MySQL Notes

What is it & What can it do:

* Structured query language
* A standard language that lets you access & manipulate databases
* An ANSI Standard (there are other versions of the SQL language)

-To be compliant they must support the major commands

USING SQL in your website:

\*What you need:

* RDBMS: data is stored in DB objects called tables
* Table:
* Column/Field:
* Row/Record:

**SYNTAX:**

* SQL keywords: not case sensitive & ‘ ; ’ at end of each statement
* SQL Statements: how most actions are performed on the DB

\*Important SQL Commands:

**SELECT** statement:

* Data returned is stored in a result table (known as result set)
* Syntax: SELECT column-name FROM table-name

SELECT \* FROM 🡪 selects all columns in that table

**SELECT DISTINCT:**

* Returns only distinct (different) values (not duplicate values)
* Syntax: SELECT DISTINCT column(s) FROM table

SELECT COUNT( DISTINCT columns(s) ) FROM table 🡪 lists # of different values

**WHERE:**

* Filters records; extracts only records that fulfill a specific condition
* Syntax: SELECT column FROM table WHERE condition
* Text Values: in single quotes; Numeric Values: no quotes
* OPERATORS: traditional operators &
* BETWEEN: between an inclusive range
* LIKE: search for a pattern
* IN: to specify multiple possible values for a column

\*\*WHERE can be combined with AND, OR, & NOT operators

* AND & OR: used to filter records based on more than one condition ex?
* NOT: displays record if condition(s) is not true ex?

\*\*Combining AND, OR, & NOT: use ‘( )’ to form complex expressions

**ORDER BY:**

* Sorts the result set in ASC (default) or DESC order; can order by multiple columns
* Syntax: SELECT column FROM table ORDER BY column

INSERT INTO: can write 2 ways 🡪 More on W3

**NULL VALUES:**

* For an optional field you can insert/update record w/out adding a value
* A field with a null value is a field with no value
* \*\*Cannot test for null values w/ comparison operators; must use ‘IS NULL’ & ‘IS NOT NULL’ operators
* WHERE column IS NULL 🡪 returns all null values
* WHERE column IS NOT NULL 🡪 returns all values except for null

UPDATE:

DELETE:

SELECT TOP:

COUNT, AVG, SUM:

**LIKE:**

* Used in WHERE clause to search for a specified pattern
* Syntax: SELECT column FROM table WHERE column LIKE pattern
* 2 wildcards used in conjunction w/ LIKE; these can be used in combinations & with AND & OR
* %: represents 0, 1, or multiple characters
* \_ : represents a single character

**WILDCARD** characters:

* Character used to substitute any other character(s) in a string 🡪 Examples on W3
* \*Wildcard chars are used in LIKE which is used in WHERE

**IN:**

* Allows you to specify multiple values in a WHERE statement
* Shorthand for multiple OR conditions
* WHERE column IN(value1, value2..)
* WHERE column IN(SELECT statement)

**BETWEEN:**

* Selects values w/in a given range (can be numbers, texts, or dates)

\*\*\*Inclusive: begin & end values are included

* WHERE column BETWEEN value1 AND value2
* WHERE column NOT BETWEEN value1 AND value2

**ALIASES:**

* Used to give a table, or column in a table a temporary name (Display Name)
* Column Syntax:

SELECT column\_name AS alias\_name

FROM table\_name;

* Table Syntax:

SELECT column\_name(s)

FROM table\_name AS alias\_name;

**JOIN:** Used to combine the rows from two or more tables, based on a related column between them

SELECT column\_name(s)  
FROM table1  
JOIN table2 ON table1.column\_name = table2.column\_name;

     

**INNER JOIN:**

* Returns records that have matching values in both tables
* SELECT FROM 1st table INNER JOIN
* Joining 2 tables:

SELECT Orders.OrderID, Customers.CustomerName  
FROM Orders  
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID;

* Joining 3 tables:

SELECT Orders.OrderID, Customers.CustomerName, Shippers.ShipperName  
FROM ((Orders  
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID)  
INNER JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID);

**LEFT JOIN:**

* Returns all records from the Left (1st table) & only the matched records from the Right(2nd table)

**RIGHT JOIN:**

* Returns all records from the Right (2nd table) & only the matched records from the Left(1st table)

SELECT Customers.CustomerName, Orders.OrderID  
FROM Customers  
LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID  
ORDER BY Customers.CustomerName;

\*\*If there is no match on the Left/Right, the result in Null

**FULL JOIN:**

* Returns all record when there is a match in either the left or the right table

SELECT Customers.CustomerName, Orders.OrderID  
FROM Customers  
FULL OUTER JOIN Orders ON Customers.CustomerID=Orders.CustomerID  
ORDER BY Customers.CustomerName;

**SELF JOIN:**

* The table is joined with itself

SELECT column\_name(s)  
FROM table1 T1, table1 T2  
WHERE condition;

SELECT A.CustomerName AS CustomerName1, B.CustomerName AS CustomerName2, A.City  
FROM Customers A, Customers B  
WHERE A.CustomerID <> B.CustomerID  
AND A.City = B.City

**UNION:**

* Operator used to combine the result-set of 2 or more SELECT statements 🡪 More on W3

SQL Server Queries Videos:

Calculated Columns:

* Mathematically manipulating 2 columns and then giving that column an alias name

Joins:

SQL DATABASE

CREATE DATABASE:

* Statement that creates a new SQL database
* Syntax: CREATE DATABASE databasename;

\*\*Make sure you have admin privilege ??

DROP DATABASE:

* Statement

CREATE TABLE:

* Statement that creates a new table in a database
* Syntax:

\*\*Create a table using another table:

DROP TABLE:

* Statement
* Syntax:

\*\*TRUNCATE TABLE: used to delete data inside a table but not the table itself

* Syntax:

ALTER TABLE:

* Statement used to add, delete, or modify columns in an existing table
* Also, used to add/drop various constraints on existing table

Syntax for ADD, DROP, & MODIFY

CONSTRIANTS:

* Used to specify rules for data in a table & limits the type of data that can go into a table

\*\*Ensures accuracy & reliability of the data

* CREATE TABLE – Syntax:
* ALTER TABLES – Syntax:

\*\*Column level or Table level

**COMMONLY USED CONSTRAINTS:** syntax examples on W3Schools

* NOT NULL
* Enforces a column to not accept Null values
* Field must always contain a value
* UNIQUE:
* Ensures all values in a column are different (primary key does this also)
* Can have many per table
* PRIMARY KEY:
* Uniquely identifies each record in a table; must contain unique values - no Null values
* Only 1 per table; may consist of 1 or 1+ fields
* FOREIGN KEY:
* A field (or collection) in one table that refers to the primary key in another table
* Used to link two tables together
* \*\*Child Table: contains the foreign key
* \*\*Parent (Referenced) Table:
* CHECK: 🡪 More on W3
* Used to limit the value range that can be placed in a column
* DEFAULT:
* Used to provide a default value for a column
* Will be added to all new records IF no other value is specified
* INDEX:
* CREATE INDEX: statement that creates indexes in tables
* Create Index Syntax 🡪 duplicate values allowed
* Create Unique Index Syntax 🡪duplicate value not allowed
* Drop Index Syntax:
* INDEXES: used to retrieve data from the DB very fast; speeds up searches & queries

\*\*Users cannot see them

\*\*\*Only create indexes on columns that will be frequently searched

AUTO INCREMENT:

* Field that allows a unique number to be generated automatically when a new record is added to the table

\*\*Usually the Primary Key

* Syntax: uses ‘AUTO\_INCREMENT’ keyword; default starts w/ 1

SQL DATES:

\*\*Most difficult part is ensuring the format of the date you’re trying to insert matches the format of the date column

* MySQL Date Data Types: copy from W3
* If data contains only date portion 🡪 good
* If time portion is involved 🡪 more complicated

VIEWS:

* A virtual table based on the result set of a SQL Statement
* A View contains rows & columns same as a real table
* The fields in a view are fields from 1 or 1+ in the real DB 🡪 more on W3

SQL Injection, Hosting