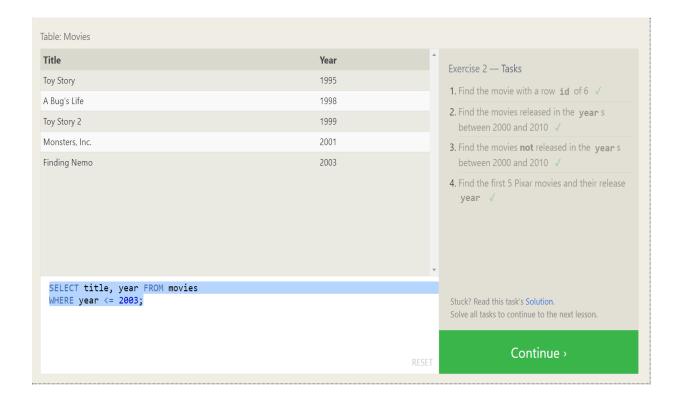
# MySQL TASK

# **SQL Lesson 1: SELECT queries 101**

A To M Fi	Toy Story A Bug's Life Toy Story 2 Monsters, Inc. Tinding Nemo	John Lasseter John Lasseter John Lasseter Pete Docter Andrew Stanton	1995 1998 1999 2001 2003	81 95 93 92		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
To M Fi	Toy Story 2  Monsters, Inc.  Finding Nemo	John Lasseter Pete Docter	1999 2001	93	1	2. Find the director of each film ✓
M Fi	Monsters, Inc.	Pete Docter	2001		1	
Fi	inding Nemo			92		3. Find the title and director of each film
		Andrew Stanton	2002			✓
TI	The Incredibles		2003	107		4. Find the title and year of each film ✓
	The incredibles	Brad Bird	2004	116		5. Find all the information about each film
C	Cars	John Lasseter	2006	117		
R	Ratatouille	Brad Bird	2007	115		
W	WALL-E	Andrew Stanton	2008	104		
) U	Jp	Pete Docter	2009	101	*	
SELE	CCT * FROM movies;					Stuck? Read this task's <b>Solution</b> . Solve all tasks to continue to the next lesson.

- 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)**



- 1. SELECT \* FROM movies WHERE Id = 6;
- 2. SELECT \* FROM movies WHERE year BETWEEN 2000 AND 2010;
- 3. SELECT \* FROM movies WHERE Year NOT BETWEEN 2000 AND 2010;
- 4. SELECT title, year FROM movies WHERE year <= 2003;

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

	Title	Director	Year	Length_minutes	<b>A</b>	
	WALL-E	Andrew Stanton		104		Exercise 3 — Tasks
			2008			1. Find all the Toy Story movies ✓
7	WALL-G	Brenda Chapman	2042	97		2. Find all the movies directed by John Lasseter $\ensuremath{\checkmark}$
						3. Find all the movies (and director) not directed by John Lasseter ✓
					*	<b>4.</b> Find all the WALL-* movies ✓
SEI	LECT * FROM	movies WHERE Title LIM	KE 'WALL-%';			Stuck? Read this task's Solution. Solve all tasks to continue to the next lesson.
					RESET	Continue >

- 1. SELECT \* 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**

Table	: Movies					
Id	Title	Director	Year	Length_minutes	A	Exercise 4 — Tasks
14	Monsters University	Dan Scanlon	2013	110		
9	Monsters, Inc.	Pete Docter	2001	92		<ol> <li>List all directors of Pixar movies         (alphabetically), without duplicates √     </li> </ol>
13	Ratatouille	Brad Bird	2007	115		2. List the last four Pixar movies released
11	The Incredibles	Brad Bird	2004	116		(ordered from most recent to least) $\ \ \checkmark$
1	Toy Story	John Lasseter	1995	81		3. List the <b>first</b> five Pixar movies sorted alphabetically ✓
					¥	<b>4.</b> List the <b>next</b> five Pixar movies sorted alphabetically √
FR	LECT * OM movies DER BY Title ASC MIT 5 OFFSET 5;					Stuck? Read this task's <b>Solution</b> . Solve all tasks to continue to the next lesson.
I					RESET	Continue >

### **Solutions:**

1. SELECT DISTINCT Director FROM movies ORDER BY Director ASC;

2. SELECT \*
FROM movies
ORDER BY Year DESC
LIMIT 4;

3. SELECT \*
FROM movies
ORDER BY Title ASC
LIMIT 5;

4. SELECT \*
FROM movies
ORDER BY Title ASC
LIMIT 5 OFFSET 5;

# **SQL Review: Simple SELECT Queries**

City	Population	Review 1 — Tasks
Chicago	2718782	
Houston	2195914	<ol> <li>List all the Canadian cities and their populations √</li> </ol>
		2. Order all the cities in the United States by thei latitude from north to south $\ensuremath{\checkmark}$
		3. List all the cities west of Chicago, ordered from west to east $\ensuremath{\checkmark}$
		<b>4.</b> 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 ✓
SELECT City, Popul FROM North_america WHERE Country = 'U ORDER BY Population	an_cities United States' on DESC	Stuck? Read this task's Solution. Solve all tasks to continue to the next lesson.
LIMIT 2 OFFSET 2;		Continue >

#### **Solutions:**

1. SELECT City, Population FROM North\_american\_cities WHERE Country = 'Canada';

2. SELECT City, Latitude FROM North\_american\_cities WHERE Country = 'United States' ORDER BY Latitude DESC;

3. SELECT City, Longitude
FROM North\_american\_cities
WHERE Longitude < (SELECT Longitude FROM North\_american\_cities
WHERE City = 'Chicago')
ORDER BY Longitude ASC;

4. SELECT City, Population FROM North\_american\_cities WHERE Country = 'Mexico' ORDER BY Population DESC LIMIT 2;

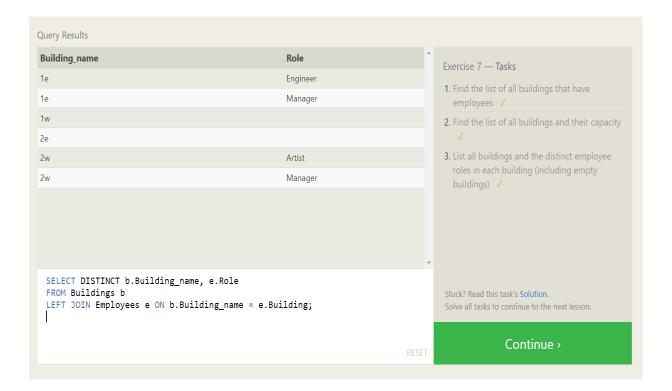
5. SELECT City, Population FROM North\_american\_cities WHERE Country = 'United States' ORDER BY Population DESC LIMIT 2 OFFSET 2;

# **SQL Lesson 6: Multi-table queries with JOINs**

9 WALL-E Andrew S 11 Toy Story 3 Lee Unkri 11 Toy Story John Lass 10 Up Pete Doct 15 Finding Nemo Andrew S 14 Monsters, Inc. Pete Doct 18 Ratatouille Brad Bird 19 WALL-E Andrew S 10 Up Pete Doct 10 Pete Doct 11 Peter Doct 12 Peter Doct 13 Ratatouille Brad Bird 15 The Incredibles Brad Bird	ch 2010 eter 1995 ter 2009 tanton 2003 ter 2001	104 103 81 101 107 92	8.5 8.4 8.3 8.3 8.2 8.1	Exercise 6 — Tasks  1. Find the domestic and ir each movie   2. Show the sales numbers did better internationally domestically   3. List all the movies by the	s for each movie tha y rather than
Toy Story John Lass  Up Pete Doct Finding Nemo Andrew S Monsters, Inc. Pete Doct Ratatouille Brad Bird	eter 1995 ter 2009 tanton 2003 ter 2001	81 101 107 92	8.3 8.3 8.2	each movie ✓  2. Show the sales numbers did better internationally domestically ✓	s for each movie tha y rather than
0 Up Pete Doct Finding Nemo Andrew S Monsters, Inc. Pete Doct Ratatouille Brad Bird	ter 2009 tanton 2003 ter 2001	101 107 92	8.3	did better internationally domestically √	y rather than
Finding Nemo Andrew S  Monsters, Inc. Pete Doct  Ratatouille Brad Bird	tanton 2003 eer 2001	107 92	8.2	domestically 🗸	
Monsters, Inc. Pete Doct Ratatouille Brad Bird	ter 2001	92			eir ratings in
Ratatouille Brad Bird			8.1	3. List all the movies by the	eir ratings in
	2007			descending order ✓	, and the second
The leave diluter Doe d Died		115	8	acoconaing oraci	
The incredibles Brad Bird	2004	116	8		
Toy Story 2 John Lass	eter 1999	93	7.9		
4 Monsters University Dan Scanl	lon 2013	110	7.4	•	
,, _	Stuck? Read this task's Solution Solve all tasks to continue to t	•			

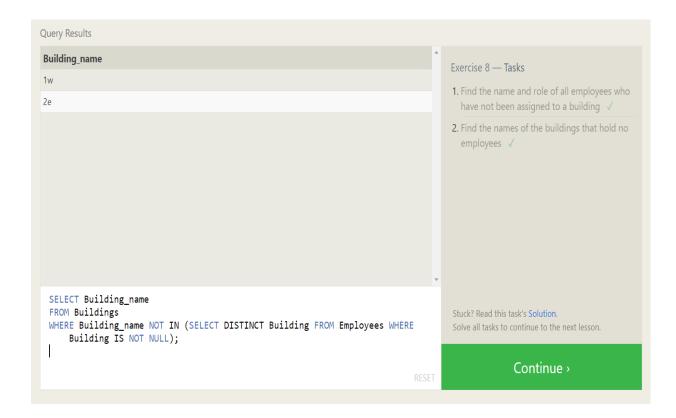
- 1. SELECT Movies.Id, Title, Director, Year, Length\_minutes, Domestic\_sales, International\_sales FROM Movies
  - INNER JOIN BoxOffice ON Movies.Id = BoxOffice.Movie\_id;
- 2. SELECT Movies.Id, Title, Director, Year, Length\_minutes, Domestic\_sales, International\_sales FROM Movies INNER JOIN BoxOffice ON Movies.Id = BoxOffice.Movie\_id WHERE International sales > Domestic sales;
- 3. SELECT Movies.Id, Title, Director, Year, Length\_minutes, Rating FROM Movies
  - INNER JOIN BoxOffice ON Movies.Id = BoxOffice.Movie\_id ORDER BY Rating DESC;

### **SQL Lesson 7: OUTER JOINs**



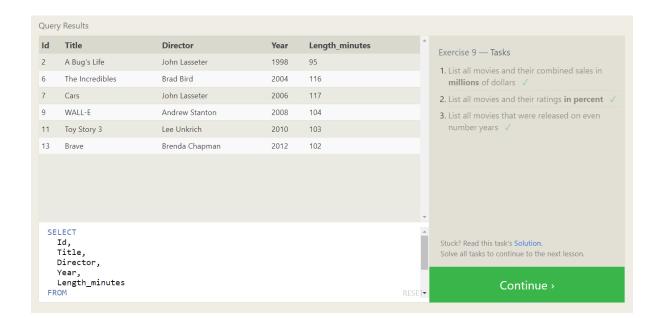
- 1. SELECT DISTINCT e.Building
  FROM Employees e
  LEFT JOIN Buildings b ON e.Building = b.Building\_name
  WHERE e.Building IS NOT NULL;
- 2. SELECT b.Building\_name, b.Capacity
  FROM Buildings b
  LEFT JOIN Employees e ON b.Building\_name = e.Building
  GROUP BY b.Building\_name, b.Capacity;
- 3. SELECT DISTINCT b.Building\_name, e.Role FROM Buildings b LEFT JOIN Employees e ON b.Building\_name = e.Building;

# **SQL Lesson 8: A short note on NULLs**



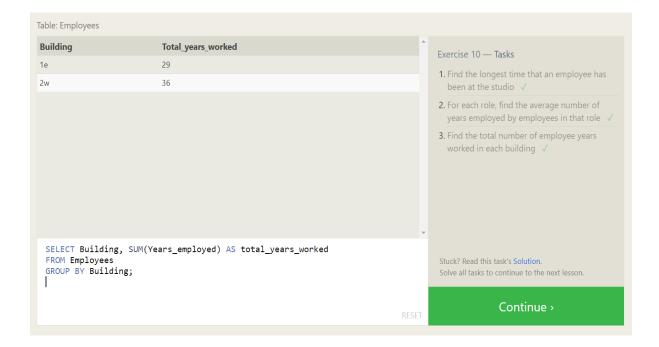
- 1. SELECT Name, Role FROM Employees WHERE Building IS NULL;
- 2. SELECT Building\_name FROM Buildings WHERE Building\_name NOT IN (SELECT DISTINCT Building FROM Employees WHERE Building IS NOT NULL);

# **SQL Lesson 9: Queries with expressions**



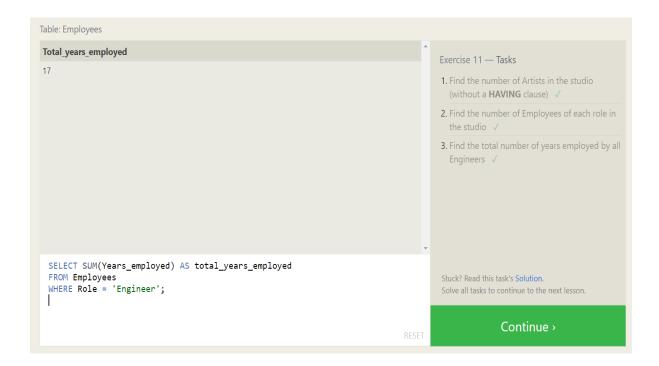
```
SELECT
 m.Id,
 m.Title,
 m.Director,
 m.Year,
 m.Length_minutes,
(b.Domestic_sales + b.International_sales) / 1000000 AS combined_sales_millions
FROM
          Movies m
JOIN
          Boxoffice b ON m.Id = b.Movie_id;
SELECT
 m.Id,
 m.Title,
 m.Director,
 m.Year,
 m.Length_minutes,
 b.Rating * 10 AS rating_percent
FROM
 Movies m
JOIN
 Boxoffice b ON m.Id = b.Movie_id;
SELECT
 Id,
 Title,
 Director,
 Year,
 Length_minutes
FROM
 Movies
WHERE
 Year \% 2 = 0;
```

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



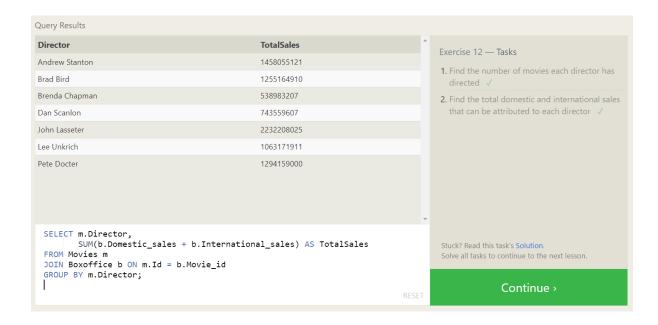
- 1. SELECT MAX(Years\_employed) AS longest\_time FROM Employees;
- 2. SELECT Role, AVG(Years\_employed) AS avg\_years\_employed FROM Employees GROUP BY Role;
- 3. SELECT Building, SUM(Years\_employed) AS total\_years\_worked FROM Employees GROUP BY Building;

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



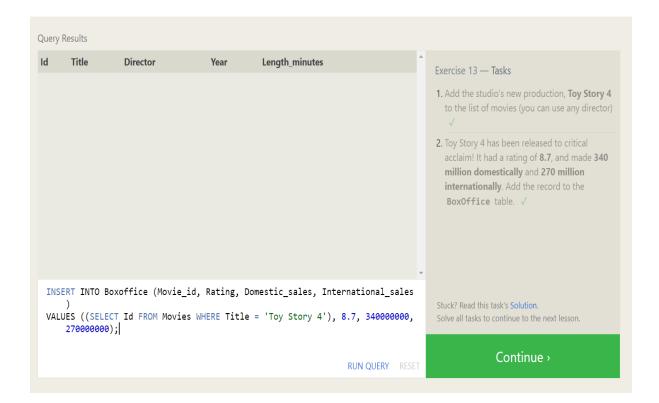
- SELECT COUNT(\*) AS num\_artists
   FROM Employees
   WHERE Role = 'Artist';
- 2. SELECT Role, COUNT(\*) AS num\_employees FROM Employees GROUP BY Role;
- 3. SELECT SUM(Years\_employed) AS total\_years\_employed FROM Employees
  WHERE Role = 'Engineer';

# SQL Lesson 12: Order of execution of a Query



- 1. SELECT Director, COUNT(\*) AS NumMoviesDirected FROM Movies GROUP BY Director;
- 2. SELECT m.Director,
   SUM(b.Domestic\_sales + b.International\_sales) AS TotalSales
  FROM Movies m
  JOIN Boxoffice b ON m.Id = b.Movie\_id
  GROUP BY m.Director;

# **SQL Lesson 13: Inserting rows**



- 1. INSERT INTO Movies (Title, Director, Year, Length\_minutes) VALUES ('Toy Story 4', 'Any Director', 2023, 100);
- 2. INSERT INTO Boxoffice (Movie\_id, Rating, Domestic\_sales, International\_sales)
  VALUES ((SELECT Id FROM Movies WHERE Title = 'Toy Story 4'), 8.7, 340000000, 270000000);

# **SQL Lesson 14: Updating rows**

ld	Title	Director	Year	Length_minutes	Î	Exercise 14 — Tasks
1	Toy Story	John Lasseter	1995	81		2.0.000
2	A Bug's Life	John Lasseter	1998	95		<ol> <li>The director for A Bug's Life is incorrect, it was actually directed by John Lasseter √</li> </ol>
3	Toy Story 2	John Lasseter	1999	93		2. The year that Toy Story 2 was released is
4	Monsters, Inc.	Pete Docter	2001	92		incorrect, it was actually released in 1999 🗸
5	Finding Nemo	Andrew Stanton	2003	107		3. Both the title and director for Toy Story 8 is
6	The Incredibles	Brad Bird		incorrect! The title should be "Toy Story 3" and it was directed by <b>Lee Unkrich</b> ✓		
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	•	
SE	DATE Movies T Title = 'Toy Stor' ERE Title = 'Toy St	y 3', Director = 'Lee ory 8';		Stuck? Read this task's Solution. Solve all tasks to continue to the next lesson.		
			ESET	Continue >		

#### **Solutions:**

1. UPDATE Movies

SET Director = 'John Lasseter' WHERE Title = 'A Bug''s Life';

2. UPDATE Movies

**SET Year = 1999** 

WHERE Title = 'Toy Story 2';

3. UPDATE Movies

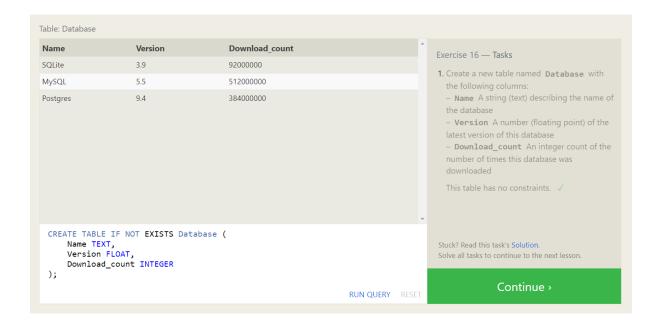
SET Title = 'Toy Story 3', Director = 'Lee Unkrich' WHERE Title = 'Toy Story 8';

# **SQL Lesson 15: Deleting rows**

Table	: Movies							
ld	Title	Director	Year	Length_minutes	<b>A</b>	Exercise 15 — Tasks		
7	Cars	John Lasseter	2006	117				
8	Ratatouille	Brad Bird	2007	115		<ol> <li>This database is getting too big, lets remove all movies that were released before 2005.</li> </ol>		
10	Up	Pete Docter	2009	101		√ 		
11	Toy Story 3	Lee Unkrich	2010	103		2. Andrew Stanton has also left the studio, so		
12	Cars 2	John Lasseter	120		please remove all movies directed by him.   ✓			
13	Brave	Brenda Chapman	2012	102				
14	Monsters University	Dan Scanlon	2013	110				
					~			
	LETE FROM Movies							
WH	ERE Director = 'Andrew S	Stanton';			Stuck? Read this task's Solution. Solve all tasks to continue to the next lesson.			
				RUN QUERY RES	SET	Continue >		

- 1. DELETE FROM Movies WHERE Year < 2005;
- 2. DELETE FROM Movies
  WHERE Director = 'Andrew Stanton';

# **SQL Lesson 16: Creating tables**



### **Solutions:**

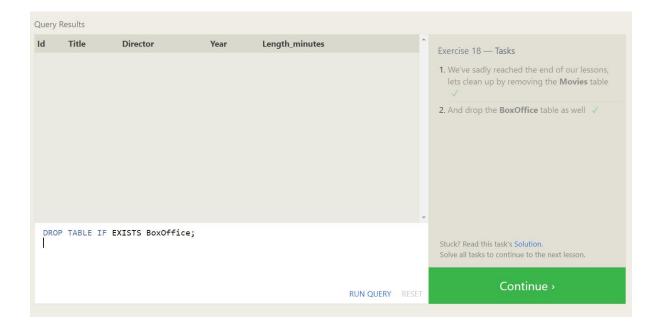
```
    CREATE TABLE IF NOT EXISTS Database (
        Name TEXT,
        Version FLOAT,
        Download_count INTEGER
        );
```

# **SQL Lesson 17: Altering tables**

Tabl	e: Movies								
Id	Title	Director	Year	Length_minutes	Aspect_ratio	Langua	ge	1	Exercise 17 — Tasks
1	Toy Story	John Lasseter	1995	81		English			
2	A Bug's Life	John Lasseter	1998	95		English			Add a column named <b>Aspect_ratio</b> with a <b>FLOAT</b> data type to store the aspect-ratio each
3	Toy Story 2	John Lasseter	1999	93		English			movie was released in. ✓
4	Monsters, Inc.	Pete Docter	2001	92		English			2. Add another column named <b>Language</b> with a
5	Finding Nemo	Andrew Stanton	2003	107		English			<b>TEXT</b> data type to store the language that the movie was released in. Ensure that the default
6	The Incredibles	Brad Bird	2004	116		English			for this language is <b>English</b> . $\checkmark$
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	,	·	
	LTER TABLE Movies DD Language TEXT		sh';						Stuck? Read this task's Solution. Solve all tasks to continue to the next lesson.
					RUN	QUERY	RESET		Continue >

- 1. ALTER TABLE Movies ADD Aspect\_ratio FLOAT;
- 2. ALTER TABLE Movies ADD Language TEXT DEFAULT 'English';

# **SQL Lesson 18: Dropping tables**



- 1. DROP TABLE IF EXISTS Movies;
- 2. DROP TABLE IF EXISTS BoxOffice;



**SQLBolt**Learn SQL with simple, interactive exercises.

SQL Lesson X: To infinity and beyond!



You've finished the tutorial!