

LESSION-1:

1. SELECT title FROM movies;
2. SELECT director FROM movies;
3. SELECT title,director FROM movies;
4. SELECT title,year FROM movies;
5. SELECT * FROM movies;

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 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 ›

RESET

LESSION-2

1. SELECT * FROM movies where id=6;
2. SELECT * FROM movies where year between 2000 and 2010;
3. SELECT * FROM movies where year not between 2000 and 2010;
4. SELECT title FROM movies limit 5 offset 0

Table: Movies

Title
Toy Story
A Bug's Life
Toy Story 2
Monsters, Inc.
Finding Nemo

```
SELECT title FROM movies limit 5 offset 0;
```

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 ›

RESET

LESSION-3:

1. SELECT * FROM movies where title like '%Toy Story%';
2. SELECT * FROM movies where director like '%john lasseter%';
3. SELECT * FROM movies where director not like '%john lasseter%';
4. SELECT * FROM movies where title like '%wall%'

Table: Movies

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

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.

RESET

Continue ›

LESSION-4

- 1.SELECT DISTINCT director FROM Movies ORDER BY director;
2. SELECT * FROM movies ORDER BY year DESC LIMIT 4;
3. SELECT * FROM movies ORDER BY title asc LIMIT 5;
4. SELECT * FROM movies ORDER BY title asc LIMIT 5 offset 5;

Table: Movies

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

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.

RESET

Continue ›

LESSION-5:

1. SELECT city, population FROM north_american_cities WHERE country = "Canada";
2. SELECT city,latitude,longitude FROM north_american_cities where country="United States";
3. SELECT * FROM north_american_cities WHERE longitude < - 87.629798 ORDER BY longitude;
4. SELECT city FROM north_american_cities where country ="Mexico"order by population desc limit 2;
5. SELECT city FROM north_american_cities WHERE country = "United States" ORDER BY population DESC LIMIT 2 OFFSET 2;

Table: North_american_cities

City
Chicago
Houston

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 >

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

LESSION-6:

1. SELECT title, domestic_sales, international_sales FROM movies INNER JOIN boxoffice ON movies.id = boxoffice.movie_id;
2. SELECT title, domestic_sales, international_sales FROM movies INNER JOIN boxoffice ON movies.id = boxoffice.movie_id where domestic_sales<international_sales;
3. SELECT title,rating FROM movies INNER JOIN boxoffice ON movies.id = boxoffice.movie_id order by rating desc;

Query Results

Title	Rating
WALL-E	8.5
Toy Story 3	8.4
Toy Story	8.3
Up	8.3
Finding Nemo	8.2
Monsters, Inc.	8.1
Ratatouille	8
The Incredibles	8
Toy Story 2	7.9
Monsters University	7.4

```
SELECT title, rating FROM movies INNER JOIN boxoffice ON movies.id = boxoffice.movie_id order by rating desc;
```

RESET

Exercise 6 — Tasks

- Find the domestic and international sales for each movie ✓
- Show the sales numbers for each movie that did better internationally rather than domestically ✓
- 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.

Continue >

LESSION-7:

1. SELECT distinct building FROM employees;
2. SELECT * FROM buildings;
3. SELECT DISTINCT building_name, role FROM buildings LEFT JOIN employees ON building_name = employees.building;

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 = employees.building;
```

RESET

Exercise 7 — Tasks

- Find the list of all buildings that have employees ✓
- Find the list of all buildings and their capacity ✓
- 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 >

LESSION-8:

1. SELECT NAME,ROLE FROM employees WHERE BUILDING IS NULL;

2. SELECT DISTINCT BUILDING_NAME FROM BUILDINGS LEFT JOIN EMPLOYEES ON BUILDING_NAME = EMPLOYEES.BUILDING WHERE EMPLOYEES.BUILDING IS NULL;

Query Results

Building_name
1w
2e

```
SELECT DISTINCT BUILDING_NAME FROM BUILDINGS LEFT JOIN EMPLOYEES ON BUILDING_NAME = EMPLOYEES.BUILDING WHERE EMPLOYEES.BUILDING IS NULL;
```

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 >

LESSION-9:

1. SELECT title,(domestic_sales+international_sales)/1000000 as combinedsales FROM movies left join Boxoffice on movies.id=boxoffice.movie_id;
2. SELECT title,rating * 10 as present FROM movies left join Boxoffice on movies.id=boxoffice.movie_id;
3. SELECT title,year as year FROM movies left join Boxoffice on movies.id=boxoffice.movie_id where year%2==0;

Query Results

Title	Year
A Bug's Life	1998
The Incredibles	2004
Cars	2006
WALL-E	2008
Toy Story 3	2010
Brave	2012

```
SELECT title,year as year FROM movies left join Boxoffice on movies.id=boxoffice.movie_id where year%2==0;
```

Exercise 9 — Tasks

1. List all movies and their combined sales in **millions** of dollars ✓
2. List all movies and their ratings **in percent** ✓
3. List all movies that were released on even number years ✓

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

Continue >

LESSION-10:

1. SELECT max(years_employed) FROM employees;
2. SELECT role,Avg(years_employed) FROM employees GROUP BY role;
3. SELECT building, sum(years_employed) FROM employees group by building;

Table: Employees

Building	Sum(Years_employed)
1e	29
2w	36

Exercise 10 — Tasks

1. Find the longest time that an employee has been at the studio ✓
2. For each role, find the average number of years employed by employees in that role ✓
3. Find the total number of employee years worked in each building ✓

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

Continue ›

```
SELECT building, sum(years_employed) FROM employees group by building;
```

RESET

LESSION-11:

1. SELECT count(*) FROM employees where role = "Artist";
2. SELECT role,count(*) FROM employees group by role;
3. SELECT sum(years_employed) from employees where role ='Engineer';

Table: Employees

Sum(Years_employed)
17

Exercise 11 — Tasks

1. Find the number of Artists in the studio (without a **HAVING** clause) ✓
2. Find the number of Employees of each role in the studio ✓
3. Find the total number of years employed by all Engineers ✓

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

Continue ›

```
SELECT sum(years_employed) from employees where role ='Engineer';
```

RESET

LESSION-12:

1. `SELECT count(*)title,director FROM movies group by director;`
2. `SELECT director, sum(domestic_sales+international_sales) FROM movies inner join Boxoffice on movies.id = boxoffice.movie_id group by director;`

Query Results

Director	Sum(Domestic_sales+International_sales)
Andrew Stanton	1458055121
Brad Bird	1255164910
Brenda Chapman	538983207
Dan Scanlon	743559607
John Lasseter	2232208025
Lee Unkrich	1063171911
Pete Docter	1294159000

```
SELECT director, sum(domestic_sales+international_sales) FROM movies inner
join Boxoffice on movies.id = boxoffice.movie_id group by director;
```

RESET

Exercise 12 — Tasks

1. Find the number of movies each director has directed ✓
2. Find the total domestic and international sales that can be attributed to each director ✓

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

Continue >

LESSION-13:

1. `insert into Movies (title,director,year,length_minutes) values('Toy Story 4','John Lasseter',' 1999','81');`
2. `insert into Boxoffice (movie_id,rating,domestic_sales,international_sales) values('15','8.7','340000000','270000000')`

Query Results

Movie_id	Rating	Domestic_sales	International_sales
3	7.9	245852179	239163000
1	8.3	191796233	170162503
2	7.2	162798565	200600000
15	8.7	340000000	270000000

```
insert into Boxoffice (movie_id, rating, domestic_sales, international_sales)
values ('15', '8.7', '340000000', '270000000');
```

Exercise 13 — Tasks

1. Add the studio's new production, **Toy Story 4** to the list of movies (you can use any director) ✓
2. 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. ✓

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

RUN QUERY RESET

Continue >

LESSION-14:

1. UPDATE movies SET director = "John Lasseter" WHERE title = "A Bug's Life";
2. UPDATE movies SET year = "1999" WHERE title = "Toy Story 2";
3. UPDATE movies SET title = 'Toy Story 3', director = 'Lee Unkrich' WHERE title = 'Toy Story 8';

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

```
UPDATE movies SET title = 'Toy Story 3', director = 'Lee Unkrich' WHERE
title = 'Toy Story 8';
```

Exercise 14 — Tasks

1. The director for A Bug's Life is incorrect, it was actually directed by **John Lasseter** ✓
2. The year that Toy Story 2 was released is incorrect, it was actually released in **1999** ✓
3. 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** ✓

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

RUN QUERY RESET

Continue >

LESSION-15:

1. Delete FROM movies where year < 2005;
2. DELETE FROM movies WHERE director = 'Andrew Stanton';

Table: Movies

Id	Title	Director	Year	Length_minutes
----	-------	----------	------	----------------

Exercise 15 — Tasks

1. This database is getting too big, lets remove all movies that were released **before** 2005. ✓
2. Andrew Stanton has also left the studio, so please remove all movies directed by him. ✓

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

```
DELETE FROM movies WHERE director = 'Andrew Stanton';
```

RUN QUERY

RESET

Continue >

LESSION-16:

1. create table Database(
 Name varchar(255),
 Version int,
 Download_count int
)

Table: Database

Name	Version	Download_count
SQLite	3.9	92000000
MySQL	5.5	512000000
Postgres	9.4	384000000

Exercise 16 — Tasks

1. 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. ✓

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

```
create table Database(  
  Name varchar(255),  
  Version int,  
  Download_count int  
)
```

RUN QUERY

RESET

Continue >

LESSION-17:

1. alter table movies

add aspect_ratio float;

2. alter table movies

add Language varchar(255) default 'English';

Table: Movies

Id	Title	Director	Year	Length_minutes	Aspect_ratio	Language
1	Toy Story	John Lasseter	1995	81		English
2	A Bug's Life	John Lasseter	1998	95		English
3	Toy Story 2	John Lasseter	1999	93		English
4	Monsters, Inc.	Pete Docter	2001	92		English
5	Finding Nemo	Andrew Stanton	2003	107		English
6	The Incredibles	Brad Bird	2004	116		English
7	Cars	John Lasseter	2006	117		English
8	Ratatouille	Brad Bird	2007	115		English
9	WALL-E	Andrew Stanton	2008	104		English
10	Up	Pete Docter	2009	101		English

```
alter table movies
  add Language varchar(255) default 'English';
```

[RUN QUERY](#) [RESET](#)

Exercise 17 — Tasks

1. Add a column named **Aspect_ratio** with a **FLOAT** data type to store the aspect-ratio each movie was released in. ✓
2. 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**. ✓

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

[Continue >](#)

LESSION-18:

1. drop table movies;

2. drop table Boxoffice;

Query Results

Id	Title	Director	Year	Length_minutes

```
drop table Boxoffice;
```

[RUN QUERY](#) [RESET](#)

Exercise 18 — Tasks

1. We've sadly reached the end of our lessons, lets clean up by removing the **Movies** table
✓
2. And drop the **BoxOffice** table as well ✓

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

[Continue >](#)