EXERCISE 1:

select title from movie;

select director from movies;

select title,director from movies;

select title,year from movies;

select \* from movies;

EXERCISE 2:

select \* from movies where id=6;

select \* from movies where year between 2000 and 2010;

select \* from movies where year not between 2000 and 2010;

select \* from movies where year limit 5;

EXERCISE 3:

select \* from movies where title like '%story%'

select \* from movies where director like 'john%'

select \* from movies where director != 'John Lasseter' ;

select \* from movies where title like '%WALL-%'

EXERCISE 4:

SELECT distinct director from movies order by director asc;

SELECT distinct title from movies order by year desc limit 4;

SELECT \* from movies order by title limit 5;

SELECT \* from movies order by title limit 5 offset 5 ;

EXERCISE 5:

SELECT \* FROM north\_american\_cities where country like '%canada';

SELECT \* FROM north\_american\_cities where country like '%United States' order by latitude desc;

SELECT \* FROM north\_american\_cities where longitude < -87.6298 order by longitude ;

SELECT city from North\_american\_cities where country = 'Mexico' order by Population desc limit 2;

SELECT city from North\_american\_cities where country = 'United States' order by Population desc limit 2 offset 2 ;

EXERCISE 6:

select title, domestic\_sales ,international\_sales from movies inner join boxoffice on movies.id = boxoffice.movie\_id;

select title, domestic\_sales, international\_sales

from movies inner join boxoffice on movies.id = boxoffice.movie\_id where international\_sales > domestic\_sales;

select title, rating

from movies inner join boxoffice on movies.id = boxoffice.movie\_id

order by rating desc;

EXERCISE 7:

SELECT DISTINCT building FROM employees;

SELECT \* FROM buildings;

SELECT DISTINCT building\_name, role

FROM buildings

LEFT JOIN employees

ON building\_name = employees.building;

EXERCISE 8:

SELECT name, role FROM employees WHERE building IS NULL;

SELECT DISTINCT building\_name

FROM buildings

LEFT JOIN employees

ON building\_name = employees.building

WHERE employees.building IS NULL;

EXERCISE 9:

SELECT DISTINCT

title,

(domestic\_sales + international\_sales) / 1000000 AS sales

FROM movies

INNER JOIN boxoffice

ON movies.id = boxoffice.movie\_id;

SELECT DISTINCT

title,

(rating \* 10) AS rate\_percent

FROM movies

INNER JOIN boxoffice

ON movies.id = boxoffice.movie\_id;

SELECT title FROM movies WHERE year % 2 = 0;

EXERCISE 10:

SELECT name, MAX(years\_employed) FROM employees;

SELECT role, AVG(years\_employed) as Average\_years\_employed

FROM employees

GROUP BY role;

SELECT building, SUM(years\_employed) FROM employees GROUP BY building;

EXERCISE 11:

SELECT COUNT(\*) FROM employees WHERE role LIKE 'artist';

SELECT role, COUNT(name) FROM employees GROUP BY role;

SELECT role, SUM(years\_employed) FROM employees

GROUP BY role HAVING role LIKE 'engineer';

EXERCISE 12:

SELECT director, COUNT(\*) FROM movies GROUP BY director;

SELECT director, SUM(domestic\_sales) + SUM(international\_sales) AS Total FROM movies

LEFT JOIN boxoffice ON movies.id = boxoffice.movie\_id

GROUP BY director;

EXERCISE 13:

INSERT INTO movies (title, director, year, length\_minutes)

VALUES ('Toy Story 4', 'John Lasseter', 2019, 123);

INSERT INTO boxoffice (movie\_id, rating, domestic\_sales, international\_sales)

VALUES (15, 8.7, 340000000, 270000000);

EXERCISE 14:

UPDATE Movies

SET Director = "John Lasseter"

WHERE Id = 2;

UPDATE Movies

SET Year = "1999" WHERE Id = 3;

UPDATE Movies

SET Title = "Toy Story 3", Director = "Lee Unkrich"

WHERE Id = 11;

EXERCISE 15:

DELETE FROM Movies

WHERE Year < 2005;

DELETE FROM Movies

WHERE Director = "Andrew Stanton";

EXERCISE 16:

CREATE TABLE Database (

Name TEXT,

Version FLOAT,

Download\_Count INTEGER);

EXERCISE 17:

ALTER TABLE Movies

ADD COLUMN Aspect\_ratio FLOAT DEFAULT 3;

ALTER TABLE Movies

ADD COLUMN Language TEXT DEFAULT "English";

EXERCISE 18:

DROP TABLE Movies;

DROP TABLE BoxOffice;