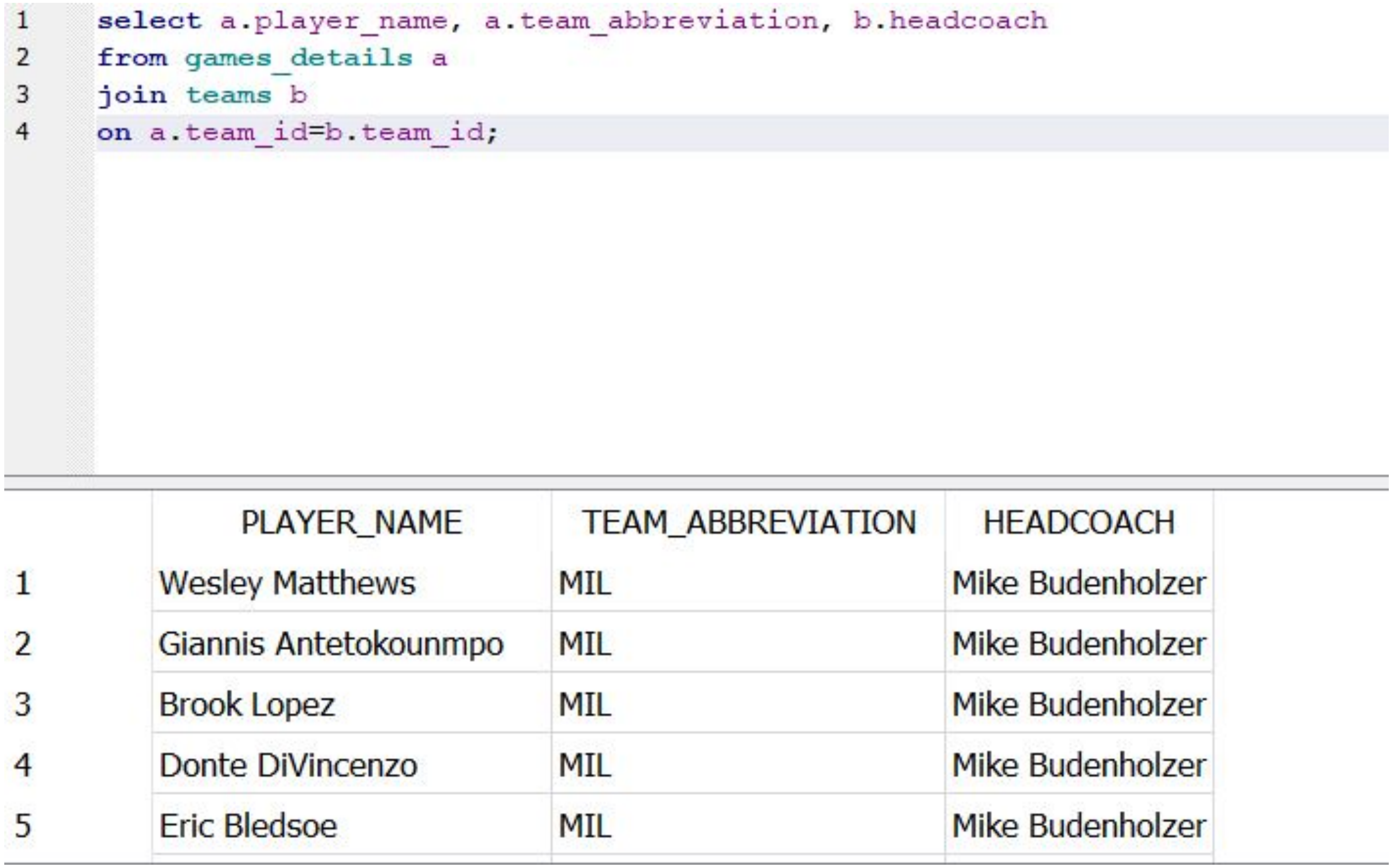
```
SELECT *
FROM A
RIGHT JOIN B
ON A.id = B.id
```



```
SELECT *
FROM A
INNER JOIN B
ON A.id = B.id
```

SQLite DEMO

- Remember that problem we had...
    - How do you grab head coaches, players, and teams?
        - In the teams table:
            - Head coaches
        - In the games_details table:
            - Players and teams
    - How can you use a JOIN statement to get all 3 of these columns?
        - Is there a column value that the teams and games_details tables share?



```sql
select a.player_name, a.team_abbreviation, b.headcoach
from games_details a
join teams b
on a.team_id=b.team_id;
```

	PLAYER_NAME	TEAM_ABBREVIATION	HEADCOACH
1	Wesley Matthews	MIL	Mike Budenholzer
2	Giannis Antetokounmpo	MIL	Mike Budenholzer
3	Brook Lopez	MIL	Mike Budenholzer
4	Donte DiVincenzo	MIL	Mike Budenholzer
5	Eric Bledsoe	MIL	Mike Budenholzer

PUTTING IT ALL TOGETHER...

- Combine all the statements & keywords you've learned today (and others from the internet) into one giant query!!
- Sequence of these statements does matter (sometimes)
  - SELECT always starts the query
  - <AGGREGATE FUNCTIONS>
  - FROM <table name>
  - any JOINs occur at the end directly after the FROM <table> <JOIN.....s>
  - WHERE, ORDER BY, LIMIT
  - ;

DB BROWSER for SQLite

DOWNLOAD:

https://sqlitebrowser.org/

OTHER RESOURCES

SQLite Documentation:
https://sqlite.org/lang.html and https://sqlite.org/docs.html

SQL Tutorial:
https://www.w3schools.com/sql/default.asp

CMU Database Systems, Advanced SQL, Fall 2019:
https://www.youtube.com/watch?v=6VCHuLqfmV8, 3:19-55:32