**SQL Commands**

1) CREATE TABLE meals (id INTEGER, name TEXT, rating INTEGER); **(Creates a table meal with those properties)**

2) SELECT \* FROM celebs; **(Selects every column from the SQL DATA list “celebs”)**

3) ALTER TABLE celebs ADD COLUMN twitter\_handle TEXT; **(adds new Column twitter\_handle to “celebs”)**

4) UPDATE celebs

SET twitter\_handle = '@taylorswift13'

WHERE id = 4; **(Updates Data list “celebs” and sets the twitter\_handle to be that text);**

5) DELETE FROM celebs WHERE twitter\_handle IS NULL; **(deletes rows with a null value for twitter handle)**

6) SELECT \* FROM movies WHERE

6.5) SELECT DISTINCT genre FROM movies; **(selects different genres with no repeats)**

7) SELECT \* FROM movies WHERE imdb\_rating > 8; **(selects all columns where the rating is above 8)**

8) SELECT \* FROM movies WHERE name LIKE “%man%; **(selects all columns where the name contains the value ‘man’ in the name)**

9) SELECT \* FROM movies

WHERE name BETWEEN 'A' AND 'J'; **(selects all columns between ‘A’ and ‘J’)**

10) SELECT \* FROM movies

WHERE genre = ‘comedy’

OR year < 1980 **(selects all columns either with comedy or years below 1980)**

11) SELECT \* FROM movies

ORDER BY imdb\_rating DESC; **(orders columns from highest rating to lowest imdb rating)**

12) SELECT \* FROM movies

ORDER BY imdb\_rating ASC

LIMIT 3; **(gets 3 columns with the lowest imdb ratings)**

13) SELECT COUNT(\*) FROM fake\_apps; **(counts the amount of columns in fake\_apps)**

14) SELECT COUNT(\*) FROM fake\_apps

WHERE price = 0; **(counts the amount of columns where price is 0)**

15) SELECT price, COUNT(\*) FROM fake\_apps

GROUP BY price; **(organizing by price etc 0-0.98, 0.99-1.99, etc and gives the count for how many fit in each group)**

16) SELECT SUM(downloads) FROM fake\_apps; **(gets total # of downloads in fake\_apps)**

17) SELECT category, SUM(downloads) FROM fake\_apps

GROUP BY category; **(gets total downloads for each category)**

18) SELECT MAX(downloads) FROM fake\_apps; **(gets MAX downloads of highest fake\_apps)**

19) SELECT name, category, MAX(downloads) FROM fake\_apps

GROUP by category; **(gets Max downloads for each fake\_app and categorizes it by category)**

20) SELECT AVG(downloads) FROM fake\_apps; **(gets avg download for all things in fake\_apps)**

21) SELECT price, ROUND(AVG(downloads), 2) FROM fake\_apps

GROUP BY price; **(gets avg downloads rounded to 2 decimals for fake\_apps, if the 2 isn’t provided for rounded, value will just be rounded to nearest integer)**

22) SELECT albums.name, albums.year, artists.name FROM albums, artists; **(gets album name, album year and artist name and puts it into 1 query result search, taking data from 2 tables)**

23) SELECT

\*

FROM

albums

JOIN artists ON

albums.artist\_id = artists.id; **(select all columns where albums artist\_id = artists.id and JOIN means that it adds on artists onto the query result so it gets showed on display)**

23)