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

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

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

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

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

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;
```

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

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

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;
```

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

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

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 ›

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=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

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

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

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

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 limit 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

Table: Movies

Title
Toy Story
Toy Story 2
Toy Story 3

```
SELECT title FROM movies WHERE title LIKE "Toy Story%";
```

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)

Table: Movies

Title
Toy Story
A Bug's Life
Toy Story 2
Cars
Cars 2

```
SELECT title FROM movies WHERE director = "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

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

Table: Movies

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

```
SELECT title FROM movies WHERE director != "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

Table: Movies

Title
WALL-E
WALL-G

```
SELECT title 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)

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;
```

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)

might see in real life. Try and use the necessary keywords and clauses introduced above in your queries.

Table: Movies

Title
Monsters University
Brave
Cars 2
Toy Story 3

```
SELECT DISTINCT title 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)

might see in real life. Try and use the necessary keywords and clauses introduced above in your queries.

Table: Movies

Title
A Bug's Life
Brave
Cars
Cars 2
Finding Nemo

```
SELECT title 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)

might see in real life. Try and use the necessary keywords and clauses introduced above in your queries.

Table: Movies

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

```
SELECT title FROM movies ORDER BY title 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 ›

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

Table: North_american_cities

City	Population
Toronto	2795060
Montreal	1717767

```
SELECT city, 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

Next – SQL Lesson 6: Multi-table queries with JOINs
Previous – SQL Lesson 4: Filtering and sorting Query results

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;
```

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

Table: North_american_cities

City
Los Angeles
Phoenix
Guadalajara
Mexico City
Ecatepec de Morelos
Houston

```
SELECT city
FROM north_american_cities
WHERE longitude < -87.629798
ORDER BY longitude;
```

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

Table: North_american_cities

City
Mexico City
Ecatepec de Morelos

```
SELECT city
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

Next – SQL Lesson 6: Multi-table queries with JOINs
Previous – SQL Lesson 4: Filtering and sorting Query results

Table: North_american_cities

City
Chicago
Houston

```
SELECT city
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 ›

Next – SQL Lesson 6: Multi-table queries with JOINs
Previous – SQL Lesson 4: Filtering and sorting Query results

6	The Incredibles	Brad Bird	2004	116	6	8	261441092	370001000

Query Results

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
Up	293004164	438338580
Monsters, Inc.	289916256	272900000
A Bug's Life	162798565	200600000
Brave	237283207	301700000

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

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

6	The Incredibles	Brad Bird	2004	116	6	8	261441092	370001000

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
INNER JOIN boxoffice
    ON movies.id = boxoffice.movie_id
WHERE international_sales > domestic_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

6	The Incredibles	Brad Bird	2004	116	6	8	261441092	370001000

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

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 ›

2w	20

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

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

2w	20

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 = employees.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 ›

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

Next – SQL Lesson 9: Queries with expressions
Previous – SQL Lesson 7: OUTER JOINs

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;
```

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 ›

Next – SQL Lesson 9: Queries with expressions
Previous – SQL Lesson 7: OUTER JOINs

6	The Incredibles	Brad Bird	2004	116	6	8	261441092	370001000

Query Results

Title	Sales
Finding Nemo	936.743261
Monsters University	743.559607
Ratatouille	623.722818
Cars 2	559.852396
Toy Story 2	485.015179
The Incredibles	631.442092
WALL-E	521.31186
Toy Story 3	1063.171911
Toy Story	361.958736
Cars	461.983149

```
SELECT DISTINCT
    title,
    (domestic_sales + international_sales) / 1000000 AS sales
FROM movies
INNER JOIN boxoffice
    ON movies.id = boxoffice.movie_id;
```

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.

Finish above Tasks

Next – SQL Lesson 10: Queries with aggregates (Pt. 1)
Previous – SQL Lesson 8: A short note on NULLs

6	The Incredibles	Brad Bird	2004	116	6	8	261441092	370001000

Query Results

Title	Rate_percent
Finding Nemo	82
Monsters University	74
Ratatouille	80
Cars 2	64
Toy Story 2	79
The Incredibles	80
WALL-E	85
Toy Story 3	84
Toy Story	83
Cars	72

```
SELECT DISTINCT
    title,
    (rating * 10) AS rate_percent
FROM movies
INNER JOIN boxoffice
    ON movies.id = boxoffice.movie_id;
```

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.

Finish above Tasks

Next – SQL Lesson 10: Queries with aggregates (Pt. 1)
Previous – SQL Lesson 8: A short note on NULLs

6	The Incredibles	Brad Bird	2004	116	6	8	261441092	370001000

Query Results

Title
A Bug's Life
The Incredibles
Cars
WALL-E
Toy Story 3
Brave

```
SELECT title 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 ›

metrics about the teams. Go ahead and give it a shot.

Table: Employees

Name	MAX(Years_employed)
Scott K.	9

```
SELECT name, MAX(years_employed) FROM employees;
```

RESET

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.

Finish above Tasks

Next – SQL Lesson 11: Queries with aggregates (Pt. 2)
Previous – SQL Lesson 9: Queries with expressions

metrics about the teams. Go ahead and give it a shot.

Table: Employees

Role	Average_years_employed
Artist	6
Engineer	3.4
Manager	6

```
SELECT role, AVG(years_employed) as Average_years_employed
  FROM employees
  GROUP BY role;
```

RESET

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.

Finish above Tasks

Next – SQL Lesson 11: Queries with aggregates (Pt. 2)
Previous – SQL Lesson 9: Queries with expressions

metrics about the teams. Go ahead and give it a shot.

Table: Employees

Building	SUM(Years_employed)
1e	29
2w	36

```
SELECT building, SUM(years_employed) FROM employees GROUP BY building
```

RESET

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 ›

Next – SQL Lesson 11: Queries with aggregates (Pt. 2)
Previous – SQL Lesson 9: Queries with expressions

COUNT(*)
5

```
SELECT COUNT(*) FROM employees WHERE role LIKE 'artist';
```

RESET

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.

Finish above Tasks

Next – SQL Lesson 12: Order of execution of a Query
Previous – SQL Lesson 10: Queries with aggregates (Pt. 1)

Role	COUNT(Name)
Artist	5
Engineer	5
Manager	3

```
SELECT role, COUNT(name) FROM employees GROUP BY role;
```

RESET

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.

Finish above Tasks

Next – SQL Lesson 12: Order of execution of a Query
Previous – SQL Lesson 10: Queries with aggregates (Pt. 1)

Role	SUM(Years_employed)
Engineer	17

```
SELECT role, SUM(years_employed) FROM employees
GROUP BY role HAVING role LIKE 'engineer';
```

RESET

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 ›

Next – SQL Lesson 12: Order of execution of a Query
Previous – SQL Lesson 10: Queries with aggregates (Pt. 1)

Query Results

Director	COUNT(*)
Andrew Stanton	2
Brad Bird	2
Brenda Chapman	1
Dan Scanlon	1
John Lasseter	5
Lee Unkrich	1
Pete Docter	2

```
SELECT director, COUNT(*) FROM movies 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.

Finish above Tasks

Next – SQL Lesson 13: Inserting rows
Previous – SQL Lesson 11: Queries with aggregates (Pt. 2)

Query Results

Director	Total
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) + SUM(international_sales) AS Total
    FROM movies
   LEFT 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 ›

Next – SQL Lesson 13: Inserting rows
Previous – SQL Lesson 11: Queries with aggregates (Pt. 2)

Query Results

Id	Title	Director	Year	Length_minutes

```
INSERT INTO movies (title, director, year, length_minutes)
 VALUES ('Toy Story 4', 'John Lasseter', 2019, 123);
```

RUN QUERY RESET

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.

Finish above Tasks

Next – SQL Lesson 14: Updating rows
Previous – SQL Lesson 12: Order of execution of a Query

Exercise

It looks like some of the information in our **Movies** database might be incorrect, so go ahead and fix them through the exercises below.

Table: Movies

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
11	Toy Story 8	El Directore	2010	103
12	Cars 2	John Lasseter	2011	120
13	Brave	Brenda Chapman	2012	102

```
UPDATE movies SET director = "John Lasseter" WHERE title = "A Bug's Life";
```

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.

Finish above Tasks

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

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 ›

Next – SQL Lesson 15: Deleting rows
Previous – SQL Lesson 13: Inserting rows

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 director = "Lee Unkrich", title = "Toy Story 3" WHERE
    title = "Toy Story 8";
```

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 ›

Next – SQL Lesson 15: Deleting rows
Previous – SQL Lesson 13: Inserting rows

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

```
DELETE FROM movies WHERE year < 2005;
```

RUN QUERY RESET

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.

Finish above Tasks

Next – SQL Lesson 16: Creating tables
Previous – SQL Lesson 14: Updating rows

Table: Movies

Id	Title	Director	Year	Length_minutes
7	Cars	John Lasseter	2006	117
8	Ratatouille	Brad Bird	2007	115
10	Up	Pete Docter	2009	101
11	Toy Story 3	Lee Unkrich	2010	103
12	Cars 2	John Lasseter	2011	120
13	Brave	Brenda Chapman	2012	102
14	Monsters University	Dan Scanlon	2013	110

Row(s) deleted

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

RUN QUERY RESET

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.

Continue ›

Next – SQL Lesson 16: Creating tables
Previous – SQL Lesson 14: Updating rows

In this exercise, you'll need to create a new table for us to insert some new rows into.

Table: Database

Missing table...

```
CREATE TABLE database (id INTEGER PRIMARY KEY, name TEXT, version FLOAT,
    download_count INTEGER );
```

RUN QUERY RESET

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.

Continue ›

Next – SQL Lesson 17: Altering tables
Previous – SQL Lesson 15: Deleting rows

Table: Movies

Id	Title	Director	Year	Length_minutes	Aspect_ratio
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	
New column added		Pete Docter	2009	101	

```
ALTER TABLE movies ADD aspect_ratio FLOAT;
```

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.

Finish above Tasks

Next – SQL Lesson 18: Dropping tables
Previous – SQL Lesson 16: Creating tables

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 TEXT 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 ›

Next – SQL Lesson 18: Dropping tables
Previous – SQL Lesson 16: Creating tables

Query Results

Id	Title	Director	Year	Length_minutes

Table dropped

```
DROP TABLE IF EXISTS movies;
```

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.

Finish above Tasks

Next – SQL Lesson X: To infinity and beyond!
Previous – SQL Lesson 17: Altering tables

Query Results

Id	Title	Director	Year	Length_minutes

Table dropped

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

RUN QUERY RESET

Exercise 18 — Tasks

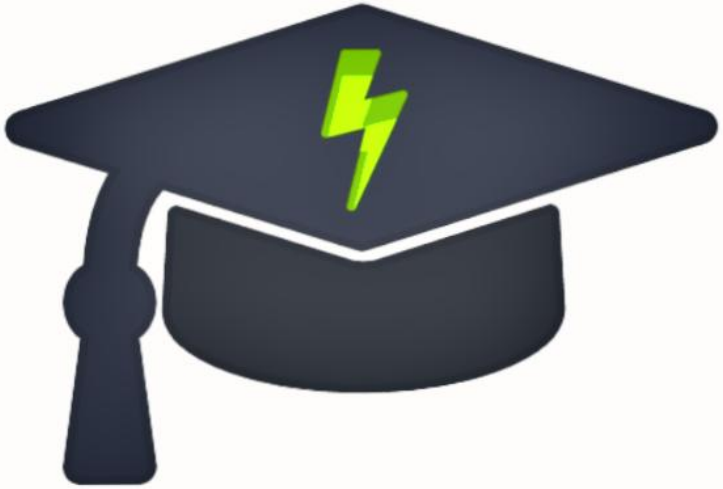
1. We've sadly reached the end of our lessons, lets clean up by removing the **Movies** table ✓
2. And drop the **BoxOffice** table as well ✓

Stuck? Read this task's Solution.
Solve all tasks to continue to the next lesson.

Continue ›

Next – SQL Lesson X: To infinity and beyond!
Previous – SQL Lesson 17: Altering tables

SQL Lesson X: To infinity and beyond!



You've finished the tutorial!

We hope the lessons have given you a bit more experience with SQL and a bit more confidence to use SQL with your own data.

We've just brushed the surface of what SQL is capable of, so to get a better idea of how SQL can be used in