SUMMARY OF UDACITY COURSE

SQL FOR DATA ANALYSIS

RATINGS: /5

LESSON 1 – BASIC SQL

UNIT 1: SQL Introduction

- SQL has many uses in Software Development

UNIT 2: The Parch & Posey Database

- Using data from Parch & Posey for the course, it is simulated data

UNIT 3: The Parch & Posey Database

- Entity Relationship Diagram (ERD) is a common way to view data in a database, snap shots
- Tables == Spreadsheets
- Link to DB: https://video.udacity-data.com/topher/2020/May/5eb5533b_parch-and-posey/parch-and-posey.sql

UNIT 4: Quiz – ERD Fundamentals

- Snap shots

UNIT 5: Map of SQL Content

- Learn to SQL to interact with a database
- SQL is an extremely in demand skill

UNIT 6: Why SQL

- SQL is popular because of its interaction with DBs
- A lot of the world's data sits in Databases, snap shots
- NoSQL: Not Only SQL
- NoSQL is particularly popular for web based data
- Snap shot

UNIT 7: How Databases Store Data

- Data in DB can be thought of just like Excel
- All data type in the same columns must match in terms of data type

UNIT 8: Types of Databases

- Many different types of SQL databases designed for different purposes

- Be using “Postgress” within this course, snap shot
- Each of these SQL DBs have subtle differences in syntax and available functions. Always checkout their documentations
- Checkout: https://www.postgresql.org/

UNIT 9: Types of Statements

- Key to SQL is “statements”
- “CREATE TABLE”: statement that creates a new table in a DB
- “DROP TABLE”: statement that removes a table in a DB
- “SELECT”: statement that allows you to read data and display it, it is also called “query”
- “SELECT” is the common statement used by analyst

UNIT 10: Quiz – Statements

- Snap shots

UNIT 11: SELECT & FROM

- SELECT statement should as fill out a form to get a set of results
- “SELECT <column name(s)> FROM <table name>”

UNIT 12: Your First Queries in SQL Workspace

- Snap shots
- “Mistakes are opportunities to learn”

UNIT 13: Solution – Your First Queries

UNIT 14: Formatting Best Practices

- SQL queries are not case-sensitive
- It is common and best practice to capitalize all SQL commands and keep everything else in your query lower case
- Avoid spaces in Table and Variable Names
- It is considered best practice to put a semicolon at the end of each statement

UNIT 15: Limit

- “LIMIT”: statement used to limit number of rows of a table to display
- “SELECT <column name(s)> FROM <table name> LIMIT <no_of_rows>”
- “LIMIT” is always the very last part of a query

UNIT 16: Quiz – Limit

- Snap shot

UNIT 17: Solution – Limit

UNIT 18: ORDER BY

- “ORDER BY”: allows us to sort our results using the data in any columns, it must come after SELECT, FROM and before LIMIT
- “ORDER BY <column_name> DESC (to change order to descending)”
- Sorting with “ORDER BY” is temporary

UNIT 19: Quiz – Order By

- Snap shots

UNIT 20: Solution – Order By

- Snap shots

UNIT 21: ORDER BY Part 2

- We can use ORDER BY on multiple columns, the sorting occurs using the leftmost column in your list first, then the next column from the left and so on

UNIT 22: Quiz – Order By Part 2

- Snap shots

UNIT 23: Solution – Order By Part 2

- Snap shots

UNIT 24: Where

- WHERE: allows us to filter a set of results based on specific criteria
- WHERE command can be seen as “filtering the data”
- Snap shot

UNIT 25: Quiz – Where

UNIT 26: Solution – Where

- Snap shots

UNIT 27: WHERE with Non-Numeric Data

- If you are using an operator with values that non-numeric you need to put the value in single quote

UNIT 28: Quiz – WHERE with Non-Numeric Data

UNIT 29: Solution – WHERE with Non-Numeric Data

- Snap shot

UNIT 30: Arithmetic Operators

- “Derived/calculated/computed Columns”: creating a new column that is a combination of existing columns using “AS” to give it an alias name
- Remember your mathematical order of operations

UNIT 31: Quiz – Arithmetic Operator

UNIT 32: Solution – Arithmetic Operator

- Snap shot

UNIT 33: Introduction to Logical Operators

- Snap shots

UNIT 34: Like

- “LIKE” operator is extremely useful when working with text, you will use LIKE within a WHERE clause
- “%” wild cards used to specify a number of characters (similar to REGEX)
- Lower and Uppercase letters are not the same within strings
- Snap shots

UNIT 35: Quiz – Like

UNIT 36: Solution – Like

- Snap shots

UNIT 37: In

- “IN” operator is useful for working with both numeric and text columns
- Snap shots

UNIT 38: Quiz – In

UNIT 39: Solution – In

- Snap shots

UNIT 40: Not

- “NOT” operator provides the inverse results for IN, LIKE and similar operators
- It is an extremely useful operator for working with IN and LIKE

UNIT 41: Quiz – Not

UNIT 42: Solution – Not

- Snap shots

UNIT 43: AND and BETWEEN

- "AND" operator used within a WHERE statement to consider more than one logical clause at a time
- To make a cleaner statement we can use BETWEEN in some cases instead of AND, snap shots

UNIT 44: Quiz – AND and BETWEEN

UNIT 45: Solution – AND and BETWEEN

UNIT 46: Or

- "OR" operator can combine multiple statements, similar to the "AND" operator
- "OR" can be combined with other operators by using parentheses

UNIT 47: Quiz – Or

UNIT 48: Solution – Or

- Snap shots

UNIT 49: Recap and Looking Ahead

- Snap shots

LESSON 2 – SQL JOINS

UNIT 1: Motivation

- Working with different tables at once is one of the power of SQL
- SQL is one of the most powerful environment for working with data

UNIT 2: Why would we want to split data into separate Tables

- It helps to organize data effectively
- It also allows queries to be faster
- Database Normalization: thinking about how data will be stored, snap shot

UNIT 3: Introduction to JOINs

- The whole purpose of JOIN statements is to allow us to pull data from more than one table at a time
- We use "ON" clause to specify a "JOIN" condition

UNIT 4: Quiz – Your First JOIN

- Snap shots

UNIT 5: Solution – Your First JOIN

- Snap shot

UNIT 6: Entity Relationship Diagrams Reminder

- ERD is a common way to view data in a database, it is also a key element to understanding how to pull data from multiple tables
- "PK": Primary key exists in every table; it is a column that has a unique value for every row

UNIT 7: Primary and Foreign Keys

- It is common that the PK is the first columns in our tables in most databases
- "FK": Foreign key is a column in one table that is primary key in a different table
- Snap shot

UNIT 8: Quiz – Primary Foreign Key Relationship

- Snap shot

UNIT 9: JOIN Revisited

- Join tables by linking the PK and the FK
- Snap shot

UNIT 10: Alias

- When we join tables together, it is nice to give each table an alias
- Snap shot

UNIT 11: Quiz - JOIN Questions Part 1

UNIT 12: Solutions – JOIN Questions Part 1

- Snap shots, using alias name is important for columns with similar names

UNIT 13: Motivation for Other JOINS

- Traditional databases do not allow for many to many relationship, cause of this there is need for other types of JOINs

UNIT 14: LEFT and RIGHT JOINs

- INNER JOIN: returns only rows that appear in both tables
- Use a Venn diagram to visualize diff types of JOINs
- Snap shots
- Types of JOINS: Left join, Right join and Full Outer join, snap shots
- FROM table_name > is referred to as the LEFT table
- LEFT JOIN table_name > is referred to as the RIGHT table

- LEFT JOIN: returns inner join results and rows in the left table that don't appear in both tables
- LEFT and RIGHT JOINs are interchangeable
- Consistency is important when writing lot of queries

UNIT 15: Other JOIN Notes

- Snap shots
- Use case for FULL OUTER JOIN (OUTER JOIN) is very rare.

UNIT 16: Quiz – LEFT and RIGHT JOIN

- Snap shots

UNIT 17: Solutions – LEFT and RIGHT JOIN

- Snap shots

UNIT 18: JOINs and Filtering

- JOINs are a means to do other types of analysis
- Using ON …. AND will filter data before joining
- Snap shot

UNIT 19: Quiz – Last Check

- Always use "alias" for complex queries

UNIT 20: Solutions – Last Check

- Snap shots

UNIT 21: Recap & Looking Ahead

- Snap shots

LESSON 3 – SQL AGGREGATIONS

UNIT 1: Introduction to Aggregation

- Databases are great at aggregating data
- COUNT, SUM, MIN, MAX, AVERAGE, they operate down columns and not across rows

UNIT 2: Introduction to NULLs

- NULLs are a datatype that specifies where no data exists in SQL, they are often ignored in our aggregation functions

UNIT 3: NULLs and Aggregation

- Need to write “IS NULL” instead of “= NULL” for the WHERE clause
- Snap shots

UNIT 4: First Aggregation – COUNT

- “SELECT COUNT(* or column_name)”

UNIT 5: COUNT & NULLs

- COUNT does not consider rows that have NULL values
- COUNT can be used on any column in a table

UNIT 6: SUM

- It is similar to COUNT
- SUM can only be used on numeric column; it will also ignore NULL values (treat them as zeroes)

UNIT 7: Quiz – SUM

UNIT 8: Solution – SUM

- Snap shots

UNIT 9: MIN & MAX

- MIN & MAX have similar syntax like COUNT, they also ignore NULL values, they can be used on non-numerical columns

UNIT 10: AVG

- AVG returns the mean of the data, it ignores the NULL values in both numerator and denominator
- Finding MEDIAN is difficult Using SQL alone (usually asked in interviews)

UNIT 11: Quiz - MIN, MAX, & AVERAGE

- Snap shot

UNIT 12: Solutions – MIN. MAX, & AVERAGE

- Snap shots

UNIT 13: GROUP BY

- GROUP BY used to aggregate data within subsets of the data
- Snap shots

UNIT 14: Quiz – GROUP BY

UNIT 15: Solution – GROUP BY

- Snap shots

UNIT 16: GROUP BY Part 2

- We can GROUP BY with multiple columns
- Any columns that is not within an aggregation must show up in GROUP BY statement
- Snap shots

UNIT 17: Quiz – GROUP BY Part 2

UNIT 18: Solution – GROUP BY Part 2

- Snap shots

UNIT 19: DISTINCT

- DISTINCT is always used in SELECT statements, and it provides the unique rows for all columns written in the SELECT statement
- You only use DISTINCT once in any particular SELECT statement
- DISTINCT = unique
- Using DISTINCT can slow down queries a bit

UNIT 20: Quiz – DISTINCT

UNIT 21: Solution – DISTINCT

- Snap shots

UNIT 22: HAVING

- HAVING is a “clean way” to filter a query that has been aggregated
- Instead of using WHERE on an aggregated query, you use HAVING

UNIT 23: Quiz – HAVING

- Snap shots

UNIT 24: Solution – HAVING

- Snap shots

UNIT 25: DATE Functions

- Databases order YYYY-MM-DD
- US used the format MM-DD-YY
- GROUPing BY s date column is not usually useful in SQL, as these columns tend to have transaction data down to a second

UNIT 26: DATE Functions 2

- DATE_TRUNC: allows us to truncate our date to a particular part of the date-time column, “day”, “month”, “year”
- DATE_PART: can be useful for pulling a specific portion of a date, but notice pulling “month” or “day” means that you are no longer keeping the years in order
- Postgresql documentation: https://www.postgresql.org/docs/9.1/functions-datetime.html
- Snap shot

UNIT 27: Quiz – DATE Functions

UNIT 28: Solutions – DATE Functions

UNIT 29: CASE Statements

- CASE: always goes in the SELECT clause
- It is SQL version of if/ else statements
- Snap shots

UNIT 30: CASE & Aggregations

- There are some advantages to separation data into separate columns, but this level of separation might be easier to do in another programming language rather than SQL

UNIT 31: Quiz – CASE

UNIT 32: Solutions – CASE

- Snap shots

UNIT 33: Recap

- The more you practice the better, but you also don’t want to get stuck on the same problem for an extended period of time

LESSON 4 – SQL Subqueries & Temporary Tables

UNIT 1: Introduction

- Subqueries, Table expressions, Persistent Derived Tables
- Subqueries and Table expressions are methods for being able to write a query that creates a table, then write a query that interact with this newly created table

UNIT 2: Introduction to Subqueries

- When we need to use existing tables to create a new table that we then want to query again, this is an indication that we need to use some sort of “subquery”

UNIT 3: Quiz – Write your first Subquery

- When using a “Subquery” it needs an “Alias”
- “Inner query” runs first followed by the “outer query”

UNIT 4: Solutions – Write your First Subquery

- Snap shots

UNIT 5: Subquery Formatting

- Formatting SQL will help with understanding your code, snap shot

UNIT 6: More On Subqueries

- You should not include an alias when you write a subquery in a conditional statement, because here subquery is treated as an individual value (or set of values in the case of IN) rather than as a table

UNIT 7: Quiz – More On Subqueries

UNIT 8: Solutions – More On Subqueries

- Snap shots

UNIT 9: Quiz – Subquery Mania

- Very very tough

UNIT 10: Solutions – Subquery Mania

- Snap shots

UNIT 11: WITH

- WITH statement is often called a “Common Table Expression (CTE)”
- They serve the exact same purpose as “subqueries”, they are common in practice, as they tend to be cleaner for a future reader to follow the logic
- CTE need to use “Alias”
- We have to define CTE at the beginning of a query, to be able to use it in the final query

UNIT 12: Quiz – WITH vs Subquery

- Snap shots

UNIT 13: Quiz – WITH

UNIT 14: Solutions – WITH

- Snap shots

UNIT 15: Subquery Conclusion

- The advances features of “Subqueries” and “CTEs” are the most widely used in an analytics role within a company
- We have now covered all of the main SQL topics we are likely to use on a day to day basis

LESSON 5 – SQL Data Cleaning

UNIT 1: Introduction to SQL Data Cleaning

- Clean and re-structure messy data
- Convert columns to different data types
- Tricks for manipulating NULLs

UNIT 2: LEFT & RIGHT

- LEFT: pulls a specified number of characters for each row in a specified columns starting at the beginning (left), “LEFT(phone_number, 3)”
- RIGHT: similar to LEFT but starts at the end (right)
- LENGTH: provides the number of characters for each row of a specified column, “LENGTH (phone_number)”

UNIT 3: Quiz – LEFT & RIGHT

UNIT 4: Solutions – LEFT & RIGHT

- Snap shots

UNIT 5: POSITION, STRPOS, & SUBSTR

- POSITION: takes a character and a column, and provides the index where that character is for each row, “POSITION(‘,’ IN city_state)”
- STRPOS: provides the same result as POSITION, but the syntax for achieving those results is a bit different, “STRPOS(city_state, ‘,’)”
- They ae both case-sensitive

UNIT 6: Quiz – POSITION & STRPOS

UNIT 7: Solutions – POSITION & STRPOS

- Snap shots

UNIT 8: CONCAT

- CONCAT: allows us to combine columns together across rows, “CONCAT(first_name, ‘ ‘ , last_name)”
- Piping: same function as CONCAT, “first_name || ‘ ‘ || last_name”

UNIT 9: Quiz – CONCAT

- Documentation: https://www.postgresql.org/docs/8.1/functions-string.html

UNIT 10: Solutions – CONCAT

- Snap shots

UNIT 11: CAST

- CAST: allows us to change columns from one data type to another, especially turning them to strings and dates, "CAST (data_columns AS DATE)" or "data_column :: DATE"
- Snap shot
- Check out the PostgreSQL literature for more functions

UNIT 12: Quiz – CAST

UNIT 13: Solutions – CAST

- Snap shots

UNIT 14: COALESCE

- COALESCE: returns the first non-null value passed for each row, "COALESCE (primary_poc, "no_poc")"

UNIT 15: Quiz – COALESCE

UNIT 16: Solutions – COALESCE

- Snap shots

UNIT 17: Recap

- "Cleaning data is a necessary evil"

LESSON 6 – SQL WINDOW FUNCTIONS (ADVANCED)

UNIT 1: Introduction to Window Functions

- Allows to compare one row to another without Joining them

UNIT 2: Window Functions 1

- Documentation: https://www.postgresql.org/docs/9.1/tutorial-window.html
- "PARTITION BY", "OVER"

UNIT 3: Quiz – Window Functions 1

UNIT 4: Solutions – Window Functions 1

- Snap shots

UNIT 5: Quiz – Window Functions 2

UNIT 6: Solutions – Window Functions 2

- Snap shot

UNIT 7: ROW_NUMBER & RANK

- "ROW_NUMBER()…OVER()", "RANK()", "DENSE_RANK()"
- RANK: gives duplicates the same rank unlike ROW_NUMBER

UNIT 8: Quiz- ROW_NUMBER & RANK

UNIT 9: Solutions – ROW_NUMBER & RANK

- Snap shot

UNIT 10: Aggregates in Window Functions

- Using aggregates with Window Functions (SUM, AVG, MIN, MAX)

UNIT 11: Quiz – Aggregates in Window Functions

UNIT 12: Solutions – Aggregates in Window Functions

- Snap shot

UNIT 13: Aliases for Multiple Window Functions

- "WINDOW alias_name AS (window function)"

UNIT 14: Quiz – Aliases for Multiple Window Functions

UNIT 15: Solutions – Aliases for Multiple Window Functions

- Snap shots

UNIT 16: Comparing a Row to Previous Row

- "LAG()": returns the value from a previous row to the current row in the table
- "LEAD()": returns the value from the row following the current row in the table
- Snap shots

UNIT 17: Quiz – Comparing a Row to Previous Row

UNIT 18: Solutions – Comparing a Row to Previous Row

- Snap shots

UNIT 19: Introduction to Percentiles

- Look at percentiles

UNIT 20: Percentiles

- "NTILE(no_of_buckets)", 4 means Quantile, 100 means percentile, 5 quintiles
- Snap shots

UNIT 21: Quiz – Percentiles

UNIT 22: Solutions – Percentiles

- Snap shots

UNIT 23: Recap

- Takes real practice to really nail Window functions

LESSON 7 – (ADVANCED) SQL ADVANCED JOINs & Performance Tuning

UNIT 1: Introduction to Advanced SQL

- More advanced JOINs, you will not use this functions day in day out but they are quite important

UNIT 2: FULL OUTER JOIN

- "FULL OUTER JOIN *** ON"
- Snap shot

UNIT 3: Quiz – FULL OUTER JOIN

UNIT 4: Solution – FULL OUTER JOIN

UNIT 5: JOINs with Comparison Operators

- "Inequality JOINs"

UNIT 6: Quiz – JOINs with Comparison Operators

- Snap shot

UNIT 7: Solutions: JOINs with Comparison Operators

UNIT 8: Self JOINs

- This comes up pretty common in job interviews
- Really have to use Alias for self JOINs
- Snap shot

UNIT 9: Quiz – Self JOINs

UNIT 10: Solutions – Self JOINs

- Snap shot

UNIT 11: UNION

- UNION operator is used to combine the result sets of 2 or more SELECT statements. It removes duplicate rows between the various SELECT statements
- Snap shots

UNIT 12: Quiz – UNION

UNIT 13: Solutions – UNION

UNIT 14: Performance Tuning Motivation

- When and how to improve our queries

UNIT 15: Performance Tuning 1

- "Table size", JOINs, Aggregations are high level things that affect query runtime
- Snap shots
- For explorations subsets of data are okay to be used
- Optimize query where it will run first, like using LIMITs in subqueries

UNIT 16: Performance Tuning 2

- Try to aggregate before joining
- But worry about the accuracy of your queries before optimizing for speed

UNIT 17: Performance Tuning 3

- "EXPLAIN": to get a query plan (order of events)

UNIT 18: JOINing Subqueries

- Joining subqueries can help optimizing our queries

UNIT 19: SQL Completion Congratulations

- "go and apply the skills in the real world"