**Selecting columns**

SELECT \*

FROM people;

SELECT age, employee

FROM people;

SELECT DISTINCT language

FROM people;

SELECT COUNT(\*)

FROM people;

To exclude non-missing values:

SELECT COUNT(birthdate)

FROM people;

To count unique values:

SELECT COUNT(DISTINCT birthdate)

FROM people;

**Filtering rows**

SELECT title

FROM films

WHERE title <> 'Metropolis';

Use ISO date format:

SELECT name, birthdate

FROM people

WHERE birthdate = '1974-11-11';

SELECT name, birthdate

FROM people

WHERE birthdate = '1974-11-11';

SELECT \*

FROM films

WHERE language = 'Spanish'

AND release\_year > 2000

AND release\_year < 2010;

SELECT title

FROM films

WHERE (release\_year = 1994 OR release\_year = 1995)

AND (certification = 'PG' OR certification = 'R');

SELECT title

FROM films

WHERE release\_year = 1994

OR release\_year = 2000;

SELECT \*

FROM films

WHERE release\_year <> 2015

ORDER BY duration

Or can use between function:

SELECT title

FROM films

WHERE release\_year

BETWEEN 1994 AND 2000;

SELECT title, release\_year

FROM films

WHERE (release\_year >= 1990 AND release\_year < 2000)

AND (language = 'French' OR language = 'Spanish');

To replace multiple ‘OR’ function:

SELECT title, language

FROM films

WHERE language IN ('English', 'Spanish', 'French');

SELECT name

FROM people

WHERE deathdate IS NULL

To find wildcards:

SELECT name

FROM companies

WHERE name LIKE 'DataC\_mp';

Or

SELECT name

FROM companies

WHERE name NOT LIKE 'Data%';

Finding names with ‘r’ as second letter:

SELECT name

FROM people

WHERE name LIKE '\_r%';

**Aggregate**

SELECT MAX(duration)

FROM films

AVG, Min, Sum

**Arithmetic**

SELECT (4 \* 3)

Or

Select (4.0 / 3.0)

**Aliasing**

SELECT title AS title,

(GROSS-BUDGET) AS net\_profit

FROM films;

Or

SELECT AVG(duration) / 60.0 AS avg\_duration\_hours

FROM films;

**Order by**

SELECT title

FROM films

ORDER BY release\_year, name DESC;

**Group by**

FROM before GROUP BY before ORDER BY

In order to GROUP BY, SELECT field must be “calculated”

SELECT release\_year, count(\*)

FROM films

GROUP BY release\_year;

SELECT release\_year, country, MAX(budget)

FROM films

GROUP BY release\_year, country

ORDER BY release\_year, country;

**Having count**

Aggregate cant be used with WHERE i.e.

SELECT release\_year

FROM films

GROUP BY release\_year

~~WHERE COUNT(title) > 10;~~

HAVING COUNT(title) > 200;

SELECT release\_year,

AVG(budget) AS avg\_budget,

AVG(gross) AS avg\_gross

FROM films

WHERE release\_year > 1990~~,~~

~~avg\_budget > 60000000~~

GROUP BY release\_year

HAVING AVG(budget) > 60000000

ORDER BY avg\_gross

LIMIT 5