Wharton
University of Pennsylvania

Workshop 2: Intro to SQL and R

WUDAC

Spring 2018

Roadmap

I. What is SQL and why is it important

II. What is R and why is it important

III. Code!

Sign-in here: https://goo.gl/forms/wY5WhaDqzwDPEU863

About me

What is SQL

What is SQL

What you can do

- Make data frames
- **Pull data from tables extremely quickly and logicly**

What you can't do

- Plots
- "Functions"
- ML

“I can’t even make plots?!! SQL sounds so pointless!”

Why is SQL important

How are hotels in Philadelphia converting from searches?

Searches	Locations	Hotels
100,000,000,000 rows	5000 rows	1,000,000 rows

Why is it important

- Helps you find data quickly
- Commands are like english, so it's easy to read and write
- Almost every database needs SQL to pull information from it
- 80% of DS code is SQL

- Extraordinarily easy to learn

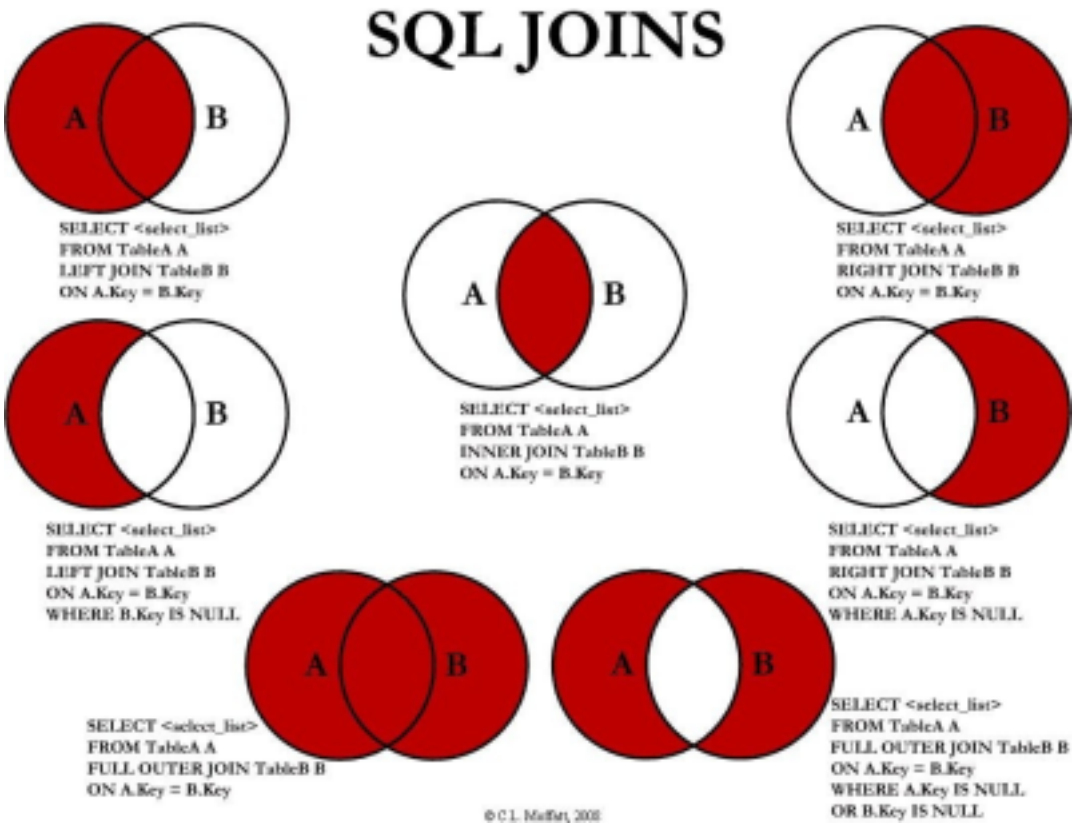
Code

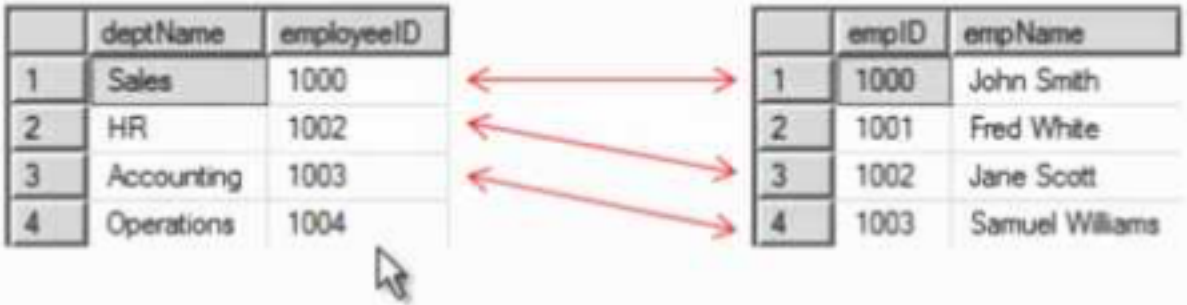
Summary so Far

SQL Queries

- Select * from ___
- Select ___ from ___
- Select ___ from ___ where ___
- Select ___ from ___ where ___ order by ___

Joins


SQL JOINS
A
B
SELECT <select_list>
FROM TableA A
LEFT 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
RIGHT 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



	deptName	employeeID
1	Sales	1000
2	HR	1002
3	Accounting	1003
4	Operations	1004

	empID	empName
1	1000	John Smith
2	1001	Fred White
3	1002	Jane Scott
4	1003	Samuel Williams

Why is SQL important

How are hotels in Philadelphia converting from searches?

Searches	Locations	Hotels
100,000,000,000 rows	5000 rows	1,000,000 rows

Bad Join

Select football.gameId, Football.yards, football2.gameId,football2.yards

From football

Join football2

On football.gameId = football2.gameId

football

GameID	Yards
1	5
1	6
2	10
2	4

football2

GameID	Yards
1	5
1	6
2	10
2	4

=

Football. GameID	Football. yards	Football2. yards
1	5	5
1	5	6
1	6	5
1	6	6

Good Join

```
football_join <- sqldf('Select
  a.*,
  yrdline100,
  PosTeamScore,
  DefTeamScore,
  ScoreDiff from football_clean a
      join football2 b
          on a.gameID = b.gameID
          and a.TimeSecs = b.TimeSecs

')
```

Reference to everything in table 'a'

Only in football2

Joining tables

Join clauses that are equal in both tables

SQL CHEAT SHEET http://www.sqltutorial.org

QUERYING DATA FROM A TABLE

```
SELECT c1, c2 FROM t;
```
Query data in columns c1, c2 from a table

```
SELECT * FROM t;
```
Query all rows and columns from a table

```
SELECT c1, c2 FROM t
WHERE condition;
```
Query data and filter rows with a condition

```
SELECT DISTINCT c1 FROM t
WHERE condition;
```
Query distinct rows from a table

```
SELECT c1, c2 FROM t
ORDER BY c1 ASC [DESC];
```
Sort the result set in ascending or descending order

```
SELECT c1, c2 FROM t
ORDER BY c1
LIMIT n OFFSET offset;
```
Skip *offset* of rows and return the next n rows

```
SELECT c1, aggregate(c2)
FROM t
GROUP BY c1;
```
Group rows using an aggregate function

```
SELECT c1, aggregate(c2)
FROM t
GROUP BY c1
HAVING condition;
```
Filter groups using HAVING clause

QUERYING FROM MULTIPLE TABLES

```
SELECT c1, c2
FROM t1
INNER JOIN t2 ON condition;
```
Inner join t1 and t2

```
SELECT c1, c2
FROM t1
LEFT JOIN t2 ON condition;
```
Left join t1 and t1

```
SELECT c1, c2
FROM t1
RIGHT JOIN t2 ON condition;
```
Right join t1 and t2

```
SELECT c1, c2
FROM t1
FULL OUTER JOIN t2 ON condition;
```
Perform full outer join

```
SELECT c1, c2
FROM t1
CROSS JOIN t2;
```
Produce a Cartesian product of rows in tables

```
SELECT c1, c2
FROM t1, t2;
```
Another way to perform cross join

```
SELECT c1, c2
FROM t1 A
INNER JOIN t2 B ON condition;
```
Join t1 to itself using INNER JOIN clause

USING SQL OPERATORS

```
SELECT c1, c2 FROM t1
UNION [ALL]
SELECT c1, c2 FROM t2;
```
Combine rows from two queries

```
SELECT c1, c2 FROM t1
INTERSECT
SELECT c1, c2 FROM t2;
```
Return the intersection of two queries

```
SELECT c1, c2 FROM t1
MINUS
SELECT c1, c2 FROM t2;
```
Subtract a result set from another result set

```
SELECT c1, c2 FROM t1
WHERE c1 [NOT] LIKE pattern;
```
Query rows using pattern matching %, _

```
SELECT c1, c2 FROM t
WHERE c1 [NOT] IN value_list;
```
Query rows in a list

```
SELECT c1, c2 FROM t
WHERE  c1 BETWEEN low AND high;
```
Query rows between two values

```
SELECT c1, c2 FROM t
WHERE  c1 IS [NOT] NULL;
```
Check if values in a table is NULL or not

Pass v Runs

Run

Pass

https://public.tableau.com/profile/jack.soslow#!/vizhome/PredictedLikelihoodofRunningvsDownDistance/Dashboard1