**DATABASE**

Retrieve, insert, update and delete data in databases.

**WHAT WE’LL COVER**

Tables – Relationships – Joins – Subqueries – Regular Expressions

**What is DATABASE?**

Is a collection of data stored in a format that can easily be accessed

**DATABASE MANAGEMENT SYSTEM (DBMS)**

**Relational – NoSQL**

NoSQL systems don’t understand SQL

**RELATIONAL DATABASES**

Customers – Orders – Products = each data stores data about specific object and so on.

**STRUCTURED QUERY LANGUAGE**

**SELECT**\*

**FROM** products

**WHERE** category =’food’

**ORDER BY** price

**RDBMS**

**MySQL**

**SQL Server**

**Oracle**

**SEQUEL – Structured English Query Language**

**SQL – Structured Query Language**

**Installing MySQL on Mac**

**Creating the Databases**

**SELECT\***

last\_name,

first\_name,

points,

(points + 10) \*100 AS discount\_factor

**FROM customers**

2 Categories of DBMS:

• Relational

Stores data that are link to each other using relationship. Structured Query Language (SQL) is the language that is used

Relational Database Management Systems (RDBMS)

MySQL SQL Server Oracle

• NoSQL No tables or relationship. Don’t understand SQL.

SEQUEL Structured English Query Language was originally developed by IBM in 70’s and back. But they change to SQL Structured Query Language.

**Installing MySQL on computer.**

https://dev.mysql.com/downloads/ Overview of workbench interface On top left: Tool Bar – creating new tab for writing SQL code.

Opening a file, creating database, table and so on.

Left side: Navigator panel with two tabs;

Administration- starting or stopping server, importing and exporting and so on.

And Schemas- shows the databases in the current server.

In the middle: query editor window

Right side: context help and snippets

Top right side: showing or hiding these panel button.

**Creating Databases**

(download the zip file attached below the video)

USE sql\_store;

SELECT \* FROM sql\_store.customers;

--we can see all the data in this table this is comment

**SELECT \* FROM sql\_store.orders;**

--order table

**SELECT Statement**

**USE sql\_store;**

**SELECT \***

**FROM customers**

select all the customers given in the table

--using two clause

**WHERE customer\_id = 1**

--this is the where clause

--only get the customer id 1

ORDER BY first\_name

–specify the columns that were going to sort

SELECT CLAUSE

SELECT last\_name , first\_name

FROM customers

select only the lastname and firstname

(another example)

**SELECT state**

**FROM customer**

— Removing duplicates

**SELECT DISTINCT** state

**FROM** customers

WHERE Clause

We use the WHERE clause to filter data.

Comparison operators:

• Greater than: >

• Greater than or equal to: >=

• Less than: <

• Less than or equal to: <=

• Equal: =

• Not equal: <>

• Not equal: !=

**SELEC**T \*

**FROM** **customers**

**WHERE state = ‘VA’**

**Logical Operators**

— AND (both conditions must be True)

**SELECT \***

**FROM customers**

WHERE birthdate > ‘1990-01-01’ AND points > 1000

—- OR (at least one condition must be True)

SELECT \*

FROM customers

WHERE birthdate > ‘1990-01-01’ OR points > 1000

— NOT (to negate a condition)

SELECT \*

FROM customers

WHERE NOT (birthdate > ‘1990-01-01’)

Exercise

--From the order\_items table, get items

-- for order #6

-- where the total price is greater than 30

Solution

SELECT \*

FROM order\_items

WHERE order\_id = 6 AND unit\_price \* quantity>30

IN Operator

—- Returns customers in any of these states: VA, NY, CA

--in a shorter way

SELECT \*

FROM customers

WHERE state IN (‘VA’, ‘NY’, ‘CA’)

SELECT \*

FROM customers

WHERE state NOT IN (‘VA’, ‘NY’, ‘CA’)

Exercise

--Return products with

-- quantity in stock equal to 49, 38, 72

Solution

SELECT \*

FROM products

WHERE quantity in stock IN (49, 38, 72)