CIS 9440 - Data Warehousing and Analytics

Class #2

Week 2 Class Overview:

1. Last Week Review
2. What is SQL?
3. Hands-on SQL Workshop

Week 2 Class Overview:

1. **Last Week Review**
2. What is SQL?
3. Hands-on SQL Workshop

What’s on Blackboard?

- Slides of Class #1 on Blackboard
- Getting Started with BigQuery resources
- Final Project survey

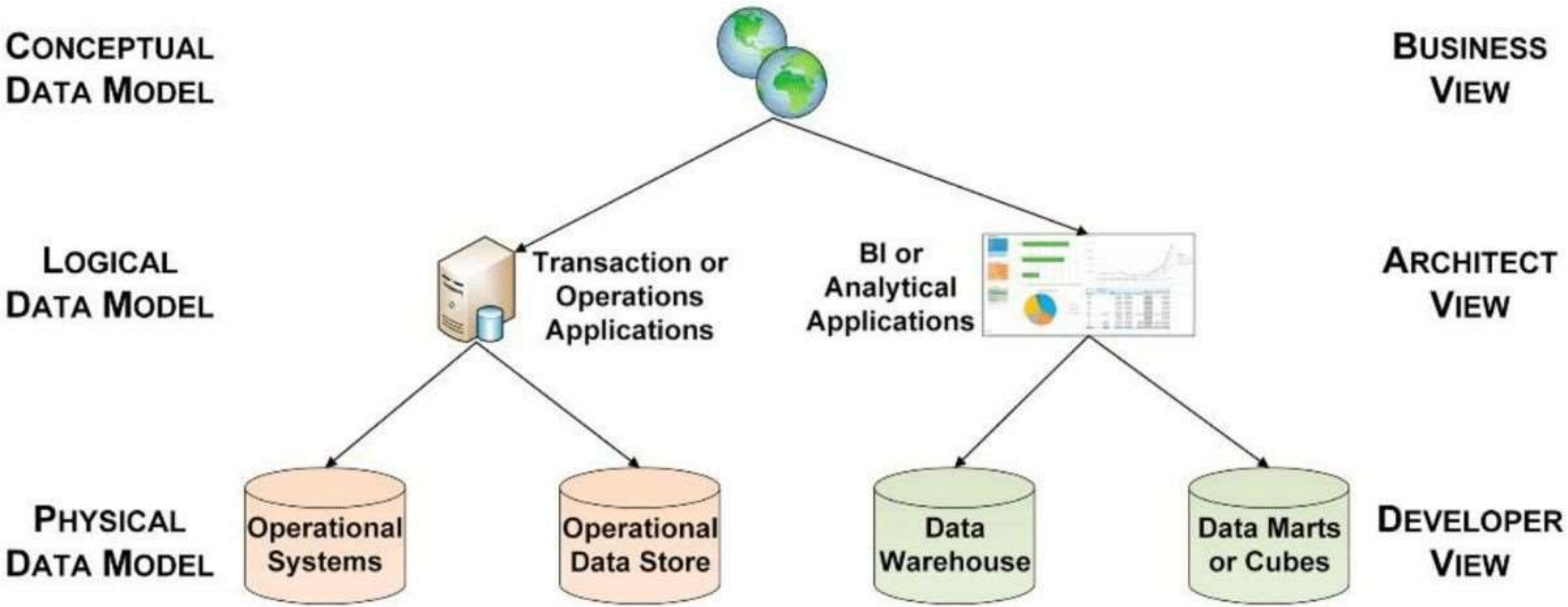
Reading, BIG Chapter 8 all topics

- Introduction to Data Modeling
- Three levels of Data Models
- Modeling workflow
- Where data modeling is used
- ER Modeling Overview (Referential Integrity)
- Normalization

Reading, BIG Chapter 8 review topics

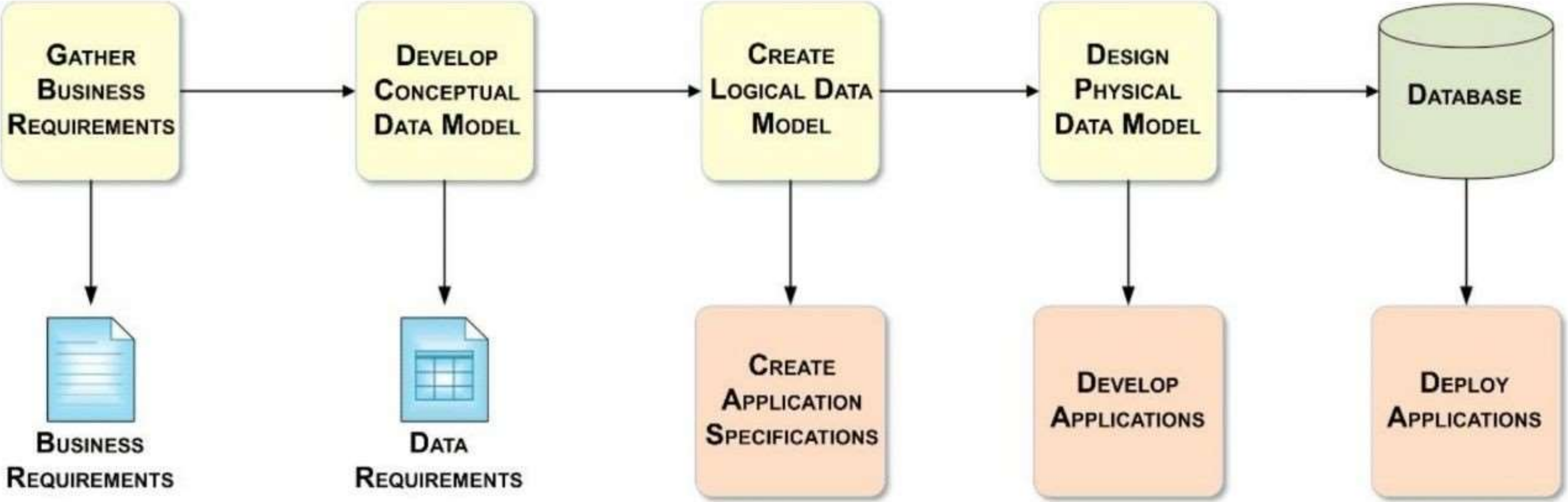
- Introduction to Data Modeling
- **Three levels of Data Models**
- **Modeling workflow**
- Where data modeling is used
- **ER Modeling Overview (Referential Integrity)**
- Normalization

Three levels of data models



f08-01

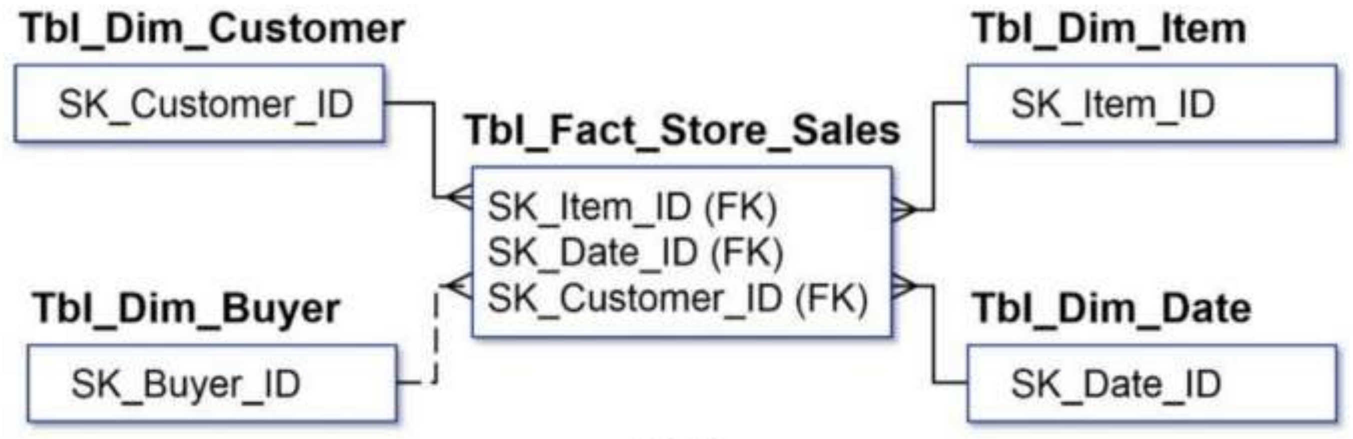
Data modeling workflow



Gather Business Requirements
Develop Conceptual Data Model
Create Logical Data Model
Design Physical Data Model
Database
Business Requirements
Data Requirements
Create Application Specifications
Develop Applications
Deploy Applications

f08-02

Referential integrity



f08-13

- Referential Integrity (RI) includes enforcing relationship cardinality rules.
  - Example: if you have a one or many relationship you must have at least one non-null record
- To enforce RI use foreign key constraints in your ETL process. This will guarantee RI for each insert, update, or delete in the database.

Week 2 Class Overview:

1. Last Week Review
2. **What is SQL?**
3. Hands-on SQL Workshop

What is SQL?

SQL (Structured Query Language) is the language used to communicate with a relational database (rows and columns).

- It’s been the standard since 1987
- It was created at IBM in the 70’s by Raymond Boyce and Donald Chamberlin.

- Do we always use relational databases?

What does SQL allow you to do?

- SQL allows an Analyst to interact with an organization's data.
  - Insert new rows
  - Query the database for data
  - Delete rows
  - Change Permissions
  - etc.

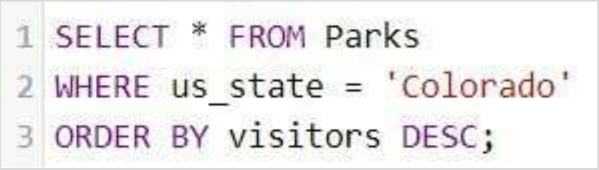
Types of SQL Queries

There are 4 main types of SQL statements:

Type of SQL Statement	**Purpose**	**Common Queries**
Data Manipulation Language (**DML**)	Manage data within a table	SELECT, INSERT, UPDATE, DELETE
Data Definition Language (**DDL**)	Define database structure or table	CREATE, ALTER, DROP, RENAME
Transaction Control Statement (**TCS**)	Save permanent changes	COMMIT, ROLLBACK, SAVEPOINT
Data Control Language (**DCL**)	Give privileges to users	GRANT, REVOKE, AUDIT

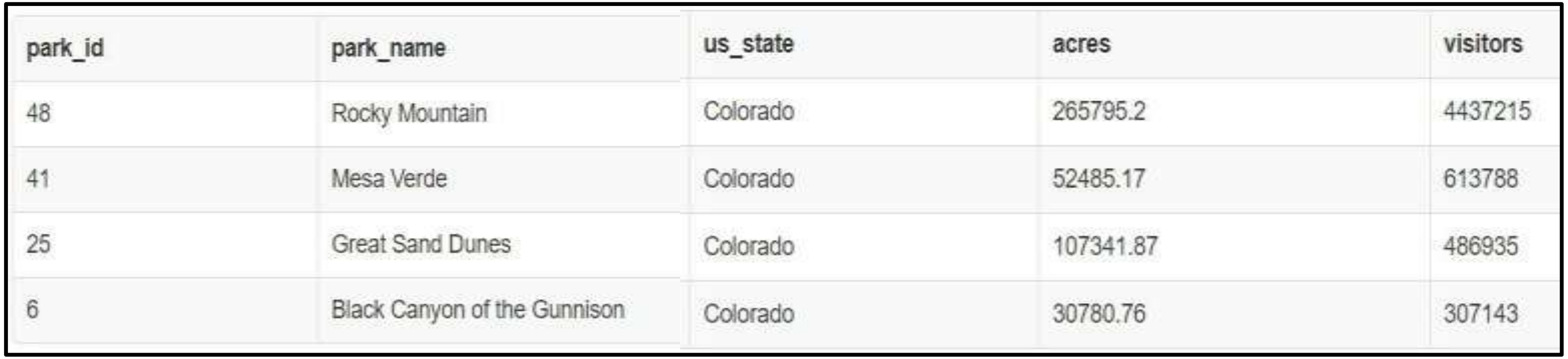
SQL, Common Example

Here, a user writes a simple SELECT query to view data from the table “Parks”



```
SELECT * FROM Parks
WHERE us_state = 'Colorado'
ORDER BY visitors DESC;
```

Intuitive!



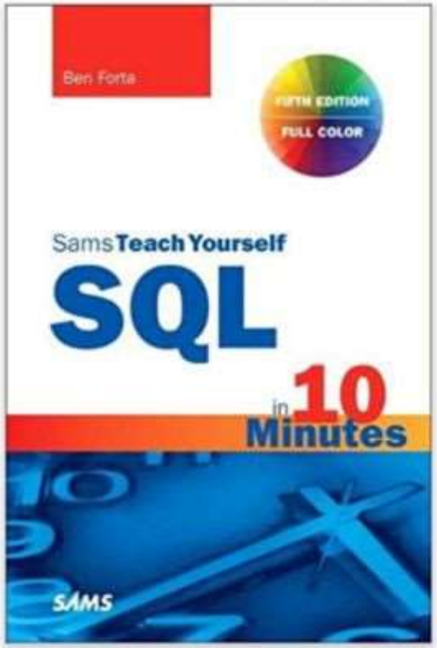
park_id	park_name	us_state	acres	visitors
48	Rocky Mountain	Colorado	265795.2	4437215
41	Mesa Verde	Colorado	52485.17	613788
25	Great Sand Dunes	Colorado	107341.87	486935
6	Black Canyon of the Gunnison	Colorado	30780.76	307143

Where is SQL in Data Warehousing?

- SQL may appear in ETL/ELT during Data Integration
  - May be built into a recurring script
- SQL can be used by Analysts to pull clean data from the BI Layer for custom/ad-hoc analyses
  - Even sometimes in BI Applications like “freeform MicroStrategy”

SQL Reference book for Analysts

Sams Teach Yourself SQL in 10 Minutes, by Ben Forta



Dialects of SQL

SQL has many versions. All versions are similar; if you know one you can learn all easily. But, different versions are referenced often so it’s valuable to know the differences:

- SQL Server
- MySQL
- PostgreSQL
- Sqlite
- And many more

Week 2 Class Overview:

1. Last Week Review
2. What is SQL?
3. **Hands-on SQL Workshop**

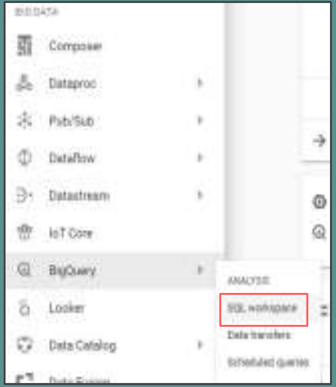
Exercise: Practice SQL with BigQuery

(90 minutes)

Go to https://cloud.google.com/

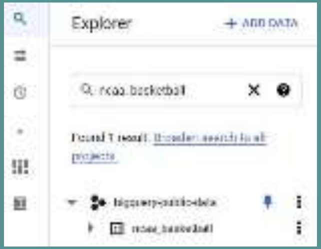
If you do not yet have a BigQuery account, use the BigQuery sandbox: https://cloud.google.com/bigquery/docs/sandbox

1. Click on “Go to console” button
2. On the left-most navigation pane, go to “BigQuery -> SQL workspace”



3.

4. In the search bar, type in “ncaa_basketball” and click enter



ncaa_basketball

Found 1 result. Broaden search to all projects

Practice SQL queries

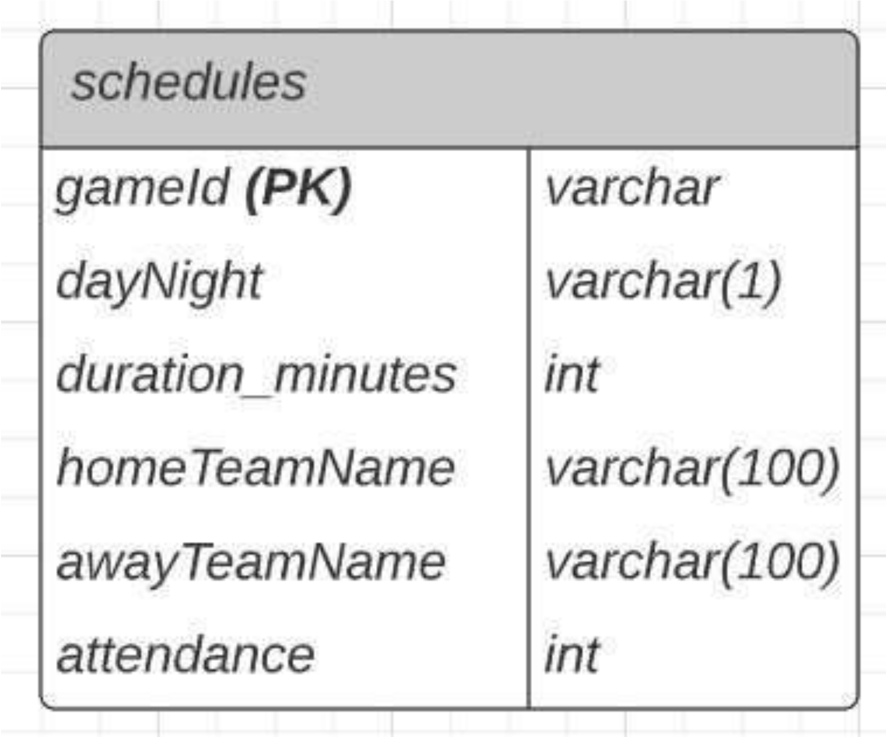
Now, time to practice!

Please type out all following queries, try to avoid copy and paste

SQL tips for today:

1. SQL is very logical, so each query can be broken down into steps. Thus, break down each practice question and solve it step-by-step. No need to write the entire query at once.
2. Use semicolons to end each query
3. Use ALL CAPS for SQL keywords like SELECT
4. We will start very basic and ramp up to analytical SQL quickly, be patient!
5. SQL is not picky about white spaces, except during subqueries

Data Model for first examples



schedules	
gameId **(PK)**	varchar
dayNight	varchar(1)
duration_minutes	int
homeTeamName	varchar(100)
awayTeamName	varchar(100)
attendance	int

*Made with Lucidchart

SELECT *

Select all from a table

This will view all rows from table schedules

How many rows? What is the table size? Primary Key?

SELECT Column

Select specific columns from a table

Try 2 columns, then 3

LIMIT

Limit the number of rows returned by a query

*in other versions of SQL, you may use **TOP rather than LIMIT**

Add a Comment

Add a comment for users to read, not read by SQL

Add a Long Comment

Add a comment that is longer than 1 line

ORDER BY (sorting)

Order your query results in ascending or descending order by a specified column(s)

ORDER BY DESC (sorting)

Enter the following into the right-most window, then click “Run”:

ORDER BY + LIMIT (get top or bottom rows)

Get the top or bottom `x` amount of rows from a table

Practice Question 1:

(3 minutes)

Select the `homeTeamName` and `attendance` columns, order the query in descending by `attendance` , and limit the query to only 5 rows.

Practice Question 2:

(3 minutes)

In minutes, what was the longest game played?

Practice Question 3:

(3 minutes)

Which `homeTeamName` played a game with the lowest `attendance`?

DISTINCT

Return only the unique values from a specified row

COUNT

Count the number of rows in a SELECT statement

This will return the count of rows in the table

COUNT DISTINCT

Count the unique values in a specified column

WHERE (filtering)

Filter based on a condition.

WHERE (filtering)

Filter based on a *greater than* condition

Practice Question Set #1

Use keywords such as **COUNT**, **DISTINCT**, **LIMIT**, **ORDER BY**, and **WHERE** to answer the following:

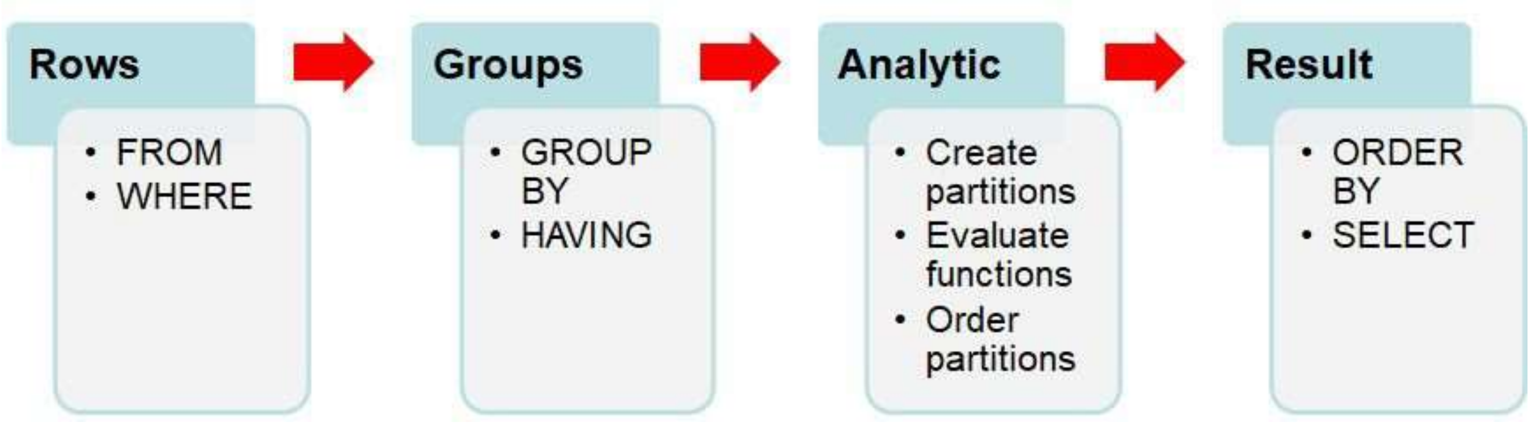
1. How many games had an `attendance` greater than 52,000?
2. What are the distinct `dayNight` values?
3. For only “D” games (night games), what are the 3 homeTeamName with the lowest duration_minutes?
4. How many homeTeamName have had at least 1 game with attendance > 49,000?

10 Minute Break

(7:30pm)

Query Clause Evaluation Order

SQL queries are run in a specific order:



Rows

- FROM
- WHERE

Groups

- GROUP BY
- HAVING

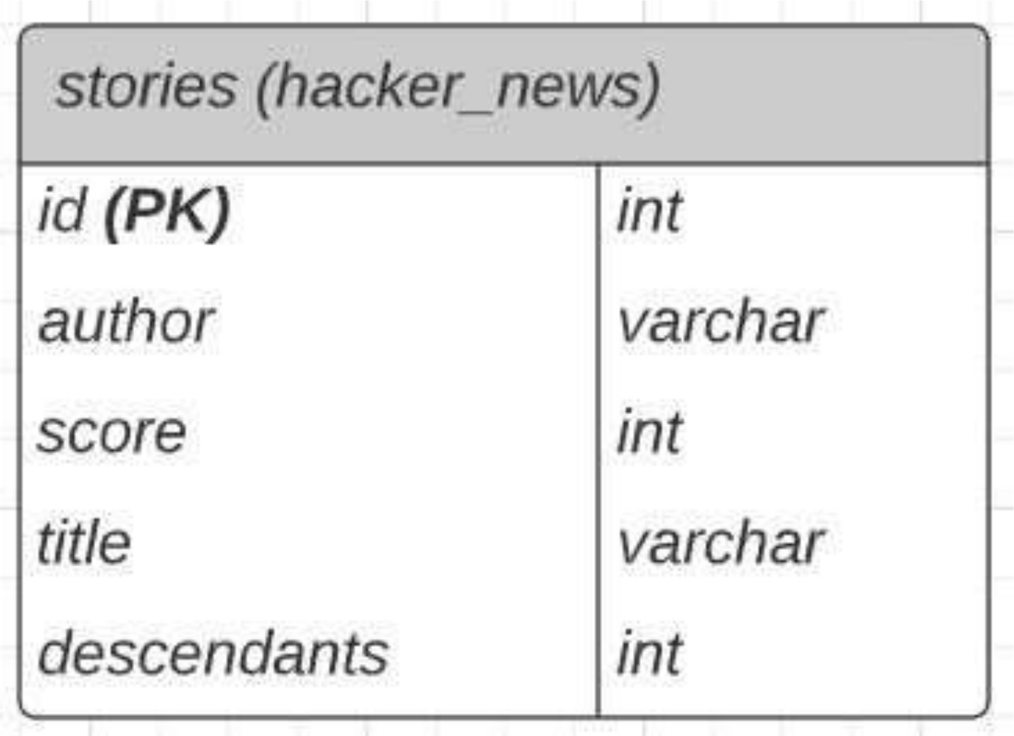
Analytic

- Create partitions
- Evaluate functions
- Order partitions

Result

- ORDER BY
- SELECT

Data Model for next examples



stories (hacker_news)	
id ***(PK)***	*int*
author	*varchar*
score	*int*
title	*varchar*
descendants	*int*

*Made with Lucidchart

IN (filtering)

Filter for only values in a specified list

View only markets where the venue is in New York or New Jersey

NOT IN (filtering)

Filter for values not in a specified list

LIKE (wildcard filtering)

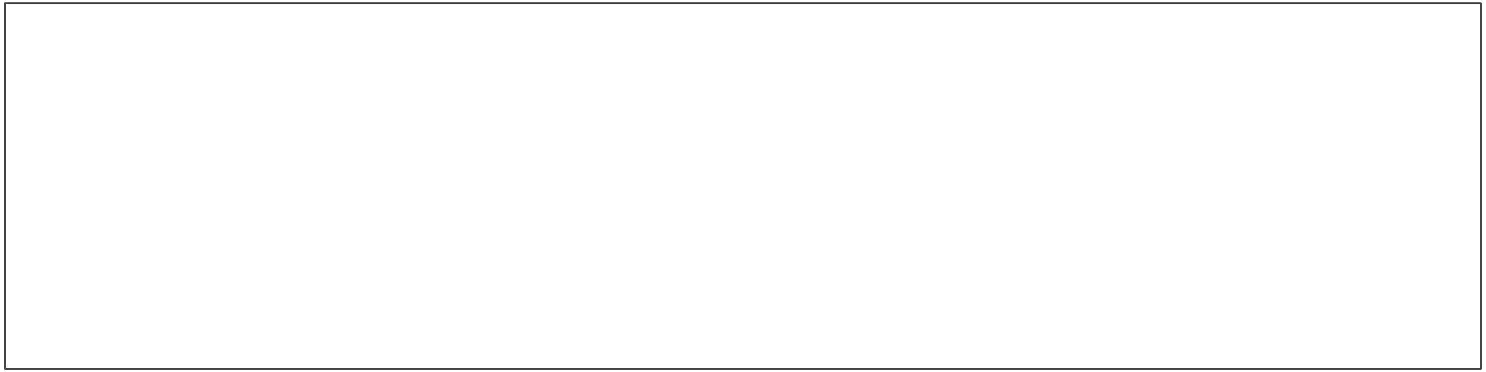
Filter for values containing certain characters

Practice Question 4:

Select all distinct `author` names that contain the word queen

LENGTH

Find the length of a field



Practice Question 5:

Select all `title` where the length of title is 10

Practice Question 5:

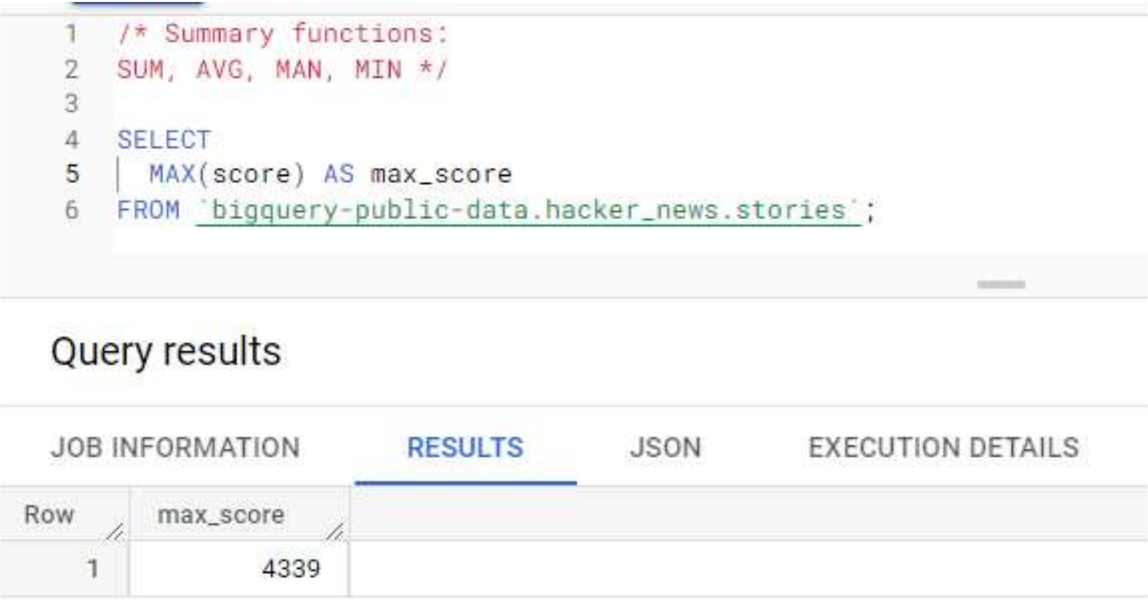
How many author have a length = 15?

Calculated Field

Create a new column based on a calculation

Summary Functions

Use SUM to get the sum of a table column



```
/* Summary functions:
SUM, AVG, MAN, MIN */

SELECT
  MAX(score) AS max_score
FROM `bigquery-public-data.hacker_news.stories`;
```

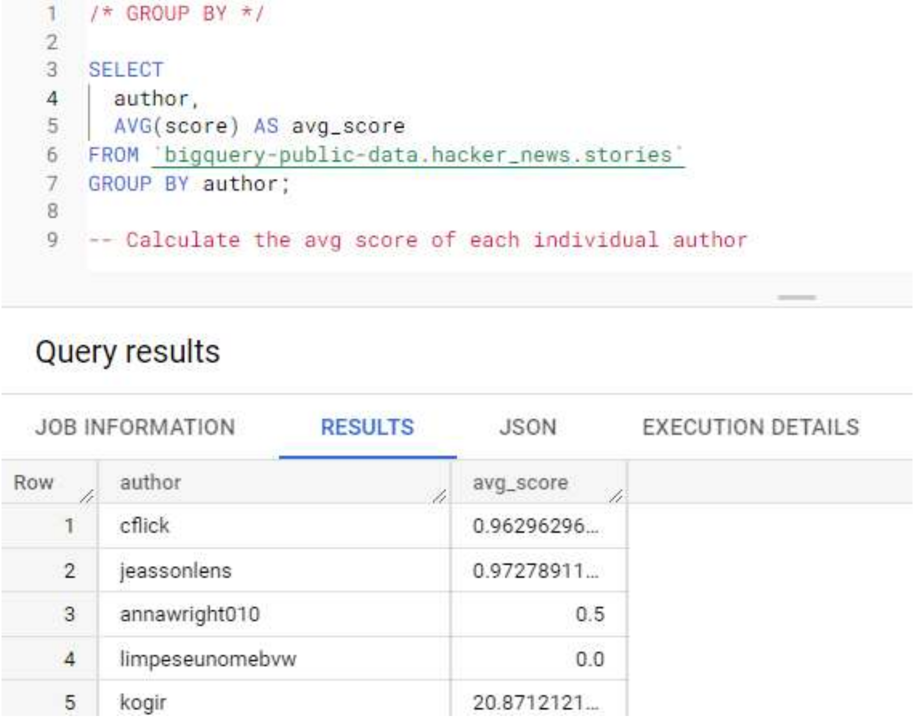
Query results

Row	max_score
1	4339

GROUP BY

If you want to group a summary function by a specific column(s)

How could we change this query to order by highest averages?



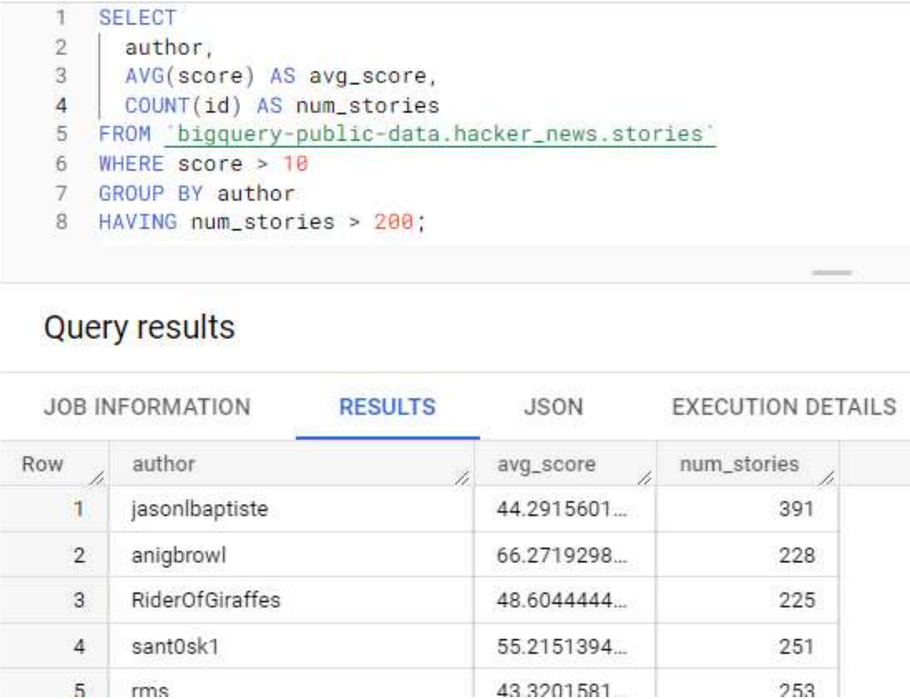
```
/* GROUP BY */

SELECT
  author,
  AVG(score) AS avg_score
FROM `bigquery-public-data.hacker_news.stories`
GROUP BY author;

-- Calculate the avg score of each individual author
```

Query results

Row	author	avg_score
1	cflick	0.96296296...
2	jeassonlens	0.97278911...
3	annawright010	0.5
4	limpeseunomebvw	0.0
5	kogir	20.8712121...

HAVING (WHERE for GROUP BY’s)

Use HAVING to apply a WHERE filtered to GROUPED BY rows



```
SELECT
  author,
  AVG(score) AS avg_score,
  COUNT(id) AS num_stories
FROM `bigquery-public-data.hacker_news.stories`
WHERE score > 10
GROUP BY author
HAVING num_stories > 200;
```

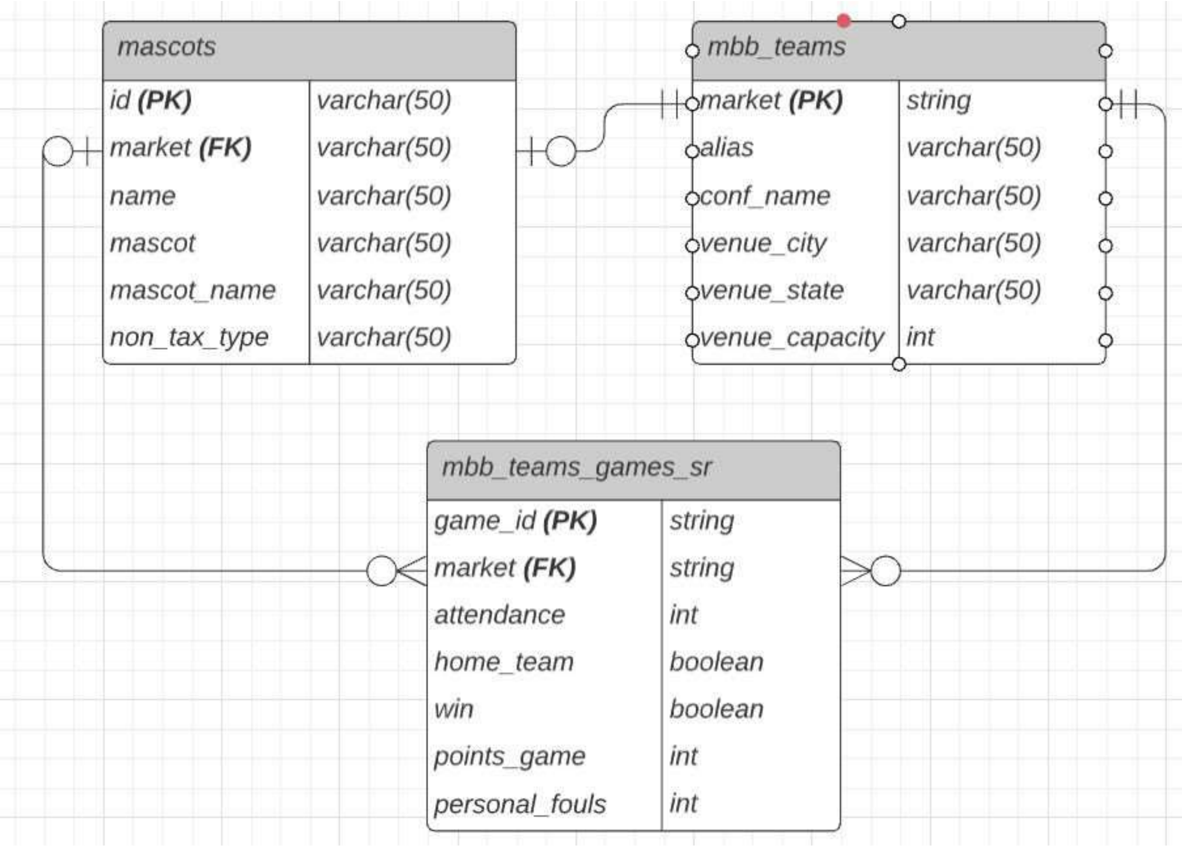
Row	author	avg_score	num_stories
1	jasonlbaptiste	44.2915601...	391
2	anigbrowl	66.2719298...	228
3	RiderOfGiraffes	48.6044444...	225
4	sant0sk1	55.2151394...	251
5	rms	43.3201581...	253

Practice Question Set #2

Use keywords such as **COUNT**, **DISTINCT**, **LIMIT**, **ORDER BY**, and **WHERE** to answer the following:

1. How many distinct `titles` are in the stories table?
2. Who is the `author` with the most total `descendants`?
3. How many `titles` contain the word “bigquery”?
4. How many `authors` have an average `score >` 12 and have at least 15 stories?

Data Model for today’s class



mascots	
id **(PK)**	varchar(50)
market **(FK)**	varchar(50)
name	varchar(50)
mascot	varchar(50)
mascot_name	varchar(50)
non_tax_type	varchar(50)

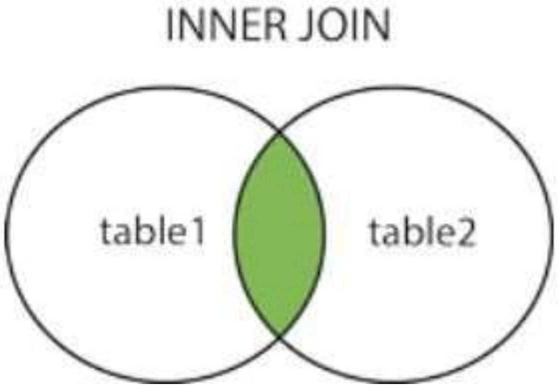
mbb_teams	
market **(PK)**	string
alias	varchar(50)
conf_name	varchar(50)
venue_city	varchar(50)
venue_state	varchar(50)
venue_capacity	int

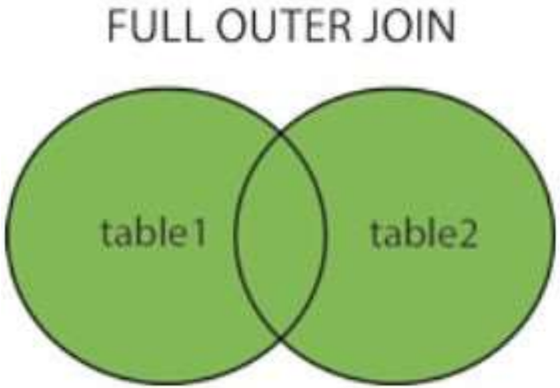
mbb_teams_games_sr	
game_id **(PK)**	string
market **(FK)**	string
attendance	int
home_team	boolean
win	boolean
points_game	int
personal_fouls	int

*Made with Lucidchart

What is a JOIN in SQL?

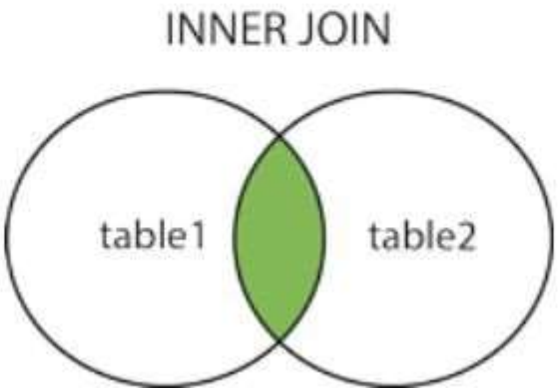
We use a JOIN to query columns from multiple tables in SQL. We specify how we link a JOIN. For example, if we wanted information about a college from both the mascots and mbb_teams table, we could JOIN these tables on the college name.

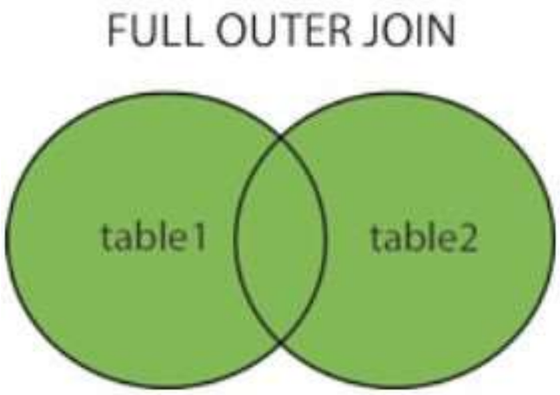




Types of Joins

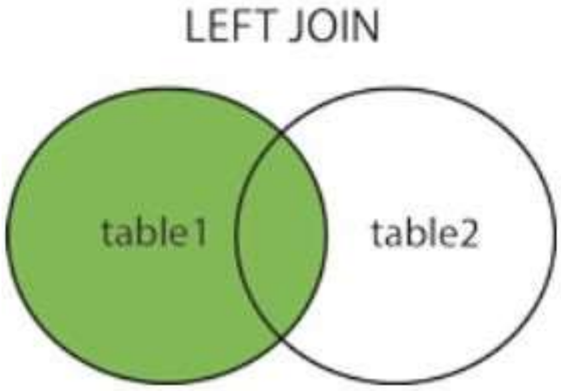
- Inner Joins return records that have matching values in both tables.
- Outer Joins return all records for both left and right tables.

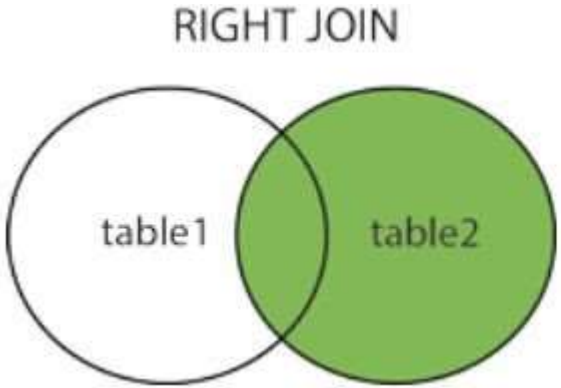




Types of Joins (continued)

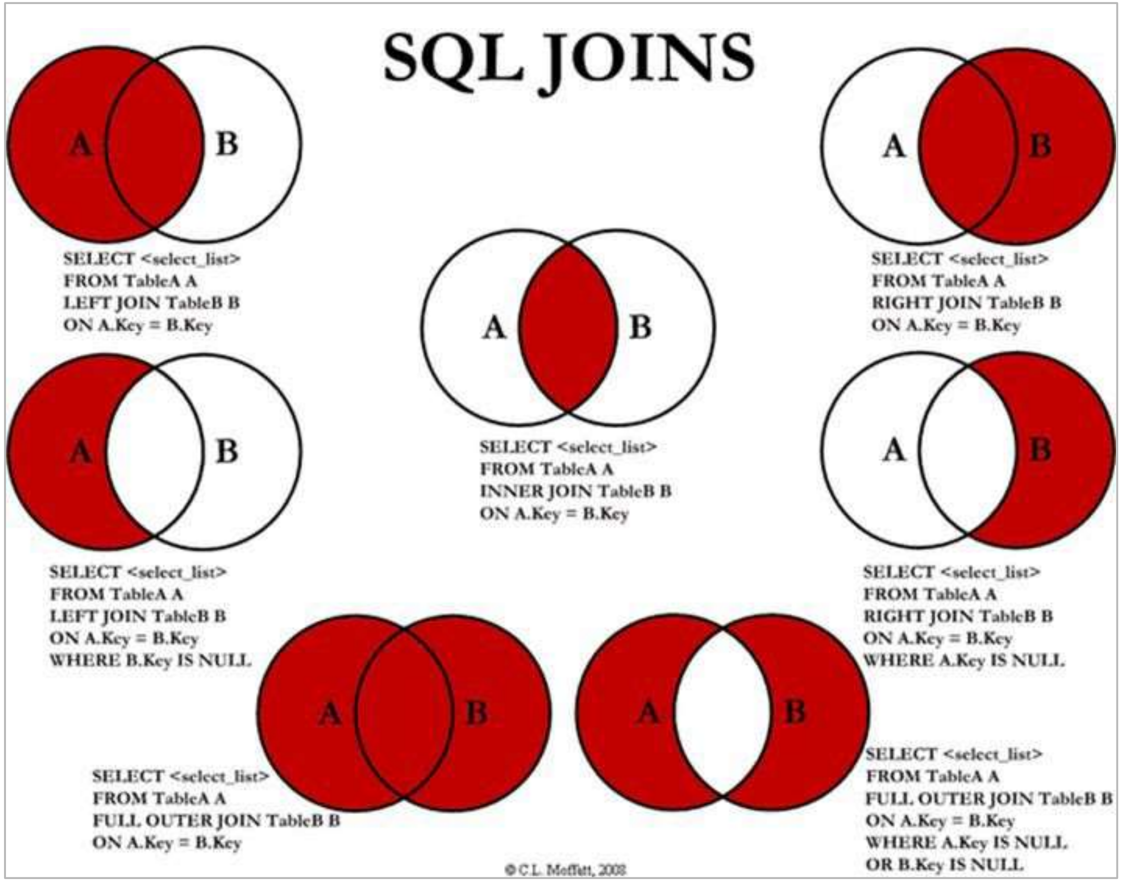
- Left Joins return all records from left table, and matching records from right table.
- Right Joins return all records from right table, and matching records from left table.





Joins Syntax (good to print)

Notice the “ON” statement in each JOIN.



```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```

```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```

```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```

```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```

```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```

```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```

```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

© C.L. Moffatt, 2008

JOIN

When you have multiple tables that have fields in common and you want to see attributes from both tables.

```
SELECT a.market,
a.mascot_name,
b.conf_name
FROM `bigquery-public-data.ncaa_basketball.mascots` a
JOIN `bigquery-public-data.ncaa_basketball.mbb_teams` b
ON a.market = b.market;
```

JOIN (continued)

When you have multiple tables that have fields in common and you want to see attributes from both tables.

```
SELECT a.market,
a.mascot_name,
b.conf_name
FROM `bigquery-public-data.ncaa_basketball.mascots` a
JOIN `bigquery-public-data.ncaa_basketball.mbb_teams` b
ON a.market = b.market
WHERE b.conf_name IN ("Ivy", "Big Sky")
ORDER BY b.venue_capacity DESC;
```

JOIN (continued)

Top 5 teams and mascots with most total points scored

```
SELECT a.market,
a.mascot,
sum(b.points_game) total_points
FROM `bigquery-public-data.ncaa_basketball.mascots` a
JOIN `bigquery-public-data.ncaa_basketball.mbb_teams_games_sr` b
ON a.market = b.market
GROUP BY a.market, a.mascot
ORDER BY 3 DESC
LIMIT 5;
```

Practice Question 10:

(5 minutes)

Write a query that returns the `market`, `mascot`, and `mascot_name` from all colleges with `conf_name` equal to “Big 12”.

Practice Question Set #3

Use keywords such as **COUNT, DISTINCT, LIMIT, ORDER BY, WHERE, IN, NOT IN, AND, OR, AVG, SUM, MAX, GROUP BY, JOIN** to answer the following:

(*Refer to the ER Diagram)

1. What is the market, mascot, and venue_state of the 5 teams with the largest venue_capacity?
2. Which conf_name had the most total wins?

CASE statement

Create an IF statement in SQL

```
SELECT market, venue_name,
CASE WHEN venue_capacity > 15000 THEN 'large arena'
WHEN venue_capacity > 7500 THEN 'medium arena'
ELSE 'small arena'
END AS arena_size
FROM `bigquery-public-data.ncaa_basketball.mbb_teams`
```

RANK

Rank the values of a column in a specified order

```
SELECT market,
   RANK() OVER (ORDER BY SUM(personal_fouls) DESC) total_fouls_rank,
SUM(personal_fouls) total_fouls
FROM `bigquery-public-data.ncaa_basketball.mbb_teams_games_sr`
GROUP BY market
ORDER BY 2 ASC;
```

Rank the markets by total personal fouls

RANK (example 2)

```
SELECT market, conf_name,
   RANK() OVER (PARTITION BY conf_name ORDER BY SUM(personal_fouls) DESC) conf_fouls_rank
FROM `bigquery-public-data.ncaa_basketball.mbb_teams_games_sr`
--WHERE conf_name IN ("Centennial Conference", "Big East")
GROUP BY market, conf_name
ORDER BY 3 ASC;
```

Total Personal Fouls rank within the team’s conference

Practice Question 11:

(5 minutes)

Write a query that returns the `scheduled_date`, `market`, and a CASE statement that returns “Ejected player” if `ejections` > 0 and “No ejected player if `ejections` = 0 from the `mbb_teams_games_sr table.`

UNION ALL

Combine columns into a single column

https://chartio.com/resources/tutorials/how-to-union-queries-in-google-bigquery/#using-the-union-option-in-standard-sql

```
SELECT market

FROM  `bigquery-public-data.ncaa_basketball.mbb_teams`

UNION ALL

SELECT mascot

FROM  `bigquery-public-data.ncaa_basketball.mascots`
```

Subquery

INSERT

Enter the following into the right-most window, then click “Run”:

```
INSERT INTO parks

VALUES

        (59, 'Baruch College', 32, 2, '1919-01-01');



SELECT * FROM parks;
```

Insert a new value into the Parks table

UPDATE

Enter the following into the right-most window, then click “Run”:

```
UPDATE Parks_Rating
SET rating = 5.0
WHERE park_id = 3;


SELECT * FROM Parks_Rating
```

Update a rating in the Parks_Rating table

DELETE

Enter the following into the right-most window, then click "Run":

```
DELETE FROM Parks_Rating

WHERE tent_campers < 100;


SELECT * FROM Parks_Rating
```

CREATE VIEW

A view is a virtual table that dynamically retrieves data each time it is called.

```
CREATE VIEW `myproject.mydataset.top_3_points`
AS SELECT market,
SUM(points_game)
FROM `bigquery-public-data.ncaa_basketball.mbb_teams_games_sr`
ORDER BY 2 DESC
LIMIT 3;
```

```
SELECT * FROM `myproject.mydataset.top_3_points`
```

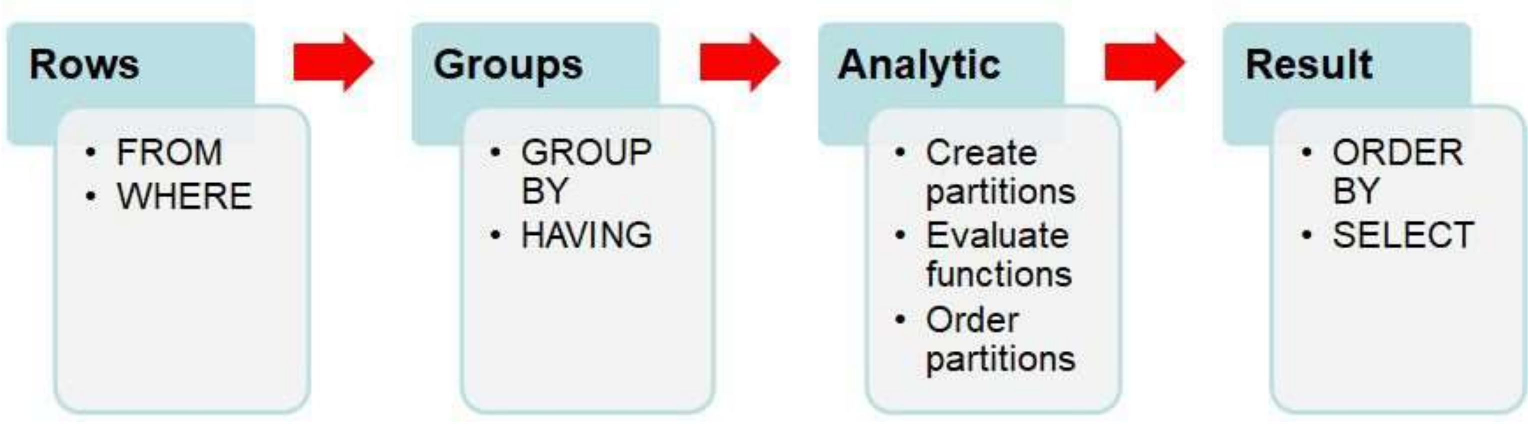
STORED PROCEDURE

A Stored Procedure is a prepared chunk of SQL code that you can save, and pass values into, so the code can be reused at any time with new values.

```
CREATE FUNCTION delete_parks_before (before_date DATE)
LANGUAGE plpgsql
AS $$
BEGIN
    DELETE FROM parks
    WHERE founded < before_date;
END;
$$;
```

Query Clause Evaluation Order

SQL queries are run in a specific order:



Rows

- FROM
- WHERE

Groups

- GROUP BY
- HAVING

Analytic

- Create partitions
- Evaluate functions
- Order partitions

Result

- ORDER BY
- SELECT

Practice Question Set #4

Use keywords such as **COUNT, DISTINCT, LIMIT, ORDER BY, WHERE, IN, NOT IN, AND, OR, AVG, SUM, MAX, GROUP BY, JOIN, INSERT, DELTE, UPDATE, VIEW** to answer the following:

1. Rank the conferences with the highest average attendance during losses. Which conferences rank first, second, and third?
2. Create a view to select the smallest 5 venue_capacity.

Homework:

1. Ensure your Google BigQuery account is setup
2. Homework #1 assigned on Blackboard
3. Final Project Survey on Blackboard in Class 1 folder
4. Reading BI Guidebook, Chapters 1 and 2