

This Exercise from <https://sqlbolt.com/>

SQL Lesson 1: SELECT queries 101

need for each task.

Table: Movies

4	Monsters, Inc.	Pete Docter	2001	92
5	Finding Nemo	Andrew Stanton	2003	107
6	The Incredibles	Brad Bird	2004	116
7	Cars	John Lasseter	2006	117
8	Ratatouille	Brad Bird	2007	115
9	WALL-E	Andrew Stanton	2008	104
10	Up	Pete Docter	2009	101
11	Toy Story 3	Lee Unkrich	2010	103
12	Cars 2	John Lasseter	2011	120
13	Brave	Brenda Chapman	2012	102
14	Monsters University	Dan Scanlon	2013	110

`SELECT * FROM movies;`

RESET

Exercise 1 — Tasks

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

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

Finish above Tasks

Answers :

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;`

SQL Lesson 2: Queries with constraints (Pt. 1)

Exercise

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

Table: Movies

4	Monsters, Inc.	Pete Docter	2001	92
5	Finding Nemo	Andrew Stanton	2003	107
6	The Incredibles	Brad Bird	2004	116
7	Cars	John Lasseter	2006	117
8	Ratatouille	Brad Bird	2007	115
9	WALL-E	Andrew Stanton	2008	104
10	Up	Pete Docter	2009	101
11	Toy Story 3	Lee Unkrich	2010	103
12	Cars 2	John Lasseter	2011	120
13	Brave	Brenda Chapman	2012	102
14	Monsters University	Dan Scanlon	2013	110

```
SELECT * FROM movies;
```

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

Answers :

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 * FROM movies WHERE id <=5;`

SQL Lesson 3: Queries with constraints (Pt. 2)

Answers :

1. `SELECT title FROM movies where title Like "Toy Story%";`

2. `SELECT *FROM movies where director= "John Lasseter";`
3. `SELECT *FROM movies where director!= "John Lasseter";`
4. `SELECT * FROM movies where title like "WALL-%";`

SQL Lesson 4: Filtering and sorting Query results

smajaas (Mohammed Ajaas) x SQLBolt - Learn SQL - SQL Le x New Tab x Find the first 5 Pixar movies x SQL query practice : 6002526 x +

sqlbolt.com/lesson/filtering_sorting_query_results

Table: Movies

Id	Title	Director	Year	Length_minutes
1	Up	Pete Docter	2009	101
2	Ratatouille	Brad Bird	2007	115
3	The Incredibles	Brad Bird	2004	116
4	A Bug's Life	John Lasseter	1998	95
5	Finding Nemo	Andrew Stanton	2003	107
6	Cars 2	John Lasseter	2011	120
7	WALL-E	Andrew Stanton	2008	104
8	Monsters University	Dan Scanlon	2013	110
9	Monsters, Inc.	Pete Docter	2001	92
10	Cars	John Lasseter	2006	117

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

RESET

SELECT * FROM movies;

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

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

Type here to search

31°C Mostly sunny 11:20 19-Nov-21

Answers :

1. `SELECT distinct director FROM movies
order by director;`
2. `SELECT DISTINCT title FROM movies
ORDER BY year DESC LIMIT 4;`
3. `SELECT title FROM movies
ORDER BY title LIMIT 5;`

4. SELECT title FROM movies
ORDER BY title LIMIT 5 OFFSET 5;

Lesson:5 SQL Review: Simple SELECT Queries

Answers :

1. SELECT city, population
FROM north_american_cities
WHERE country = "Canada";
2. SELECT city FROM north_american_cities WHERE country = "United States" ORDER BY latitude DESC
3. SELECT city FROM north_american_cities WHERE longitude < -87.629798
BY longitude ASC;
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;

SQL Lesson 6: Multi-table queries with JOINS

smajaas (Mohammed Ajaas) x SQLBolt - Learn SQL - SQL Le... x New Tab x Find the first 5 Pixar movies x SQL query practice : 6002526 x +

← → ↻ sqlbolt.com/lesson/select_queries_with_joins
table: movies (read-only)

Id	Title	Director	Year	Length_minutes
1	Toy Story	John Lasseter	1995	81
2	A Bug's Life	John Lasseter	1998	95
3	Toy Story 2	John Lasseter	1999	93
4	Monsters, Inc.	Pete Docter	2001	92
5	Finding Nemo	Andrew Stanton	2003	107
6	The Incredibles	Brad Bird	2004	116
7	Cars	John Lasseter	2006	117
8	Ratatouille	Brad Bird	2007	115
9	WALL-E	Andrew Stanton	2008	104
10	The Incredibles 2	Pete Docter	2018	118

table: boxoffice (read-only)

Movie_id	Rating	Domestic_sales	International_sales
5	8.2	380843261	555900000
14	7.4	268492764	475066843
8	8	206445654	417277164
12	6.4	191452396	368400000
3	7.9	245852179	239163000
6	8	261441092	370001000
7	6.8	233618701	373618701
1	7.6	309369881	439037614
2	6.9	162803442	285393442
4	6.8	219497321	220497321

Query Results

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	The Incredibles 2	Pete Docter	2018	118

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

Answers :

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 international_sales > domestic_sales;`
3. `SELECT title, rating FROM movies INNER JOIN boxoffice ON movies.id = boxoffice.movie_id ORDER BY rating DESC;`

SQL Lesson 7: OUTER JOINS

The screenshot shows a web browser window with the URL `sqlbolt.com/lesson/select_queries_with_outer_joins`. The page displays two tables: 'Table: Buildings (Read-Only)' and 'Table: Employees (Read-Only)'. Below these tables, the 'Query Results' section shows a table with columns 'Role', 'Name', 'Building', and 'Years_employed'. To the right of the query results, there is a section titled 'Exercise 7 — Tasks' with three tasks listed.

Table: Buildings (Read-Only)

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

Table: Employees (Read-Only)

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

Query Results

Role	Name	Building	Years_employed
Engineer	Becky A.	1e	4
Engineer	Dan B.	1e	2
Engineer	Sharon F.	1e	6
Engineer	Dan M.	1e	4
Engineer	Malcom S.	1e	1
Artist	Tylar S.	2w	2
Artist	Sherman D.	2w	8
Artist	Jakob J.	2w	6
Artist	Lillia A.	2w	7

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)

Answers :

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;

SQL Lesson 8: A short note on NULLs

The screenshot shows the SQLBolt website interface. On the left, a table titled 'Query Results' displays data for employees. The table has four columns: Role, Name, Building, and Years_employed. Below the table, a text input field contains the SQL query 'SELECT * FROM employees;', and a 'RESET' button is located to its right. On the right side of the interface, a panel titled 'Exercise 8 — Tasks' lists two tasks: '1. Find the name and role of all employees who have not been assigned to a building' and '2. Find the names of the buildings that hold no employees'. Below the tasks, there is a link to a 'Solution' and a prompt to 'Solve all tasks to continue to the next lesson.' At the bottom of this panel is a 'Finish above Tasks' button. The website's footer includes navigation links for 'Next - SQL Lesson 9: Queries with expressions' and 'Previous - SQL Lesson 7: OUTER JOINS', a donation request for 'Donating (\$4) via Paypal', and the current date and time: '19:15 20-Nov-21'.

Role	Name	Building	Years_employed
Engineer	Becky A.	1e	4
Engineer	Dan B.	1e	2
Engineer	Sharon F.	1e	6
Engineer	Dan M.	1e	4
Engineer	Malcom S.	1e	1
Artist	Tylar S.	2w	2
Artist	Sherman D.	2w	8
Artist	Jakob J.	2w	6
Artist	Lillia A.	2w	7
Artist	Brandon J.	2w	7

SELECT * FROM employees;

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](#)

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

19:15 20-Nov-21

Answers :

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

```
SELECT * FROM employees where building is null
```

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

```
SELECT Building_name from buildings left join employees on building_name=building where role is null;
```

SQL Lesson 9: Queries with expressions:

The screenshot shows the SQLBolt website interface. At the top, there's a browser tab titled 'SQLBolt - Learn SQL - SQL Lesson' and a URL 'sqlbolt.com/lesson/select_queries_with_expressions'. Below the browser window, there's a table with 10 rows and 5 columns: Id, Title, Director, Year, and Length_minutes. The table contains data for movies like 'Toy Story', 'A Bug's Life', 'Toy Story 2', 'Monsters, Inc.', 'Finding Nemo', 'The Incredibles', 'Cars', 'Ratatouille', 'WALL-E', and 'Up'. Below the table, there's a text input field with the SQL query 'SELECT * FROM movies;' and a 'RESET' button. To the right of the table, there's a sidebar with 'Exercise 9 — Tasks' containing three tasks: '1. List all movies and their combined sales in millions of dollars', '2. List all movies and their ratings in percent', and '3. List all movies that were released on even number years'. At the bottom of the sidebar, there's a 'Finish above Tasks' button. The bottom of the screenshot shows a Windows taskbar with various icons and a system tray showing '30°C Partly cloudy' and '19:28 20-Nov-21'.

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

Query Results

```
SELECT * FROM movies;
```

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

Answers :

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

SELECT title,(Domestic_sales+International_sales)/1000000 as Total_sales_In_Million FROM movies inner join Boxoffice on Movies.id = movie_id;

2. List all movies and their ratings in percent:

SELECT title, rating*10 as Percentage FROM movies inner join Boxoffice on Movies.id = movie_id;

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

SELECT title FROM movies where year %2=0;

SQL Lesson 10: Queries with aggregates (Pt. 1)

The screenshot shows the SQLBolt website interface. On the left, there is a table named 'Employees' with the following data:

Role	Name	Building	Years_employed
Engineer	Becky A.	1e	4
Engineer	Dan B.	1e	2
Engineer	Sharon F.	1e	6
Engineer	Dan M.	1e	4
Engineer	Malcom S.	1e	1
Artist	Tylar S.	2w	2
Artist	Sherman D.	2w	8
Artist	Jakob J.	2w	6
Artist	Lillia A.	2w	7
Artist	Brandon J.	2w	7

Below the table, there is a text input field containing the SQL query: `SELECT * FROM employees;` and a 'RESET' button.

On the right side, there is a section titled 'Exercise 10 — Tasks' with the following 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

Below the tasks, there is a link to 'Solution' and a button to 'Finish above Tasks'.

At the bottom of the page, there is a navigation bar with links to 'Next - SQL Lesson 11: Queries with aggregates (Pt. 2)' and 'Previous - SQL Lesson 9: Queries with expressions'. There is also a footer with a donation link and the date '20-Nov-21'.

Answers :

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

**SELECT distinct name,Max(Years_employed) as longest_service
FROM employees;**

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

**SELECT role,avg(Years_employed) as Average_service FROM
employees group by role;**

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

**SELECT building,sum(Years_employed) as Sum_of_years FROM
employees group by building;**

SQL Lesson 11: Queries with aggregates (Pt. 2)

SQLBolt - Learn SQL - SQL Lesson 11: Queries with aggregates (Pt. 2)

Exercise

For this exercise, you are going to dive deeper into **Employee** data at the film studio. Think about the different clauses you want to apply for each task.

Table: Employees

Role	Name	Building	Years_employed
Engineer	Becky A.	1e	4
Engineer	Dan B.	1e	2
Engineer	Sharon F.	1e	6
Engineer	Dan M.	1e	4
Engineer	Malcom S.	1e	1
Artist	Tylar S.	2w	2
Artist	Sherman D.	2w	8
Artist	Jakob J.	2w	6
Artist	Lillia A.	2w	7
Artist	Brandon J.	2w	7

SELECT * FROM employees;

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.

Type here to search

30°C Partly cloudy 20:04 20-Nov-21

Answers :

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

```
SELECT role, count(role) FROM employees where Role="Artist";
```

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

```
SELECT role, count(role) FROM employees group by role;
```

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

```
SELECT role, sum(years_employed) FROM employees where Role="Engineer";
```

SQL Lesson 12: Order of execution of a Query

SQLBolt - Learn SQL - SQL Lesson 12: Order of execution of a Query

Query Results

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

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

RESET

Answers :

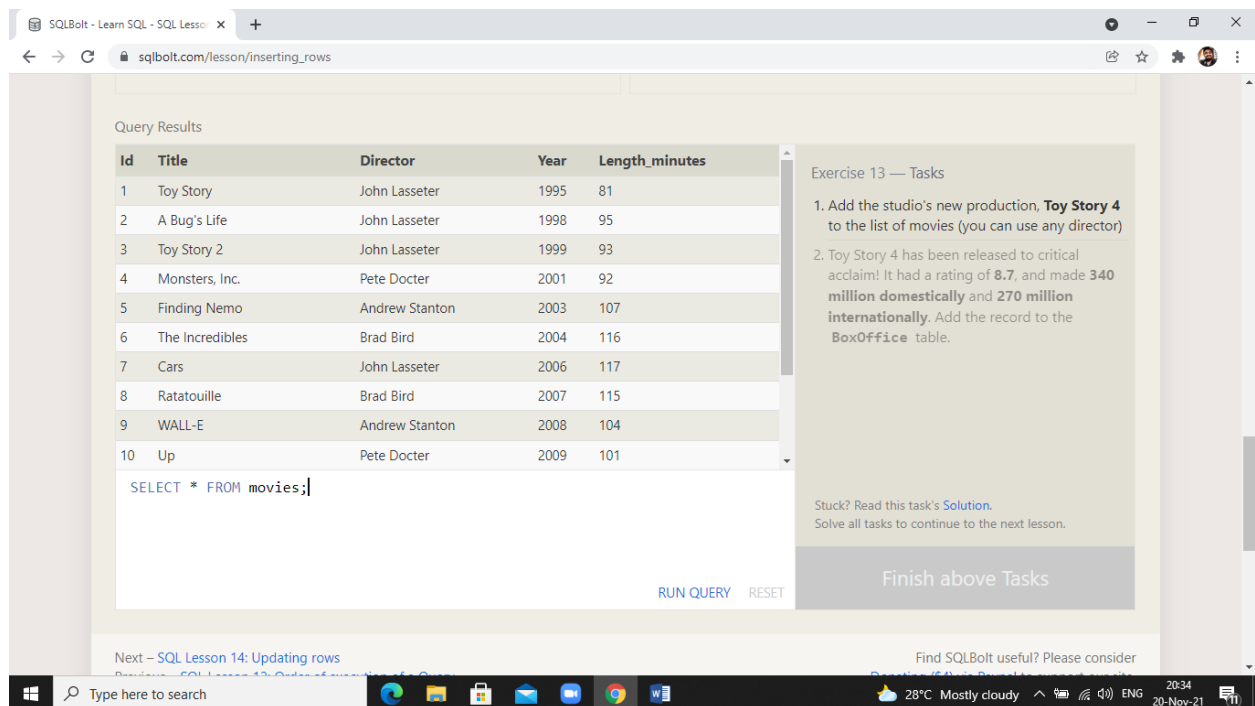
1. Find the number of movies each director has directed

SELECT Director, count(Director) as count FROM movies group by director order by count desc ;

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

SELECT director, sum(Domestic_sales+International_sales) /1000000 as Total_Collection FROM movies inner join Boxoffice on Movies.id=Movie_id group by director order by Total_Collection desc ;

SQL Lesson 13: Inserting rows



The screenshot shows the SQLBolt website interface for Lesson 13. On the left, a table titled 'Query Results' lists 10 movies with columns: Id, Title, Director, Year, and Length_minutes. Below the table is a text input field containing the SQL query: `SELECT * FROM movies;`. To the right of the table is a sidebar for 'Exercise 13 — Tasks' containing two tasks. At the bottom of the sidebar is a 'Finish above Tasks' button. The bottom of the page shows a navigation bar with 'Next - SQL Lesson 14: Updating rows' and a footer with 'Find SQLBolt useful? Please consider'.

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

Query Results

SELECT * FROM movies;

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

Find SQLBolt useful? Please consider

Answers :

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

INSERT INTO Movies

VALUES (4, "Toy Story 4", "Ajaas", 2021, 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.

INSERT INTO Boxoffice

VALUES (4,8.7,340000000,270000000)

SQL Lesson 14: Updating rows:

SQLBolt - Learn SQL - SQL Lesson 14: Updating rows

sqlbolt.com/lesson/updating_rows

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

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

`SELECT * FROM movies;`

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.

Answers:

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

update movies

set director = "John Lasseter"

where id=2;

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

update movies

set year = 1999

where id=3;

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

UPDATE Movies

SET Title = "Toy Story 3",

Director = "Lee Unkrich"

where id = 11;

SQL Lesson 15: Deleting rows

The screenshot shows the SQLBolt website interface. On the left, a table named 'Movies' is displayed with the following data:

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

Below the table, the SQL query editor contains the text: `SELECT * FROM movies;`

On the right side, the 'Exercise 15 — Tasks' section lists two 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.

Below the tasks, there is a link to the solution: [Stuck? Read this task's Solution.](#) and a button labeled 'Finish above Tasks'.

At the bottom of the page, there are navigation links: 'Next - SQL Lesson 16: Creating tables' and 'Previous - SQL Lesson 14: Updating rows'. There is also a footer with a donation request: 'Find SQLBolt useful? Please consider Donating (\$4) via Paypal to support our site.'

Answers:

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

DELETE FROM Movies

where title > 2005;

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

DELETE FROM Movies

where director = "Andrew Stanton";

SQL Lesson 16: Creating tables

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 floating point,
  Download_Count integer);
```

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 >](#)

Waiting for googleads.g.doubleclick.net...

Type here to search

28°C Mostly cloudy 22:12 20-Nov-21

Answers :

CREATE TABLE Database (
Id Integer Primary Key,
Name text,
Version floating point,
Download_Count integer);

SQL Lesson 17: Altering tables:

Exercise

Our exercises use an implementation that only support adding new columns, so give that a try 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
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;`

[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](#)

Answers :

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

ALTER TABLE Movies

ADD 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**.

ALTER TABLE Movies

ADD Language Text

DEFAULT English;

SQL Lesson 18: Dropping tables

The screenshot shows the SQLBolt website interface for Lesson 18. The browser address bar shows the URL `sqlbolt.com/lesson/dropping_tables`. The main content area is divided into two sections: 'Query Results' on the left and 'Exercise 18 — Tasks' on the right.

Query Results: A table with 10 rows and 5 columns: **Id**, **Title**, **Director**, **Year**, and **Length_minutes**.

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

Below the table, there is a text input field containing the SQL query: `SELECT * FROM movies;`. To the right of the input field are two buttons: **RUN QUERY** and **RESET**.

Exercise 18 — Tasks: A list of two 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

Below the tasks, there is a link: [Stuck? Read this task's Solution.](#) and a text: **Solve all tasks to continue to the next lesson.**

At the bottom of the task section, there is a button: **Finish above Tasks**.

Footer: The footer contains navigation links: [Next – SQL Lesson X: To infinity and beyond!](#) and [Previous – SQL Lesson 17: Altering tables](#). On the right, there is a message: **Find SQLBolt useful? Please consider Donating (\$4) via Paypal to support our site.**

The Windows taskbar at the bottom shows the system clock as 22:21 on 20-Nov-21, with a weather forecast of 27°C Mostly cloudy.

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

DROP TABLE IF EXISTS MOVIES

2. And drop the BoxOffice table as well
-

DROP TABLE IF EXISTS BOXOFFICE