Database – MySQL

Student Name: Netravati

Batch: B34 WD English

Exercise

We will be using a database with data about some of Pixar's classic movies for most of our exercises. This first exercise will only involve the **Movies** table, and the default query below currently shows all the properties of each movie. To continue onto the next lesson, alter the query to find the exact information we need for each task.

Exercise 1 — Tasks

1. Find the `title` of each film ✓

Table: Movies

Title
Toy Story
A Bug's Life
Toy Story 2
Monsters, Inc.
Finding Nemo
The Incredibles
Cars
Ratatouille
WALL-E
Up

```
SELECT title FROM movies;
```

2. Find the `director` of each film ✓

Table: Movies

Director
John Lasseter
John Lasseter
John Lasseter
Pete Docter
Andrew Stanton
Brad Bird
John Lasseter
Brad Bird
Andrew Stanton
Pete Docter

```
SELECT director FROM movies;
```

3. Find the `title` and `director` of each film

Table: Movies

Title	Director
Toy Story	John Lasseter
A Bug's 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

```
SELECT title, director FROM movies;
```

4. Find the `title` and `year` of each film ✓

Table: Movies

Title	Year
Toy Story	1995
A Bug's Life	1998
Toy Story 2	1999
Monsters, Inc.	2001
Finding Nemo	2003
The Incredibles	2004
Cars	2006
Ratatouille	2007
WALL-E	2008
Up	2009

```
SELECT title, year FROM movies;
```

5. Find `all` the information about each film ✓

Table: Movies

Id	Title	Director	Year	Length_minutes
1	Toy Story	John Lasseter	1995	81
2	A Bug's 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

```
SELECT * FROM movies;
```

Exercise

Using the right constraints, find the information we need from the **Movies** table for each task below.

Exercise 2 — Tasks

1. Find the movie with a row `id` of 6 ✓

Table: Movies

Id	Title	Director	Year	Length_minutes
6	The Incredibles	Brad Bird	2004	116

```
SELECT * FROM movies where id=6;
```

3. Find the movies **not** released in the `year`s between 2000 and 2010 ✓

Table: Movies

Id	Title	Director	Year	Length_minutes
1	Toy Story	John Lasseter	1995	81
2	A Bug's Life	John Lasseter	1998	95
3	Toy Story 2	John Lasseter	1999	93
12	Cars 2	John Lasseter	2011	120
13	Brave	Brenda Chapman	2012	102
14	Monsters University	Dan Scanlon	2013	110

```
SELECT * FROM movies Where year NOT BETWEEN 2000 AND 2010;
```

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

Table: Movies

Id	Title	Director	Year	Length_minutes
1	Toy Story	John Lasseter	1995	81
2	A Bug's 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

```
SELECT * FROM movies limit 5;
```

Exercise

Here's the definition of a query with a **WHERE** clause again, go ahead and try and write some queries with the operators above to limit the results to the information we need in the tasks below.

Exercise 3 — Tasks

1. Find all the Toy Story movies ✓

Table: Movies

Id	Title	Director	Year	Length_minutes
1	Toy Story	John Lasseter	1995	81
3	Toy Story 2	John Lasseter	1999	93
11	Toy Story 3	Lee Unkrich	2010	103

```
SELECT * FROM movies where Title like "%Toy%";
```

2. Find all the movies directed by John Lasseter ✓

Table: Movies

Id	Title	Director	Year	Length_minutes
1	Toy Story	John Lasseter	1995	81
2	A Bug's Life	John Lasseter	1998	95
3	Toy Story 2	John Lasseter	1999	93
7	Cars	John Lasseter	2006	117
12	Cars 2	John Lasseter	2011	120

```
SELECT * FROM movies where Director Like "%John Lasseter%";
```

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

Table: Movies

Id	Title	Director	Year	Length_minutes
4	Monsters, Inc.	Pete Docter	2001	92
5	Finding Nemo	Andrew Stanton	2003	107
6	The Incredibles	Brad Bird	2004	116
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
13	Brave	Brenda Chapman	2012	102
14	Monsters University	Dan Scanlon	2013	110
87	WALL-G	Brenda Chapman	2042	97

```
SELECT * FROM movies where Director NOT Like "%John Lasseter%";
```

4. Find all the WALL-* movies ✓

Table: Movies

Id	Title	Director	Year	Length_minutes
9	WALL-E	Andrew Stanton	2008	104
87	WALL-G	Brenda Chapman	2042	97

```
SELECT * FROM movies where Title Like "%WALL-%";
```

Exercise 4 — Tasks

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

Table: Movies

Director
Andrew Stanton
Brad Bird
Brenda Chapman
Dan Scanlon
John Lasseter
Lee Unkrich
Pete Docter

```
SELECT distinct(Director) FROM movies order by Director ASC ;
```

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

Table: Movies

Id	Title	Director	Year	Length_minutes
11	Monsters University	Dan Scanlon	2013	110
8	Brave	Brenda Chapman	2012	102
10	Cars 2	John Lasseter	2011	120
9	Toy Story 3	Lee Unkrich	2010	103

```
SELECT * FROM movies order by year DESC limit 4 ;
```

3. List the **first** five Pixar movies sorted alphabetically ✓

Table: Movies

Id	Title	Director	Year	Length_minutes
4	Toy Story	John Lasseter	1995	81
1	A Bug's Life	John Lasseter	1998	95
5	Toy Story 2	John Lasseter	1999	93
14	Monsters, Inc.	Pete Docter	2001	92
12	Finding Nemo	Andrew Stanton	2003	107

```
SELECT * FROM movies order by year ASC,Title ASC, year limit 5;
```

4. List the **next** five Pixar movies sorted alphabetically ✓

Table: Movies

Title
Monsters University
Monsters, Inc.
Ratatouille
The Incredibles
Toy Story

```
SELECT title FROM movies ORDER BY title ASC LIMIT 5 OFFSET 5;
```

Exercise

In the exercise below, you will be working with a different table. This table instead contains information about a few of the most populous cities of North America[1] including their population and geo-spatial location in the world.

> Did you know?
>
> Positive latitudes correspond to the northern hemisphere, and positive longitudes correspond to the eastern hemisphere. Since North America is north of the equator and west of the prime meridian, all of the cities in the list have positive latitudes and negative longitudes.

Try and write some queries to find the information requested in the tasks you know. You may have to use a different combination of clauses in your query for each task. Once you're done, continue onto the next lesson to learn about queries that span multiple tables.

Review 1 — Tasks

1. 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";
```

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

Table: North_american_cities

City	Latitude
Chicago	41.878114
New York	40.712784
Philadelphia	39.952584
Los Angeles	34.052234
Phoenix	33.448377
Houston	29.760427

```
SELECT city, latitude 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 ✓

Table: North_american_cities

City	Longitude
Los Angeles	-118.243685
Phoenix	-112.074037
Guadalajara	-103.349609
Mexico City	-99.133208
Ecatepec de Morelos	-99.050674
Houston	-95.369803

```
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) ✓

Table: North_american_cities

City	Population
Mexico City	8555500
Ecatepec de Morelos	1742000

```
SELECT city, population FROM north_american_cities WHERE country LIKE
    "Mexico" ORDER BY population DESC LIMIT 2;
```

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

Table: North_american_cities

City	Population
Chicago	2718782
Houston	2195914

```
SELECT city, population FROM north_american_cities WHERE country LIKE
    "United States" ORDER BY population DESC LIMIT 2 OFFSET 2;
```