



# SQL: Structured Query Language

Adam Peterson

University of Michigan: Department of Biostatistics

materials found at  
[https://github.com/apeterson91/computing\\_workshops/workshop\\_2](https://github.com/apeterson91/computing_workshops/workshop_2)



# Agenda

- ① Motivation
- ② Keywords:
  - 'SELECT'
  - 'WHERE'
  - 'GROUP BY'
- ③ Inner queries
- ④ Joins
  - Inner
  - Outer
  - Left, Right

# Motivation

- What is SQL and why is it important?



# Motivation



- What is SQL and why is it important?
- SQL is a programming language that allows one to programmatically access data in databases

# Motivation



- What is SQL and why is it important?
- SQL is a programming language that allows one to programmatically access data in databases
- i.e. With SQL we can *query* a database for just the information we want and nothing else.



# Set-Up:1

Table : Student\_Table

S_ID	First_Name	Last_Name	Student_Age	Student_Major
1	John	Smith	23	Biostatistics
2	Anne	Doroughty	21	Biostatistics
3	Anthony	Jones	19	Statistics



## Set-Up:2

Loading Data into SAS for PROC SQL exercises



## SELECT Keyword

# SELECT

## SELECT: Formulaic

```
SELECT <ColumnNames> FROM <TableName>
```

## SELECT: Example

```
SELECT First_Name FROM Student_Table;
```

First_Name
John
Anne
Anthony





SELECT Keyword

# SELECT example- SAS

Code fairly simple...



SELECT Keyword

# SELECT example- SAS

Code fairly simple...

```
PROC SQL;  
SELECT First_Name  
FROM Student_Table;  
QUIT;
```



WHERE Keyword

# Where

## WHERE: Formulaic

```
SELECT <ColumnNames> FROM <TableName> WHERE  
<Condition>
```

## WHERE: Example

```
SELECT First_Name FROM Student_Table WHERE  
Student_Age<22;
```

First_Name
Anne
Anthony



# WHERE example- SAS

```
PROC SQL;  
SELECT First_Name  
FROM Student_Table  
WHERE Student_Age < 22; QUIT;
```



## GROUP BY Keyword

## GROUP BY: Formulaic

```
SELECT < Aggregate_Function(ColumnNames) > FROM  
<TableName> GROUPBY <GroupColumnName>
```

## GROUP BY: Example

```
SELECT Student_Major, SUM(Student_Age) FROM Student_Table  
WHERE Student_Age >19 GROUP BY Student_Major;
```

Student_Major	SUM(Student_Age)
Biostatistics	44



# Group By example- SAS

```
PROC SQL;  
SELECT SUM(Student_Age)  
FROM Student_Table  
WHERE Student_Age > 19  
GROUP BY Student_Major;  
QUIT;
```



## Inner Query: Formulaic

```
SELECT <ColumnNames> FROM (SELECT <ColumnNames>  
FROM <TableName> )
```

## Inner Query: Example

```
SELECT First_Name FROM (SELECT * FROM Student_Table  
WHERE Student_Age<22) ;
```

First_Name
Anne
Anthony



# Inner Query By example- SAS

```
PROC SQL;  
SELECT First_Name FROM  
(SELECT * FROM Student_Table WHERE Student_Age < 22) ;  
QUIT;
```





## Quick Aside: Relational Databases

Let's talk a bit about how data is stored in tables in a relational database

- Unique Identifiers



## Quick Aside: Relational Databases

Let's talk a bit about how data is stored in tables in a relational database

- Unique Identifiers
- One-To-Many Relationship
  - One student, multiple classes

## Quick Aside: Relational Databases

Let's talk a bit about how data is stored in tables in a relational database

- Unique Identifiers
- One-To-Many Relationship
  - One student, multiple classes

StuClass\_Table

S_ID	Class_Num	Class_Name	Class_Dept
1	602	Statistical Inference II	Biostatistics
1	651	Applied Linear Regression II	Biostatistics
2	516	Epidemiology II	Epidemiology
3	601	Statistical Inference I	Biostatistics
3	531	Analysis of Time Series	Statistics

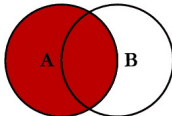


# Type of Joins

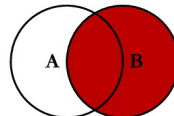
- Left Outer
  - Get everything from 'left' table, matching items from right
- Right Outer
  - Get everything from 'right' table, matching items from left
- Inner
  - Only items that are found in both tables
- More...

# Type of Joins

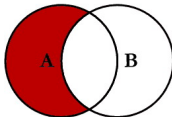
## SQL JOINS



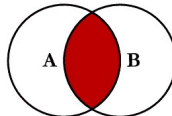
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



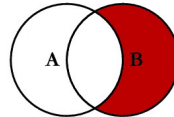
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



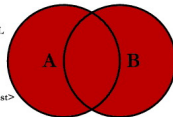
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



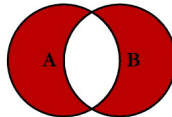
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
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.J. McElroy 2008



# Join: Example

## Join: Formulaic

```
SELECT Table_1.<ColumnNames>, Table_2.<ColumnNames>  
FROM <TableName> <Type> JOIN
```

## Join: Example

```
SELECT Student_Table.First_Name StuClass_Table.Class_Name  
FROM Student_Table INNER JOIN StuClass_Table ON  
Student_Table.S_ID = StuClass_Table.S_ID;
```

# Join: Example

## Join: Example

```
SELECT Student_Table.First_Name, StuClass_Table.Class_Name  
FROM Student_Table INNER JOIN StuClass_Table ON  
Student_Table.S_ID = StuClass_Table.S_ID;
```

Student_ID	First_Name	Class_Name
1	John	Statistical Inference II
1	John	Applied Linear Regression II
2	Anne	Epidemiology II
3	Anthony	Statistical Inference I
3	Anthony	Analysis of Time Series



# SAS - Join Example

```
PROC SQL;  
SELECT Student_Table.First_Name,  
       StuClass_Table.Class_Name  
FROM Student_Table  
INNER JOIN StuClass_Table  
ON Student_Table.S_ID = StuClass_Table.S_ID ;  
QUIT;
```





# R, Python and beyond

SQL can be found anywhere there is data:

- Server databases: Postgresql, MySQL, MS SQL Server, ...
- 'Local' Databases - SQLite - cellphones, computers

We can query either of these using Python, R, or SAS using SQL



# Resources for Further Learning

- [pandasql](#)
- [sqldf](#)
- [CodeAcademy's SQL Course](#)
- [W3](#) - great reference



# Questions

Any Questions?