**Excersise 1:**

1. SELECT Title FROM movies;
2. SELECT Director FROM movies;
3. SELECT Title,Director FROM movies;

4 . SELECT Title,Year FROM movies;

5. SELECT \* FROM movies;

**Excersise 2:**

1. SELECT \* FROM movies where id=6;
2. SELECT \* FROM movies where year between 2000 and 2010;
3. SELECT \* FROM movies where year not between 2000 and 2010;
4. SELECT \* FROM movies Where id between 1 and 5;

**Excersise 3:**

1. SELECT \* FROM movies Where Title LIKE "Toy Story%";
2. SELECT \* FROM movies Where Director LIKE "John Lasseter";
3. SELECT \* FROM movies Where Director Not LIKE "John Lasseter";
4. SELECT \* FROM movies Where Title LIKE "WALL-%";

**Excersise 4:**

1. SELECT DISTINCT director FROM movies

ORDER BY director ASC;

1. pSELECT \* FROM movies

ORDER BY year DESC

LIMIT 4;

1. SELECT \* FROM movies

ORDER BY title ASC

LIMIT 5;

1. SELECT \* FROM movies

ORDER BY title ASC

LIMIT 5 OFFSET 5;

**Excersise 5:**

1. SELECT city, population FROM north\_american\_cities WHERE country like "canada";
2. SELECT city, latitude FROM north\_american\_cities

WHERE country = "United States"

ORDER BY latitude DESC;

1. SELECT city, longitude FROM north\_american\_cities

WHERE longitude < -87.629798

ORDER BY longitude ASC;

1. SELECT city, population FROM north\_american\_cities
2. WHERE country LIKE "Mexico"

ORDER BY population DESC

LIMIT 2;

**Excersise 6:**

1. SELECT title, domestic\_sales, international\_sales

FROM movies JOIN boxoffice

ON movies.id = boxoffice.movie\_id;

1. SELECT \*

FROM movies JOIN boxoffice

ON movies.id = boxoffice.movie\_id

where International\_sales > Domestic\_sales;

1. SELECT \*

FROM movies JOIN boxoffice

ON movies.id = boxoffice.movie\_id

order by rating DESC;

**Excersise 7:**

1. SELECT Distinct Building FROM employees;
2. SELECT \* FROM Buildings;
3. SELECT Distinct Building\_name,Role FROM

Buildings left join employees

on Building\_name=Building;

**Excersise 8:**

1. SELECT name, role FROM employees

where Building is null;

1. SELECT \* FROM

Buildings left join Employees

on Building\_name= Building

where Role is null;

**Excersise 9:**

1. SELECT title, (domestic\_sales + international\_sales) / 1000000 AS gross\_sales\_millions

FROM movies

JOIN boxoffice

ON movies.id = boxoffice.movie\_id;

1. SELECT title, ((rating \*10)) as Rating\_percent

FROM movies

JOIN boxoffice

ON movies.id = boxoffice.movie\_id;

1. SELECT title, Year as even\_year

FROM movies where year%2==0;

**Excersise 10:**

1. SELECT max(years\_employed) FROM employees;
2. SELECT Role,AVG(Years\_employed) FROM employees

Group by Role;

1. SELECT Building, sum(Years\_employed) FROM employees

Group by Building;

**Excersise 11:**

1. SELECT role, COUNT(\*) as Number\_of\_artists

FROM employees

WHERE role = "Artist";

1. SELECT role, COUNT(\*) as Number\_of\_employees

FROM employees

group by role;

1. SELECT role, SUM(years\_employed) as Number\_of\_years\_employed

FROM employees

where role="Engineer";

**Excersise 12:**

1. SELECT Director,Count(Title) as number\_of\_movies FROM movies

group by director;

1. SELECT Director,SUM(Domestic\_sales+International\_sales) as total\_sales FROM

movies join Boxoffice

on id=movie\_id

group by director;

**Excersise 13:**

1. INSERT INTO movies VALUES (4, "Toy Story 4", "El Directore", 2015, 90);
2. INSERT INTO Boxoffice VALUES (4, 8.7,34000000,27000000);

**Excersise 14:**

1. UPDATE movies

SET director = "John Lasseter"

WHERE id = 2;

2. UPDATE movies

SET year = 1999

WHERE id = 3;

3. UPDATE movies

SET Title = "Toy Story 3",

Director= "Lee Unkrich"

WHERE id = 11;

**Excersise 15:**

1. Delete from Movies

where year<2005;

1. Delete from Movies

where Director ="Andrew Stanton";

**Excersise 16:**

1. CREATE TABLE Database (

Name TEXT,

Version INTEGER,

Download\_count INTEGER );

**Excersise 17:**

1. ALTER TABLE Movies

ADD A spect\_ratio Float;

1. ALTER TABLE Movies

ADD Language Text

default English;

**Exercise 18:**

1. DROP TABLE IF EXISTS Movies;
2. DROP TABLE IF EXISTS Boxoffice;