

Introduction to MySQL

Session II

Oct 9, 2014
Coalition for Queens

Dan Goldin
dan@dangoldin.com

Quick review

- SELECT
 - How many games are being played in week 9?
 - How many players have score projections in week 7?
 - What's the most projected passing touchdowns in week 11?

Two useful WHERE clauses

- LIKE
 - SELECT [fields] FROM [table] WHERE [field] 'pattern';
- IN
 - SELECT [fields] FROM [table] WHERE [field] in ([LIST]);

LIKE clause

- `SELECT * FROM players WHERE name LIKE '%John%';`
- `SELECT * FROM players WHERE name LIKE 'John %';`
- `SELECT * FROM players WHERE name LIKE '%John';`

IN clause

- `SELECT * FROM positions WHERE name IN ('RB', 'WR');`
- `SELECT * FROM stats WHERE week IN (1,3,5,7);`

Exercises

- <https://github.com/dangoldin/mysql-class/blob/master/session2/exercises2.md>

GROUP BY

- What if we want to run the aggregate functions over a group?
 - Total points by player
 - Number of games per week
 - Number of players by position
 - Number of games for each home/away team pair

GROUP BY

- SELECT [fields], [aggregate_function] FROM [table] GROUP BY [fields];
- SELECT [fields], [aggregate_function] FROM [table] WHERE [condition] GROUP BY [fields];
- SELECT [fields], [aggregate_function] FROM [table] WHERE [condition] GROUP BY [fields] HAVING [condition2];

GROUP BY

- `SELECT player_id, sum(points) FROM stats GROUP BY player_id;`
- `SELECT week, count(*) FROM schedule GROUP BY week;`
- `SELECT position_id, count(*) FROM players GROUP BY position_id;`
- `SELECT home_id, away_id, count(*) FROM schedule GROUP BY home_id, away_id;`
- `SELECT player_id, sum(points) FROM stats GROUP BY player_id HAVING sum(points) > 250;`

Exercises

- <https://github.com/dangoldin/mysql-class/blob/master/session2/exercises2.md>

JOIN

- So far, we've only been doing queries on a single table.

JOIN

- `SELECT [fields] FROM [table1] JOIN [table2] ON [join condition];`
- `SELECT [fields] FROM [table1] JOIN [table2] ON [join condition] JOIN [table3] ON [join condition];`
- `SELECT [fields] FROM [table1] JOIN [table2] ON [join condition] JOIN [table3] ON [join condition] WHERE [condition] ORDER BY [field];`

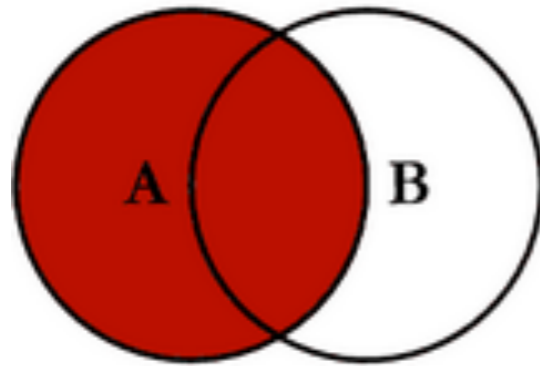
JOIN

- `SELECT * FROM players JOIN positions ON players.position_id = positions.id;`
- `SELECT players.name, positions.name FROM players JOIN positions ON players.position_id = positions.id;`
- `SELECT p.name, pos.name as position FROM players as p JOIN positions as pos ON p.position_id = pos.id;`
- `SELECT p.name, pos.name as position FROM players as p JOIN positions as pos ON p.position_id = pos.id WHERE pos.name in ('RB', 'WR') ORDER BY p.name;`

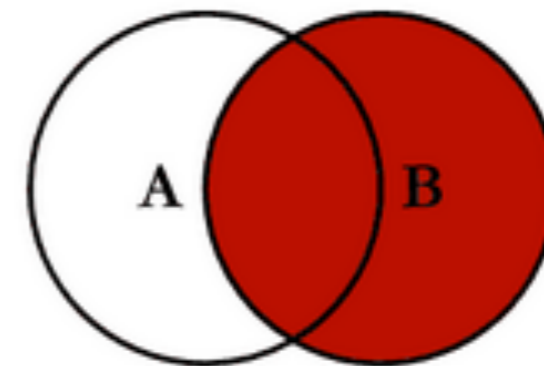
Exercises

- <https://github.com/dangoldin/mysql-class/blob/master/session2/exercises2.md>

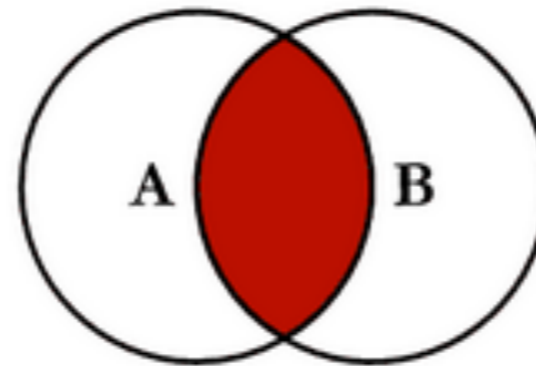
SQL JOINS



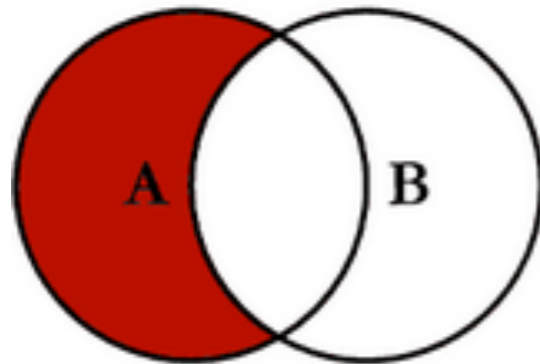
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



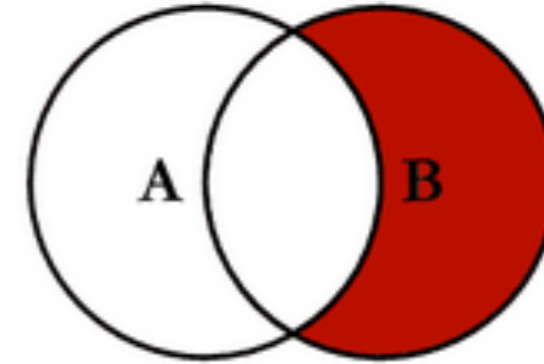
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



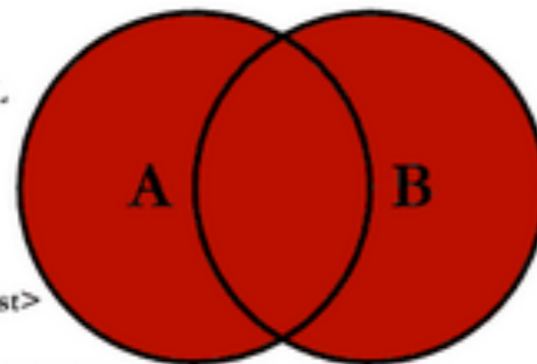
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



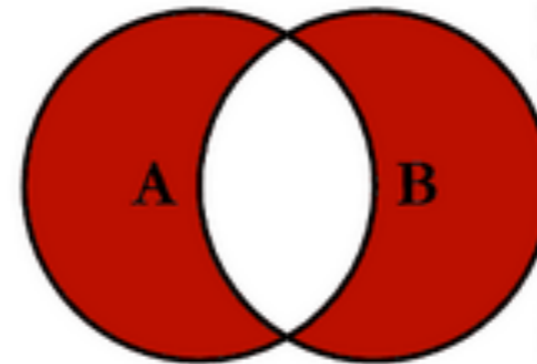
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



```
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

- From: <http://www.codeproject.com/Articles/33052/Visual-Representation-of-SQL-Joins>