**SELECT name, genre**

**FROM movies;**

Queries names and genres from movies.

**SELECT name AS ‘Titles’**

**FROM movies;**

Queries names and relabels it ‘Titles’.

**SELECT DISTINCT tools**

**FROM inventory;**

Queries strictly unique tools. Does not repeat duplicate tools.

**SELECT \***

**FROM movies**

**WHERE imdb\_rating > 8;**

Queries all movies if (imdb\_rating > 8). Acts as if-statements.

**SELECT \***

**FROM movies**

**WHERE name LIKE ‘Se\_en’;**

Queries all movies if their name is like ‘Se\_en’.

‘\_’ is a wildcard character that matches any other character.

‘Seven’, ‘Se7en’ are LIKE ‘Se\_en’.

**SELECT \***

**FROM movies**

**WHERE name LIKE ‘A%’;**

Queries all movies if their name is like ‘A ...’.

‘%’ is a wildcard character that matches (>= 0) characters.

‘A%’ matches to ‘Aaaaa’, ‘Abced’, ‘A’, all that begin with ‘A’.

‘%a’ matches to ‘sjhdjkasbdka, ‘ZZZZZZa’, ‘a’, all that end with ‘a’.

**SELECT name**

**FROM movies**

**WHERE imdb\_rating IS NOT NULL;**

Queries names if their imdb\_rating rating is not null. Self-explanatory.

**SELECT \***

**FROM movies**

**WHERE year BETWEEN 1990 AND 1999;**

Queries all movies if (1990 <= year <= 1999).

**SELECT \***

**FROM movies**

**WHERE name BETWEEN ‘A’ AND ‘J’;**

Queries all movies if their first letter in name between ‘A’ and (not including) ‘J’ (only includes ‘J’ itself).

**SELECT \***

**FROM movies**

**WHERE year BETWEEN 1990 AND 1999**

**AND/OR genre = ‘romance’;**

Queries all movies if (1990 <= year <= 1999) and/or (genre = ‘romance’). Joins two where-statements.

**SELECT \***

**FROM movies**

**ORDER BY name DESC/ASC;**

Queries all movies ordered by names (ascending (low -> high, A-Z) or descending (high -> low, Z-A)). Goes after WHERE statements.

**SELECT \***

**FROM movies**

**LIMIT 10;**

Queries all movies but limits results at 10 movies. Sets the maximum number of rows the result set has. Always goes at the end of a query.

**SELECT name,**

**CASE**

**WHEN imdb\_rating > 8 THEN ‘Fantastic’**

**WHEN imdb\_rating > 6 THEN ‘Mediocre’**

**ELSE ‘Avoid at all costs’**

**END AS ‘Reviews’**

**FROM movies;**

Queries names, accompanied by another column, whose values are determined by the WHEN-statements (like multiple if-else statements). Names the new column as ‘Reviews’.

