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;

1. Find the title and year of each film

SELECT title, year

FROM movies;

1. Find all the information about each film

SELECT \* FROM movies;

1. Find the movie with a row id of 6

SELECT title from movies where id=6

SELECT title from movies having id=6

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

SELECT title, year FROM movies

**where** year **between** 2000 AND 2010;

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

SELECT title, year FROM movies

WHERE year not BETWEEN 2000 AND 2010;

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

* SELECT title, year
* FROM movies

WHERE year <= 2003;

* SELECT title, year

FROM movies

order by year asc

limit 5

1. Find all the Toy Story movies

SELECT title, director

FROM movies

WHERE title LIKE "Toy Story%";

1. Find all the movies directed by John Lasseter

SELECT title, director

FROM movies

WHERE director = "John Lasseter";

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

SELECT title, director

FROM movies

WHERE director != "John Lasseter";

1. Find all the WALL-\* movies

SELECT \* FROM movies

WHERE title LIKE "WALL-\_";

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

SELECT DISTINCT director

FROM movies

ORDER BY director ASC;

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

SELECT title, year

FROM movies

ORDER BY year DESC

LIMIT 4;

1. List the first five Pixar movies sorted alphabetically

SELECT title, year

FROM movies

ORDER BY title aSC

LIMIT 5;

1. List the next five Pixar movies sorted alphabetically

SELECT title, year

FROM movies

ORDER BY title aSC

LIMIT 5 offset 5;

1. List all the Canadian cities and their populations

SELECT city, population

FROM north\_american\_cities

WHERE country = "Canada";

1. 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;

1. 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;

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

SELECT city, population FROM north\_american\_cities

WHERE country = "Mexico"

ORDER BY population DESC

LIMIT 2;

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

SELECT city, population FROM north\_american\_cities

WHERE country = "United States"

ORDER BY population DESC

LIMIT 2 offset 2;

1. Find the domestic and international sales for each movie ✓

SELECT title, domestic\_sales, international\_sales

FROM movies

JOIN boxoffice

ON movies.id = boxoffice.movie\_id

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

SELECT title,international\_sales,domestic\_sales

FROM movies

JOIN boxoffice

ON movies.id = boxoffice.movie\_id

WHERE international\_sales > domestic\_sales;

SELECT domestic\_sales, international\_sales

FROM boxoffice

where domestic\_sales<international\_sales

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

SELECT title, rating

FROM movies

JOIN boxoffice

ON movies.id = boxoffice.movie\_id

order by rating desc;

1. Find the list of all buildings that have employees

SELECT distinct building

FROM employees;

1. Find the list of all buildings and their capacity

SELECT building\_name,capacity

FROM buildings;

1. 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 building\_name = building;

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

SELECT name, role

FROM employees

WHERE building IS NULL;

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

SELECT DISTINCT building\_name

FROM buildings

LEFT JOIN employees

ON building\_name = building

WHERE name IS NULL;

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

SELECT title, (international\_sales + domestic\_sales) / 1000000 AS combined\_sales\_millions

FROM movies

JOIN boxoffice

ON movies.id = boxoffice.movie\_id;

1. List all movies and their ratings in percent

SELECT title, (rating)\*10 AS rating\_in\_percent

FROM movies

JOIN boxoffice

ON movies.id = boxoffice.movie\_id;

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

SELECT title, year

FROM movies

WHERE year % 2 = 0

order by year asc;

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

SELECT MAX(years\_employed) as longest\_employementtime

FROM employees;

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

SELECT role, AVG(years\_employed) as Avg\_yearsemployed

FROM employees

group by role;

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

SELECT building,

SUM(years\_employed) as Totalyears\_employed

FROM employees

GROUP BY building;

1. Find the number of Artists in the studio (without a HAVING clause) ✓

SELECT role,

COUNT() as Num\_of\_artists

FROM employees

WHERE role = "Artist";

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

SELECT role, COUNT()

FROM employees

GROUP BY role;

1. Find the total number of years employed by all Engineers

SELECT role, SUM(years\_employed)

FROM employees

GROUP BY role

having role = "Engineer";

1. Find the number of movies each director has directed   
   SELECT director, COUNT(id) as Numof\_movies  
   FROM movies  
   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 totalsales\_fromall\_movies

FROM movies

join boxoffice

on movies.id = boxoffice.movie\_id

GROUP BY director;

1. Add the studio's new production, Toy Story 4 to the list of movies (you can use any director)   
    INSERT INTO movies VALUES (19, "Toy Story 4", "Ismavayrus", 2016, 110);
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 (19, 8.7,340,270);
3. The director for A Bug's Life is incorrect, it was actually directed by John Lasseter/  
   Update movies  
   Set director = "John Lasseter"  
   where id = 2
4. The year that Toy Story 2 was released is incorrect, it was actually released in 1999/  
   Update movies  
   Set year = 1999  
   where id = 3;
5. Both the title and director for Toy Story 8 is incorrect! The title should be "Toy Story 3" and it was directed by Lee Unkrich /

Update movies

Set title = "Toy Story 3", director = "Lee Unkrich"

Where id = 11;

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";

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

This table has no constraints. ✓

Create table Database (

Name TEXT,

Version FLOAT,

download\_count INTEGER

);

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 Column Aspect\_ratio FLOAT DEFAULT 2.41;
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 ;
3. We've sadly reached the end of our lessons, lets clean up by removing the Movies table  
   Drop Table Movies;
4. And drop the BoxOffice table as well

Drop Table boxofice;