Step-1 : Creation of database <myDatabase> and table <movies>.

```
create database myDatabase;

create table movies
    -> (
    -> Id int(20) unsigned not null auto_increment,
    -> Title varchar(40) not null,
    -> Director varchar(30) not null,
    -> Year int(10) not null,
    -> Length_minutes int(10) not null,
    -> primary key (Id) );
```

```
mysql> create database myDatabase;
Query OK, 1 row affected (0.01 sec)

mysql> use myDatabase;
Database changed
mysql> create table movies
    -> (
    -> Id int(20) unsigned not null auto_increment,
    -> Title varchar(40) not null,
    -> Director varchar(30) not null,
    -> Year int(10) not null,
    -> Length_minutes int(10) not null,
    -> primary key (Id) );
Query OK, 0 rows affected, 3 warnings (0.04 sec)

mysql> describe movies;
+----------------+--------------+------+-----+---------+----------------+
| Field          | Type         | Null | Key | Default | Extra          |
+----------------+--------------+------+-----+---------+----------------+
| Id             | int unsigned | NO   | PRI | NULL    | auto_increment |
| Title          | varchar(40)  | NO   |     | NULL    |                |
| Director       | varchar(30)  | NO   |     | NULL    |                |
| Year           | int          | NO   |     | NULL    |                |
| Length_minutes | int          | NO   |     | NULL    |                |
+----------------+--------------+------+-----+---------+----------------+
5 rows in set (0.01 sec)
```

```
mysql> insert into movies(Title,Director,Year,Length_minutes) values
    -> ("Toy Story","John Lasseter",1995,81),
    -> ("A Bugs Life","John Lasseter",1998,95);
Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> select * from movies;
+----+-------------+---------------+------+----------------+
| Id | Title       | Director      | Year | Length_minutes |
+----+-------------+---------------+------+----------------+
|  1 | Toy Story   | John Lasseter | 1995 |             81 |
|  2 | A Bugs Life | John Lasseter | 1998 |             95 |
+----+-------------+---------------+------+----------------+
2 rows in set (0.00 sec)
```

```
mysql> insert into movies(Title,Director,Year,Length_minutes) values
    -> ("Toy Story 2","John Lasseter",1999,93),
    -> ("Monsters,Inc.","Pete Docter",2001,92),
    -> ("Finding Nemo","Andrew Stanton",2003,107),
    -> ("The Incredibles","Brad Bird",2004,116),
    -> ("Cars","John Lasseter",2006,117),
    -> ("Ratatouille","Brad Bird",2007,115),
    -> ("WALL-E","Andrew Stanton",2008,104),
    -> ("Up","Pete Docter",2009,101),
    -> ("Toy Story 3","Lee Unkrich",2010,103),
    -> ("Cars 2","John Lasseter",2011,120),
    -> ("Brave","Brenda Chapman",2012,102),
    -> ("Monsters University","Dan Scanlon",2013,110);
Query OK, 12 rows affected (0.00 sec)
Records: 12  Duplicates: 0  Warnings: 0

mysql> select * from movies;
+----+--------------------+---------------+------+----------------+
| Id | Title              | Director      | Year | Length_minutes |
+----+--------------------+---------------+------+----------------+
|  1 | Toy Story          | John Lasseter | 1995 |             81 |
|  2 | A Bugs Life        | John Lasseter | 1998 |             95 |
|  3 | Toy Story 2        | John Lasseter | 1999 |             93 |
|  4 | Monsters,Inc.      | Pete Docter   | 2001 |             92 |
|  5 | Finding Nemo       | Andrew Stanton| 2003 |            107 |
|  6 | The Incredibles    | Brad Bird     | 2004 |            116 |
|  7 | Cars               | John Lasseter | 2006 |            117 |
|  8 | Ratatouille        | Brad Bird     | 2007 |            115 |
|  9 | WALL-E             | Andrew Stanton| 2008 |            104 |
| 10 | Up                 | Pete Docter   | 2009 |            101 |
| 11 | Toy Story 3        | Lee Unkrich   | 2010 |            103 |
| 12 | Cars 2             | John Lasseter | 2011 |            120 |
| 13 | Brave              | Brenda Chapman| 2012 |            102 |
| 14 | Monsters University| Dan Scanlon   | 2013 |            110 |
+----+--------------------+---------------+------+----------------+
14 rows in set (0.00 sec)
```

Step 2 : Queries

SQL Lesson 1: SELECT queries

Q1.Find the title of each film

select title from movies;

```
mysql> select title from movies;
+---------------------+
| title               |
+---------------------+
| Toy Story           |
| A Bugs Life         |
| Toy Story 2         |
| Monsters,Inc.       |
| Finding Nemo        |
| The Incredibles     |
| Cars                |
| Ratatouille         |
| WALL-E              |
| Up                  |
| Toy Story 3         |
| Cars 2              |
| Brave               |
| Monsters University |
+---------------------+
14 rows in set (0.00 sec)
```

Q2.Find the director of each film

select director from movies;

```
mysql> select director from movies;
+----------------+
| director       |
+----------------+
| John Lasseter  |
| John Lasseter  |
| John Lasseter  |
| Pete Docter    |
| Andrew Stanton |
| Brad Bird      |
| John Lasseter  |
| Brad Bird      |
| Andrew Stanton |
| Pete Docter    |
| Lee Unkrich    |
| John Lasseter  |
| Brenda Chapman |
| Dan Scanlon    |
+----------------+
14 rows in set (0.00 sec)
```

Q3.Find the title and director of each film

select title,director from movies;

```
mysql> select title,director from movies;
+---------------------+----------------+
| title               | director       |
+---------------------+----------------+
| Toy Story           | John Lasseter  |
| A Bugs Life         | John Lasseter  |
| Toy Story 2         | John Lasseter  |
| Monsters,Inc.       | Pete Docter    |
| Finding Nemo        | Andrew Stanton |
| The Incredibles     | Brad Bird      |
| Cars                | John Lasseter  |
| Ratatouille         | Brad Bird      |
| WALL-E              | Andrew Stanton |
| Up                  | Pete Docter    |
| Toy Story 3         | Lee Unkrich    |
| Cars 2              | John Lasseter  |
| Brave               | Brenda Chapman |
| Monsters University | Dan Scanlon    |
+---------------------+----------------+
14 rows in set (0.00 sec)
```

Q4.Find the title and year of each film

select title,year from movies;

```
mysql> select title,year from movies;
+---------------------+------+
| title               | year |
+---------------------+------+
| Toy Story           | 1995 |
| A Bugs Life         | 1998 |
| Toy Story 2         | 1999 |
| Monsters,Inc.       | 2001 |
| Finding Nemo        | 2003 |
| The Incredibles     | 2004 |
| Cars                | 2006 |
| Ratatouille         | 2007 |
| WALL-E              | 2008 |
| Up                  | 2009 |
| Toy Story 3         | 2010 |
| Cars 2              | 2011 |
| Brave               | 2012 |
| Monsters University | 2013 |
+---------------------+------+
14 rows in set (0.00 sec)
```

Q5.Find all the information about each film

select * from movies;

```
mysql> select * from movies;
+----+--------------------+----------------+------+----------------+
| Id | Title              | Director       | Year | Length_minutes |
+----+--------------------+----------------+------+----------------+
|  1 | Toy Story          | John Lasseter  | 1995 |             81 |
|  2 | A Bugs Life        | John Lasseter  | 1998 |             95 |
|  3 | Toy Story 2        | John Lasseter  | 1999 |             93 |
|  4 | Monsters,Inc.      | Pete Docter    | 2001 |             92 |
|  5 | Finding Nemo       | Andrew Stanton | 2003 |            107 |
|  6 | The Incredibles    | Brad Bird      | 2004 |            116 |
|  7 | Cars               | John Lasseter  | 2006 |            117 |
|  8 | Ratatouille        | Brad Bird      | 2007 |            115 |
|  9 | WALL-E             | Andrew Stanton | 2008 |            104 |
| 10 | Up                 | Pete Docter    | 2009 |            101 |
| 11 | Toy Story 3        | Lee Unkrich    | 2010 |            103 |
| 12 | Cars 2             | John Lasseter  | 2011 |            120 |
| 13 | Brave              | Brenda Chapman | 2012 |            102 |
| 14 | Monsters University | Dan Scanlon   | 2013 |            110 |
+----+--------------------+----------------+------+----------------+
14 rows in set (0.00 sec)
```

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

Q1.Find the movie with a row id of 6

select id,title from movies where id=6;

```
mysql> select id,title from movies where id=6;
+----+-----------------+
| id | title           |
+----+-----------------+
|  6 | The Incredibles |
+----+-----------------+
1 row in set (0.00 sec)
```

Q2.Find the movies released in the years between 2000 and 2010

select id,title from movies where year between 2000 and 2010;

```
mysql> select id,title from movies where year between 2000 and 2010;
+----+-----------------+
| id | title           |
+----+-----------------+
|  4 | Monsters,Inc.   |
|  5 | Finding Nemo    |
|  6 | The Incredibles |
|  7 | Cars            |
|  8 | Ratatouille     |
|  9 | WALL-E          |
| 10 | Up              |
| 11 | Toy Story 3     |
+----+-----------------+
8 rows in set (0.00 sec)
```

Q3.Find the movies not released in the years between 2000 and 2010

select id,title from movies where year not between 2000 and 2010;

```
mysql> select id,title from movies where year not between 2000 and 2010;
+----+--------------------+
| id | title              |
+----+--------------------+
|  1 | Toy Story          |
|  2 | A Bugs Life        |
|  3 | Toy Story 2        |
| 12 | Cars 2             |
| 13 | Brave              |
| 14 | Monsters University |
+----+--------------------+
6 rows in set (0.00 sec)
```

Q4.Find the first 5 Pixar movies and their release year

select id,title from movies where id<6;

```
mysql> select id,title,year from movies where id<6;
+----+--------------+------+
| id | title        | year |
+----+--------------+------+
|  1 | Toy Story    | 1995 |
|  2 | A Bugs Life  | 1998 |
|  3 | Toy Story 2  | 1999 |
|  4 | Monsters,Inc. | 2001 |
|  5 | Finding Nemo | 2003 |
+----+--------------+------+
5 rows in set (0.00 sec)
```

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

Q1.Find all the Toy Story movies

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

```
mysql> SELECT title FROM movies where title like "%Toy Story%";
+-------------+
| title       |
+-------------+
| Toy Story   |
| Toy Story 2 |
| Toy Story 3 |
+-------------+
3 rows in set (0.00 sec)
```

Q2.Find all the movies directed by John Lasseter

SELECT id,title FROM movies where director="John Lasseter";

```
mysql> SELECT id,title FROM movies where  director="John Lasseter";
+----+-------------+
| id | title       |
+----+-------------+
|  1 | Toy Story   |
|  2 | A Bugs Life |
|  3 | Toy Story 2 |
|  7 | Cars        |
| 12 | Cars 2      |
+----+-------------+
5 rows in set (0.00 sec)
```

Q3.Find all the movies (and director) not directed by John Lasseter

SELECT id,title,director FROM movies where director!="John Lasseter";

```
mysql> SELECT id,title,director FROM movies where  director!="John Lasseter";
+----+--------------------+----------------+
| id | title              | director       |
+----+--------------------+----------------+
|  4 | Monsters,Inc.      | Pete Docter    |
|  5 | Finding Nemo       | Andrew Stanton |
|  6 | The Incredibles    | Brad Bird      |
|  8 | Ratatouille        | Brad Bird      |
|  9 | WALL-E             | Andrew Stanton |
| 10 | Up                 | Pete Docter    |
| 11 | Toy Story 3        | Lee Unkrich    |
| 13 | Brave              | Brenda Chapman |
| 14 | Monsters University | Dan Scanlon    |
+----+--------------------+----------------+
9 rows in set (0.00 sec)
```

Q4.Find all the WALL-* movies

SELECT id,title,director FROM movies where title like "WALL-_";

```
mysql> SELECT id,title,director FROM movies where  title like "WALL-_";
+----+--------+----------------+
| id | title  | director       |
+----+--------+----------------+
|  9 | WALL-E | Andrew Stanton |
+----+--------+----------------+
1 row in set (0.00 sec)
```

SQL Lesson 4: Filtering and sorting Query results

Q1.List all directors of Pixar movies (alphabetically), without duplicates

SELECT distinct director FROM movies order by director asc;

```
mysql> SELECT distinct director FROM movies order by director asc;
+----------------+
| director       |
+----------------+
| Andrew Stanton |
| Brad Bird      |
| Brenda Chapman |
| Dan Scanlon    |
| John Lasseter  |
| Lee Unkrich    |
| Pete Docter    |
+----------------+
7 rows in set (0.00 sec)
```

Q2.List the last four Pixar movies released (ordered from most recent to least)

SELECT title,year FROM movies order by year desc limit 4;

```
mysql> SELECT title,year FROM movies order by year desc limit 4;
+---------------------+------+
| title               | year |
+---------------------+------+
| Monsters University | 2013 |
| Brave               | 2012 |
| Cars 2              | 2011 |
| Toy Story 3         | 2010 |
+---------------------+------+
4 rows in set (0.00 sec)
```

Q3.List the first five Pixar movies sorted alphabetically

SELECT title FROM movies order by title asc limit 5;

```
mysql> SELECT title FROM movies order by title asc limit 5;
+--------------+
| title        |
+--------------+
| A Bugs Life  |
| Brave        |
| Cars         |
| Cars 2       |
| Finding Nemo |
+--------------+
5 rows in set (0.00 sec)
```

Q4.List the next five Pixar movies sorted alphabetically

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

```
mysql> SELECT title FROM movies order by title asc limit 5 offset 5;
+---------------------+
| title               |
+---------------------+
| Monsters University |
| Monsters,Inc.       |
| Ratatouille         |
| The Incredibles     |
| Toy Story           |
+---------------------+
5 rows in set (0.00 sec)
```

SQL Review: Simple SELECT Queries

Q1.List all the Canadian cities and their populations

Table: North_american_cities

City	Population
Toronto	2795060
Montreal	1717767

```
SELECT city,population FROM north_american_cities where country="Canada";
```

Q2.Order all the cities in the United States by their latitude from north to south

Table: North_american_cities

City	Country	Population	Latitude	Longitude
Chicago	United States	2718782	41.878114	-87.629798
New York	United States	8405837	40.712784	-74.005941
Philadelphia	United States	1553165	39.952584	-75.165222
Los Angeles	United States	3884307	34.052234	-118.243685
Phoenix	United States	1513367	33.448377	-112.074037
Houston	United States	2195914	29.760427	-95.369803

```
SELECT * FROM north_american_cities where country="United States" order by
    latitude desc;
```

Q3.List all the cities west of Chicago, ordered from west to east

Table: North_american_cities

City	Country	Population	Latitude	Longitude
Los Angeles	United States	3884307	34.052234	-118.243685
Phoenix	United States	1513367	33.448377	-112.074037
Guadalajara	Mexico	1500800	20.659699	-103.349609
Mexico City	Mexico	8555500	19.432608	-99.133208
Ecatepec de Morelos	Mexico	1742000	19.601841	-99.050674
Houston	United States	2195914	29.760427	-95.369803

```
SELECT * FROM north_american_cities where longitude<-87.629798 order by
    longitude asc;
```

Q4.List the two largest cities in Mexico (by population)

Table: North_american_cities

City	Country	Population	Latitude	Longitude
Mexico City	Mexico	8555500	19.432608	-99.133208
Ecatepec de Morelos	Mexico	1742000	19.601841	-99.050674

```
SELECT * FROM north_american_cities where country="Mexico" order by
    population desc limit 2;
```

Q5.List the third and fourth largest cities (by population) in the United States and their population

Table: North_american_cities

City	Country	Population	Latitude	Longitude
Chicago	United States	2718782	41.878114	-87.629798
Houston	United States	2195914	29.760427	-95.369803

```
SELECT * FROM north_american_cities where country="United States" order by
    population desc limit 2 offset 2;
```

SQL Lesson 6: Multi-table queries with JOINs

Q1.Find the domestic and international sales for each movie

SELECT id,title,domestic_sales,internaltional_sales FROM boxoffice inner join movies where id=movie_id order by id asc

```
mysql> SELECT id,title,domestic_sales,internaltional_sales FROM boxoffice inner join movies where id=movie_id order by id asc
    -> ;
+----+--------------------+----------------+---------------------+
| id | title              | domestic_sales | internaltional_sales |
+----+--------------------+----------------+---------------------+
|  1 | Toy Story          |      191796233 |           170162503 |
|  2 | A Bugs Life        |      162798565 |           200600000 |
|  3 | Toy Story 2        |      245852179 |           239163000 |
|  4 | Monsters,Inc.      |      289916256 |           272900000 |
|  5 | Finding Nemo       |      380843261 |           555900000 |
|  6 | The Incredibles    |      261441092 |           370001000 |
|  7 | Cars               |      244082982 |           217900167 |
|  8 | Ratatouille        |      206445654 |           417277164 |
|  9 | WALL-E             |      223808164 |           297503696 |
| 10 | Up                 |      293004164 |           438338580 |
| 11 | Toy Story 3        |      415004880 |           648167031 |
| 12 | Cars 2             |      191452396 |           368400000 |
| 13 | Brave              |      237283207 |           301700000 |
| 14 | Monsters University |      268492764 |           475066843 |
+----+--------------------+----------------+---------------------+
14 rows in set (0.00 sec)
```

Q2.Show the sales numbers for each movie that did better internationally rather than domestically

SELECT id,title,domestic_sales,internaltional_sales FROM boxoffice inner join movies where id=movie_id and internaltional_sales>domestic_sales order by id;

```
mysql> SELECT id,title,domestic_sales,internaltional_sales FROM boxoffice inner join movies where id=movie_id and internaltional_sales>domestic_sales order by id;
+----+--------------------+---------------+----------------------+
| id | title              | domestic_sales | internaltional_sales |
+----+--------------------+---------------+----------------------+
|  2 | A Bugs Life        |      162798565 |            200600000 |
|  5 | Finding Nemo       |      380843261 |            555900000 |
|  6 | The Incredibles    |      261441092 |            370001000 |
|  8 | Ratatouille        |      206445654 |            417277164 |
|  9 | WALL-E             |      223808164 |            297503696 |
| 10 | Up                 |      293004164 |            438338580 |
| 11 | Toy Story 3        |      415004880 |            648167031 |
| 12 | Cars 2             |      191452396 |            368400000 |
| 13 | Brave              |      237283207 |            301700000 |
| 14 | Monsters University |      268492764 |            475066843 |
+----+--------------------+---------------+----------------------+
10 rows in set (0.00 sec)
```

Q3.List all the movies by their ratings in descending order

SELECT id,title,rating,domestic_sales,internaltional_sales FROM boxoffice inner join movies where id=movie_id order by rating desc;

```
mysql> SELECT id,title,rating,domestic_sales,internaltional_sales FROM boxoffice inner join movies where id=movie_id order by rating desc;
+----+--------------------+--------+---------------+----------------------+
| id | title              | rating | domestic_sales | internaltional_sales |
+----+--------------------+--------+---------------+----------------------+
|  9 | WALL-E             |  8.500 |      223808164 |            297503696 |
| 11 | Toy Story 3        |  8.400 |      415004880 |            648167031 |
|  1 | Toy Story          |  8.300 |      191796233 |            170162503 |
| 10 | Up                 |  8.300 |      293004164 |            438338580 |
|  5 | Finding Nemo       |  8.200 |      380843261 |            555900000 |
|  4 | Monsters,Inc.      |  8.100 |      289916256 |            272900000 |
|  6 | The Incredibles    |  8.000 |      261441092 |            370001000 |
|  8 | Ratatouille        |  8.000 |      206445654 |            417277164 |
|  3 | Toy Story 2        |  7.900 |      245852179 |            239163000 |
| 14 | Monsters University |  7.400 |      268492764 |            475066843 |
|  2 | A Bugs Life        |  7.200 |      162798565 |            200600000 |
|  7 | Cars               |  7.200 |      244082982 |            217900167 |
| 13 | Brave              |  7.200 |      237283207 |            301700000 |
| 12 | Cars 2             |  6.400 |      191452396 |            368400000 |
+----+--------------------+--------+---------------+----------------------+
14 rows in set (0.00 sec)
```

SQL Lesson 7: OUTER JOINs

Q1.Find the list of all buildings that have employees

SELECT distinct building FROM employees left join buildings on building=building_name;

Query Results

Building
1e
2w

```
SELECT distinct building FROM employees left join buildings on building
    =building_name;
```

Q2.Find the list of all buildings and their capacity

select building_name,capacity from buildings;

Query Results

Building_name	Capacity
1e	24
1w	32
2e	16
2w	20

```
select building_name,capacity from buildings;
```

Q3.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=building_name;

Query Results

Building_name	Role
1e	Engineer
1e	Manager
1w	
2e	
2w	Artist
2w	Manager

```
SELECT distinct building_name,role FROM buildings left join employees on
    building=building_name;
```

SQL Lesson 8: A short note on NULLs

Q1.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;

Name	Role
Yancy I.	Engineer
Oliver P.	Artist

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

Q2.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;

Building_name
1w
2e

```
SELECT DISTINCT building_name FROM buildings left join employees on
    building_name = building WHERE role IS NULL;
```

SQL Lesson 9: Queries with expressions

Q1.List all movies and their combined sales in millions of dollars

SELECT title, (domestic_sales + internaltional_sales) / 1000000 AS gross_sales_millions from movies join boxoffice on movies.id = boxoffice.movie_id;

```
mysql> SELECT title, (domestic_sales + internaltional_sales) / 1000000 AS gross_sales_millions from movies join boxoffice on movies.id = boxoffice.movie_id;
+--------------------+----------------------+
| title              | gross_sales_millions |
+--------------------+----------------------+
| Toy Story          |             361.9587 |
| A Bugs Life        |             363.3986 |
| Toy Story 2        |             485.0152 |
| Monsters,Inc.      |             562.8163 |
| Finding Nemo       |             936.7433 |
| The Incredibles    |             631.4421 |
| Cars               |             461.9831 |
| Ratatouille        |             623.7228 |
| WALL-E             |             521.3119 |
| Up                 |             731.3427 |
| Toy Story 3        |            1063.1719 |
| Cars 2             |             559.8524 |
| Brave              |             538.9832 |
| Monsters University |             743.5596 |
+--------------------+----------------------+
14 rows in set (0.00 sec)
```

Q2.List all movies and their ratings in percent

SELECT title, rating * 10 AS rating_percentage FROM movies join boxoffice on movies.id = boxoffice.movie_id;

```
mysql> SELECT title, rating * 10 AS rating_percentage FROM movies join boxoffice on movies.id = boxoffice.movie_id;
+--------------------+-------------------+
| title              | rating_percentage |
+--------------------+-------------------+
| Toy Story          |            83.000 |
| A Bugs Life        |            72.000 |
| Toy Story 2        |            79.000 |
| Monsters,Inc.      |            81.000 |
| Finding Nemo       |            82.000 |
| The Incredibles    |            80.000 |
| Cars               |            72.000 |
| Ratatouille        |            80.000 |
| WALL-E             |            85.000 |
| Up                 |            83.000 |
| Toy Story 3        |            84.000 |
| Cars 2             |            64.000 |
| Brave              |            72.000 |
| Monsters University |            74.000 |
+--------------------+-------------------+
14 rows in set (0.00 sec)
```

Q3.List all movies that were released on even number years

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

```
mysql> SELECT title, year from movies where year % 2 = 0;
+------------------+------+
| title            | year |
+------------------+------+
| A Bugs Life      | 1998 |
| The Incredibles  | 2004 |
| Cars             | 2006 |
| WALL-E           | 2008 |
| Toy Story 3      | 2010 |
| Brave            | 2012 |
| Toy Story 4      | 2014 |
+------------------+------+
7 rows in set (0.00 sec)
```

SQL Lesson 10/11: Queries with aggregates

Q1.Find the longest time that an employee has been at the studio ✓

SELECT MAX(years_employed) as max_years_employed from employees;

Table: Employees

Max_years_employed
9

```
SELECT MAX(years_employed) as max_years_employed from employees;
```

Q2.For each role, find the average number of years employed by employees in that role

SELECT role, AVG(years_employed) as average_years_employed

from employees group by role;

Role	Average_years_employed
Artist	6
Engineer	3.4
Manager	6

```
SELECT role, AVG(years_employed) as average_years_employed
from employees group by role;
```

Q3.Find the total number of employee years worked in each building

SELECT building, SUM(years_employed) as total_years_employed

from employees group by building;

Building	Total_years_employed
1e	29
2w	36

```
SELECT building, SUM(years_employed) as total_years_employed
from employees group by building;
```

SQL Lesson 12: Order of execution of a Query

Q1.Find the number of movies each director has directed ✓

SELECT director, COUNT(id) as num_movies_directed

from movies group by director;

```
mysql> SELECT director, COUNT(id) as num_movies_directed
    -> from movies group by director;
+----------------+---------------------+
| director       | num_movies_directed |
+----------------+---------------------+
| John Lasseter  |                   2 |
| Brad Bird      |                   1 |
| Pete Docter    |                   1 |
| Lee Unkrich    |                   1 |
| Brenda Chapman |                   1 |
| Dan Scanlon    |                   1 |
+----------------+---------------------+
6 rows in set (0.00 sec)
```

Q2.Find the total domestic and international sales that can be attributed to each director

SELECT director, SUM(domestic_sales + internaltional_sales) as Cumulative_sales_from_all_movies

FROM movies inner join boxoffice ON movies.id = boxoffice.movie_id group by director;

```
mysql> SELECT director, SUM(domestic_sales + internaltional_sales) as Cumulative_sales_from_all_movies FROM movies inner join boxoffice ON movies.id = boxoffice.movie_id group by director;
+---------------+-----------------------------------+
| director      | Cumulative_sales_from_all_movies |
+---------------+-----------------------------------+
| John Lasseter |                        1021835545 |
| Brad Bird     |                         623722818 |
| Pete Docter   |                         731342744 |
| Lee Unkrich   |                        1063171911 |
| Brenda Chapman|                         538983207 |
| Dan Scanlon   |                         743559607 |
+---------------+-----------------------------------+
6 rows in set (0.00 sec)
```

SQL Lesson 13: Inserting rows

Q1.Add the studio's new production, Toy Story 4 to the list of movies (you can use any director)

insert into movies(title,director,year,length_minutes) values ("Toy Story 4","John Lasseter",2014,112);

```
mysql> insert into Movies(title,director,year,length_minutes) values ("Toy Story 4","Andrew Stanton",2014,112);
Query OK, 1 row affected (0.00 sec)

mysql> select * from movies;
+----+---------------------+----------------+------+----------------+
| Id | Title               | Director       | Year | Length_minutes |
+----+---------------------+----------------+------+----------------+
|  1 | Toy Story           | John Lasseter  | 1995 |             81 |
|  2 | A Bugs Life         | John Lasseter  | 1998 |             95 |
|  3 | Toy Story 2         | John Lasseter  | 1999 |             93 |
|  4 | Monsters,Inc.       | Pete Docter    | 2001 |             92 |
|  5 | Finding Nemo        | Andrew Stanton | 2003 |            107 |
|  6 | The Incredibles     | Brad Bird      | 2004 |            116 |
|  7 | Cars                | John Lasseter  | 2006 |            117 |
|  8 | Ratatouille         | Brad Bird      | 2007 |            115 |
|  9 | WALL-E              | Andrew Stanton | 2008 |            104 |
| 10 | Up                  | Pete Docter    | 2009 |            101 |
| 11 | Toy Story 3         | Lee Unkrich    | 2010 |            103 |
| 12 | Cars 2              | John Lasseter  | 2011 |            120 |
| 13 | Brave               | Brenda Chapman | 2012 |            102 |
| 14 | Monsters University | Dan Scanlon    | 2013 |            110 |
| 15 | Toy Story 4         | Andrew Stanton | 2014 |            112 |
+----+---------------------+----------------+------+----------------+
15 rows in set (0.00 sec)
```

Q2.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(rating,domestic_sales,internaltional_sales) values(8.7,340000000,270000000);

```
mysql> insert into boxoffice(rating,domestic_sales,internaltional_sales) values(8.7,340000000,270000000);
Query OK, 1 row affected (0.00 sec)

mysql> select * from boxoffice;
+----------+--------+---------------+----------------------+
| movie_id | Rating | Domestic_sales | Internaltional_sales |
+----------+--------+---------------+----------------------+
|        1 |  8.300 |      191796233 |            170162503 |
|        2 |  7.200 |      162798565 |            200600000 |
|        3 |  7.900 |      245852179 |            239163000 |
|        4 |  8.100 |      289916256 |            272900000 |
|        5 |  8.200 |      380843261 |            555900000 |
|        6 |  8.000 |      261441092 |            370001000 |
|        7 |  7.200 |      244082982 |            217900167 |
|        8 |  8.000 |      206445654 |            417277164 |
|        9 |  8.500 |      223808164 |            297503696 |
|       10 |  8.300 |      293004164 |            438338580 |
|       11 |  8.400 |      415004880 |            648167031 |
|       12 |  6.400 |      191452396 |            368400000 |
|       13 |  7.200 |      237283207 |            301700000 |
|       14 |  7.400 |      268492764 |            475066843 |
|       15 |  8.700 |      340000000 |            270000000 |
+----------+--------+---------------+----------------------+
15 rows in set (0.00 sec)
```

SQL Lesson 14: Updating rows

Q1.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;

```
mysql> UPDATE movies set director = "John Lasseter" where id = 2;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1  Changed: 0  Warnings: 0
```

Q2.The year that Toy Story 2 was released is incorrect, it was actually released in 1999 ✓

UPDATE movies SET year = 1999 where id=3;

```
mysql> UPDATE movies SET year = 1999 where id=3;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1  Changed: 0  Warnings: 0
```

Q3.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;

```
mysql> UPDATE movies SET title="Toy Story 3",director="Lee Unkrich" where id=11;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1  Changed: 0  Warnings: 0
```

SQL Lesson 15: Deleting rows

Q1.This database is getting too big, lets remove all movies that were released before 2005.

delete from movies where year<2005;

```
mysql> select * from movies;
+----+---------------------+----------------+------+----------------+
| Id | Title               | Director       | Year | Length_minutes |
+----+---------------------+----------------+------+----------------+
|  1 | Toy Story           | John Lasseter  | 1995 |             81 |
|  2 | A Bugs Life         | John Lasseter  | 1998 |             95 |
|  3 | Toy Story 2         | John Lasseter  | 1999 |             93 |
|  4 | Monsters,Inc.       | Pete Docter    | 2001 |             92 |
|  5 | Finding Nemo        | Andrew Stanton | 2003 |            107 |
|  6 | The Incredibles     | Brad Bird      | 2004 |            116 |
|  7 | Cars                | John Lasseter  | 2006 |            117 |
|  8 | Ratatouille         | Brad Bird      | 2007 |            115 |
|  9 | WALL-E              | Andrew Stanton | 2008 |            104 |
| 10 | Up                  | Pete Docter    | 2009 |            101 |
| 11 | Toy Story 3         | Lee Unkrich    | 2010 |            103 |
| 12 | Cars 2              | John Lasseter  | 2011 |            120 |
| 13 | Brave               | Brenda Chapman | 2012 |            102 |
| 14 | Monsters University | Dan Scanlon    | 2013 |            110 |
| 15 | Toy Story 4         | Andrew Stanton | 2014 |            112 |
+----+---------------------+----------------+------+----------------+
15 rows in set (0.00 sec)

mysql> delete from movies where year<2005;
Query OK, 6 rows affected (0.00 sec)

mysql> select * from movies;
+----+---------------------+----------------+------+----------------+
| Id | Title               | Director       | Year | Length_minutes |
+----+---------------------+----------------+------+----------------+
|  7 | Cars                | John Lasseter  | 2006 |            117 |
|  8 | Ratatouille         | Brad Bird      | 2007 |            115 |
|  9 | WALL-E              | Andrew Stanton | 2008 |            104 |
| 10 | Up                  | Pete Docter    | 2009 |            101 |
| 11 | Toy Story 3         | Lee Unkrich    | 2010 |            103 |
| 12 | Cars 2              | John Lasseter  | 2011 |            120 |
| 13 | Brave               | Brenda Chapman | 2012 |            102 |
| 14 | Monsters University | Dan Scanlon    | 2013 |            110 |
| 15 | Toy Story 4         | Andrew Stanton | 2014 |            112 |
+----+---------------------+----------------+------+----------------+
9 rows in set (0.00 sec)
```

Q2.Andrew Stanton has also left the studio, so please remove all movies directed by him.

delete from movies where director="Andrew Stanton";

```
mysql> delete from movies where director="Andrew Stanton";
Query OK, 2 rows affected (0.00 sec)

mysql> select * from movies;
+----+---------------------+----------------+------+----------------+
| Id | Title               | Director       | Year | Length_minutes |
+----+---------------------+----------------+------+----------------+
|  7 | Cars                | John Lasseter  | 2006 |            117 |
|  8 | Ratatouille         | Brad Bird      | 2007 |            115 |
| 10 | Up                  | Pete Docter    | 2009 |            101 |
| 11 | Toy Story 3         | Lee Unkrich    | 2010 |            103 |
| 12 | Cars 2              | John Lasseter  | 2011 |            120 |
| 13 | Brave               | Brenda Chapman | 2012 |            102 |
| 14 | Monsters University | Dan Scanlon    | 2013 |            110 |
+----+---------------------+----------------+------+----------------+
7 rows in set (0.00 sec)
```

SQL Lesson 16: Creating tables

```
mysql> insert into boxoffice(rating, domestic_sales, internaltional_sales) values
    -> (8.3,191796233,170162503);
Query OK, 1 row affected (0.00 sec)

mysql> select * from boxoffice;
+----------+--------+----------------+----------------------+
| movie_id | Rating | Domestic_sales | Internaltional_sales |
+----------+--------+----------------+----------------------+
|        1 |  8.300 |      191796233 |            170162503 |
+----------+--------+----------------+----------------------+
1 row in set (0.00 sec)
```

```
mysql> insert into boxoffice(rating, domestic_sales, internaltional_sales) values
    -> (7.2,162798565,200600000),
    -> (7.9,245852179,239163000),
    -> (8.1,289916256,272900000),
    -> (8.2,380843261,555900000),(8,261441092,370001000),(7.2,244082982,217900167),(8,206445654,417277164),
    -> (8.5,223808164,297503696),(8.3,293004164,438338580),(8.4,415004880,648167031),(6.4,191452396,368400000),
    -> (7.2,237283207,301700000),(7.4,268492764,475066843);
Query OK, 13 rows affected (0.00 sec)
Records: 13  Duplicates: 0  Warnings: 0

mysql> select * from boxoffice;
+----------+--------+----------------+----------------------+
| movie_id | Rating | Domestic_sales | Internaltional_sales |
+----------+--------+----------------+----------------------+
|        1 |  8.300 |      191796233 |            170162503 |
|        2 |  7.200 |      162798565 |            200600000 |
|        3 |  7.900 |      245852179 |            239163000 |
|        4 |  8.100 |      289916256 |            272900000 |
|        5 |  8.200 |      380843261 |            555900000 |
|        6 |  8.000 |      261441092 |            370001000 |
|        7 |  7.200 |      244082982 |            217900167 |
|        8 |  8.000 |      206445654 |            417277164 |
|        9 |  8.500 |      223808164 |            297503696 |
|       10 |  8.300 |      293004164 |            438338580 |
|       11 |  8.400 |      415004880 |            648167031 |
|       12 |  6.400 |      191452396 |            368400000 |
|       13 |  7.200 |      237283207 |            301700000 |
|       14 |  7.400 |      268492764 |            475066843 |
+----------+--------+----------------+----------------------+
14 rows in set (0.00 sec)
```

SQL Lesson 17: Altering tables

Q1.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;

```
mysql> ALTER TABLE movies ADD Aspect_ratio float;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select * from movies;
+----+----------------------+----------------+------+----------------+--------------+
| Id | Title                | Director       | Year | Length_minutes | Aspect_ratio |
+----+----------------------+----------------+------+----------------+--------------+
|  7 | Cars                 | John Lasseter  | 2006 |            117 |         NULL |
|  8 | Ratatouille          | Brad Bird      | 2007 |            115 |         NULL |
| 10 | Up                   | Pete Docter    | 2009 |            101 |         NULL |
| 11 | Toy Story 3          | Lee Unkrich    | 2010 |            103 |         NULL |
| 12 | Cars 2               | John Lasseter  | 2011 |            120 |         NULL |
| 13 | Brave                | Brenda Chapman | 2012 |            102 |         NULL |
| 14 | Monsters University  | Dan Scanlon    | 2013 |            110 |         NULL |
+----+----------------------+----------------+------+----------------+--------------+
7 rows in set (0.00 sec)
```

Q2.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 Language varchar(20) default "English";

```
mysql> ALTER TABLE movies ADD Language varchar(20) default "English";
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select * from movies;
+----+----------------------+----------------+------+----------------+--------------+----------+
| Id | Title                | Director       | Year | Length_minutes | Aspect_ratio | Language |
+----+----------------------+----------------+------+----------------+--------------+----------+
|  7 | Cars                 | John Lasseter  | 2006 |            117 |         NULL | English  |
|  8 | Ratatouille          | Brad Bird      | 2007 |            115 |         NULL | English  |
| 10 | Up                   | Pete Docter    | 2009 |            101 |         NULL | English  |
| 11 | Toy Story 3          | Lee Unkrich    | 2010 |            103 |         NULL | English  |
| 12 | Cars 2               | John Lasseter  | 2011 |            120 |         NULL | English  |
| 13 | Brave                | Brenda Chapman | 2012 |            102 |         NULL | English  |
| 14 | Monsters University  | Dan Scanlon    | 2013 |            110 |         NULL | English  |
+----+----------------------+----------------+------+----------------+--------------+----------+
7 rows in set (0.00 sec)
```

SQL Lesson 18: Dropping tables

DROP TABLE Movies;

```
DROP TABLE IF EXISTS boxoffice;
```