Introduction to MySQL
Session III

Oct 16, 2014
Coalition for Queens

Dan Goldin
dan@dangoldin.com

Agenda

1. Review GROUP BY/JOINs
2. Subqueries in an IN
3. Subselect JOINs
4. Exercises

GROUP BY

- Allows us to aggregate and group our data
  - SELECT [fields], [aggregate functions] FROM [table] GROUP BY [fields];

GROUP BY

- We need to make sure that what we're SELECTing matches what we're GROUPing BY

```
SELECT week,
  SUM(points) AS total_points,
  SUM(passing_tds) as total_passing_tds
FROM stats
GROUP BY week;
```

GROUP BY

Raw data

week	player_id	points
5	2	16.81
5	1	13.51
5	3	13.54
5	4	12.81
6	2	20.72
6	1	14.2
6	4	14.55
6	3	13.85
7	2	16.96
7	3	15.52
7	4	15.46
7	1	14.02
8	2	20.94
8	1	14.43
8	3	13.25

GROUP BY week

week	total_points
5	56.67
6	63.32
7	61.96
8	61.28

GROUP BY player_id

player_id	total_points
1	56.16
2	75.43
3	56.16
4	55.48

Exercises

- Describe the columns and data returned by the following query:

```
SELECT name AS player, count(*) AS count
FROM players
GROUP BY name;
```

Exercises

- Describe the columns and data returned by the following query:

```
SELECT week, count(*) AS num_games
FROM schedule
WHERE home_id IN (1,2,3,4,5,6)
GROUP BY week;
```

Exercises

- Describe the columns and data returned by the following query:

```
SELECT week,
  sum(rushing_tds) AS total_rushing_tds,
  max(rushing_tds) AS max_rushing_tds,
  sum(receiving_tds) AS titak_receiving_tds,
  max(receiving_tds) AS max_receiving_tds
FROM stats
WHERE points > 0
GROUP BY week;
```

Exercises

- Now write queries to do the following:
  - Retrieve the number of total projected points by week for all weeks past 11.
  - Retrieve the number of players in our dataset for each team (team_id).

JOIN

- Allows us to merge data from multiple tables (Think VLOOKUP in Excel)
  - SELECT [fields] FROM [table1] JOIN [table2] ON [join condition];

JOIN

- Without a join condition, we'll get the combination of everything.
- Question: If we join the following two tables, how many rows will the resulting table have?

positions

id	name
1	QB
2	RB
3	TE
4	WR

players

id	name	position_id	team_id
1	Carson Palmer	1	1
2	Matt Ryan	1	2
3	Joe Flacco	1	3
4	EJ Manuel	1	4
5	Cam Newton	1	5

JOIN

- To make it easier to write complicated queries, we can also alias table names.
  - Gives tables a shorter name to reference throughout a query
  - Allows us to include the same table multiple times

JOIN

```
SELECT week,
  name,
  name
FROM schedule
JOIN teams ON schedule.home_id = teams.id
JOIN teams ON schedule.away_id = teams.id;
```

```
SELECT s.week,
  t1.name AS home_team,
  t2.name AS away_team
FROM schedule AS s
JOIN teams AS t1 ON s.home_id = t1.id
JOIN teams AS t2 ON s.away_id = t2.id;
```

Exercises

- Describe the columns and data returned by the following queries:

```
SELECT
  p.name AS player_name,
  t.name AS team_name
FROM players p
JOIN teams t ON p.team_id = t.id
```

Exercises

- Describe the columns and data returned by the following queries:

```
SELECT
  p.name AS player_name,
  q.name AS position
FROM players p
JOIN positions q ON p.position_id = q.id
WHERE q.name in ('WR', 'TE');
```

Exercises

- Describe the columns and data returned by the following queries:

```
SELECT t.name AS team_name,
  count(*) as num_players
FROM teams t
JOIN players p ON t.id = p.team_id
GROUP BY t.name
ORDER BY num_players DESC;
```

Exercises

- Describe the columns and data returned by the following queries:

```
SELECT t.name, sum(s.points)
FROM stats s
JOIN players p on p.id = s.player_id
JOIN teams t on p.team_id = t.id
GROUP BY t.name;
```

Exercises

- Now write queries to do the following:
  - Retrieve the name of every home team playing in week 9.
  - Retrieve the name of every away team playing in week 6.

Subqueries

- Use a query within another query
    - SELECT [fields] FROM [table] WHERE [field] IN ([other query]);
    - SELECT [fields] FROM [table] JOIN ([other query]) ON [join condition];

Exercises

- Which is the inner query? What does it do?
- Describe the columns and data returned by the following queries:

```
SELECT name
FROM players WHERE team_id IN (
  SELECT id
  FROM teams
  WHERE name in ('NYJ', 'NYG')
);
```

Exercises

- Which is the inner query? What does it do?
- Describe the columns and data returned by the following queries:

```
SELECT sum(points)
FROM stats
WHERE week IN (
  SELECT week
  FROM schedule
  GROUP BY week
  HAVING count(*) < 16
);
```

Exercises

- Which is the inner query? What does it do?
- Describe the columns and data returned by the following queries:

```
SELECT p.name, s.total_points
FROM players AS p
JOIN (
  SELECT player_id, SUM(points) as total_points
  FROM stats
  GROUP BY player_id
  HAVING total_points > 100
) AS s on p.id = s.player_id;
```

Exercises

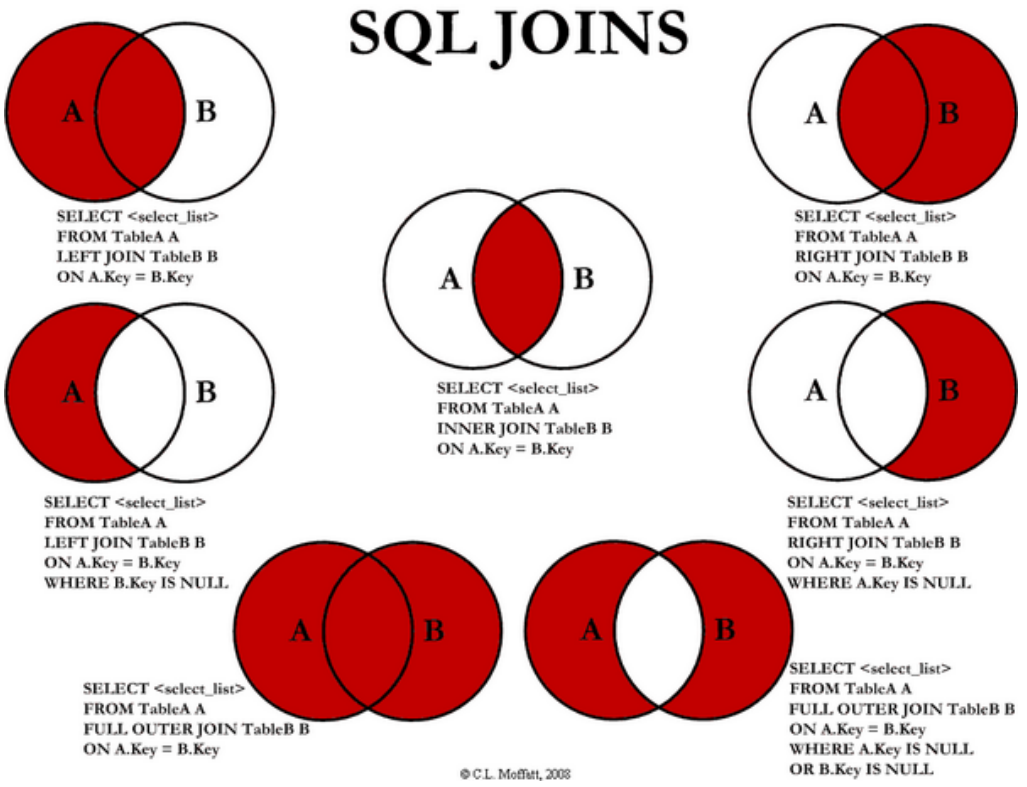
- Which is the inner query? What does it do?
- Describe the columns and data returned by the following queries:

```
SELECT t.name as team_name
FROM teams AS t
JOIN (
  SELECT team_id, count(1) as num_players
  FROM players
  GROUP BY team_id
  HAVING num_players > 10
) AS p on p.team_id = t.id;
```

Exercises

- Write queries (any way you want) to do the following:
  - Find the total projected points scored by each position.
  - Find the total projected points scored by each position by week.
  - Find the total projected points scored by each position by week for weeks > 10.

Different join types



From: [Visual Representation of Joins]