Exercise - 1

1.Find the title of each film

SELECT title FROM movies;

2.Find the director of each film

SELECT director FROM movies;

3.Find the title and director of each film

SELECT title,director FROM movies;

4.Find the title and year of each film

SELECT title,year FROM movies;

5.Find all the information about each film

SELECT \* FROM movies;

Exercise - 2

1.Find the movie with a row id of 6

SELECT \* FROM movies where id = 6;

2.Find the movies released in the years between 2000 and 2010

SELECT \* FROM movies where year between "2000"and "2010";

between is inclusive it will include 2000 and 2010

3.Find the movies not released in the years between 2000 and 2010

SELECT \* FROM movies where year not between "2000" and "2010";

4.Find the first 5 Pixar movies and their release year

SELECT title, year FROM movies where id <= 5;

Exercise - 3

1.Find all the Toy Story movies

SELECT \* FROM movies where title like "Toy Story%";

like is case insensitive

2.Find all the movies directed by John Lasseter

SELECT \* FROM movies where director = "John Lasseter";

3.Find all the movies (and director) not directed by John Lasseter

SELECT \* FROM movies where director != "John Lasseter";

4.Find all the WALL-\* movies

SELECT \* FROM movies where title like "WALL-%";

Exercise - 4

1.List all directors of Pixar movies (alphabetically), without duplicates

SELECT distinct director FROM movies order by director asc;

2.List the last four Pixar movies released (ordered from most recent to least)

SELECT \* FROM movies order by year desc limit 4;

3.List the first five Pixar movies sorted alphabetically

SELECT \* FROM movies order by title asc limit 5;

4.List the next five Pixar movies sorted alphabetically

SELECT \* FROM movies order by title asc limit 5 offset 5;

Exercise - 5

1.List all the Canadian cities and their populations

SELECT city, population FROM north\_american\_cities where country ="Canada";

2.Order all the cities in the United States by their latitude from north to south

SELECT city FROM north\_american\_cities where country = "United States" order by latitude desc;

3.List all the cities west of Chicago, ordered from west to east

SELECT city, longitude FROM north\_american\_cities where longitude < -87.629798 order by longitude asc;

4.List the two largest cities in Mexico (by population)

SELECT \* FROM north\_american\_cities where country = "Mexico" order by population desc limit 2;

5.List the third and fourth largest cities (by population) in the United States and their population

SELECT \* FROM north\_american\_cities where country = "United States" order by population desc limit 2; offset 2;

Exercise - 6

1.Find the domestic and international sales for each movie

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

2.Show the sales numbers for each movie that did better internationally rather than domestically

SELECT id, title, domestic\_sales, international\_sales FROM movies inner join boxoffice on movies.id = boxoffice.movie\_id where international\_sales > domestic\_sales;

3.List all the movies by their ratings in descending order.

SELECT id, title, rating FROM movies inner join boxoffice on id = movie\_id order by rating desc;

Exercise - 7

1.Find the list of all buildings that have employees

SELECT distinct building FROM employees;

2.Find the list of all buildings and their capacity

SELECT building\_name,capacity FROM buildings;

3.List all buildings and the distinct employee roles in each building (including empty buildings)

SELECT distinct building\_name, role FROM buildings LEFT JOIN employees on buildings.building\_name = employees.building;

Exercise - 8

1.Find the name and role of all employees who have not been assigned to a building

SELECT name ,role ,building FROM employees where building is null;

2.Find the names of the buildings that hold no employees

SELECT building\_name, role FROM buildings left join employees on building\_name = building where role is null;

Exercise - 9

1.List all movies and their combined sales in millions of dollars

SELECT title, (domestic\_sales + international\_sales) / 1000000 as sales\_millions FROM movies JOIN boxoffice ON movies.id = boxoffice.movie\_id;  
We can use alias on select column and table

2.List all movies and their ratings in percent

SELECT title, rating\*10 as Rating FROM movies JOIN boxoffice ON movies.id = boxoffice.movie\_id;

3.List all movies that were released on even number years

SELECT title, year FROM movies where year % 2 = 0;

Exercise - 10

1.Find the longest time that an employee has been at the studio

SELECT max(years\_employed) as Max\_Time FROM employees;

2.For each role, find the average number of years employed by employees in that role

SELECT role ,avg(years\_employed) as Avg\_years FROM employees group by role;

Column in group by must be in select

3.Find the total number of employee years worked in each building

SELECT building, sum(years\_employed) as Total\_years FROM employees group by building ;

Exercise – 11  
1.Find the number of Artists in the studio (without a HAVING clause)

SELECT count(\*) as total\_artists FROM employees where role="Artist";

2.Find the number of Employees of each role in the studio

SELECT role , count(\*) as total\_emp FROM employees group by role;

3.Find the total number of years employed by all Engineers  
SELECT role, SUM(years\_employed) FROM employees where role = "Engineer" group by role;

SELECT role, SUM(years\_employed) FROM employees group by role having role = "Engineer";

Exercise – 12

1.Find the number of movies each director has directed

SELECT director, count(\*) as Total\_Movies FROM movies inner join boxoffice on id = movie\_id group by director;

2.Find the total domestic and international sales that can be attributed to each director

SELECT director, sum(domestic\_sales+international\_sales) as total\_revenue FROM movies inner join boxoffice on id = movie\_id group by director;

Exercise – 13

1.Add the studio's new production, Toy Story 4 to the list of movies (you can use any director)

insert into movies (id,title, director,year,length\_minutes) values (4,"Toy Story 4","John Lasseter",2020,90);

2.Toy Story 4 has been released to critical acclaim! It had a rating of 8.7, and made 340 million domestically and 270 million internationally. Add the record to the BoxOffice table.  
insert into boxoffice values (4, 8.7, 340000000, 270000000);

Exercise – 14  
1.The director for A Bug's Life is incorrect, it was actually directed by John Lasseter

update movies set director = "John Lasseter" where title = "A Bug's Life" ;

2.The year that Toy Story 2 was released is incorrect, it was actually released in 1999

update movies set year = 1999 where title = "Toy Story 2";

3.Both the title and director for Toy Story 8 is incorrect! The title should be "Toy Story 3" and it was directed by Lee Unkrich1  
update movies set title = "Toy Story 3", director = "Lee Unkrich" where title = "Toy Story 8";

Exercise – 15  
1.This database is getting too big, lets remove all movies that were released before 2005.

delete from movies where year < 2005;

2.Andrew Stanton has also left the studio, so please remove all movies directed by him.

delete from movies where director = "Andrew Stanton";

Exercise -16   
1.Create a new table named Database with the following columns:

– Name A string (text) describing the name of the database

– Version A number (floating point) of the latest version of this database

– Download\_count An integer count of the number of times this database was downloaded  
create table database (name text, version float, download\_count integer);

Exercise- 17

1.Add a column named Aspect\_ratio with a FLOAT data type to store the aspect-ratio each movie was released in. ✓

alter table movies add Aspect\_ratio float;

2.Add another column named Language with a TEXT data type to store the language that the movie was released in. Ensure that the default for this language is English.

alter table movies add column Language text default "English";

Exercise – 18

1.We've sadly reached the end of our lessons, lets clean up by removing the Movies table

drop table movies;

2.And drop the BoxOffice table as well

drop table boxoffice;

Order of execution  
1. From  
2. Where  
3. Group by  
4.Select  
5.Order by