Nama : Wahyuni Puji Lestari

1. SQL Lesson 1: SELECT queries 101

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;

2. SQL Lesson 2: Queries with constraints (Pt. 1)

1	Find the movie with a row id of 6	SELECT title FROM movies WHERE id = 6;
2	Find the movies released in the years between 2000 and 2010	SELECT title FROM movies WHERE year BETWEEN 2000 AND 2010;
3	Find the movies **not** released in the years between 2000 and 2010	SELECT title 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 year <= 2003;

3. SQL Lesson 3: Queries with constraints (Pt. 2)

1	Find all the Toy Story movies	SELECT title FROM movies WHERE title LIKE "Toy Story%";
2	Find all the movies directed by John Lasseter	SELECT title FROM movies WHERE director = "John Lasseter";
3	Find all the movies (and director) not directed by John Lasseter	SELECT title, director FROM movies WHERE director != "John Lasseter";
4	Find all the WALL-* movies	SELECT title FROM movies WHERE title LIKE "WALL-%";

4. SQL Lesson 4: Filtering and sorting Query results

1	Find all the Toy Story movies	SELECT DISTINCT director FROM movies ORDER BY director ASC;
2	List the last four Pixar movies released (ordered from most recent to least)	SELECT title FROM movies ORDER BY YEAR DESC LIMIT 4;
3	List the first five Pixar movies sorted alphabetically	SELECT title FROM movies ORDER BY title ASC LIMIT 5;
4	List the next five Pixar movies sorted alphabetically	SELECT title FROM movies ORDER BY title ASC LIMIT 5 OFFSET 5;

5. SQL Review: Simple SELECT Queries

1	List all the Canadian cities and their populations	SELECT city, population , country 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 < -87.669006 ORDER BY longitude ASC;
4	List the two largest cities in Mexico (by population)	SELECT city 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 city, population FROM north_american_cities WHERE country="United States" ORDER BY population DESC LIMIT 2 OFFSET 2;

6. SQL Lesson 6: Multi-table queries with JOINs

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;
2	Show the sales numbers for each movie that did better internationally rather than domestically	SELECT title, domestic_sales, international_sales FROM Movies 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 title FROM Movies JOIN Boxoffice ON Movies.id=Boxoffice.movie_id ORDER BY Rating DESC;

7. SQL Lesson 7: OUTER JOINs

1	Find the list of all buildings that have employees	SELECT DISTINCT building_name FROM employees INNER JOIN Buildings ON employees.building=Buildings.building_name;
2	Find the list of all buildings and their capacity	SELECT DISTINCT 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 building_name = building;

8. SQL Lesson 8: A short note on NULLs

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;
2	Find the names of the buildings that hold no employees	SELECT DISTINCT building_name FROM buildings LEFT JOIN employees

		ON building_name = building WHERE role IS NULL;

9. SQL Lesson 9: Queries with expressions

1	List all movies and their combined sales in **millions** of dollars	SELECT title, (domestic_sales + international_sales 1000000 AS gross_sales_millions FROM movies JOIN boxoffice ON movies.id = boxoffice.movie_id;
2	List all movies and their ratings **in percent**	SELECT title, rating * 10 AS rating_percent 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;

10. SQL Lesson 10: Queries with aggregates (Pt. 1)

1	Find the longest time that an employee has been at the studio	SELECT MAX(years_employed) as Max FROM employees;
2	For each role, find the average number of years employed by employees in that role	SELECT role, AVG(years_employed) as Average FROM employees GROUP BY role;
3	Find the total number of employee years worked in each building	SELECT building, SUM(years_employed) as Total FROM employees GROUP BY building;

11. SQL Lesson 11: Queries with aggregates (Pt. 2)

1	Find the number of Artists in the studio (without a **HAVING** clause)	SELECT role, COUNT(*) as artists FROM employees WHERE role = "Artist";
2	Find the number of Employees of each role in the studio	SELECT role, COUNT(*) FROM employees GROUP BY role;
3	Find the total number of years employed by all Engineers	SELECT role, SUM(years_employed) FROM employees GROUP BY role HAVING role = "Engineer";

12. SQL Lesson 12: Order of execution of a Query

1	Find the number of movies each director has directed	SELECT director, COUNT(id) as Num 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 Cumulative FROM movies INNER JOIN boxoffice ON movies.id = boxoffice.movie_id GROUP BY director;