Heather Smallwood

10/30/2023

**SQL Database day 1 lecture notes**

1. What is a database?

--Collection of data

2. What is SQL? (Structured Query Language)

-- Standard language for storing, manipulating and retrieving data in databases.

3. RDBMS (Relational Database Management System)

-- We use a RDBMS to show data from our database.

Ex: (PostgreSQL, MySQL, SQLite, Many Others)

--------- SQL Keywords ----------

SELECT - Selects data from a database

SELECT DISTINCT - return only distinct (different) values

FROM - Specifies which table to select or delete data from

WHERE - Filters a result set to include only records that fulfill a specified condition

LIKE - Searches for a specified pattern in a column

BETWEEN - Selects values within a given range

AND - Only includes rows where both conditions is true

ORDER BY - Sorts the result set in ascending or descending order

ASC - Sorts the result set in ascending order

DESC - Sorts the result set in descending order

AS - Renames a column or table with an alias

GROUP BY - Groups the result set (used with aggregate functions: COUNT, MAX, MIN, SUM, AVG)

LIMIT - Specifies the number of records to return in the result set

--------- SQL Wildcards ----------

% - Represents zero or more characters

\_ - Represents a single character

--------- SQL Aggregation Functions ----------

MIN() - returns the smallest value of the selected column

MAX() - returns the largest value of the selected column

COUNT() - returns the number of rows that matches a specified criterion

AVG() - returns the average value of a numeric column

SUM() - returns the total sum of a numeric column

-- Verify connection to DB using the Actor Table

-- SELECT first\_name and last\_name

-- FROM actor TABLE

-- Query first\_name which equals 'Nick'

-- Using the WHERE Clause

-- Query first\_name which equals 'Nick'

-- Using LIKE clause

-- NOTE: ( = ) is looking for a literal string

-- while the LIKE Clause allows use for wildcards

-- Using LIKE Clause and WILD CARD ( % )

-- Get all J names

-- Single Character Wild Card ( \_ )

-- Query for all first\_names that start with 'K\_\_%' has two letters, anything goes after

----- Comparasion Operators:

-- = Equals to

-- > Greater Than

-- < Less Than

-- >= Greater Than or Equal to

-- <= Less Than or Equal to

-- <> Not equals

-- Using Comparasion Operators with the Payment Table

-- Query WHERE amount GREATER THAN $10.

-- Query amounts BETWEEN $10-$15

-- NOTE: When using BETWEEN both values are inclusive

-- Order the payments by amount

-- Using the ORDER BY

-- ASC for ascending order (Default)

-- DESC for decending order

-- Query all payments NOT EQUAL to 2.99

----- Aggregate Functions

-- MIN()

-- MAX()

-- SUM()

-- AVG()

-- COUNT()

-- Query the SUM of amounts GREATER THAN OR EQUAL TO 5.99

-- Query the AVG of amounts GREATER THAN OR EQUAL TO 5.99

-- Query the COUNT of amounts GREATER THAN OR EQUAL TO 5.99

-- Query to display the count of DISTINCT amounts paid

-- Query to display the MIN paid using alias

-- Query to display the MAX paid using alias

-- GROUP BY (Used with aggregate functions)

-- Query the customer that paid the most