1. Find the **title** of each film ✓

SELECT title from movies;

1. Find the **director** of each film

Select director from movies;

1. 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;

1. Find the movie with a row **id** of 6

Select \* from movies where id=6;

1. Find the movies released in the **year**s between 2000 and 2010

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

8. Find the movies **not** released in the **year**s between 2000 and 2010

Select \* 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 order by year limit 5;

1. Find all the Toy Story movies

SELECT \* FROM movies WHERE title LIKE 'Toy Story%';

1. Find all the movies directed by John Lasseter

SELECT title,director FROM movies WHERE director like "John lasseter";

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

Select title,director from movies where director not like ‘”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;

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

SELECT title,year from movies order by year desc;

1. List the **first** five Pixar movies sorted alphabetically

SELECT title from movies order by title limit 5;

1. List the **next** five Pixar movies sorted alphabetically

SELECT title from movies where title > 'Finding Nemo' order by title limit 5;

1. List all the Canadian cities and their populations

SELECT city,population from north\_american\_cities where country like "canada";

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

SELECT city,longitude from North\_american\_cities where longitude <-87.6298 order by longitude asc;

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

SELECT city,population from north\_american\_cities where country like "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 like "United states" order by population desc limit 2 offset;

22 Find the domestic and international sales for each movie

SELECT id,title,domestic\_sales,international\_Sales from boxoffice b join movies m on m.id=b.movie\_id;

23 show the sales numbers for each movie that did better internationally rather than domestically

SELECT id,title,international\_sales,domestic\_sales from movies m join boxoffice b on m.id=b.movie\_id group by title having international\_Sales>domestic\_Sales;

24 List all the movies by their ratings in descending order

select id,title,rating from movies join boxoffice on movies.id=boxoffice.movie\_id order by rating desc;

25 Find the list of all buildings that have employees

SELECT distinct building\_name from buildings join employees on buildings.building\_name=employees.building;

26 Find the list of all buildings and their capacity

SELECT \* from buildings;

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

SELECT b.building\_name,e.role FROM buildings b LEFT JOIN employees e ON b.building\_name = e.building GROUP BY b.building\_name, e.role ORDER By b.building\_name;

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

SELECT role,name from employees where building is null;

29 Find the names of the buildings that hold no employees

select building\_name from buildings where building\_name not in (SELECT building\_name from buildings join employees on employees.building=buildings.building\_name);

30 List all movies and their combined sales in **millions** of dollars

SELECT title,round(domestic\_Sales+international\_Sales/1000000,2) as combined\_Sales from movies m join Boxoffice b on m.id=b.movie\_id ;

31 List all movies and their ratings **in percent**

SELECT title,(rating/10)\*100 as rate\_percent from movies m join Boxoffice b on m.id=b.movie\_id ;

32 List all movies that were released on even number years

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

33 Find the longest time that an employee has been at the studio

SELECT max(years\_employed) from employees;

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

SELECT Name,role,avg(years\_employed) from employees group by role;

35 Find the total number of employee years worked in each building

SELECT sum(years\_employed),building from employees group by building;

36 Find the number of Artists in the studio (without a **HAVING** clause)

select count(role) from employees where role like "artist";

37 Find the total number of years employed by all Engineers

select sum(Years\_employed),role from employees where role like "Engineer";

38 Find the number of Employees of each role in the studio

select count(name),role from employees group by role ;

39 Find the number of movies each director has directed

select count(title),director from movies group by director;

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

select sum(domestic\_sales),sum(international\_Sales) from boxoffice b join movies m

on m.id=b.movie\_id group by director;

41 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);

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

Insert into movies values(4,”Toy Story 4”,” John Lasseter”,1996,90);

43 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";

44 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";

45 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" and director="Lee Unkrich" where title=

"Toy Story 8";

46 This database is getting too big, lets remove all movies that were released **before** 2005.

delete from movies where year<2005;

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

delete from movies where director like "Andrew Stanton";

48 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 REAL,Download\_count INTEGER);

49 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;

50 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 default “English”;

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

Drop table movies;

52 And drop the **BoxOffice** table as well

Drop table BoxOffice