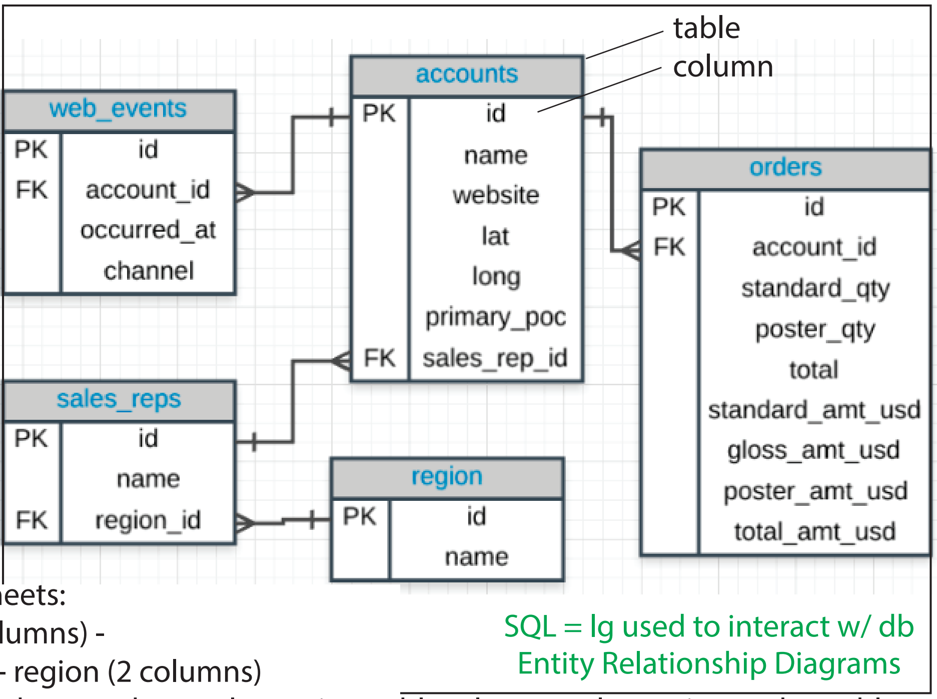
Bertelsmann Tech Scholarship In this section, you will gain knowledge about SQL basics for working with a single table. You will learn the key commands to filter a table in many different ways. Introduction to Data Analysis & Programming

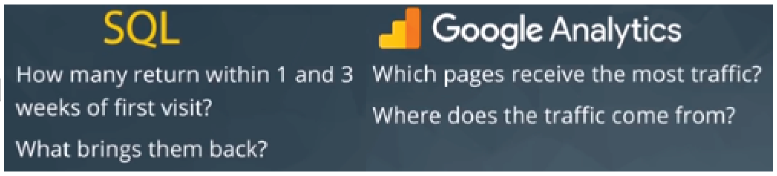
LESSON 26
Basic SQL - 1970s

26.1 Intro - SQL use in data analysis
26.2 Parch & Posey Database - paper co.
- Which product lines is worst ? regular - poster - glossy paper ?
- Which marketing channels need more investment ?
26.3 The Parch & Posey Database
- Store data in spreadsheets
- Visualize the relationship betw spreadsheets using entity relationship diagram, data for analyzing including:
 --> The names of the tables.
 --> The columns in each table.
 --> The way the tables work together.
- Each box = a spreadsheet.
- Parch & Posey db has 5 tables = 5 spreadsheets:
 --> web_event (4 columns) - accounts (7 columns) - orders (9 columns) - sales_reps (3 columns) - region (2 columns)



SQL = lg used to interact w/ db
Entity Relationship Diagrams

- The "crow's foot" connects tables together shows us how columns in 1 table relate to columns in another table. We will be learning the basics of how to work with SQL to interact with a single table. Next lesson, learn more about why these connections are so important for working with SQL and relational databases.
- SQL can query 1 or many tables, understand these relationship will insign data faster.
- SQL = A language that allows us to access data stored in a database.
- ERD = A diagram that shows how data is structured in a database.
- DB = A collection of tables that share connected data stored in a computer.
26.5 Text: Map of SQL Content
- Next 3 lessons, will write SQL to interact w/ a db here, use our br software. Will utilize SQL to analyze data and answer business questions.
- Project: at the end of 3 lessons, download a program to write code on your pc, then analyze and answer business questions using data associated with a music store by querying their db.
- Lesson Outline: 3 lessons aimed at helping to understand how to write SQL queries. The 3 lessons aim at the following components of SQL:
--> SQL Basics - get 1st taste at how SQL works, and learn basics of SQL lg. Learn to write code to interact w/ tables similar to the ones we analyzed in Excel earlier. Specifically, will learn a little about dbs, basic syntax of SQL, write 1st queries!
--> SQL Joins - learn real power of SQL, learn Entity Relationship Diagrams (ERDs), join multiple tables together from a relational db. Join tables made companies to adopt this approach to holding data.
--> SQL Aggregations - learn advanced features of SQL, ability to summarize data from multiple tables in a db.
* End of 3 lessons, will tackle a project. Project aims to assure you mastered 3 topics, and more advanced queries not covered in this course. Meant to introduce advanced material, but don't feel discouraged if you didn't get these - they were beyond the scope of the class, and they are not required to pass the project!
26.6 Video: Why SQL
--> Intro: why SQL queries, SQL and the dbs that utilize SQL so popular. SQL = lg, used all over the place beyond dbs. SQL popular for its interaction w/ dbs. DB = excel files all sitting in one place. Not all databases like that.



SQL
How many return within 1 and 3 weeks of first visit?
What brings them back?

Google Analytics
Which pages receive the most traffic?
Where does the traffic come from?

--> Why Do Data Analysts Use SQL?
- SQL has fns for users read, change, manuplate data. Excel querry max at 1 million rows >< SQL @ billion rows @ a time.

- Using traditional relational dbs, we interact w/ using SQL.
  1. SQL is easy to understand.
  2. Traditional databases allow us to access data directly.
  3. Traditional databases allow us to audit and replicate our data.
  4. SQL is a great tool for analyzing multiple tables at once.
  5. SQL allows you to analyze more complex questions than dashboard tools like Google Analytics.

SQL vs. NoSQL = not only SQL ~ MongoDB. NoSQL write code that interacts w/ data a bit differently than SQL. NoSQL envs particularly popular for web based data, less popular for data lives in spreadsheets.

--> Why Do Businesses Choose SQL?

26.7 Video: How Databases Store Data

A few key points about data stored in SQL databases:

1. Data in dbs is stored in tables = spreadsheets. Each spreadsheet has rows and columns. Each row holds data on a transaction, a person, a company, etc., each column holds data pertaining to a particular aspect of one of the rows you care about like a name, location, a unique id, etc.
2. All data in a column must match data type.
3. Consistent column types = working w/ dbs is fast.

26.8 Text + Quiz: Types of Databases

- Many different types of SQL dbs designed for different purposes. We will use PostgreSQL.
- Most popular dbs: MySQL, Access, Oracle, Microsoft SQL Server, Postgres

You can also write SQL within other programming frameworks like Python, Scala, and HaDoop.

- Each SQL databases may subtle differences in syntax and available fns.
- This article "https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems" compares SQLite, PostgreSQL, and MySQL

26.9 Video: Types of Statements

- SQL has few elements. The key to SQL is understanding statements. Statement tell db what you want to do w/ data. A few statements include:

-> CREATE TABLE is a statement that creates a new table in a database.

-> DROP TABLE is a statement that removes a table in a database.

-> SELECT allows you to read data and display it. This is called a query, we will learn all about this. Most work done by reading, manipulating data.

26.11 Video: SELECT & FROM

SQL command query: SELECT ... FROM ....

- SELECT indicates which column(s) you want to be given the data for. All columns in the table, use "*"
- FROM specifies from which table(s) you want to select the columns. Notice the columns need to exist in this table.
- A few additional statements and operators used along with them to ask more advanced questions of data.

26.14 Formatting Best Practices

- SQL queries are not case-sensitive. But text data stored in SQL tables is case-sensitive.
- Avoid Spaces in Table and Var Names: use underscores and avoid spaces in column names. Postgres refer to these columns/tables with double quotes around them (Ex: FROM "Table Name"). Other env is square brackets instead (Ex: FROM [Table Name]).
- Semicolons > your query may need a semicolon at the end to execute each statement, which also allows you to run multiple queries at once if your environment allows this.

26.15 Video: LIMIT clause > SELECT (to choose columns) and FROM (to choose tables) statements. The LIMIT (to limit rows)

26.16 SELECT occurred_at, account_id, channel FROM web_events LIMIT 15;

- Using Atom to Write SQL Queries and Save Your Notes, save SQL query files in Atom w/ a .sql extension to get highlighting support w/ SQL syntax.

26.18 Video: ORDER BY

- ORDER BY statement = sort our results using the data in any column = similar to sorting a sheet using a column, but SQL query only has temporary effects, for the results of that query. This highlights the meaning and function of a SQL "query."

- ORDER BY comes in a query after SELECT and FROM statements, but before the LIMIT statement. As you learn additional commands, the order of these statements will matter more. Default is to sort in ascending order.

26.19 Quiz: ORDER BY

```
SELECT id, account_id, total_amt_usd
 FROM orders
 ORDER BY total_amt_usd DESC
 LIMIT 5;
```

Operators symbols use w/ WHERE' stat

1. `>` (greater than)
2. `<` (less than)
3. `>=` (greater than or equal to)
4. `<=` (less than or equal to)
5. `=` (equal to)
6. `!=` (not equal to)

26.21 Video: ORDER BY Part II
- Can ORDER BY more than 1 col at a time. In a list of cols in an ORDER BY command, the sorting occurs using leftmost col in your list first, then next col from left, & so on.

26.24 Video: WHERE > statement, display subsets of tables based on conditions that must be met. WHERE comm as filtering the data.

26.27 Video: WHERE with Non-Numeric Data
- Use = and != operators. Use single quotes w/ text data, not double quotes.

```
    //  WHERE name = 'America'
```

26.30 Video: Arithmetic Operators
- Derived Columns = alias = Creating a new col that is a combination of existing cols (or "calculated" or "computed" column), using the AS keyword. This column only temporary, existing just for the duration of your query.

```
    SELECT id, (standard_amt_usd/total_amt_usd)*100 AS std_percent, total_amt_usd
```

26.33 Text: Introduction to Logical Operators
LIKE = perform operations similar to WHERE & =, but for cases when not know exactly what you are looking for.
IN = allows you to perform operations similar to using WHERE and =, but for more than one condition.
NOT, used w/ IN & LIKE to select all rows NOT LIKE or NOT IN a certain condition.
AND & BETWEEN, combine operations where all combined conditions must be true.
OR combine operation where at least one combined conditions must be true.

```
SELECT name
 FROM accounts
 WHERE name LIKE '%one%';
```

26.34 Video: LIKE > LIKE operator working with text, within a WHERE clause, used with %.
The % = any number of characters leading up to a particular set of characters or following a certain set of characters. Use single quotes for the text. Searching for 'T' is not the same as searching for 't'.
Think about use these types of applications to identify phone numbers from a certain region, or an individual where you can't quite remember the full name.

26.37 Video: IN operator > working w/ both numeric and text columns.
- Allows to use an =, but for more than one item of that particular column.

```
SELECT *
FROM web_events
WHERE channel IN ('organic', 'adwords');
```

We can check one, two or many column values for which we want to pull data, but all within the same query.
- OR operator allow us to perform these tasks, but the IN operator is a cleaner way to write these queries.

26.40 Video: NOT operator = extremely useful operator for working w/ IN and LIKE. By specifying NOT LIKE or NOT IN, we can grab all of the rows that do not meet a particular criteria.

26.43 Video: AND and BETWEEN
- AND is used within WHERE to consider more than one logical clause at a time. Each time link a new statement with an AND, need to specify the column. May link many statements at the same time. This operator works with all of the operations we have seen so far including arithmetic operators (+, *, -, /). LIKE, IN, and NOT logic can linked together using AND.
- BETWEEN Operator > make a cleaner statement than using AND. Particularly when using the same column for different parts of AND statement.

```
 WHERE column >= 6 AND column <= 10   ---------------   WHERE column BETWEEN 6 AND 10
```

```
SELECT *
FROM orders
WHERE standard_qty > 1000 AND
poster_qty = 0 AND gloss_qty = 0;
```

```
SELECT *
FROM web_events
WHERE channel IN ('organic', 'adwords') AND occurred_at
BETWEEN '2016-01-01' AND '2017-01-01'
ORDER BY occurred_at DESC;
```

```
SELECT occurred_at, gloss_qty
FROM orders
WHERE gloss_qty BETWEEN 24 AND 29;
```

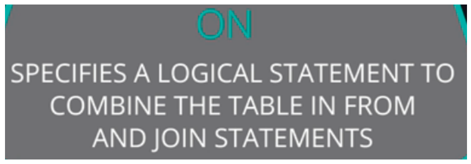
```
SELECT name
FROM accounts
WHERE name NOT LIKE 'C%' AND name LIKE '%s';
```

26.46 OR >> Similar to AND, OR operator can combine multiple statements. Each time you link a new statement with an OR, you need to specify column you looking at. You may link as many statements as you would like to consider at the same time. This operator works with all of the operations we have seen so far including arithmetic operators (+, *, -, /), LIKE, IN, NOT, AND, and BETWEEN logic can all be linked together using the OR operator. We may need parentheses to assure that logic we want to perform is being executed correctly.

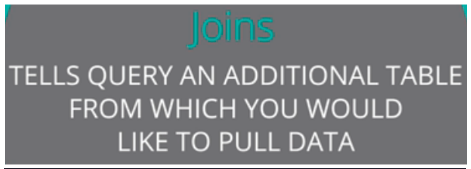
```
SELECT id, gloss_qty, poster_qty
FROM orders
WHERE gloss_qty > 4000 or poster_qty > 4000;
```

```
SELECT id, gloss_qty, poster_qty, standard_qty
FROM orders
WHERE (gloss_qty > 1000 or poster_qty > 1000)
AND standard_qty = 0;
```

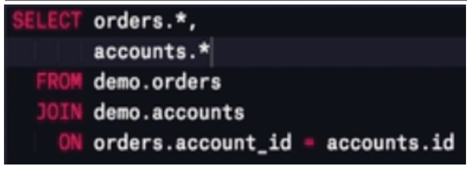
```
SELECT *
FROM accounts
WHERE (name like 'C%' or name like 'W%')
        AND ((primary_poc like '%ana%' OR primary_poc like '%Ana%')
        AND primary_poc NOT LIKE '%eana%');
```


ON

SPECIFIES A LOGICAL STATEMENT TO COMBINE THE TABLE IN FROM AND JOIN STATEMENTS


Joins

TELLS QUERY AN ADDITIONAL TABLE FROM WHICH YOU WOULD LIKE TO PULL DATA


```
SELECT orders.*,
       accounts.*
  FROM demo.orders
  JOIN demo.accounts
    ON orders.account_id = accounts.id
```

LESSON 27 SQL Joins

COMBINE DATA FROM MULTIPLE TABLES TOGETHER

27.01 Video: Motivation > Relational DB = table within it relate to one another. Contain common identifiers, allow info from table combined. Use SQL to link tables, work w/ data by using Joints.

27.2 Video: Why Would We Want to Split Data Into Separate Tables?

DB Normalization > When creating a db, must think about how data will be stored, known as normalization, to set up a new db, important to thorough understanding of db normalization. There are 3 ideas aimed at db normalization: 1. Are the tables storing logical groupings of the data?
2. Can I make changes in a single location, rather than in many tables for the same information?
3. Can I access and manipulate data quickly and efficiently? *Reading: SQL by Design: Why You Need DB Normalization. https://www.itprotoday.com/sql-server/sql-design-why-you-need-database-normalization*

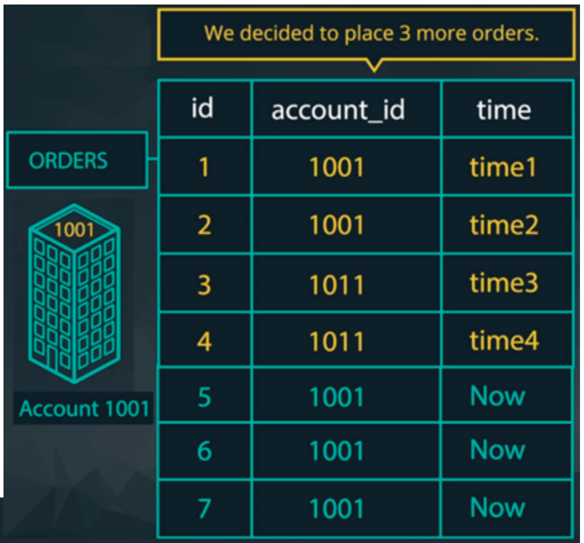
However, most analysts work w/ a db, already set up w/ necessary properties in place. As analysts of data, you don't think too much about data normalization. Just need to be able to pull data from db, so you can start making insights. This is our focus. Why in RD orders table doesn't have name of customer, only account_id ? Joint to connect this data to names.

2 reasons: 1. Orders & accounts store different types of obj, separate to easy organize.
2. This multi-table structurre allows queries execute quickly.

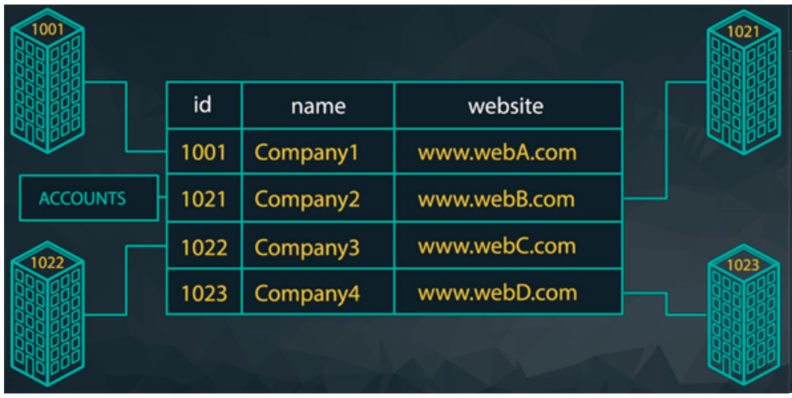
Now, how to connect Orders & accounts data ? Using Join

27.3 JOINs Statements allow to pull data from more than 1 table at a time. This is simple & powerful all at the same time. Addition of JOIN statement = adding ON statement. Use ON clause to specify a JOIN condition which is a logical statement to combine table in FROM and JOIN statements.

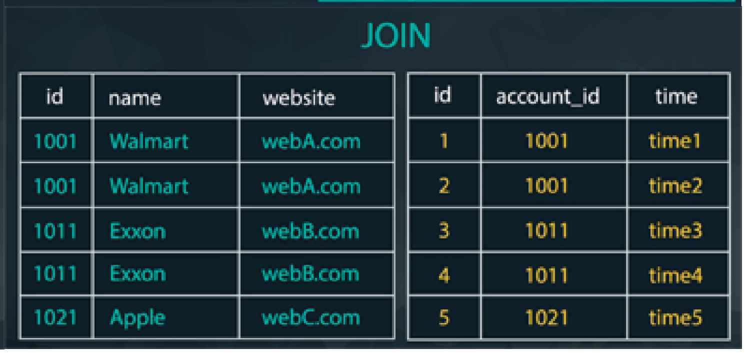
INNER Join



id	account_id	time
1	1001	time1
2	1001	time2
3	1011	time3
4	1011	time4
5	1001	Now
6	1001	Now
7	1001	Now



id	name	website
1001	Company1	www.webA.com
1021	Company2	www.webB.com
1022	Company3	www.webC.com
1023	Company4	www.webD.com


JOIN

id	name	website	id	account_id	time
1001	Walmart	webA.com	1	1001	time1
1001	Walmart	webA.com	2	1001	time2
1011	Exxon	webB.com	3	1011	time3
1011	Exxon	webB.com	4	1011	time4
1021	Apple	webC.com	5	1021	time5