VISUALIZING SQL FUNCTIONALITY

Matthew Hall

WHAT IS SQL?

- **Structured Query Language (SQL)**
- **Made For Processing & Communicating with Relational Databases**
- **What is a Relational Database?**
  - Tables which can have pre-defined relationships

WHAT IS SQL USED FOR?

- **4 General Networking Commands**
  - Get, Put, Update, Delete
- **Interfacing with Databases!**
  - We write queries to interact with data
- **Analysis of Data**
  - Our queries can manipulate the underlying data too!

WHY IS THIS IMPORTANT FOR PYTHON?

- **Pandas Has Similar Functionality On and Between DataFrames!**

- **SQL-Like Operations Often Integral in Analysis of Data**
  - Summary Statistics, Mean, Median, Annual Totals, Quarterly Results

TYPES OF SQL OPERATIONS

And Their Counterpart in Pandas

(I'm using MySQL in the examples)

SELECT * FROM { TABLE }

- **The SELECT statement in SQL is a type of GET request**
  - We're asking for data!
- **FROM a specific TABLE in the database**
- **With All (*) Columns**
  - Rows too!

IN PYTHON / R

- **Get the data by accessing variable**

```
df
```

- **Get the data by accessing variable**

```
df
```

SELECT { COLS } FROM { TABLE }

- **Same as before, limited to selected columns!**
  - Still all rows!

IN PYTHON / R

```
df.name
```

```
df$name
```

```
df[["name","age"]]
```

```
df %>% select(name, age)
```

SELECT * FROM { } WHERE

- This is our first FILTER functionality to learn!
- When added to the previous SELECT statements, we can now filter out *rows* to return!
- We use a logical statement (predicate) to decide what to include!

SAMPLE SALES DATASET

Q1 Sales for a set of 10 stores with region info!

Date	Store ID	Region	Sales	COGS	SG&A Expenses
J-20	S-01	A	$105,000	$24,000	$35,000
J-20	S-02	A	$120,000	$35,000	$45,000
J-20	S-03	A	$115,000	$32,000	$52,000
J-20	S-04	B	$75,000	$12,000	$55,000
J-20	S-05	B	$65,000	$13,000	$62,000
J-20	S-06	C	$85,000	$23,000	$40,000
J-20	S-07	C	$98,000	$24,000	$42,000
J-20	S-08	D	$225,000	$60,000	$85,000
J-20	S-09	D	$275,000	$90,000	$105,000
J-20	S-10	D	$315,000	$128,000	$145,000
F-20	S-01	A	$115,000	$29,000	$35,000
F-20	S-02	A	$135,000	$40,000	$45,000
F-20	S-03	A	$95,000	$37,000	$52,000
F-20	S-04	B	$78,000	$17,000	$55,000
F-20	S-05	B	$98,000	$18,000	$62,000
F-20	S-06	C	$55,000	$28,000	$40,000
F-20	S-07	C	$65,000	$29,000	$42,000
F-20	S-08	D	$226,000	$65,000	$85,000
F-20	S-09	D	$255,000	$95,000	$105,000
F-20	S-10	D	$320,000	$133,000	$145,000
M-20	S-01	A	$135,000	$34,000	$35,000
M-20	S-02	A	$140,000	$45,000	$45,000
M-20	S-03	A	$135,000	$42,000	$52,000
M-20	S-04	B	$98,000	$22,000	$55,000
M-20	S-05	B	$97,000	$23,000	$62,000
M-20	S-06	C	$68,000	$33,000	$40,000
M-20	S-07	C	$55,000	$34,000	$42,000
M-20	S-08	D	$320,000	$70,000	$85,000
M-20	S-09	D	$300,000	$100,000	$105,000
M-20	S-10	D	$290,000	$138,000	$145,000

FILTER: STEP #1

Start with logical argument that can filter out rows

We want to return the ones that evaluate True!

Example:

Let's filter for Region A!

To the right is all of Region A

Date	Store ID	Region	Sales	COGS	SG&A Expenses
J-20	S-01	A	$105,000	$24,000	$35,000
J-20	S-02	A	$120,000	$35,000	$45,000
J-20	S-03	A	$115,000	$32,000	$52,000
J-20	S-04	B	$75,000	$12,000	$55,000
J-20	S-05	B	$65,000	$13,000	$62,000
J-20	S-06	C	$85,000	$23,000	$40,000
J-20	S-07	C	$98,000	$24,000	$42,000
J-20	S-08	D	$225,000	$60,000	$85,000
J-20	S-09	D	$275,000	$90,000	$105,000
J-20	S-10	D	$315,000	$128,000	$145,000
F-20	S-01	A	$115,000	$29,000	$35,000
F-20	S-02	A	$135,000	$40,000	$45,000
F-20	S-03	A	$95,000	$37,000	$52,000
F-20	S-04	B	$78,000	$17,000	$55,000
F-20	S-05	B	$98,000	$18,000	$62,000
F-20	S-06	C	$55,000	$28,000	$40,000
F-20	S-07	C	$65,000	$29,000	$42,000
F-20	S-08	D	$226,000	$65,000	$85,000
F-20	S-09	D	$255,000	$95,000	$105,000
F-20	S-10	D	$320,000	$133,000	$145,000
M-20	S-01	A	$135,000	$34,000	$35,000
M-20	S-02	A	$140,000	$45,000	$45,000
M-20	S-03	A	$135,000	$42,000	$52,000
M-20	S-04	B	$98,000	$22,000	$55,000
M-20	S-05	B	$97,000	$23,000	$62,000
M-20	S-06	C	$68,000	$33,000	$40,000
M-20	S-07	C	$55,000	$34,000	$42,000
M-20	S-08	D	$320,000	$70,000	$85,000
M-20	S-09	D	$300,000	$100,000	$105,000
M-20	S-10	D	$290,000	$138,000	$145,000

FILTER: RESULT

We're now left with all rows that are just Region A

Date	Store ID	Region	Sales	COGS	SG&A Expenses
J-20	S-01	A	$105,000	$24,000	$35,000
J-20	S-02	A	$120,000	$35,000	$45,000
J-20	S-03	A	$115,000	$32,000	$52,000
F-20	S-01	A	$115,000	$29,000	$35,000
F-20	S-02	A	$135,000	$40,000	$45,000
F-20	S-03	A	$95,000	$37,000	$52,000
M-20	S-01	A	$135,000	$34,000	$35,000
M-20	S-02	A	$140,000	$45,000	$45,000
M-20	S-03	A	$135,000	$42,000	$52,000

IN PYTHON / R

```
df[df.Region == "A"]
```

```
filter(df, Region == "A")
```

AGGREGATION

- In SQL, Aggregation Refers to Calculations Upon Common Columns Between Rows!
- Aggregation Generally Refers to a Calculation On The Entire Column
- Example: Sum of Sales

AGGREGATION: STEP #1

Start with Identifying The Data To Aggregate

We're Going To Start With Columns!

On the Right, You Can See The Color Coding of Data Points To Calculate On

Date	Store ID	Region	Sales	COGS	SG&A Expenses
J-20	S-01	A	$105,000	$24,000	$35,000
J-20	S-02	A	$120,000	$35,000	$45,000
J-20	S-03	A	$115,000	$32,000	$52,000
J-20	S-04	B	$75,000	$12,000	$55,000
J-20	S-05	B	$65,000	$13,000	$62,000
J-20	S-06	C	$85,000	$23,000	$40,000
J-20	S-07	C	$98,000	$24,000	$42,000
J-20	S-08	D	$225,000	$60,000	$85,000
J-20	S-09	D	$275,000	$90,000	$105,000
J-20	S-10	D	$315,000	$128,000	$145,000
F-20	S-01	A	$115,000	$29,000	$35,000
F-20	S-02	A	$135,000	$40,000	$45,000
F-20	S-03	A	$95,000	$37,000	$52,000
F-20	S-04	B	$78,000	$17,000	$55,000
F-20	S-05	B	$98,000	$18,000	$62,000
F-20	S-06	C	$55,000	$28,000	$40,000
F-20	S-07	C	$65,000	$29,000	$42,000
F-20	S-08	D	$226,000	$65,000	$85,000
F-20	S-09	D	$255,000	$95,000	$105,000
F-20	S-10	D	$320,000	$133,000	$145,000
M-20	S-01	A	$135,000	$34,000	$35,000
M-20	S-02	A	$140,000	$45,000	$45,000
M-20	S-03	A	$135,000	$42,000	$52,000
M-20	S-04	B	$98,000	$22,000	$55,000
M-20	S-05	B	$97,000	$23,000	$62,000
M-20	S-06	C	$68,000	$33,000	$40,000
M-20	S-07	C	$55,000	$34,000	$42,000
M-20	S-08	D	$320,000	$70,000	$85,000
M-20	S-09	D	$300,000	$100,000	$105,000
M-20	S-10	D	$290,000	$138,000	$145,000

AGGREGATION: STEP 2

Choose What Statistical Moment To Calculate for the Groups!

- **Sum, Product, Max, Min, Std, Var, etc.**

Keep In Mind The Meaning of the Data Changes!

The Sum of Sales is the Cumulate Revenue Between All 10 Stores For the First Quarter

Date	Store ID	Region	Sales	COGS	SG&A Expenses
J-20	S-01	A	$105,000	$24,000	$35,000
J-20	S-02	A	$120,000	$35,000	$45,000
J-20	S-03	A	$115,000	$32,000	$52,000
J-20	S-04	B	$75,000	$12,000	$55,000
J-20	S-05	B	$65,000	$13,000	$62,000
J-20	S-06	C	$85,000	$23,000	$40,000
J-20	S-07	C	$98,000	$24,000	$42,000
J-20	S-08	D	$225,000	$60,000	$85,000
J-20	S-09	D	$275,000	$90,000	$105,000
J-20	S-10	D	$315,000	$128,000	$145,000
F-20	S-01	A	$115,000	$29,000	$35,000
F-20	S-02	A	$135,000	$40,000	$45,000
F-20	S-03	A	$95,000	$37,000	$52,000
F-20	S-04	B	$78,000	$17,000	$55,000
F-20	S-05	B	$98,000	$18,000	$62,000
F-20	S-06	C	$55,000	$28,000	$40,000
F-20	S-07	C	$65,000	$29,000	$42,000
F-20	S-08	D	$226,000	$65,000	$85,000
F-20	S-09	D	$255,000	$95,000	$105,000
F-20	S-10	D	$320,000	$133,000	$145,000
M-20	S-01	A	$135,000	$34,000	$35,000
M-20	S-02	A	$140,000	$45,000	$45,000
M-20	S-03	A	$135,000	$42,000	$52,000
M-20	S-04	B	$98,000	$22,000	$55,000
M-20	S-05	B	$97,000	$23,000	$62,000
M-20	S-06	C	$68,000	$33,000	$40,000
M-20	S-07	C	$55,000	$34,000	$42,000
M-20	S-08	D	$320,000	$70,000	$85,000
M-20	S-09	D	$300,000	$100,000	$105,000
M-20	S-10	D	$290,000	$138,000	$145,000
		Sum	**$4,558,000**	**$1,473,000**	**$1,998,000**

AGGREGATION: RESULT

We're left with our summary statistics!

Note: The average of this column is the average monthly revenue per store, per month.

	Sales	COGS	SG&A Expenses
Sum of …	$4,558,000	$1,473,000	$1,998,000
Average of …	$151,933	$49,100	$66,600

IN PYTHON / R

```
df.sum()
```

```
colSums(df %>%
select_if(is.numeric))
```

By default, Pandas ignores non-numerical columns. We need to add the select_if argument to get just the numerical in R.

If it's all numerical:

```
colSums(df)
```

GROUP BY

- **Group By allows us to form groups in the Data Frame, then calculate aggregate statistics for the groups formed**

- **Group By allows us to create statistics for similar data points!**

- **Common Group By Parameters:**
  - Time, Locations, Categorical Info, Quarterly, Annual, etc.

GROUP BY: STEP 1

Select The Columns / Data Points We Want To Create Groups On

In The Example on the Right, I Chose To Group By Month!

I color coded the groups that are formed based upon months.

Date	Store ID	Region	Sales	COGS	SG&A Expenses
J-20	S-01	A	$105,000	$24,000	$35,000
J-20	S-02	A	$120,000	$35,000	$45,000
J-20	S-03	A	$115,000	$32,000	$52,000
J-20	S-04	B	$75,000	$12,000	$55,000
J-20	S-05	B	$65,000	$13,000	$62,000
J-20	S-06	C	$85,000	$23,000	$40,000
J-20	S-07	C	$98,000	$24,000	$42,000
J-20	S-08	D	$225,000	$60,000	$85,000
J-20	S-09	D	$275,000	$90,000	$105,000
J-20	S-10	D	$315,000	$128,000	$145,000
F-20	S-01	A	$115,000	$29,000	$35,000
F-20	S-02	A	$135,000	$40,000	$45,000
F-20	S-03	A	$95,000	$37,000	$52,000
F-20	S-04	B	$78,000	$17,000	$55,000
F-20	S-05	B	$98,000	$18,000	$62,000
F-20	S-06	C	$55,000	$28,000	$40,000
F-20	S-07	C	$65,000	$29,000	$42,000
F-20	S-08	D	$226,000	$65,000	$85,000
F-20	S-09	D	$255,000	$95,000	$105,000
F-20	S-10	D	$320,000	$133,000	$145,000
M-20	S-01	A	$135,000	$34,000	$35,000
M-20	S-02	A	$140,000	$45,000	$45,000
M-20	S-03	A	$135,000	$42,000	$52,000
M-20	S-04	B	$98,000	$22,000	$55,000
M-20	S-05	B	$97,000	$23,000	$62,000
M-20	S-06	C	$68,000	$33,000	$40,000
M-20	S-07	C	$55,000	$34,000	$42,000
M-20	S-08	D	$320,000	$70,000	$85,000
M-20	S-09	D	$300,000	$100,000	$105,000
M-20	S-10	D	$290,000	$138,000	$145,000

GROUP BY: STEP 2

Decide The Columns & Statistical Moments We Want to Run Summary Calculations On

On the Right, I chose to summarize with the sum of the sales column.

Now, The Data Represents The Total Monthly Revenue Across All 10 Stores in Each Month

Date	Store ID	Region	Sales			Sum of Sales
J-20	S-01	A	$105,000			$1,478,000
J-20	S-02	A	$120,000			$1,442,000
J-20	S-03	A	$115,000			$1,638,000
J-20	S-04	B	$75,000			
J-20	S-05	B	$65,000			
J-20	S-06	C	$85,000			
J-20	S-07	C	$98,000			
J-20	S-08	D	$225,000			
J-20	S-09	D	$275,000			
J-20	S-10	D	$315,000			
F-20	S-01	A	$115,000			
F-20	S-02	A	$135,000			
F-20	S-03	A	$95,000			
F-20	S-04	B	$78,000			
F-20	S-05	B	$98,000			
F-20	S-06	C	$55,000			
F-20	S-07	C	$65,000			
F-20	S-08	D	$226,000			
F-20	S-09	D	$255,000			
F-20	S-10	D	$320,000			
M-20	S-01	A	$135,000			
M-20	S-02	A	$140,000			
M-20	S-03	A	$135,000			
M-20	S-04	B	$98,000			
M-20	S-05	B	$97,000			
M-20	S-06	C	$68,000			
M-20	S-07	C	$55,000			
M-20	S-08	D	$320,000			
M-20	S-09	D	$300,000			
M-20	S-10	D	$290,000			

GROUP BY: RESULT

In the End, We Just Have Our Summary Statistics!

We've Went From a Raw Dataset to Monthly Sales Data About Our Company!

Sum of Sales
$1,478,000
$1,442,000
$1,638,000

IN PYTHON / R

```
df.groupby("month")["sales"].sum()
```

If You Have a DateTime Index, `"month"` will be

```
df.index.month
```

GROUP BY: EXAMPLE 2

For Our Second Example, We're Going to Group By 2 Data Points!

Month & Region!

We start by forming the same groups as before!

Date	Store ID	Region	Sales	COGS	SG&A Expenses
J-20	S-01	A	$105,000	$24,000	$35,000
J-20	S-02	A	$120,000	$35,000	$45,000
J-20	S-03	A	$115,000	$32,000	$52,000
J-20	S-04	B	$75,000	$12,000	$55,000
J-20	S-05	B	$65,000	$13,000	$62,000
J-20	S-06	C	$85,000	$23,000	$40,000
J-20	S-07	C	$98,000	$24,000	$42,000
J-20	S-08	D	$225,000	$60,000	$85,000
J-20	S-09	D	$275,000	$90,000	$105,000
J-20	S-10	D	$315,000	$128,000	$145,000
F-20	S-01	A	$115,000	$29,000	$35,000
F-20	S-02	A	$135,000	$40,000	$45,000
F-20	S-03	A	$95,000	$37,000	$52,000
F-20	S-04	B	$78,000	$17,000	$55,000
F-20	S-05	B	$98,000	$18,000	$62,000
F-20	S-06	C	$55,000	$28,000	$40,000
F-20	S-07	C	$65,000	$29,000	$42,000
F-20	S-08	D	$226,000	$65,000	$85,000
F-20	S-09	D	$255,000	$95,000	$105,000
F-20	S-10	D	$320,000	$133,000	$145,000
M-20	S-01	A	$135,000	$34,000	$35,000
M-20	S-02	A	$140,000	$45,000	$45,000
M-20	S-03	A	$135,000	$42,000	$52,000
M-20	S-04	B	$98,000	$22,000	$55,000
M-20	S-05	B	$97,000	$23,000	$62,000
M-20	S-06	C	$68,000	$33,000	$40,000
M-20	S-07	C	$55,000	$34,000	$42,000
M-20	S-08	D	$320,000	$70,000	$85,000
M-20	S-09	D	$300,000	$100,000	$105,000
M-20	S-10	D	$290,000	$138,000	$145,000

EX 2: STEP 2

Then, We Form Groups Within The Original Groups!

On The Right, We're Going to Make the Subgroups with the Region!

Note: This Works With The Dataset In Any Order, I Chose This Order to Make It Easy To See Groups!

Date	Store ID	Region	Sales	COGS	SG&A Expenses
J-20	S-01	A	$105,000	$24,000	$35,000
J-20	S-02	A	$120,000	$35,000	$45,000
J-20	S-03	A	$115,000	$32,000	$52,000
J-20	S-04	B	$75,000	$12,000	$55,000
J-20	S-05	B	$65,000	$13,000	$62,000
J-20	S-06	C	$85,000	$23,000	$40,000
J-20	S-07	C	$98,000	$24,000	$42,000
J-20	S-08	D	$225,000	$60,000	$85,000
J-20	S-09	D	$275,000	$90,000	$105,000
J-20	S-10	D	$315,000	$128,000	$145,000
F-20	S-01	A	$115,000	$29,000	$35,000
F-20	S-02	A	$135,000	$40,000	$45,000
F-20	S-03	A	$95,000	$37,000	$52,000
F-20	S-04	B	$78,000	$17,000	$55,000
F-20	S-05	B	$98,000	$18,000	$62,000
F-20	S-06	C	$55,000	$28,000	$40,000
F-20	S-07	C	$65,000	$29,000	$42,000
F-20	S-08	D	$226,000	$65,000	$85,000
F-20	S-09	D	$255,000	$95,000	$105,000
F-20	S-10	D	$320,000	$133,000	$145,000
M-20	S-01	A	$135,000	$34,000	$35,000
M-20	S-02	A	$140,000	$45,000	$45,000
M-20	S-03	A	$135,000	$42,000	$52,000
M-20	S-04	B	$98,000	$22,000	$55,000
M-20	S-05	B	$97,000	$23,000	$62,000
M-20	S-06	C	$68,000	$33,000	$40,000
M-20	S-07	C	$55,000	$34,000	$42,000
M-20	S-08	D	$320,000	$70,000	$85,000
M-20	S-09	D	$300,000	$100,000	$105,000
M-20	S-10	D	$290,000	$138,000	$145,000

EX 2: STEP 3

Select The Columns and Statistical Moments to Calculate For Each Group!

Keep In Mind The Meaning of the Data Too!

Image on Next Slide!

Date	Store ID	Region	Sales	COGS	SG&A Expenses
J-20	S-01	A	$105,000	$24,000	$35,000
J-20	S-02	A	$120,000	$35,000	$45,000
J-20	S-03	A	$115,000	$32,000	$52,000
J-20	S-04	B	$75,000	$12,000	$55,000
J-20	S-05	B	$65,000	$13,000	$62,000
J-20	S-06	C	$85,000	$23,000	$40,000
J-20	S-07	C	$98,000	$24,000	$42,000
J-20	S-08	D	$225,000	$60,000	$85,000
J-20	S-09	D	$275,000	$90,000	$105,000
J-20	S-10	D	$315,000	$128,000	$145,000
F-20	S-01	A	$115,000	$29,000	$35,000
F-20	S-02	A	$135,000	$40,000	$45,000
F-20	S-03	A	$95,000	$37,000	$52,000
F-20	S-04	B	$78,000	$17,000	$55,000
F-20	S-05	B	$98,000	$18,000	$62,000
F-20	S-06	C	$55,000	$28,000	$40,000
F-20	S-07	C	$65,000	$29,000	$42,000
F-20	S-08	D	$226,000	$65,000	$85,000
F-20	S-09	D	$255,000	$95,000	$105,000
F-20	S-10	D	$320,000	$133,000	$145,000
M-20	S-01	A	$135,000	$34,000	$35,000
M-20	S-02	A	$140,000	$45,000	$45,000
M-20	S-03	A	$135,000	$42,000	$52,000
M-20	S-04	B	$98,000	$22,000	$55,000
M-20	S-05	B	$97,000	$23,000	$62,000
M-20	S-06	C	$68,000	$33,000	$40,000
M-20	S-07	C	$55,000	$34,000	$42,000
M-20	S-08	D	$320,000	$70,000	$85,000
M-20	S-09	D	$300,000	$100,000	$105,000
M-20	S-10	D	$290,000	$138,000	$145,000

		Sum Of...		
Month	Region	Sales	COGS	SG&A
Jan	A	$340,000	$91,000	$132,000
Jan	B	$140,000	$25,000	$117,000
Jan	C	$183,000	$47,000	$82,000
Jan	D	$815,000	$278,000	$335,000

		Sum Of...		
Month	Region	Sales	COGS	SG&A
Feb	A	$345,000	$106,000	$132,000
Feb	B	$176,000	$35,000	$117,000
Feb	C	$120,000	$57,000	$82,000
Feb	D	$801,000	$293,000	$335,000

		Sum Of...		
Month	Region	Sales	COGS	SG&A
Mar	A	$410,000	$121,000	$132,000
Mar	B	$195,000	$45,000	$117,000
Mar	C	$123,000	$67,000	$82,000
Mar	D	$910,000	$308,000	$335,000

EX 2: RESULT

We're Now Left With Monthly Revenue and Expense Information On a Monthly and Regional Basis!

Data In This Format Could Help Us Determine Which Regions Are the Most Profitable During Which Time of the Year if We Had all 12 Months!

		Sum Of…		
Month	Region	Sales	COGS	SG&A
Jan	A	$340,000	$91,000	$132,000
	B	$140,000	$25,000	$117,000
	C	$183,000	$47,000	$82,000
	D	$815,000	$278,000	$335,000
Feb	A	$345,000	$106,000	$132,000
	B	$176,000	$35,000	$117,000
	C	$120,000	$57,000	$82,000
	D	$801,000	$293,000	$335,000
Mar	A	$410,000	$121,000	$132,000
	B	$195,000	$45,000	$117,000
	C	$123,000	$67,000	$82,000
	D	$910,000	$308,000	$335,000

IN PYTHON / R

```
df.groupby(["month","region"]).sum()
```

MERGE STATEMENTS

- **In Databases & The Real World, Related Data Is Often Spread Out and Disperse**
- **We Can Merge The Datasets Together Using the "Common Keys" Relating Data Points**
- **For Example:**
  - Everything in a CRM is going to be tied together by Customer ID or something similar

REGIONAL MGT INFORMATION

On The Right, We Have A Table of Information For Each Region!

Let's Put Them Together In One Dataset with The Previous Group By Result

ID	First	Last Initial	Region	Email
1	Matthew	H	A	mh@simon.edu
2	Alex	C	B	ac@simon.edu
3	Iris	D	C	id@simon.edu
4	Rex	G	D	rg@simon.edu

		Sum Of...		
Month	Region	Sales	COGS	SG&A
Jan	A	$340,000	$91,000	$132,000
	B	$140,000	$25,000	$117,000
	C	$183,000	$47,000	$82,000
	D	$815,000	$278,000	$335,000
Feb	A	$345,000	$106,000	$132,000
	B	$176,000	$35,000	$117,000
	C	$120,000	$57,000	$82,000
	D	$801,000	$293,000	$335,000
Mar	A	$410,000	$121,000	$132,000
	B	$195,000	$45,000	$117,000
	C	$123,000	$67,000	$82,000
	D	$910,000	$308,000	$335,000

STEP 1: IDENTIFY COMMON COLUMN

This is the common key between each table of data!

The columns do not have to have the same name but should have the *relational data.*

In Our Example, This Is The Regional Information to Relate Regional Sales to Regional Managers.

ID	First	Last Initial	Region	Email
1	Matthew	H	A	mh@simon.edu
2	Alex	C	B	ac@simon.edu
3	Iris	D	C	id@simon.edu
4	Rex	G	D	rg@simon.edu

		Sum Of…		
Month	Region	Sales	COGS	SG&A
Jan	A	$340,000	$91,000	$132,000
	B	$140,000	$25,000	$117,000
	C	$183,000	$47,000	$82,000
	D	$815,000	$278,000	$335,000
Feb	A	$345,000	$106,000	$132,000
	B	$176,000	$35,000	$117,000
	C	$120,000	$57,000	$82,000
	D	$801,000	$293,000	$335,000
Mar	A	$410,000	$121,000	$132,000
	B	$195,000	$45,000	$117,000
	C	$123,000	$67,000	$82,000
	D	$910,000	$308,000	$335,000

STEP 2: DATA MERGE

Once we figured out the common keys, we just need to run the merge operation!

There's a few merge types:

- **Inner**
- **Outer**
- **Left (Right)**

- **Inner Only Returns Rows with Matches From Both Sets**
- **Outer Returns All Rows From Both Data Sets**
- **Left Keeps All Rows From Left, and Adds Data From Right On Matches**

STEP 3: RESULT

After The Merge, Your Data Will Look Something Like This:

We Now Have The Manager Information Merged On The Sales Info.

This Would Be Great For Generating Reports!

Tip: Some Information Like The Index May Be Dropped Depending Upon Engine / Platform.

		Sum Of...					
Month	Region	Sales	COGS	SG&A	First	Last Initial	Email
Jan	A	$340,000	$91,000	$132,000	Matthew	H	mh@simon.edu
	B	$140,000	$25,000	$117,000	Alex	C	ac@simon.edu
	C	$183,000	$47,000	$82,000	Iris	D	id@simon.edu
	D	$815,000	$278,000	$335,000	Rex	G	rg@simon.edu
Feb	A	$345,000	$106,000	$132,000	Matthew	H	mh@simon.edu
	B	$176,000	$35,000	$117,000	Alex	C	ac@simon.edu
	C	$120,000	$57,000	$82,000	Iris	D	id@simon.edu
	D	$801,000	$293,000	$335,000	Rex	G	rg@simon.edu
Mar	A	$410,000	$121,000	$132,000	Matthew	H	mh@simon.edu
	B	$195,000	$45,000	$117,000	Alex	C	ac@simon.edu
	C	$123,000	$67,000	$82,000	Iris	D	id@simon.edu
	D	$910,000	$308,000	$335,000	Rex	G	rg@simon.edu

IN PYTHON / R

```
pd.merge(sales, mgt,
left_on="region", right_on="region")
```

FINAL THOUGHTS

- **Practice Makes “Perfect!”**
- **You Need to Practice To Get Better**
- **Focus On Building Intuition**


MELIORA