

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.

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;
```

RESET

Exercise 1 — Tasks

1. Find the **title** of each film ✓
2. Find the **director** of each film
3. Find the **title** and **director** of each film
4. Find the **title** and **year** of each film
5. Find **all** the information about each film

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Finish above Tasks

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.

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;

RESET

Exercise 1 — Tasks

- 1. Find the **title** of each film ✓
- 2. Find the **director** of each film ✓
- 3. Find the **title** and **director** of each film
- 4. Find the **title** and **year** of each film
- 5. Find **all** the information about each film

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Finish above Tasks

Next – [SQL Lesson 2: Queries with constraints \(Pt. 1\)](#)  
Previous – [Introduction to SQL](#)

Find SQLBolt useful? Please consider  
Donating (\$4) via [Paypal](#) to support our site.

?

```
SELECT *
FROM mytable;
```

This query, in particular, is really useful because it's a simple way to inspect a table by dumping all the data at once.

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.

Table: Movies

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

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

RESET

Exercise 1 — Tasks

- 1. Find the **title** of each film ✓
- 2. Find the **director** of each film ✓
- 3. Find the **title** and **director** of each film ✓
- 4. Find the **title** and **year** of each film
- 5. Find **all** the information about each film

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Finish above Tasks

```
SELECT *  
FROM mytable;
```

This query, in particular, is really useful because it's a simple way to inspect a table by dumping all the data at once.

### 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.

Table: Movies

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

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

RESET

### Exercise 1 — Tasks

1. Find the **title** of each film ✓
2. Find the **director** of each film ✓
3. Find the **title** and **director** of each film ✓
4. Find the **title** and **year** of each film ✓
5. Find **all** the information about each film

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Finish above Tasks

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.

Table: Movies

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

SELECT \* FROM movies

RESET

Exercise 1 — Tasks

1. Find the **title** of each film ✓
2. Find the **director** of each film ✓
3. Find the **title** and **director** of each film ✓
4. Find the **title** and **year** of each film ✓
5. Find **all** the information about each film ✓

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Continue >

Next — [SQL Lesson 2: Queries with constraints \(Pt. 1\)](#)  
Previous — [Introduction to SQL](#)

Find SQLBolt useful? Please consider  
[Donating \(\\$4\) via Paypal](#) to support our site.

Free ChatGPT Browser Extension



In addition to making the results more manageable to understand, writing clauses to constrain the set of rows returned also allows the query to run faster due to the reduction in unnecessary data being returned.

Did you know?

As you might have noticed by now, SQL doesn't *require* you to write the keywords all capitalized, but as a convention, it helps people distinguish SQL keywords from column and tables names, and makes the query easier to read.

Exercise

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

Table: Movies

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

```
SELECT * FROM movies WHERE id IN (6)
```

RESET

Exercise 2 — Tasks

1. Find the movie with a row **id** of 6 ✓
2. Find the movies released in the **year**s between 2000 and 2010
3. Find the movies **not** released in the **year**s between 2000 and 2010
4. Find the first 5 Pixar movies and their release **year**

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Finish above Tasks

Did you know?

As you might have noticed by now, SQL doesn't *require* you to write the keywords all capitalized, but as a convention, it helps people distinguish SQL keywords from column and tables names, and makes the query easier to read.

### Exercise

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

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
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

```
SELECT * FROM movies WHERE year BETWEEN 2000 AND 2010
```

RESET

#### Exercise 2 — Tasks

1. Find the movie with a row `id` of 6 ✓
2. Find the movies released in the `year`s between 2000 and 2010 ✓
3. Find the movies **not** released in the `year`s between 2000 and 2010
4. Find the first 5 Pixar movies and their release `year`

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Finish above Tasks

Next — SQL Lesson 3: Queries with constraints (Pt. 2)  
Previous — SQL Lesson 1: SELECT queries 101

Find SQLBolt useful? Please consider  
[Donating \(\\$4\) via Paypal](#) to support our site.

## Exercise

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

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
```

RESET

### Exercise 2 — Tasks

1. Find the movie with a row **id** of 6 ✓
2. Find the movies released in the **year** s between 2000 and 2010 ✓
3. Find the movies **not** released in the **year** s between 2000 and 2010 ✓
4. Find the first 5 Pixar movies and their release **year**

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Finish above Tasks

Next – [SQL Lesson 3: Queries with constraints \(Pt. 2\)](#)  
Previous – [SQL Lesson 1: SELECT queries 101](#)

Find SQLBolt useful? Please consider  
[Donating \(\\$4\) via Paypal](#) to support our site.





## Exercise

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

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 WHERE id <=5
```

RESET

### Exercise 2 — Tasks

1. Find the movie with a row **id** of 6 ✓
2. Find the movies released in the **year** s between 2000 and 2010 ✓
3. Find the movies **not** released in the **year** s between 2000 and 2010 ✓
4. Find the first 5 Pixar movies and their release **year** ✓

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Continue >

Next – [SQL Lesson 3: Queries with constraints \(Pt. 2\)](#)  
Previous – [SQL Lesson 1: SELECT queries 101](#)

Find SQLBolt useful? Please consider  
[Donating \(\\$4\) via Paypal](#) to support our site.



```
FROM mytable
WHERE condition
AND/OR another_condition
AND/OR ...;
```

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%'
```

RESET

### Exercise 3 — Tasks

1. Find all the Toy Story movies ✓
2. Find all the movies directed by John Lasseter
3. Find all the movies (and director) not directed by John Lasseter
4. Find all the WALL-\* movies

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Finish above Tasks

Next – SQL Lesson 4: Filtering and sorting Query results  
Previous – SQL Lesson 2: Queries with constraints (Pt. 1)

Find SQLBolt useful? Please consider  
[Donating \(\\$4\) via Paypal](#) to support our site.

90s Kids-க்கு தேவையான வினைத்து பொருட்களும்



the operators above to limit the results to the information we need in the tasks below.

```
Select query with constraints
SELECT column, another_column, ...
FROM mytable
WHERE condition
    AND/OR another_condition
    AND/OR ...;
```

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%'
```

RESET

Exercise 3 — Tasks

- 1. Find all the Toy Story movies ✓
- 2. Find all the movies directed by John Lasseter ✓
- 3. Find all the movies (and director) not directed by John Lasseter
- 4. Find all the WALL-\* movies

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Finish above Tasks

the operators above to limit the results to the information we need in the tasks below.

```
Select query with constraints
SELECT column, another_column, ...
FROM mytable
WHERE condition
    AND/OR another_condition
    AND/OR ...;
```

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%'
```

RESET

Exercise 3 — Tasks

- 1. Find all the Toy Story movies ✓
- 2. Find all the movies directed by John Lasseter ✓
- 3. Find all the movies (and director) not directed by John Lasseter ✓
- 4. Find all the WALL-\* movies

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Finish above Tasks

```
FROM mytable
WHERE condition
AND/OR another_condition
AND/OR ...;
```

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-%'
```

RESET

### Exercise 3 — Tasks

1. Find all the Toy Story movies ✓
2. Find all the movies directed by John Lasseter ✓
3. Find all the movies (and director) not directed by John Lasseter ✓
4. Find all the WALL-\* movies ✓

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Continue >

Next – SQL Lesson 4: Filtering and sorting Query results  
Previous – SQL Lesson 2: Queries with constraints (Pt. 1)

Find SQLBolt useful? Please consider  
[Donating \(\\$4\) via Paypal](#) to support our site.

90s Kids-க்கு தேவையான வினைத்து பொருட்களும்





## Exercise

There are a few concepts in this lesson, but all are pretty straight-forward to apply. To spice things up, we've gone and scrambled the **Movies** table for you in the exercise to better mimic what kind of data you might see in real life. Try and use the necessary keywords and clauses introduced above in your queries.

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 |
```

RESET

### Exercise 4 — Tasks

1. List all directors of Pixar movies (alphabetically), without duplicates ✓
2. List the last four Pixar movies released (ordered from most recent to least)
3. List the **first** five Pixar movies sorted alphabetically
4. List the **next** five Pixar movies sorted alphabetically

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Finish above Tasks

Next – [SQL Review: Simple SELECT Queries](#)  
Previous – [SQL Lesson 3: Queries with constraints \(Pt. 2\)](#)

Find SQLBolt useful? Please consider  
[Donating \(\\$4\) via Paypal](#) to support our site.



If you are curious about when the `LIMIT` and `OFFSET` are applied relative to the other parts of a query, they are generally done last after the other clauses have been applied. We'll touch more on this in [Lesson 12: Order of execution](#) after introducing a few more parts of the query.

## Exercise

There are a few concepts in this lesson, but all are pretty straight-forward to apply. To spice things up, we've gone and scrambled the **Movies** table for you in the exercise to better mimic what kind of data you might see in real life. Try and use the necessary keywords and clauses introduced above in your queries.

Table: Movies

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

```
SELECT * FROM movies ORDER BY year DESC LIMIT 4
```

RESET

### Exercise 4 — Tasks

1. List all directors of Pixar movies (alphabetically), without duplicates ✓
2. List the last four Pixar movies released (ordered from most recent to least) ✓
3. List the **first** five Pixar movies sorted alphabetically
4. List the **next** five Pixar movies sorted alphabetically

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Finish above Tasks

Next – [SQL Review: Simple SELECT Queries](#)  
Previous – [SQL Lesson 3: Queries with constraints \(Pt. 2\)](#)

Find SQLBolt useful? Please consider  
[Donating \(\\$4\) via Paypal](#) to support our site.

## Exercise

There are a few concepts in this lesson, but all are pretty straight-forward to apply. To spice things up, we've gone and scrambled the **Movies** table for you in the exercise to better mimic what kind of data you might see in real life. Try and use the necessary keywords and clauses introduced above in your queries.

Table: Movies

Id	Title	Director	Year	Length_minutes
7	A Bug's Life	John Lasseter	1998	95
2	Brave	Brenda Chapman	2012	102
1	Cars	John Lasseter	2006	117
3	Cars 2	John Lasseter	2011	120
6	Finding Nemo	Andrew Stanton	2003	107

```
SELECT * FROM movies ORDER BY title LIMIT 5
```

RESET

### Exercise 4 — Tasks

1. List all directors of Pixar movies (alphabetically), without duplicates ✓
2. List the last four Pixar movies released (ordered from most recent to least) ✓
3. List the **first** five Pixar movies sorted alphabetically ✓
4. List the **next** five Pixar movies sorted alphabetically

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Finish above Tasks

Next – [SQL Review: Simple SELECT Queries](#)  
Previous – [SQL Lesson 3: Queries with constraints \(Pt. 2\)](#)

Find SQLBolt useful? Please consider  
[Donating \(\\$4\) via Paypal](#) to support our site.



If you are confused about when the **WHERE** and **ORDER BY** are applied relative to the other parts of a query, they are generally done last after the other clauses have been applied. We'll touch more on this in [Lesson 12: Order of execution](#) after introducing a few more parts of the query.

### Exercise

There are a few concepts in this lesson, but all are pretty straight-forward to apply. To spice things up, we've gone and scrambled the **Movies** table for you in the exercise to better mimic what kind of data you might see in real life. Try and use the necessary keywords and clauses introduced above in your queries.

Table: Movies

Id	Title	Director	Year	Length_minutes
5	Monsters University	Dan Scanlon	2013	110
14	Monsters, Inc.	Pete Docter	2001	92
10	Ratatouille	Brad Bird	2007	115
8	The Incredibles	Brad Bird	2004	116
9	Toy Story	John Lasseter	1995	81

```
SELECT * from movies where title between 'Monster University' and 'Toy Story' order by title asc
```

RESET

#### Exercise 4 — Tasks

1. List all directors of Pixar movies (alphabetically), without duplicates ✓
2. List the last four Pixar movies released (ordered from most recent to least) ✓
3. List the **first** five Pixar movies sorted alphabetically ✓
4. List the **next** five Pixar movies sorted alphabetically ✓

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Continue >



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.

Table: North\_american\_cities

City	Country	Population
Toronto	Canada	2795060
Montreal	Canada	1717767

```
SELECT city, country, population FROM north_american_cities WHERE country = 'Canada'
```

RESET

Review 1 — Tasks

- 1. List all the Canadian cities and their populations ✓
- 2. Order all the cities in the United States by their latitude from north to south
- 3. List all the cities west of Chicago, ordered from west to east
- 4. List the two largest cities in Mexico (by population)
- 5. List the third and fourth largest cities (by population) in the United States and their population

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Finish above Tasks



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.

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
```

RESET

#### Review 1 — Tasks

1. List all the Canadian cities and their populations ✓
2. Order all the cities in the United States by their latitude from north to south ✓
3. List all the cities west of Chicago, ordered from west to east
4. List the two largest cities in Mexico (by population)
5. List the third and fourth largest cities (by population) in the United States and their population

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Finish above Tasks

Next – [SQL Lesson 6: Multi-table queries with JOINS](#)  
Previous – [SQL Lesson 4: Filtering and sorting Query results](#)

Find SQLBolt useful? Please consider  
[Donating \(\\$4\) via Paypal](#) to support our site.

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.

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
Chicago	-87.629798
Havana	-82.345189
Toronto	-79.383184
Philadelphia	-75.165222

```
SELECT city, longitude FROM north_american_cities WHERE longitude < -8.629798
ORDER BY longitude ASC;
```

RESET

Review 1 — Tasks

1. List all the Canadian cities and their populations ✓
2. Order all the cities in the United States by their latitude from north to south ✓
3. List all the cities west of Chicago, ordered from west to east ✓
4. List the two largest cities in Mexico (by population)
5. List the third and fourth largest cities (by population) in the United States and their population

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Finish above Tasks

Next – [SQL Lesson 6: Multi-table queries with JOINS](#)  
Previous – [SQL Lesson 4: Filtering and sorting Query results](#)

Find SQLBolt useful? Please consider  
[Donating \(\\$4\) via Paypal](#) to support our site.



11 Days Grand Australia Tour

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.

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
```

RESET

Review 1 — Tasks

1. List all the Canadian cities and their populations ✓
2. Order all the cities in the United States by their latitude from north to south ✓
3. List all the cities west of Chicago, ordered from west to east ✓
4. List the two largest cities in Mexico (by population) ✓
5. List the third and fourth largest cities (by population) in the United States and their population

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Finish above Tasks

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.

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
```

RESET

Review 1 — Tasks

- 1. List all the Canadian cities and their populations ✓
- 2. Order all the cities in the United States by their latitude from north to south ✓
- 3. List all the cities west of Chicago, ordered from west to east ✓
- 4. List the two largest cities in Mexico (by population) ✓
- 5. List the third and fourth largest cities (by population) in the United States and their population ✓

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Continue >



3	Toy Story 2	John Lasseter	1999	93	8	8	206445654	417277164
4	Monsters, Inc.	Pete Docter	2001	92	12	6.4	191452396	368400000
5	Finding Nemo	Andrew Stanton	2003	107	3	7.9	245852179	239163000
6	The Incredibles	Brad Bird	2004	116	6	8	261441092	370001000
7	Cars	John Lasseter	2006	117	0	6.5	200893154	207502100

#### Query Results

Title
Finding Nemo
Monsters University
Ratatouille
Cars 2
The Incredibles
WALL-E
Toy Story 3
Up
A Bug's Life
Brave

```
SELECT Title from Movies
INNER JOIN Boxoffice on Id = Movie_id WHERE Domestic_sales <
    International_sales
```

RESET

#### Exercise 6 — Tasks

1. Find the domestic and international sales for each movie ✓
2. Show the sales numbers for each movie that did better internationally rather than domestically ✓
3. List all the movies by their ratings in descending order

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Finish above Tasks

Next – [SQL Lesson 7: OUTER JOINS](#)  
Previous – [SQL Review: Simple SELECT Queries](#)

Find SQLBolt useful? Please consider  
[Donating \(\\$4\) via Paypal](#) to support our site.





6	The Incredibles	Brad Bird	2004	116	6	8	261441092	370001000
---	-----------------	-----------	------	-----	---	---	-----------	-----------

#### Query Results

Title
Finding Nemo
Monsters University
Ratatouille
Cars 2
The Incredibles
WALL-E
Toy Story 3
Up
A Bug's Life
Brave

```
SELECT Title from Movies
INNER JOIN Boxoffice on Id = Movie_id WHERE Domestic_sales <
International_sales
```

RESET

#### Exercise 6 — Tasks

1. Find the domestic and international sales for each movie ✓
2. Show the sales numbers for each movie that did better internationally rather than domestically ✓
3. List all the movies by their ratings in descending order

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Finish above Tasks

Next — [SQL Lesson 7: OUTER JOINs](#)  
Previous — [SQL Review: Simple SELECT Queries](#)

Find SQLBolt useful? Please consider  
[Donating \(\\$4\) via Paypal](#) to support our site.

EASY EMI  
RS.30000

10 Days London  
Scotland Vacation

gt holidays  
Travel, Vacations

## 10 Days London Scotland Tour

Book your Group Tour Packages to Australia, UK, Japan, South Africa, and

column in that table corresponds with the id column in the movies table (i.e., try and solve the tasks below using the **INNER JOIN** introduced above.

Table: Movies (Read-Only)

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

Table: Boxoffice (Read-Only)

Movie_id	Rating	Domestic_sales	International_sales
5	8.2	380843261	555900000
14	7.4	268492764	475066843
8	8	206445654	417277164
12	6.4	191452396	368400000
3	7.9	245852179	239163000
6	8	261441092	370001000

Query Results

Title
WALL-E
Toy Story 3
Toy Story
Up
Finding Nemo
Monsters, Inc.
Ratatouille
The Incredibles
Toy Story 2
Monsters University

```
SELECT Title FROM Movies
INNER JOIN Boxoffice ON Id = Movie_id
ORDER BY Rating DESC
```

Exercise 6 — Tasks

1. Find the domestic and international sales for each movie ✓
2. Show the sales numbers for each movie that did better internationally rather than domestically ✓
3. List all the movies by their ratings in descending order ✓

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

RESET

Continue >

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

Engineer	Becky A.	1e	4
Engineer	Dan B.	1e	2
Engineer	Sharon F.	1e	6
Engineer	Dan M.	1e	4
Engineer	Malcom S.	1e	1
Artist	Tylar S.	2w	2

Query Results

Building
1e
2w

```
SELECT DISTINCT building FROM employees;
```

RESET

### Exercise 7 — Tasks

1. Find the list of all buildings that have employees ✓
2. Find the list of all buildings and their capacity
3. List all buildings and the distinct employee roles in each building (including empty buildings)

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Finish above Tasks

Next – SQL Lesson 8: A short note on NULLs  
Previous – SQL Lesson 6: Multi-table queries with JOINs

Find SQLBolt useful? Please consider  
Donating (\$4) via Paypal to support our site.

1e	24	Engineer	Becky A.	1e	4
1w	32	Engineer	Dan B.	1e	2
2e	16	Engineer	Sharon F.	1e	6
2w	20	Engineer	Dan M.	1e	4
		Engineer	Malcom S.	1e	1
		Artist	Tylar S.	2w	2

Query Results

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

```
SELECT * FROM Buildings
```

RESET

Exercise 7 — Tasks

1. Find the list of all buildings that have employees ✓
2. Find the list of all buildings and their capacity ✓
3. List all buildings and the distinct employee roles in each building (including empty buildings)

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Finish above Tasks

Next – SQL Lesson 8: A short note on NULLs  
Previous – SQL Lesson 6: Multi-table queries with JOINs

Find SQLBolt useful? Please consider  
Donating (\$4) via Paypal to support our site.

Table: Buildings (Read-Only)

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

Table: Employees (Read-Only)

Role	Name	Building	Years_employed
Engineer	Becky A.	1e	4
Engineer	Dan B.	1e	2
Engineer	Sharon F.	1e	6
Engineer	Dan M.	1e	4
Engineer	Malcom S.	1e	1
Artist	Tylar S.	2w	2

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_name = building;
```

RESET

#### Exercise 7 — Tasks

1. Find the list of all buildings that have employees ✓
2. Find the list of all buildings and their capacity ✓
3. List all buildings and the distinct employee roles in each building (including empty buildings) ✓

Stuck? Read this task's [Solution](#).

Solve all tasks to continue to the next lesson.

Continue >



Table: Buildings (Read-Only)

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

Table: Employees (Read-Only)

Role	Name	Building	Years_employed
Engineer	Becky A.	1e	4
Engineer	Dan B.	1e	2
Engineer	Sharon F.	1e	6
Engineer	Dan M.	1e	4
Engineer	Malcom S.	1e	1
Artist	Tylar S.	2w	2

Query Results

Name	Role
Yancy I.	Engineer
Oliver P.	Artist

```
SELECT name, role FROM employees
WHERE building IS NULL;
```

RESET

#### Exercise 8 — Tasks

1. Find the name and role of all employees who have not been assigned to a building. ✓
2. Find the names of the buildings that hold no employees

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Finish above Tasks

Table: Buildings (Read-Only)

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

Table: Employees (Read-Only)

Role	Name	Building	Years_employed
Engineer	Becky A.	1e	4
Engineer	Dan B.	1e	2
Engineer	Sharon F.	1e	6
Engineer	Dan M.	1e	4
Engineer	Malcom S.	1e	1
Artist	Tylar S.	2w	2

Query Results

Building_name
1w
2e

```
SELECT DISTINCT building_name
FROM buildings
LEFT JOIN employees
ON building_name = building
WHERE role IS NULL;
```

RESET

Exercise 8 — Tasks

1. Find the name and role of all employees who have not been assigned to a building. ✓
2. Find the names of the buildings that hold no employees. ✓

Stuck? Read this task's [Solution](#).  
Solve all tasks to continue to the next lesson.

Continue >