**Learn SQL-**

Common data types – Integer, text, date, real(decimal)

**Create basic table –**

CREATE TABLE friends (

id INTEGER,

name TEXT,

birthday DATE

);

**Add to table-**

INSERT INTO table\_name (column1, column2, column3)

VALUES (value1, value2, value3);

INSERT INTO friends (id, name, birthday)

VALUES (1, 'Jane Doe', '1990-05-30');

SELECT \* From friends;

**Update table-**

UPDATE table\_name

SET column1 = value1

WHERE condition;

UPDATE friends

SET name = 'Jane Smith'

WHERE id = 1;

**Add Column –**

ALTER TABLE table\_name

ADD COLUMN column DATA\_TYPE;

ALTER TABLE friends

ADD COLUMN email TEXT;

**Remove from table-**

DELETE FROM friends

WHERE id = 1;

**Queries –**

**Return all -**

SELECT \* From table\_name;

**Return certain columns -**

SELECT column1, column2 FROM table\_name;

**Return columns with a new name –**

SELECT column AS ‘New\_name” FROM table\_name;

**Distinct(returns unique values, filters out duplicates)-**

SELECT DISTINCT column FROM table\_name;

**Where(only data we want)-**

SELECT \* FROM table\_name WHERE column meets you condition;

SELECT \* FROM movies WHERE imdb\_rating > 8;

Could be !=, <, >, <=, >=, or =

**LIKE(operator used with the WHERE clause to search for a specific pattern)-**

SELECT \* FROM table\_name WHERE column LIKE ‘whatever string you want’;

**the \_ is a wild card- To get all movies that start with SE and end with en-**

SELECT \* FROM movies WHERE name LIKE ‘Se\_en’;

**the % is a wild card- To get all movies that start with an A-**

SELECT \* FROM movies WHERE name LIKE ‘A%’;

**To end with a-**

SELECT \* FROM movies WHERE name LIKE ‘%a’;

**To find something with man in the name-**

SELECT \* FROM movies WHERE name LIKE ‘%man%’

**To find something start starts with a certain word-**

SELECT \* FROM movies WHERE name LIKE “The %’;

**Is NULL, IS NOT NULL(finds items with or without filled columns)-**

**Will only return items for colomnA where columnB is populated-(use IS NULL for opposite)**

SELECT columnA FROM table\_name WHERE columnB IS NOT NULL;

SELECT name FROM movies WHERE imdb\_rating IS NOT NULL;

**BETWEEN(find anything in a range) Between 2 letters is not inclusive of 2nd letter, but both numbers are inclusive.-**

SELECT \* FROM table\_name WHERE column BETWEEN valueA AND valueB;

To get from A – I, but not J

SELECT \* FROM movies WHERE name BETWEEN ‘A’ AND ‘J’;

For years 1990 – 2000, including 2000-

SELECT \* FROM movies WHERE year BETWEEN 1990 AND 1999;

**Queries cont.**

**AND(lets you combine multiple condition)-**

SELECT \* FROM table\_name WHERE columnA BETWEEN conditonA AND conditionB

AND column = condition

SELECT \* FROM movies WHERE year BETWEEN 1990 AND 1999

AND genre = ‘romance;

SELECT \* FROM movies WHERE year < 1985

AND genre = 'horror';

**OR(similar to AND) –**

SELECT \* FROM movies WHERE year > 2014

OR genre = 'action';

SELECT \* FROM movies WHERE genre = 'comedy'

OR genre = 'romance';

**Order By(sorts results in order numeric or alphabetically ASC for ascending, DESC for descending)-**

SELECT name, year FROM movies ORDER BY name;

SELECT name, year, imdb\_rating FROM movies ORDER BY imdb\_rating DESC;

**Limit(limits your results a certain number)-**

SELECT \* FROM movies LIMIT 10;

SELECT \* FROM movies ORDER BY imdb\_rating DESC LIMIT 3;

**Case(SQL’s way to handle if-then logic)-**

SELECT name,

CASE

WHEN imdb\_rating > 8 THEN 'Fantastic'

WHEN imdb\_rating > 6 THEN 'Poorly Received'

ELSE 'Avoid at All Costs'

END

FROM movies;

SELECT name,

CASE

WHEN genre = 'romance' OR genre = 'comedy'

THEN 'Chill'

ELSE 'Intense'

END AS 'Mood'

FROM movies;