SELECT queries 101

Exercise 1 - Tasks

1. Find the title of each film.

SQL Query: `SELECT title FROM movies;`

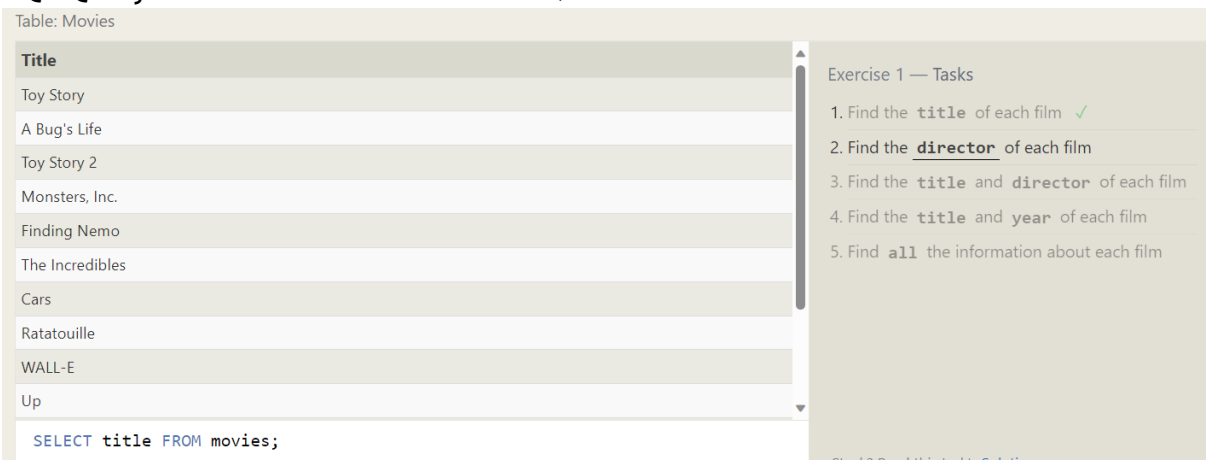


Table: Movies

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;
```

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

2. Find the director of each film.

SQL Query: `SELECT director FROM movies;`

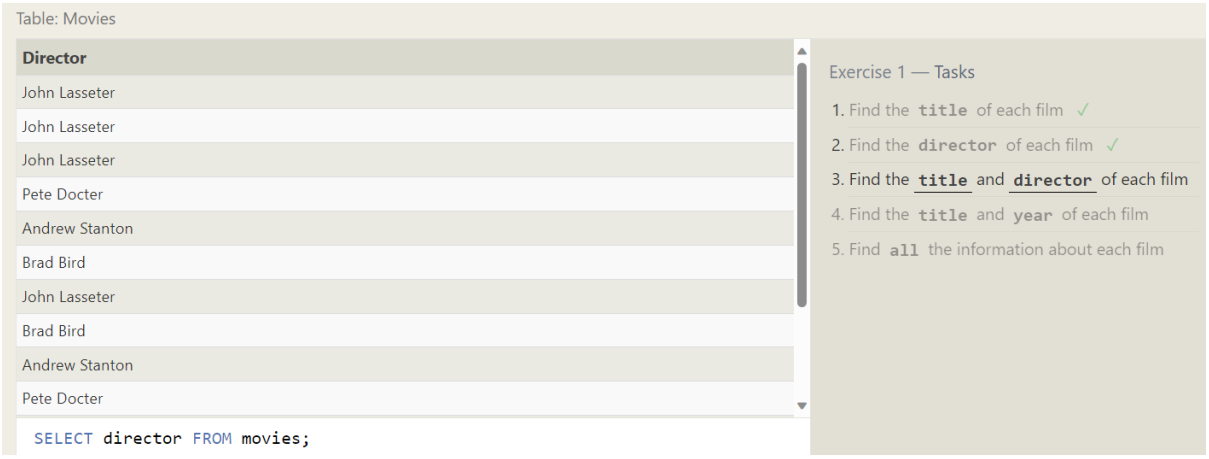


Table: Movies

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;
```

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

3. Find the title and director of each film.

SQL Query: `SELECT title, director FROM movies;`

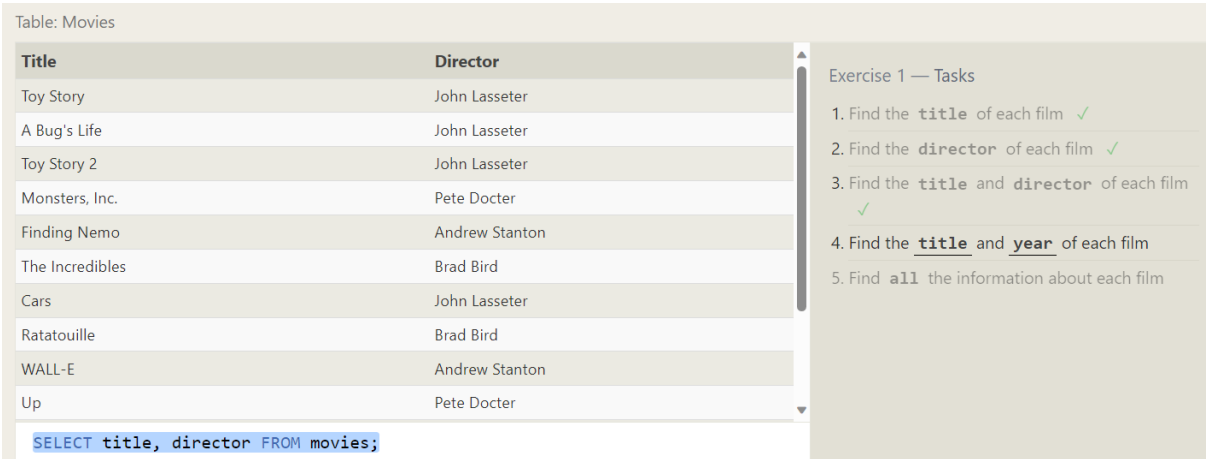


Table: Movies

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;
```

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

4. Find the title and year of each film.

SQL Query: `SELECT title, year FROM movies;`

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;
```

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

5. Find all the information about each film.

SQL Query: `SELECT title, director 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;
```

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 ›

Queries with constraints

Exercise 2 - Tasks

1. Find the movie with a row id of 6

SQL Query: `SELECT * FROM movies WHERE id = 6;`

Table: Movies

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

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

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`

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

SQL Query: `SELECT * FROM movies WHERE year BETWEEN 2000 AND 2010;`

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;
```

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`

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

SQL Query: `SELECT * FROM movies WHERE year NOT 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

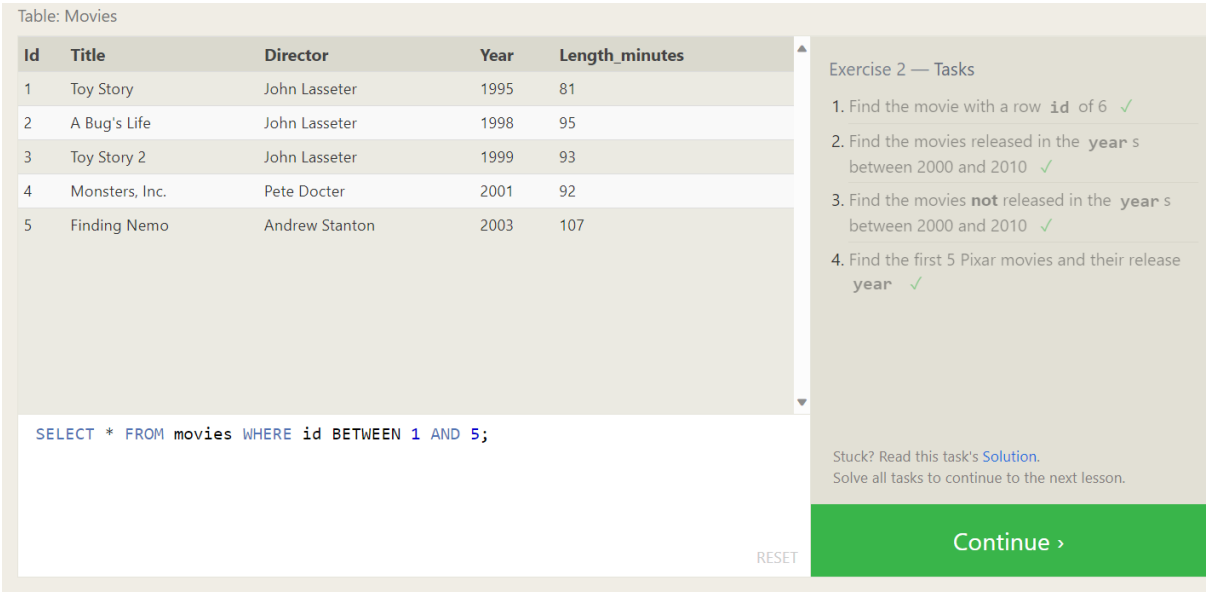
```
SELECT * FROM movies WHERE year NOT BETWEEN 2000 AND 2010;
```

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`

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

SQL Query: `SELECT * FROM movies WHERE id BETWEEN 1 and 5;`



Queries with constraints (Pt. 2)

Exercise 3 - Tasks

1. Find all the Toy Story movies.

SQL Query: `SELECT * FROM movies WHERE title LIKE "%Toy Story%";`

2. Find all the movies directed by John Lasseter.

SQL Query: `SELECT * FROM movies WHERE director LIKE "%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%";
```

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

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

SQL Query: `SELECT title, director FROM movies WHERE director NOT LIKE "%John Lasseter%";`

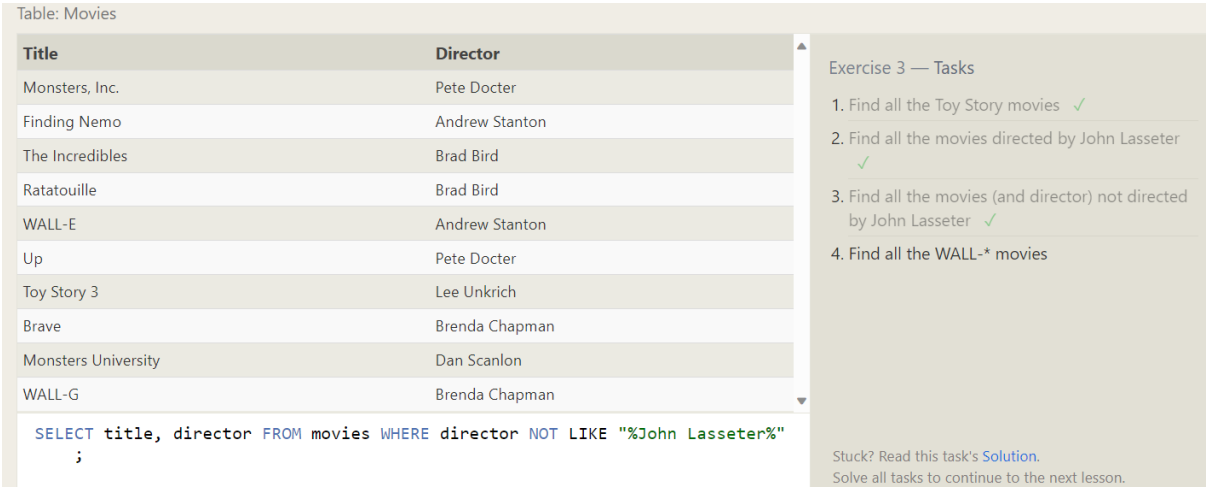


Table: Movies

Title	Director
Monsters, Inc.	Pete Docter
Finding Nemo	Andrew Stanton
The Incredibles	Brad Bird
Ratatouille	Brad Bird
WALL-E	Andrew Stanton
Up	Pete Docter
Toy Story 3	Lee Unkrich
Brave	Brenda Chapman
Monsters University	Dan Scanlon
WALL-G	Brenda Chapman

```
SELECT title, director FROM movies WHERE director NOT LIKE "%John Lasseter%"
    ;
```

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.

4. Find all the WALL-* movies

SQL Query: `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

```
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 ›

Filtering and sorting Query results

Exercise 4 - Tasks

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

SQL Query: `SELECT DISTINCT director FROM movies ORDER BY director ASC;`

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;
```

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

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

SQL Query: `SELECT * FROM movies ORDER BY year DESC LIMIT 4;`

Table: Movies

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

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

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

3. List the first five Pixar movies sorted alphabetically

SQL Query: `SELECT * FROM movies ORDER BY title ASC LIMIT 5;`

Table: Movies

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

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

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

4. List the next five Pixar movies sorted alphabetically

SQL Query: `SELECT * FROM movies ORDER BY title ASC LIMIT 5 OFFSET 5;`

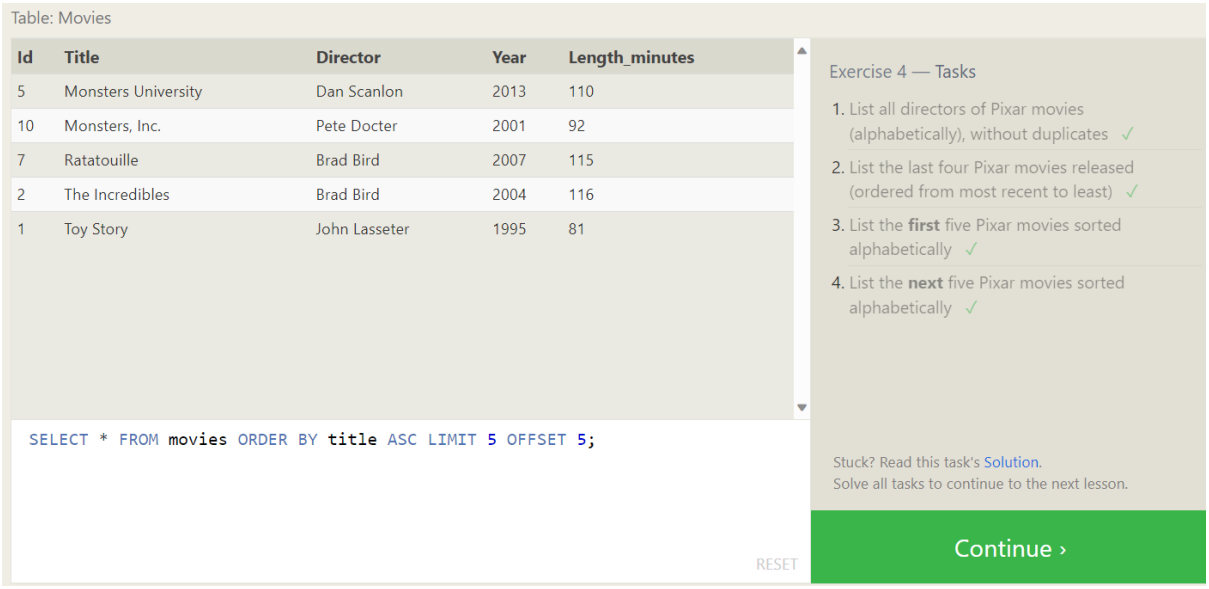


Table: Movies

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

```
SELECT * FROM movies ORDER BY title ASC LIMIT 5 OFFSET 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.

Continue ›

Simple SELECT Queries

Exercise 5 - Tasks

1. List all the Canadian cities and their populations

SQL Query: `SELECT city,population FROM north_american_cities WHERE country ="Canada";`

Table: North_american_cities

City	Population
Toronto	2795060
Montreal	1717767

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

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

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

SQL Query:
```
SELECT city FROM north_american_cities WHERE country ="United States" ORDER BY latitude DESC;
```

Table: North_american_cities

City
Chicago
New York
Philadelphia
Los Angeles
Phoenix
Houston

```
SELECT city FROM north_american_cities WHERE country ="United States" ORDER
    BY latitude DESC;
```

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.

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

SQL Query :
```
SELECT city, longitude FROM north_american_cities
WHERE longitude < -87.629798
ORDER BY longitude ASC;
```

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;
```

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.

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

SQL Query :
```
SELECT * FROM north_american_cities WHERE country ="Mexico"
ORDER BY Population desc limit 2;
```

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;
```

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.

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

SQL Query:
```
SELECT * FROM north_american_cities WHERE country ="United States" ORDER BY Population desc limit 2 OFFSET 2;
```

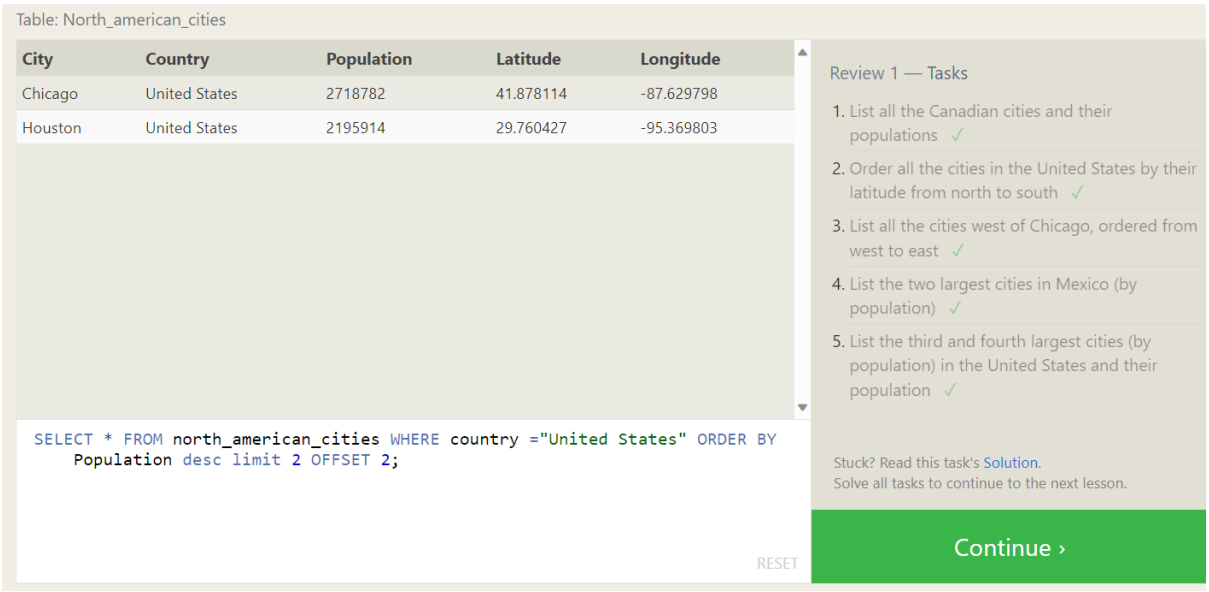


Table: North_american_cities

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;
```

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 ›

Simple SELECT Queries

Exercise 6 - Tasks

1. Find the domestic and international sales for each movie

SQL Query:
```
SELECT title, domestic_sales, international_sales
FROM movies
  JOIN boxoffice
    ON movies.id = boxoffice.movie_id;
```

Query Results

Title	Domestic_sales	International_sales
Finding Nemo	380843261	555900000
Monsters University	268492764	475066843
Ratatouille	206445654	417277164
Cars 2	191452396	368400000
Toy Story 2	245852179	239163000
The Incredibles	261441092	370001000
WALL-E	223808164	297503696
Toy Story 3	415004880	648167031
Toy Story	191796233	170162503
Cars	244082982	217900167

```
SELECT title, domestic_sales, international_sales
FROM movies
  JOIN boxoffice
    ON movies.id = boxoffice.movie_id;
```

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.

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

SQL Query: `SELECT title, domestic_sales, international_sales`

```
FROM movies
  JOIN boxoffice
    ON movies.id = boxoffice.movie_id
    WHERE international_sales >  domestic_sales;
```

Query Results

Title	Domestic_sales	International_sales
Finding Nemo	380843261	555900000
Monsters University	268492764	475066843
Ratatouille	206445654	417277164
Cars 2	191452396	368400000
The Incredibles	261441092	370001000
WALL-E	223808164	297503696
Toy Story 3	415004880	648167031
Up	293004164	438338580
A Bug's Life	162798565	200600000
Brave	237283207	301700000

```
SELECT title, domestic_sales, international_sales
FROM movies
  JOIN boxoffice
    ON movies.id = boxoffice.movie_id
    WHERE international_sales >  domestic_sales;
```

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.

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

SQL Query: `SELECT title, director, year, rating`

```
FROM movies
  JOIN boxoffice
    ON movies.id = boxoffice.movie_id
    ORDER BY rating desc
```

Query Results

Title	Director	Year	Rating
WALL-E	Andrew Stanton	2008	8.5
Toy Story 3	Lee Unkrich	2010	8.4
Toy Story	John Lasseter	1995	8.3
Up	Pete Docter	2009	8.3
Finding Nemo	Andrew Stanton	2003	8.2
Monsters, Inc.	Pete Docter	2001	8.1
Ratatouille	Brad Bird	2007	8
The Incredibles	Brad Bird	2004	8
Toy Story 2	John Lasseter	1999	7.9
Monsters University	Dan Scanlon	2013	7.4

```
SELECT title, director, year, rating
FROM movies
  JOIN boxoffice
    ON movies.id = boxoffice.movie_id
    ORDER BY rating desc
```

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.

Continue ›

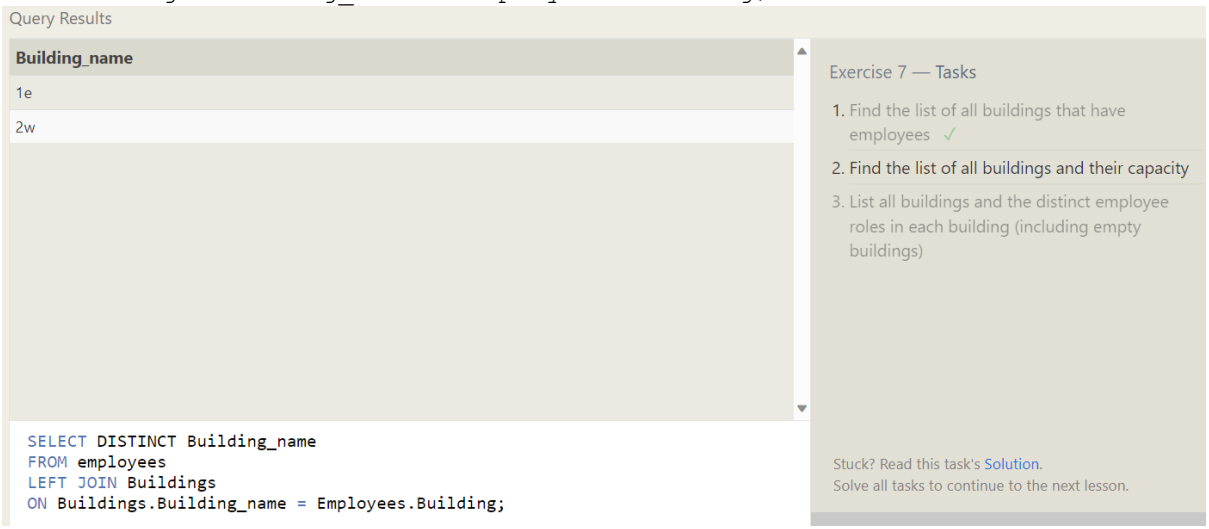
OUTER JOINs

Exercise 7 - Tasks

1. Find the list of all buildings that have employees

SQL Query: `SELECT DISTINCT Building_name`

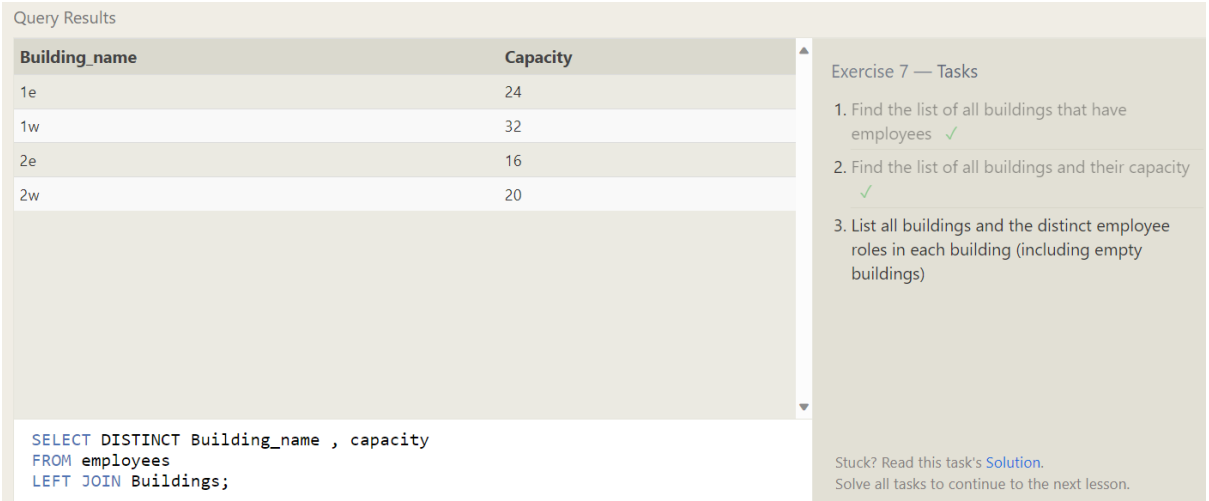
```
FROM employees
LEFT JOIN Buildings
ON Buildings.Building_name = Employees.Building;
```



2. Find the list of all buildings and their capacity

SQL Query: `SELECT DISTINCT Building_name , capacity`

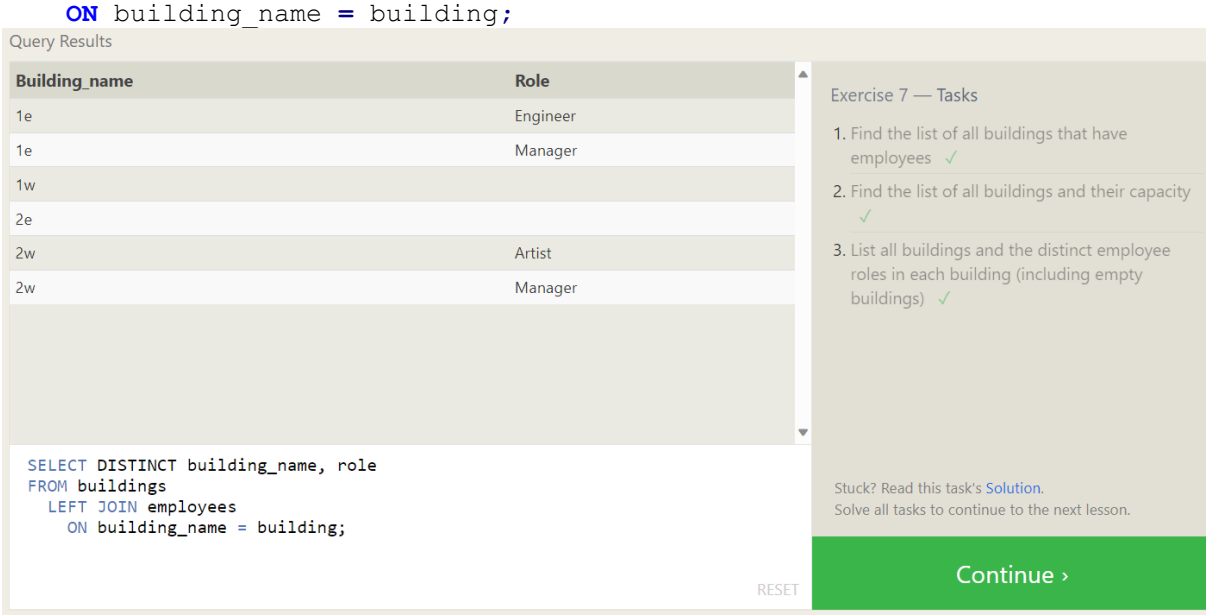
```
FROM employees
LEFT JOIN Buildings;
```



3. List all buildings and the distinct employee roles in each building (including empty buildings)

SQL Query: `SELECT DISTINCT building_name, role`

```
FROM buildings
  LEFT JOIN employees
    ON building_name = building;
```



A short note on NULLs

Exercise 8 - Tasks

1. Find the name and role of all employees who have not been assigned to a building

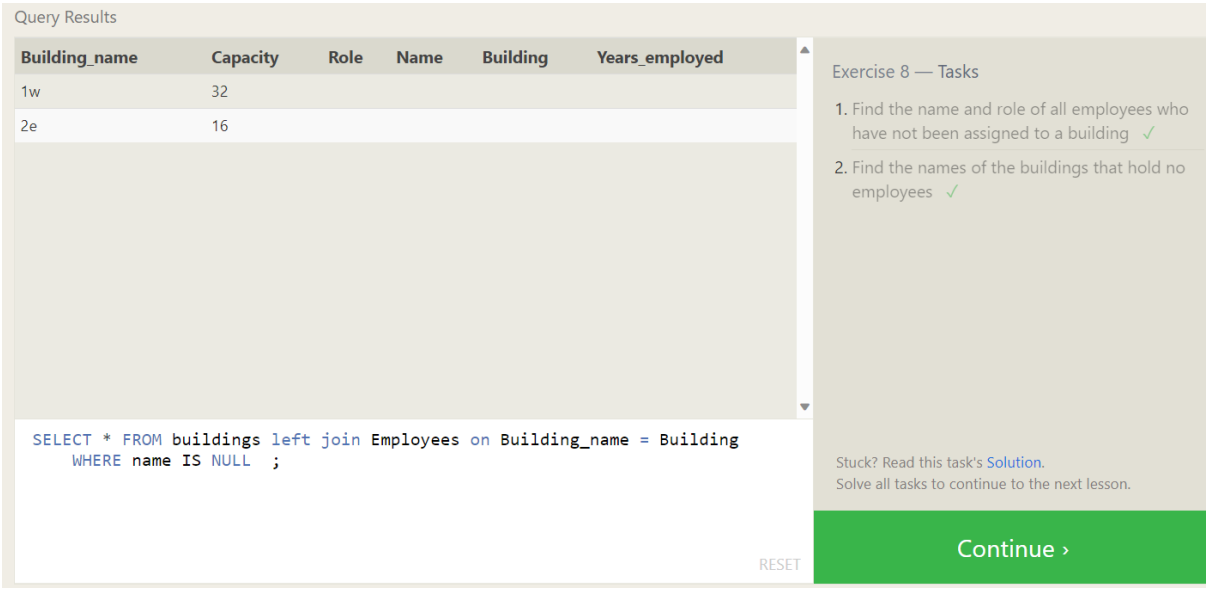
SQL Query: `SELECT name, role`

```
FROM employees
Where building IS NULL;
```

2. Find the names of the buildings that hold no employees

SQL Query: `SELECT * FROM buildings`

```
left join Employees
on Building_name = Building
WHERE name IS NULL  ;
```

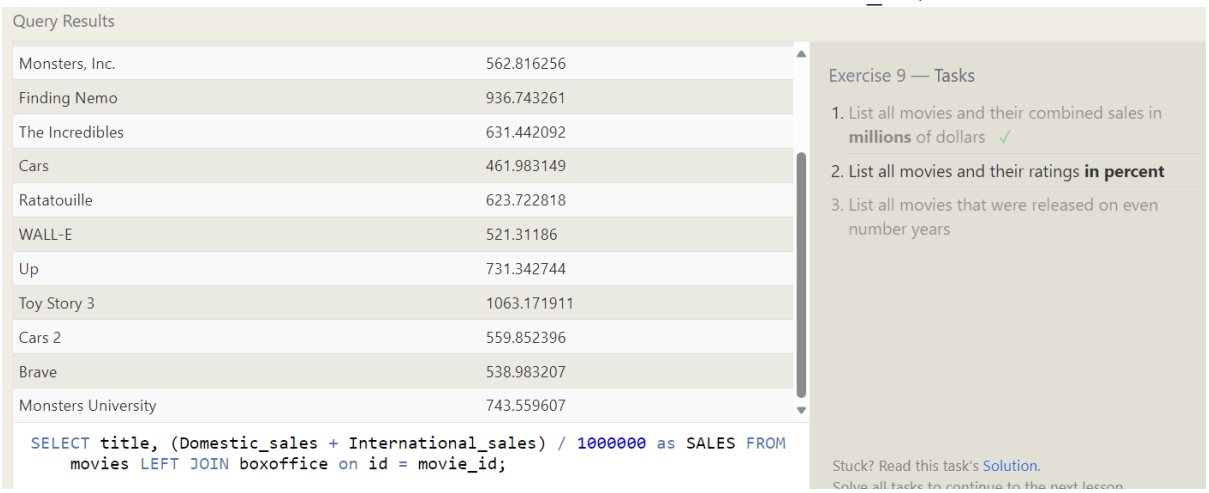


Queries with expressions

Exercise 9 - Tasks

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

SQL Query: `SELECT title, (Domestic_sales + International_sales) / 1000000 as SALES FROM movies LEFT JOIN boxoffice on id = movie_id;`



2. List all movies and their ratings in percent

SQL Query: `SELECT title, rating, (rating *10) AS percent FROM movies LEFT JOIN BoxOffice ON id = movie_id;`

Query Results

Title	Rating	Percent
Toy Story	8.3	83
A Bug's Life	7.2	72
Toy Story 2	7.9	79
Monsters, Inc.	8.1	81
Finding Nemo	8.2	82
The Incredibles	8	80
Cars	7.2	72
Ratatouille	8	80
WALL-E	8.5	85
Up	8.3	83

```
SELECT title, rating, (rating *10) AS percent FROM movies LEFT JOIN
    BoxOffice ON id = movie_id;
```

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.

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

SQL Query: `SELECT title, director, year FROM movies WHERE year%2 = 0;`

Query Results

Title	Director	Year
A Bug's Life	John Lasseter	1998
The Incredibles	Brad Bird	2004
Cars	John Lasseter	2006
WALL-E	Andrew Stanton	2008
Toy Story 3	Lee Unkrich	2010
Brave	Brenda Chapman	2012

```
SELECT title, director, year FROM movies WHERE year%2 = 0;
```

RESET

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 ›

Queries with aggregates (Pt. 1)

Exercise 10 - Tasks

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

SQL Query: `SELECT role, name, MAX(Years_Employed) as Longest_Employed FROM employees;`

Table: Employees

Role	Name	Longest_Employed
Manager	Scott K.	9

```
SELECT role, name, MAX(Years_Employed) as Longest_Employed FROM employees;
```

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.

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

SQL Query: `SELECT role, avg(years_employed) as Average_Year FROM employees group by role;`

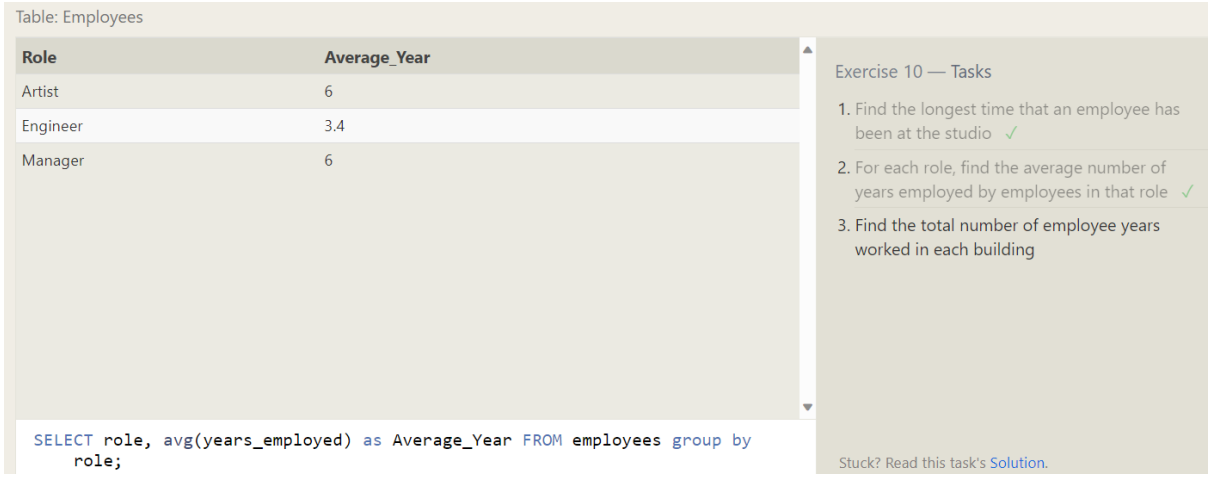


Table: Employees

Role	Average_Year
Artist	6
Engineer	3.4
Manager	6

```
SELECT role, avg(years_employed) as Average_Year FROM employees group by
    role;
```

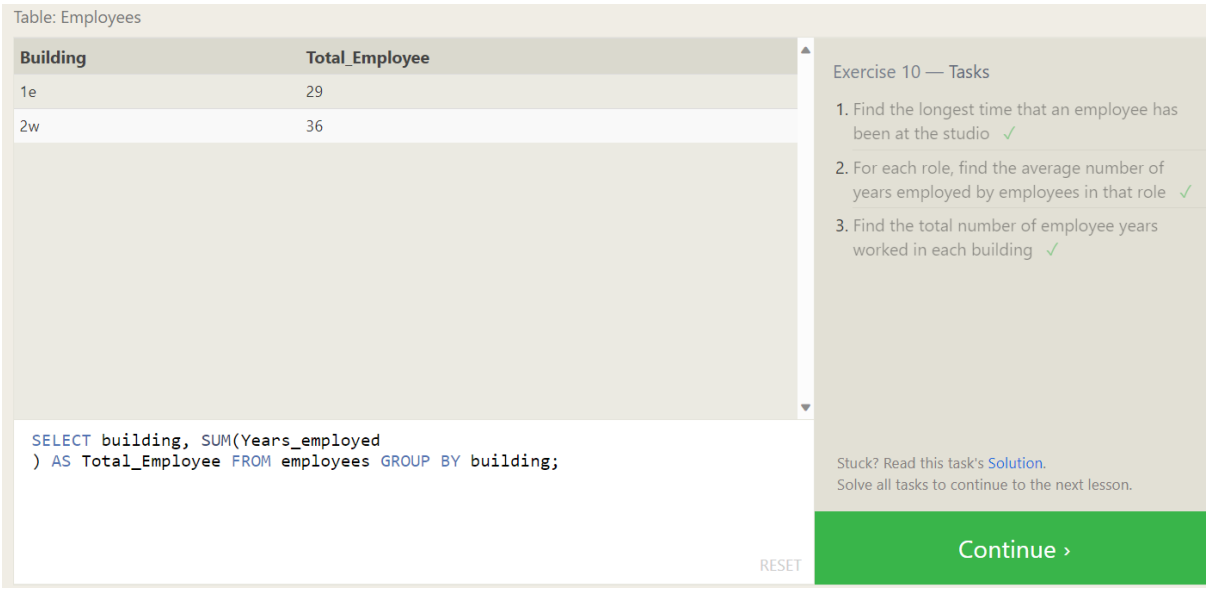
Exercise 10 — Tasks

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.

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

SQL Query: `SELECT building, SUM(Years_employed) AS Total_Employee FROM employees GROUP BY building;`



Queries with aggregates (Pt. 2)

Exercise 11 - Tasks

1. Find the number of Artists in the studio (without a HAVING clause)

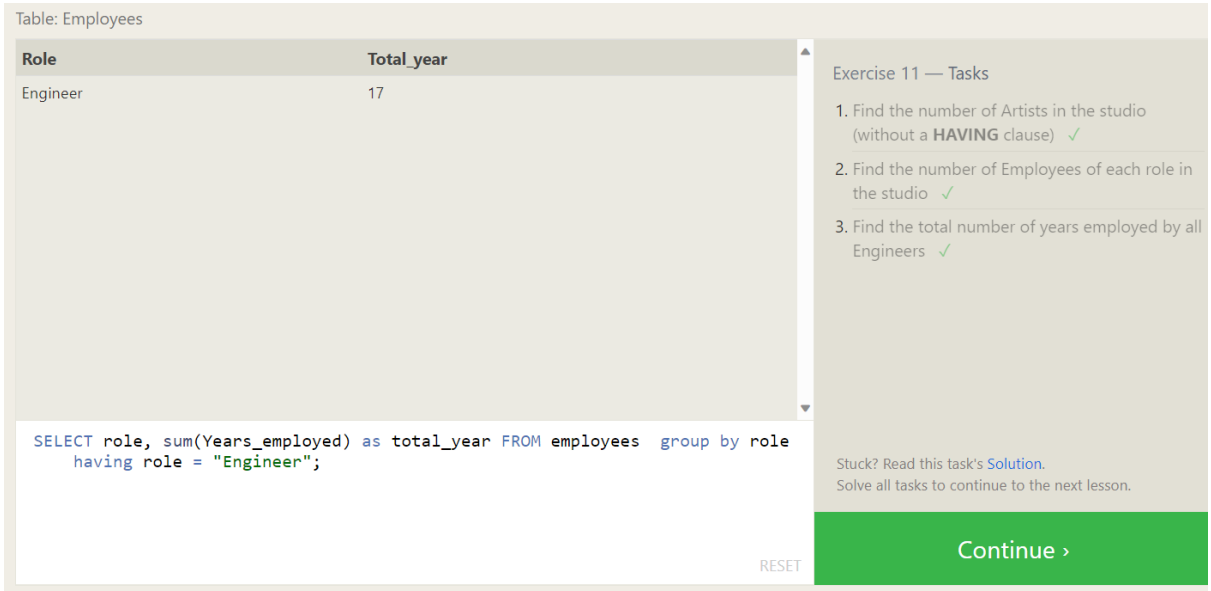
SQL Query: `SELECT role, count(role) as no_of_artist FROM employees where role = "Artist";`

2. Find the number of Employees of each role in the studio

SQL Query: `SELECT role, count(name) as No_Of_Employee FROM employees group by role;`

3. Find the total number of years employed by all Engineers

SQL Query: `SELECT role, sum(Years_employed) as total_year FROM employees group by role having role = "Engineer";`

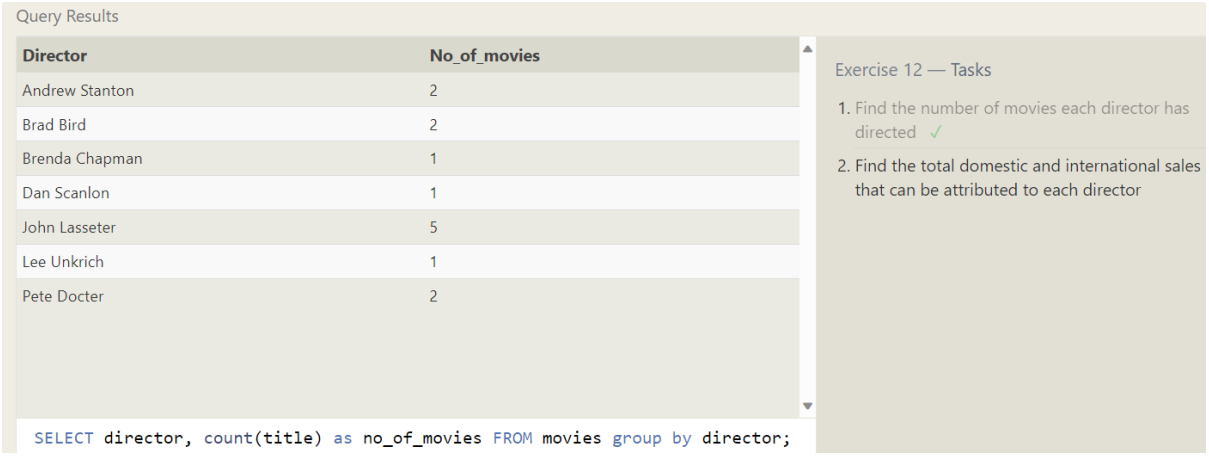


Order of execution of a Query

Exercise 12 - Tasks

1. Find the number of movies each director has directed.

SQL Query:
```
SELECT director, count(title) as no_of_movies FROM movies group by director;
```



Query Results

Director	No_of_movies
Andrew Stanton	2
Brad Bird	2
Brenda Chapman	1
Dan Scanlon	1
John Lasseter	5
Lee Unkrich	1
Pete Docter	2

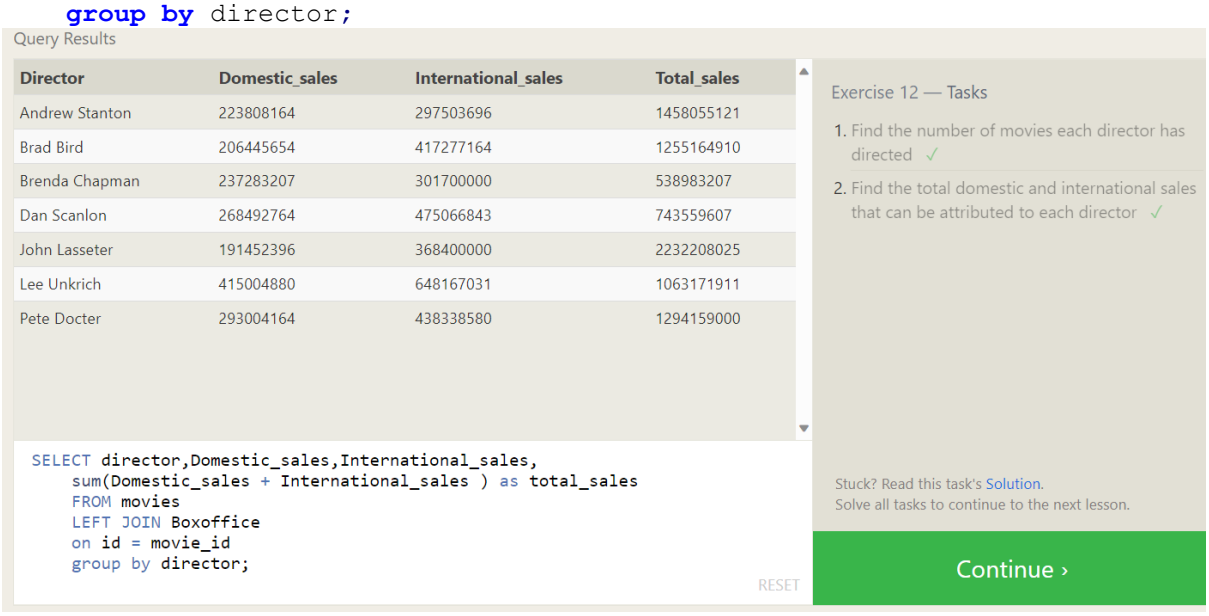
```
SELECT director, count(title) as no_of_movies FROM movies group by director;
```

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

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

SQL Query:
```
SELECT director,Domestic_sales,International_sales,
    sum(Domestic_sales + International_sales ) as total_sales
    FROM movies
    LEFT JOIN Boxoffice
    on id = movie_id
    group by director;
```



Query Results

Director	Domestic_sales	International_sales	Total_sales
Andrew Stanton	223808164	297503696	1458055121
Brad Bird	206445654	417277164	1255164910
Brenda Chapman	237283207	301700000	538983207
Dan Scanlon	268492764	475066843	743559607
John Lasseter	191452396	368400000	2232208025
Lee Unkrich	415004880	648167031	1063171911
Pete Docter	293004164	438338580	1294159000

```
SELECT director,Domestic_sales,International_sales,
    sum(Domestic_sales + International_sales ) as total_sales
    FROM movies
    LEFT JOIN Boxoffice
    on id = 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 ›

Inserting rows

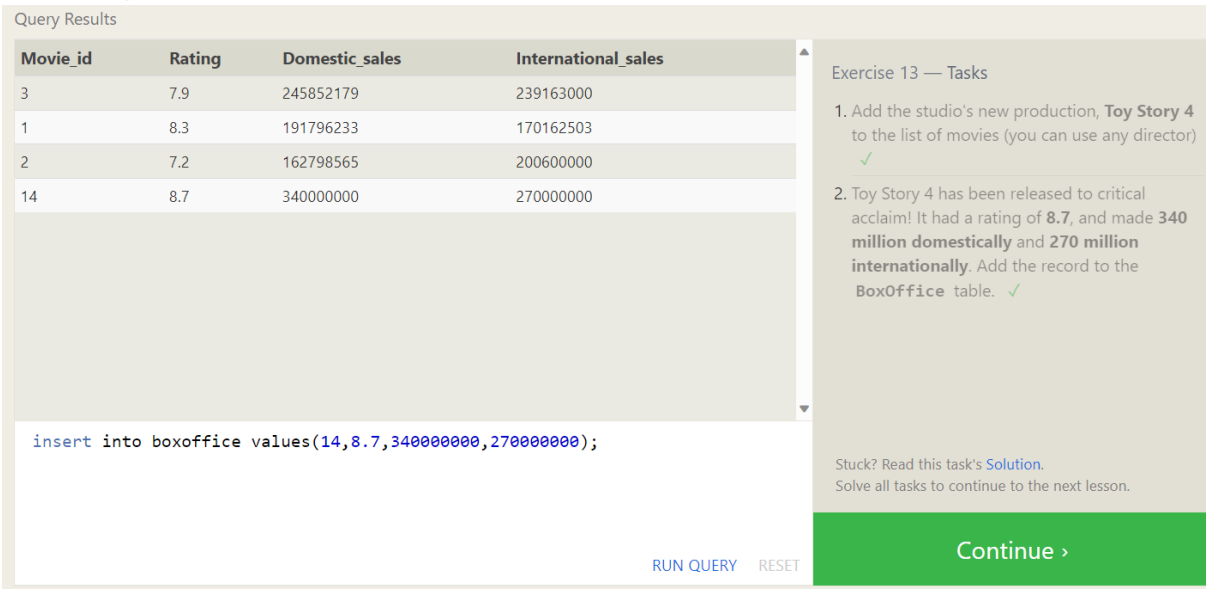
Exercise 13 - Tasks

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

SQL Query: `INSERT INTO movies values(14,"Toy Story 4","John Lasseter",2020,120);`

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.

SQL Query: `insert into boxoffice values(14,8.7,340000000,270000000);`



Updating rows

Exercise 14 - Tasks

1. The director for A Bug's Life is incorrect, it was actually directed by John Lasseter.

SQL Query: `UPDATE movies set director = "John Lasseter" where id = 2 ;`

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	1899	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 director = "John Lasseter" where id = 2 ;
```

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.

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

SQL Query: `UPDATE movies set year = 1999 where title = "Toy Story 2";`

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 year = 1999 where title = "Toy Story 2";
```

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**

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

SQL Query: `UPDATE movies set title = "Toy Story 3", director = "Lee Unkrich" where id = 11;`

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

```
UPDATE movies set title = "Toy Story 3", director = "Lee Unkrich" where id =
    11;
```

RUN QUERY RESET

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.

Continue ›

Deleting rows

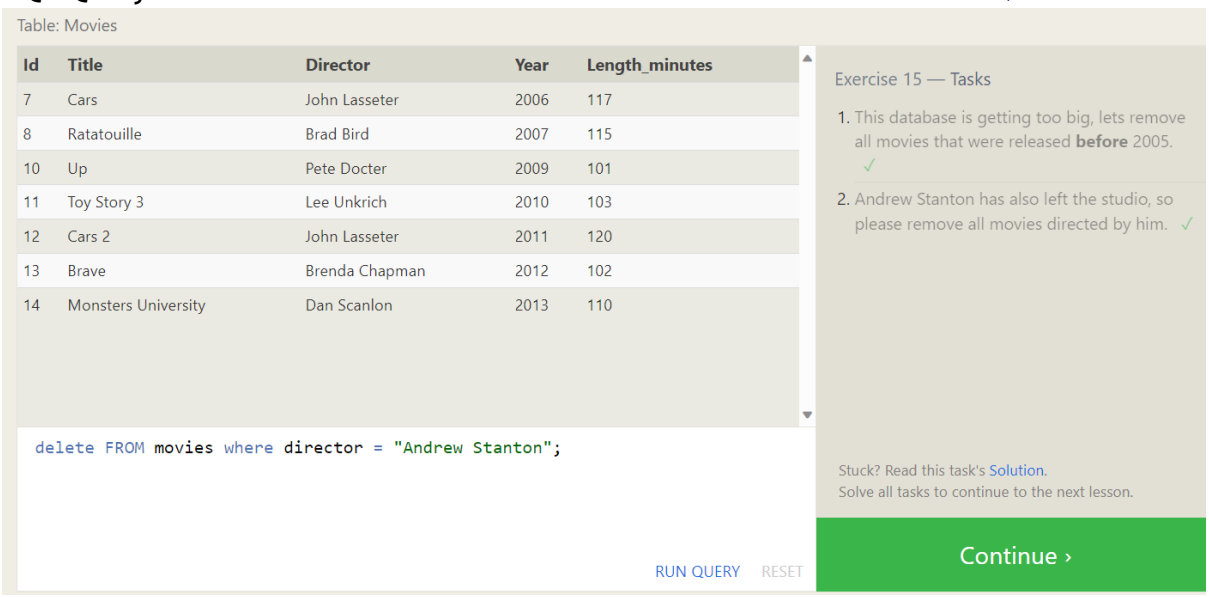
Exercise 15 - Tasks

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

SQL Query: `delete FROM movies where year < 2005;`

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

SQL Query: `delete FROM movies where director = "Andrew Stanton";`



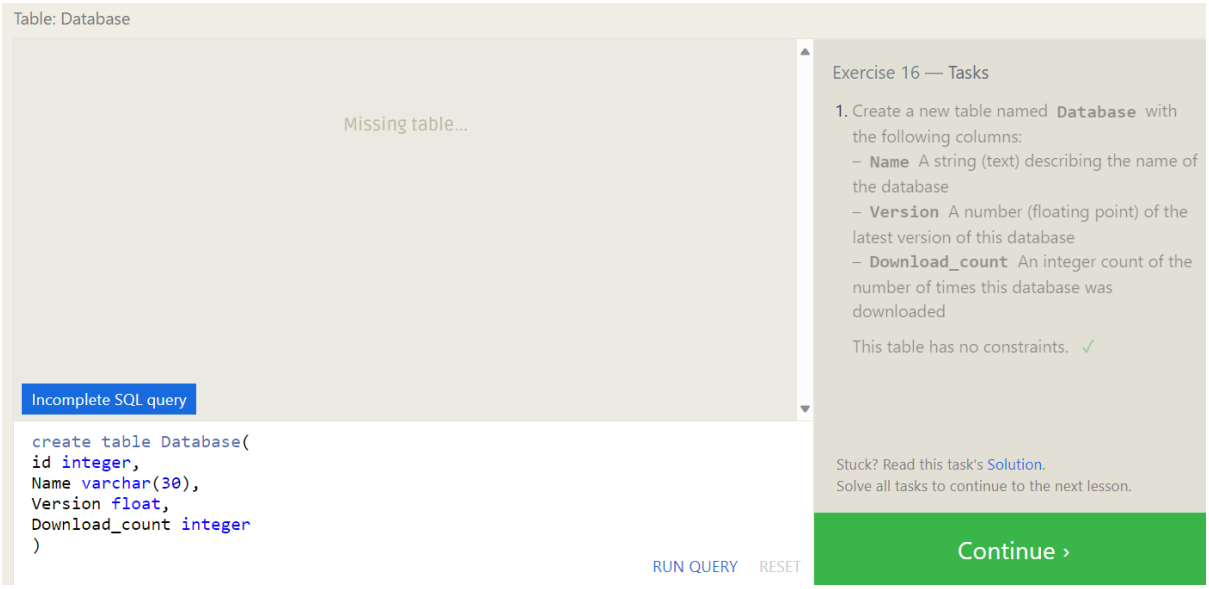
Creating tables

Exercise 16 - Tasks

1. Create a new table named Database with the following columns:
    a. - Name A string (text) describing the name of the database
    b. - Version A number (floating point) of the latest version of this database
    c. - Download_count An integer count of the number of times this database was downloaded
    d. This table has no constraints.

SQL Query: `create table Database(`

```
id integer,
Name varchar(30),
Version float,
Download_count integer
)
```

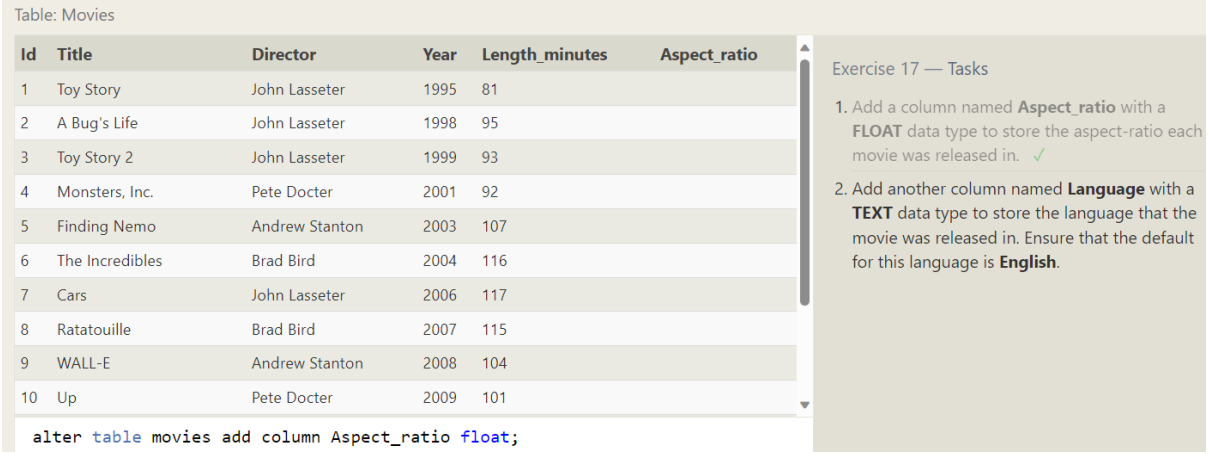


Altering tables

Exercise 17 - Tasks

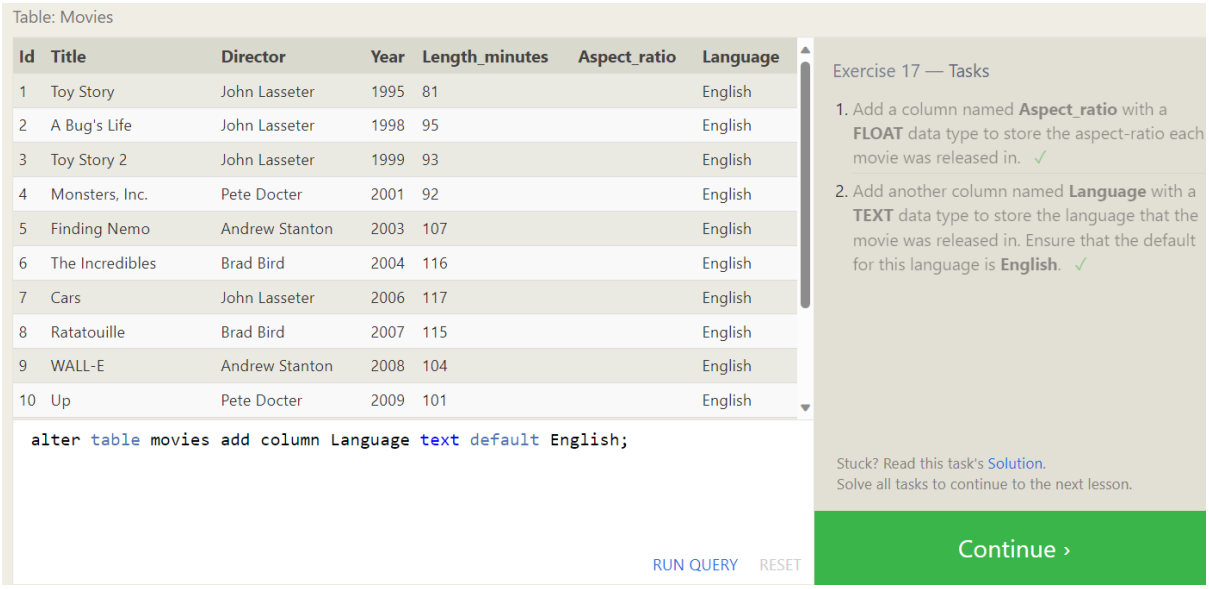
1. Add a column named Aspect_ratio with a FLOAT data type to store the aspect-ratio each movie was released in.

SQL Query: `alter table movies add column Aspect_ratio float;`



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.

SQL Query: `alter table movies add column Language text default English;`

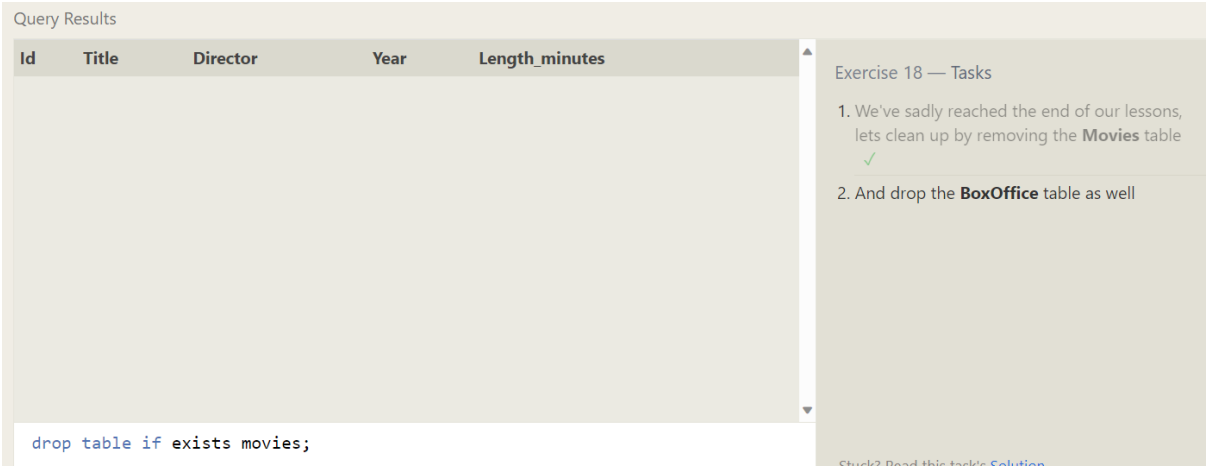


Dropping tables

Exercise 18 - Tasks

1. We've sadly reached the end of our lessons, lets clean up by removing the Movies table.

SQL Query: `drop table if exists movies;`



2. And drop the BoxOffice table as well

SQL Query: `drop table if exists boxoffice;`