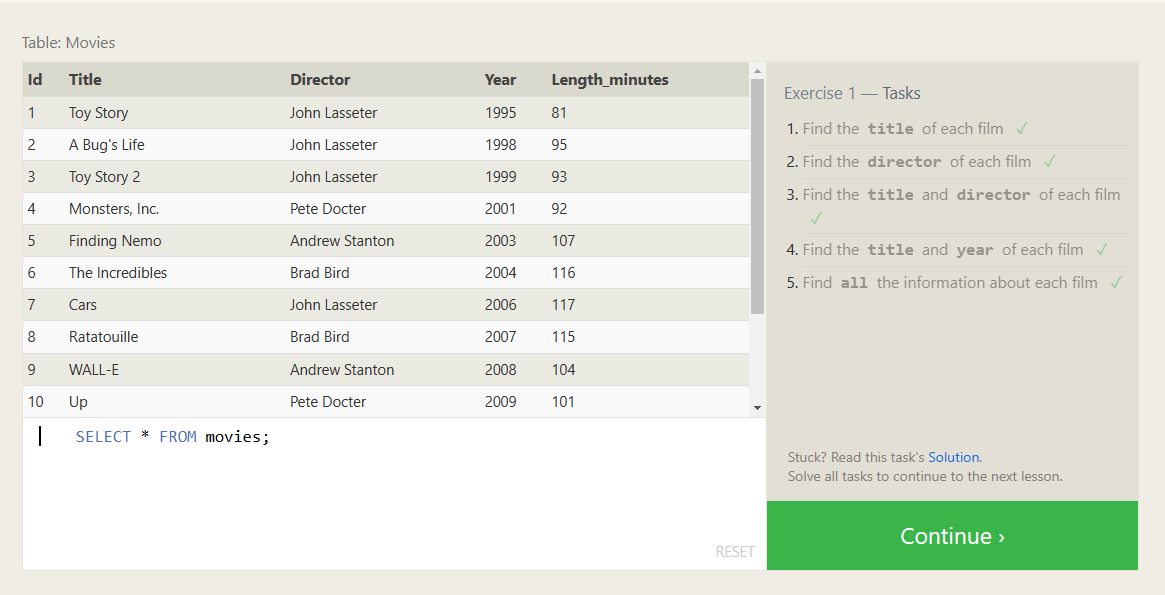
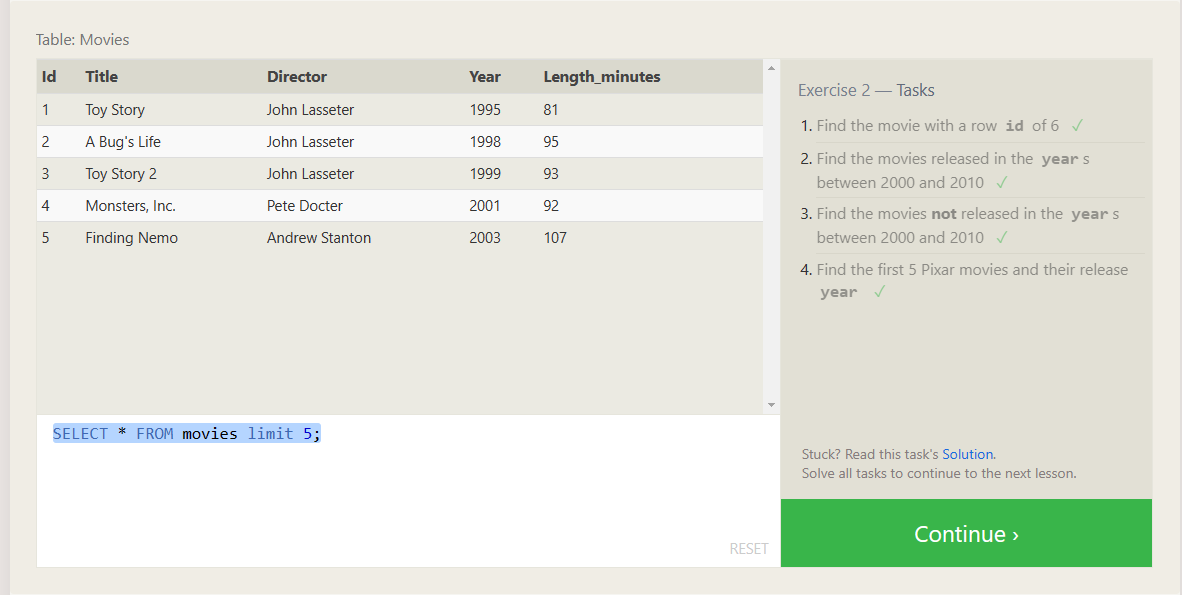
**SQL Bolt – Task:**

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



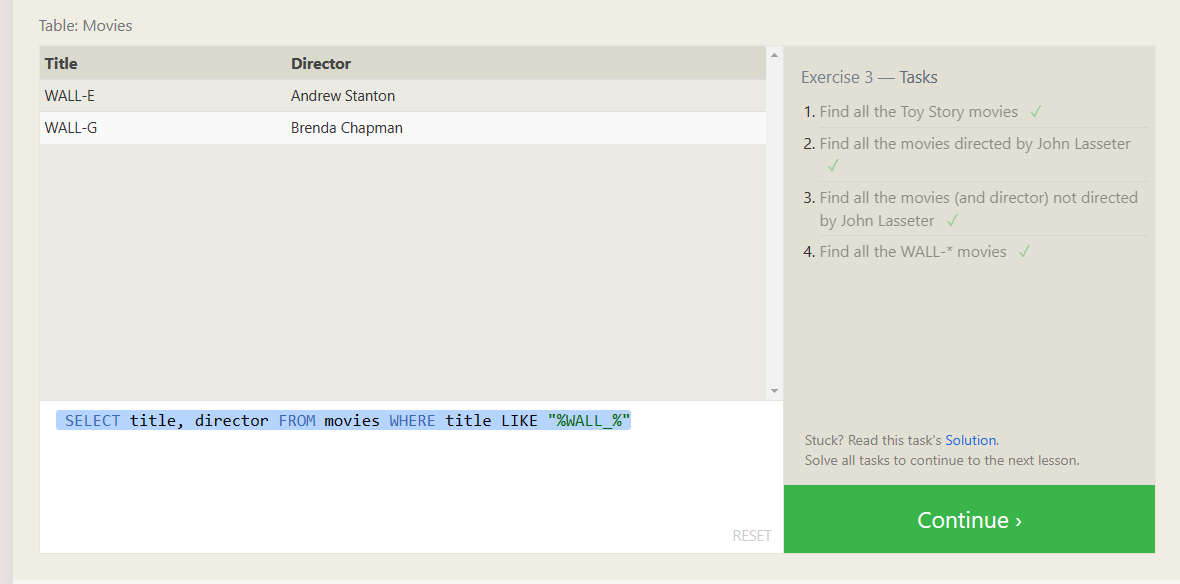
1. SELECT title FROM movies;
2. SELECT director FROM movies;
3. SELECT title, director FROM movies;
4. SELECT title, year FROM movies;
5. SELECT \* FROM movies;

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

****

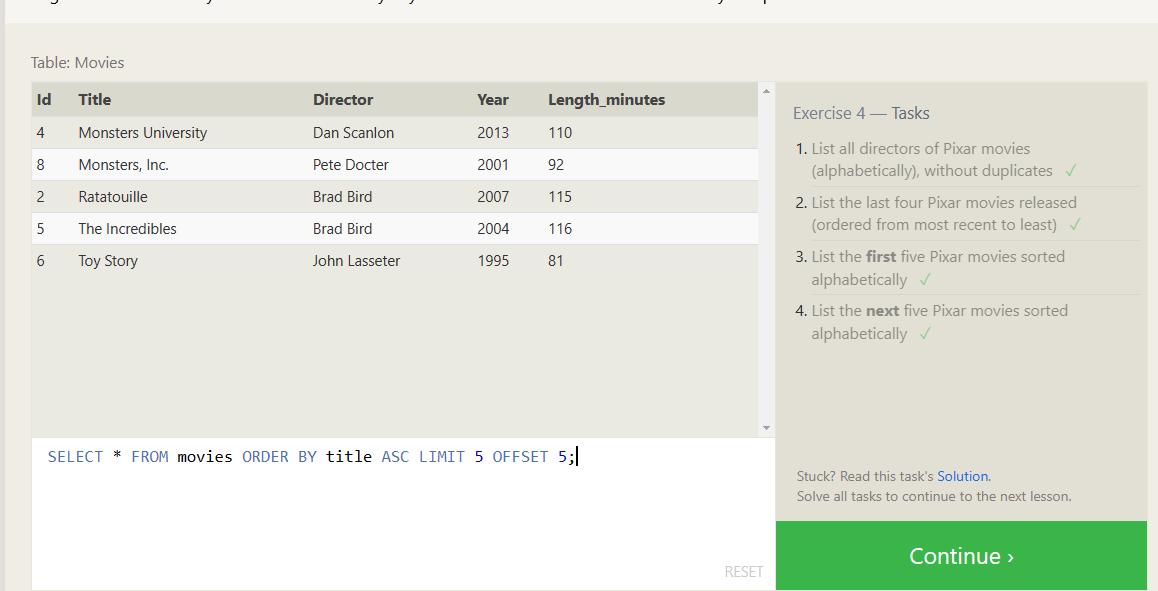
1. SELECT \* FROM movies where id = 6;
2. SELECT \* FROM movies where year between 2000 and 2010;
3. SELECT \* FROM movies where not year between 2000 and 2010;
4. SELECT \* FROM movies limit 5;

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

****

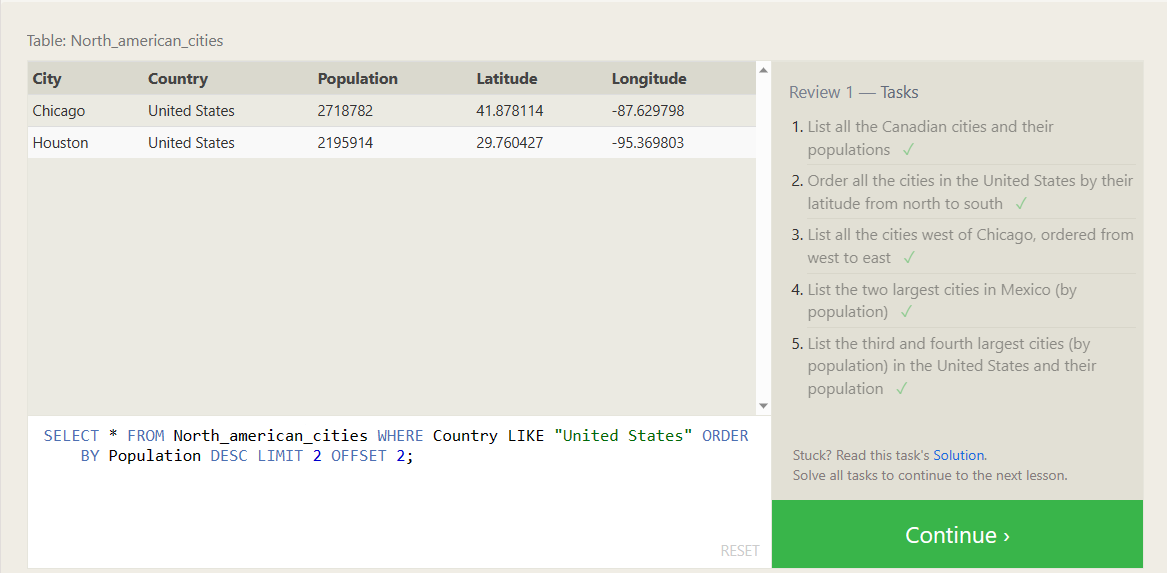
1. SELECT \* FROM movies WHERE title LIKE "%toy story%";
2. SELECT title FROM movies WHERE director = "John Lasseter";
3. SELECT title FROM movies WHERE director != "John Lasseter";
4. SELECT title, director FROM movies WHERE title LIKE "%WALL\_%"

**SQL Lesson 4: Filtering and sorting Query results.**

****

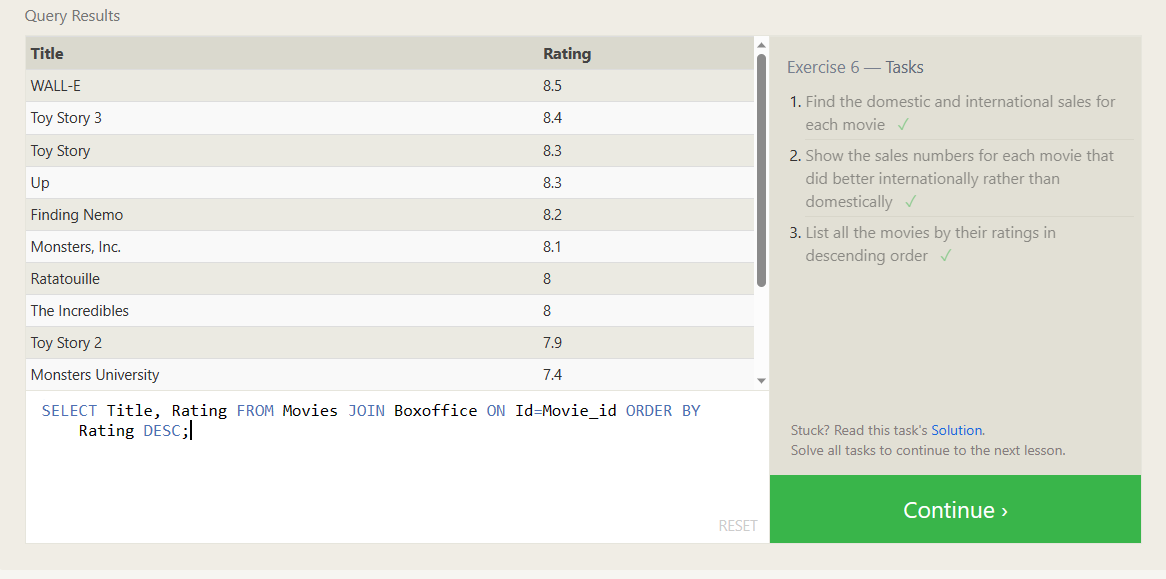
1. SELECT DISTINCT director FROM movies ORDER BY director;
2. SELECT \* FROM movies ORDER BY year DESC LIMIT 4;
3. SELECT \* FROM movies ORDER BY title ASC LIMIT 5;
4. SELECT \* FROM movies ORDER BY title ASC LIMIT 5 OFFSET 5;

**SQL Review Lesson 5: Simple SELECT Queries.**

****

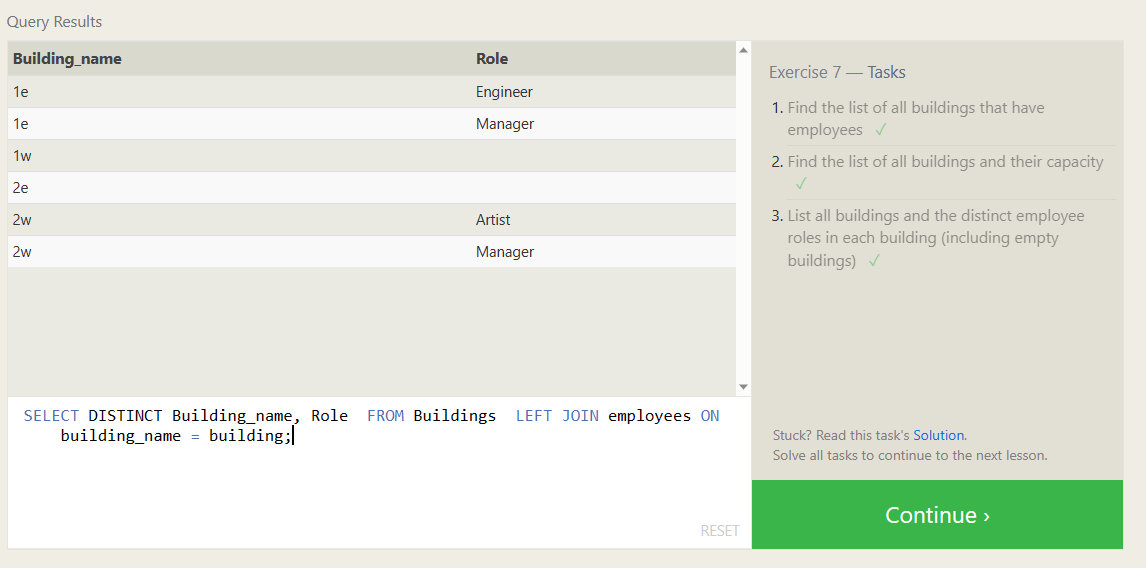
1. SELECT \* FROM North\_american\_cities WHERE Country LIKE "Canada";
2. SELECT \* FROM North\_american\_cities WHERE Country = "United States" ORDER BY Latitude DESC;
3. SELECT \* FROM North\_american\_cities WHERE Longitude < -87.69 ORDER BY Longitude ASC;
4. SELECT \* FROM North\_american\_cities WHERE Country LIKE "Mexico" ORDER BY Population DESC LIMIT 2;
5. SELECT \* FROM North\_american\_cities WHERE Country LIKE "United States" ORDER BY Population DESC LIMIT 2 OFFSET 2;

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



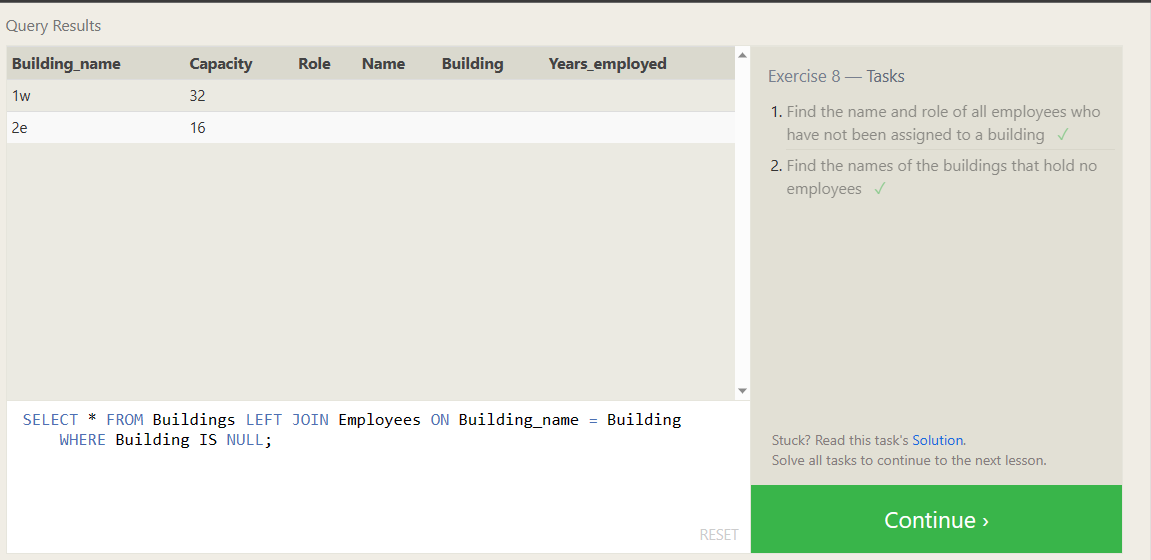
1. SELECT Title, International\_sales, Domestic\_sales FROM Movies JOIN Boxoffice ON Id=Movie\_id;
2. SELECT Title, International\_sales, Domestic\_sales FROM Movies JOIN Boxoffice ON Id=Movie\_id WHERE International\_sales > Domestic\_sales;
3. SELECT Title, Rating FROM Movies JOIN Boxoffice ON Id=Movie\_id ORDER BY Rating DESC;

**SQL Lesson 7: OUTER JOINs**

****

1. SELECT DISTINCT Building FROM Employees LEFT JOIN Buildings ON Building=Building\_name WHERE Years\_employed NOT NULL;
2. SELECT \* FROM Buildings;
3. SELECT DISTINCT Building\_name, Role FROM Buildings LEFT JOIN employees ON building\_name = building;

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

****

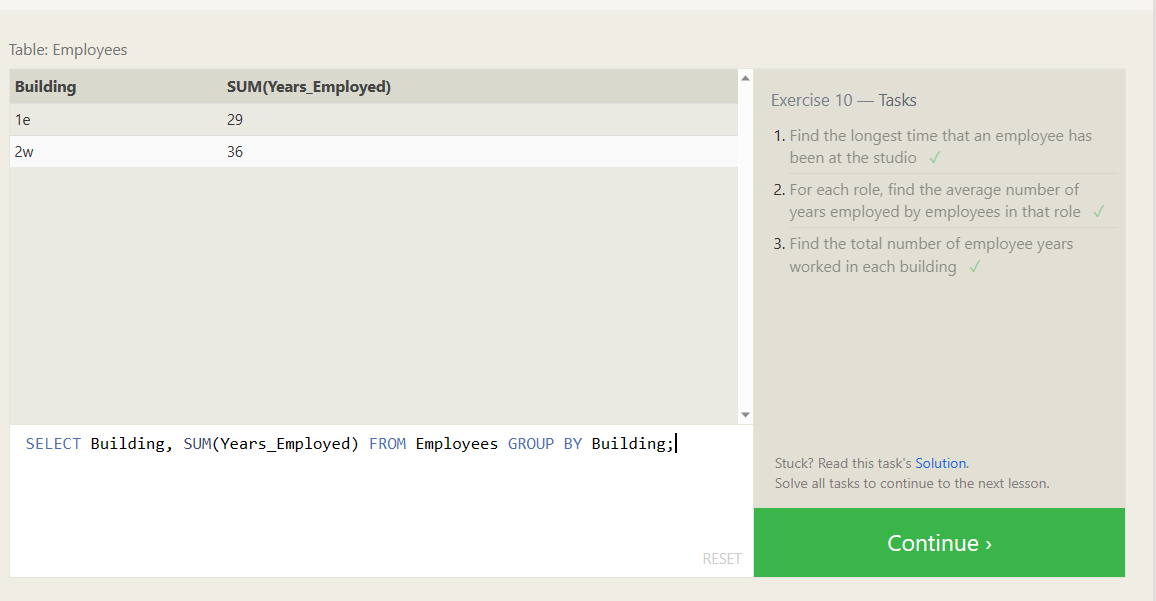
1. SELECT \* FROM Employees LEFT JOIN Buildings ON Building\_name = Building WHERE Building IS NULL;
2. SELECT \* FROM Buildings LEFT JOIN Employees ON Building\_name = Building WHERE Building IS NULL;

**SQL Lesson 9: Queries with expressions**

****

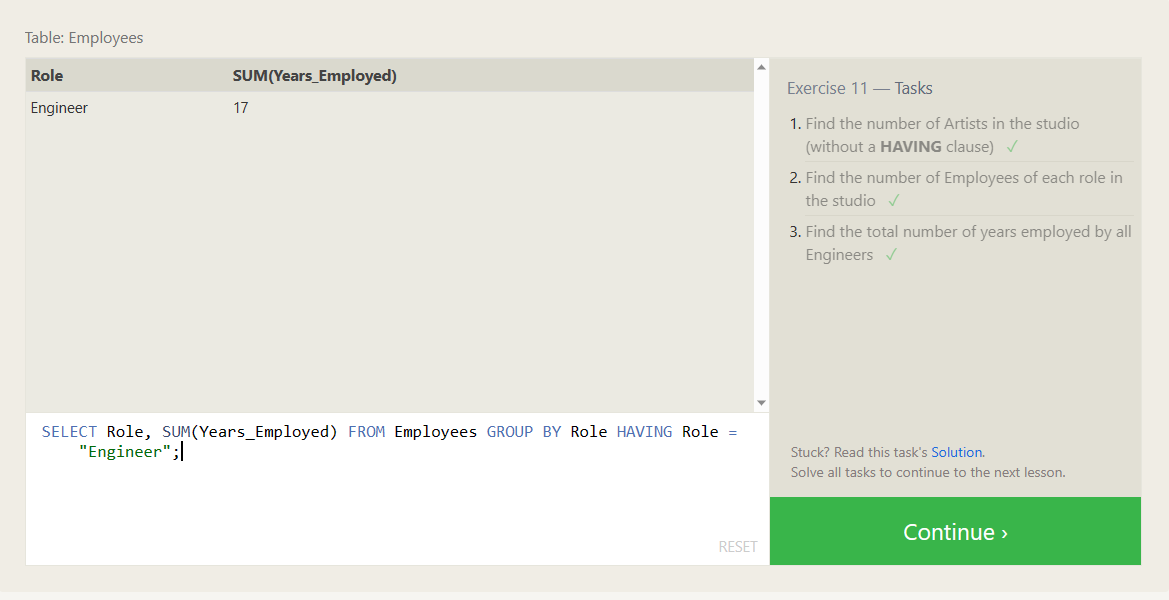
1. SELECT Title, (Domestic\_sales + International\_sales)/1000000 AS Total\_Sales\_Millions FROM Movies LEFT JOIN Boxoffice ON Id=Movie\_Id;
2. SELECT Title, Rating\*10 as Percent FROM Movies LEFT JOIN Boxoffice ON Id=Movie\_Id;
3. SELECT Title, Year FROM Movies LEFT JOIN Boxoffice ON Id=Movie\_Id WHERE Year % 2 = 0;

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

****

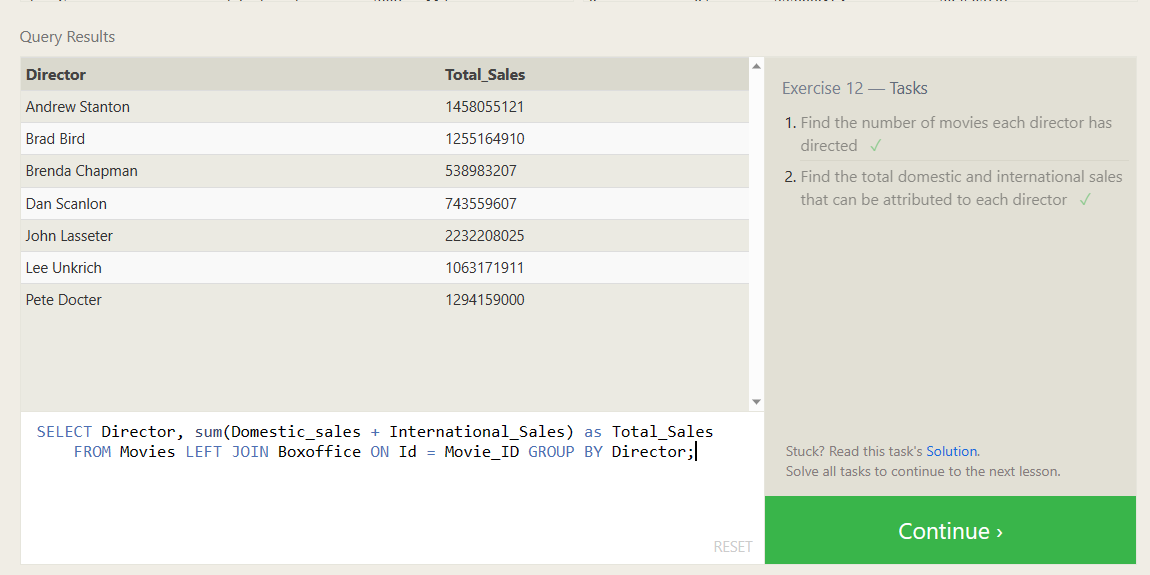
1. SELECT MAX(Years\_employed) FROM Employees;
2. SELECT Role, AVG(Years\_Employed) FROM Employees GROUP BY Role;
3. SELECT Building, SUM(Years\_Employed) FROM Employees GROUP BY Building;

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

****

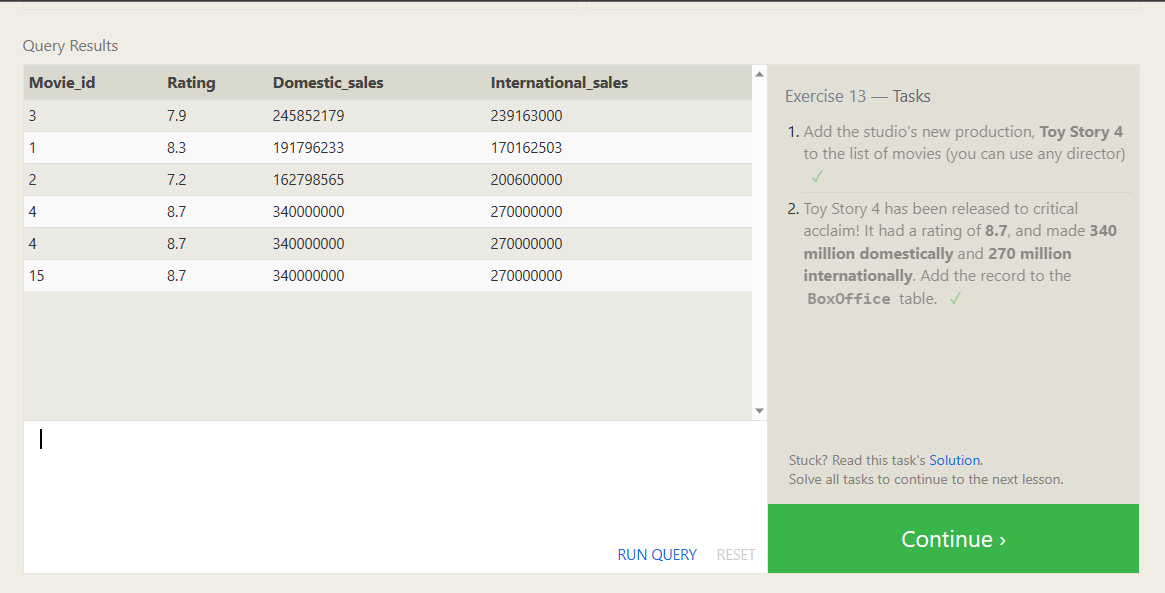
1. SELECT Role, COUNT(\*) AS Number\_of\_Artists FROM Employees WHERE Role = "Artist";
2. SELECT Role, COUNT(\*) FROM Employees GROUP BY Role;
3. SELECT Role, SUM(Years\_Employed) FROM Employees GROUP BY Role HAVING Role = "Engineer";

**SQL Lesson 12: Order of execution of a Query**

****

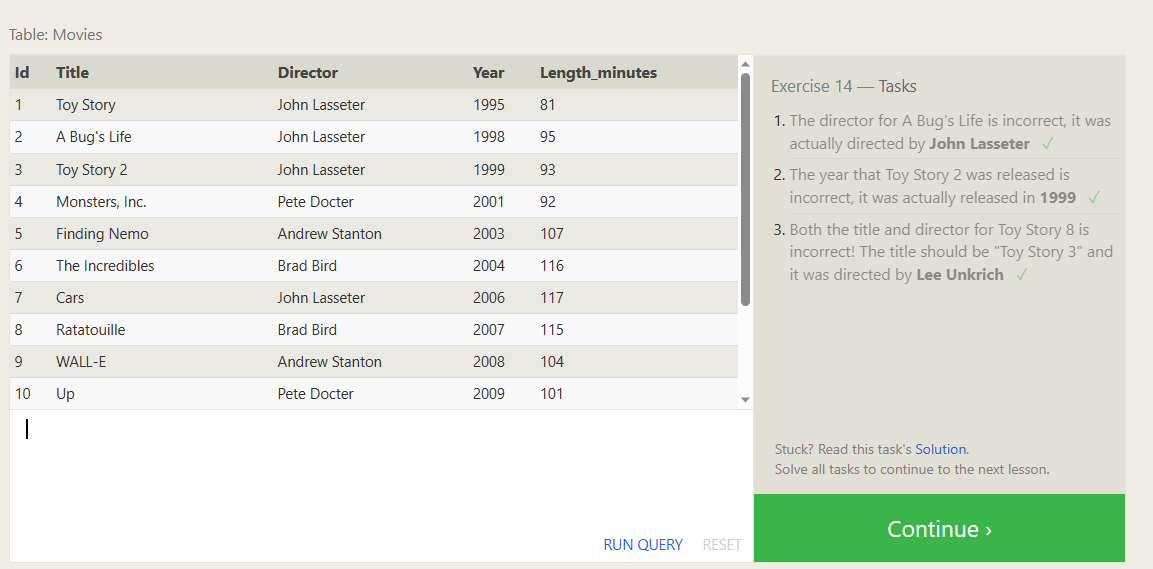
1. SELECT \*, COUNT(Title) FROM Movies GROUP BY Director;
2. SELECT Director, sum(Domestic\_sales + International\_Sales) as Total\_Sales FROM Movies LEFT JOIN Boxoffice ON Id = Movie\_ID GROUP BY Director;

**SQL Lesson 13: Inserting rows**

****

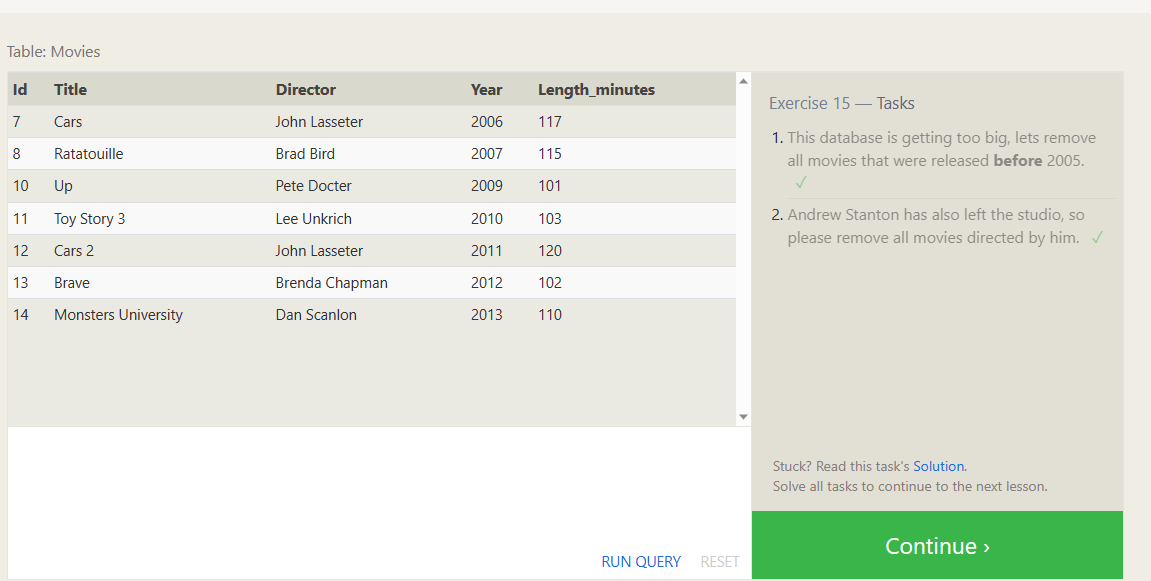
1. INSERT INTO Movies(Title,Director) VALUES ('Toy Story 4','John Lasseter')
2. INSERT INTO Boxoffice VALUES (15,8.7,340000000, 270000000)

**SQL Lesson 14: Updating rows**

****

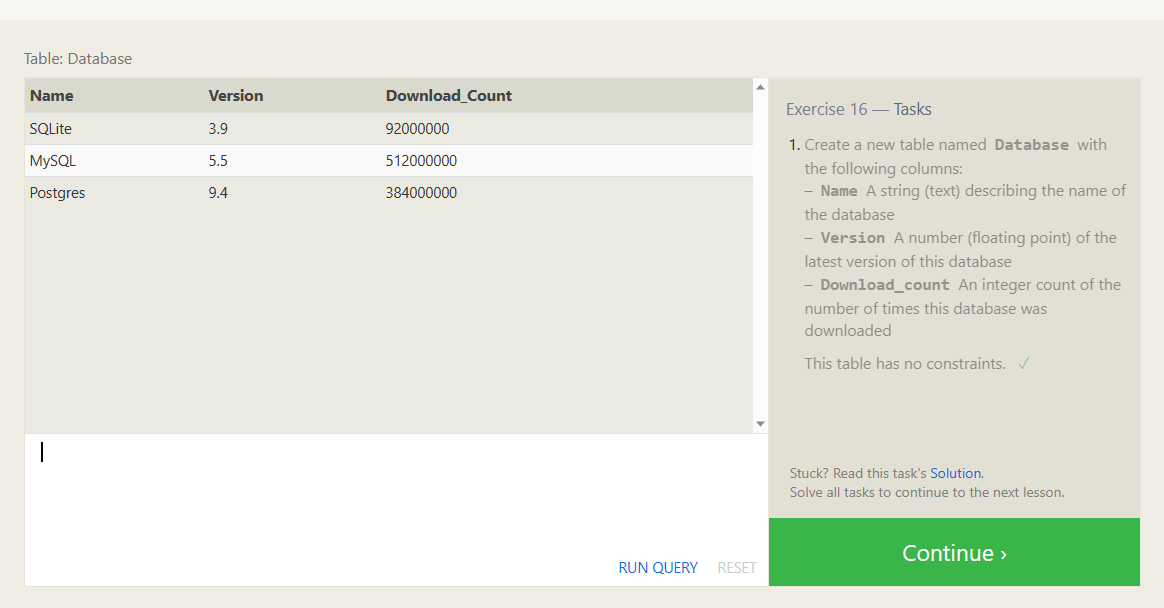
1. UPDATE Movies SET Director = "John Lasseter" WHERE Id = 2;
2. UPDATE Movies SET Year = "1999" WHERE Id = 3;
3. UPDATE Movies SET Title = "Toy Story 3", Director = "Lee Unkrich" WHERE Id = 11;

**SQL Lesson 15: Deleting rows**

****

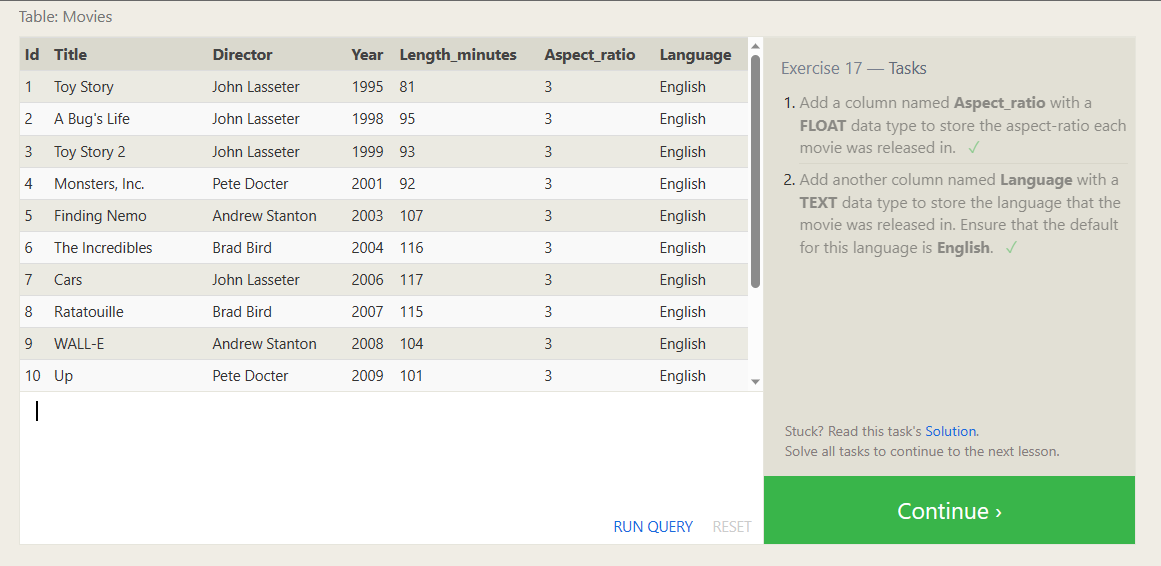
1. DELETE FROM Movies WHERE Year < 2005;
2. DELETE FROM Movies WHERE Director = "Andrew Stanton";

**SQL Lesson 16: Creating tables**

****

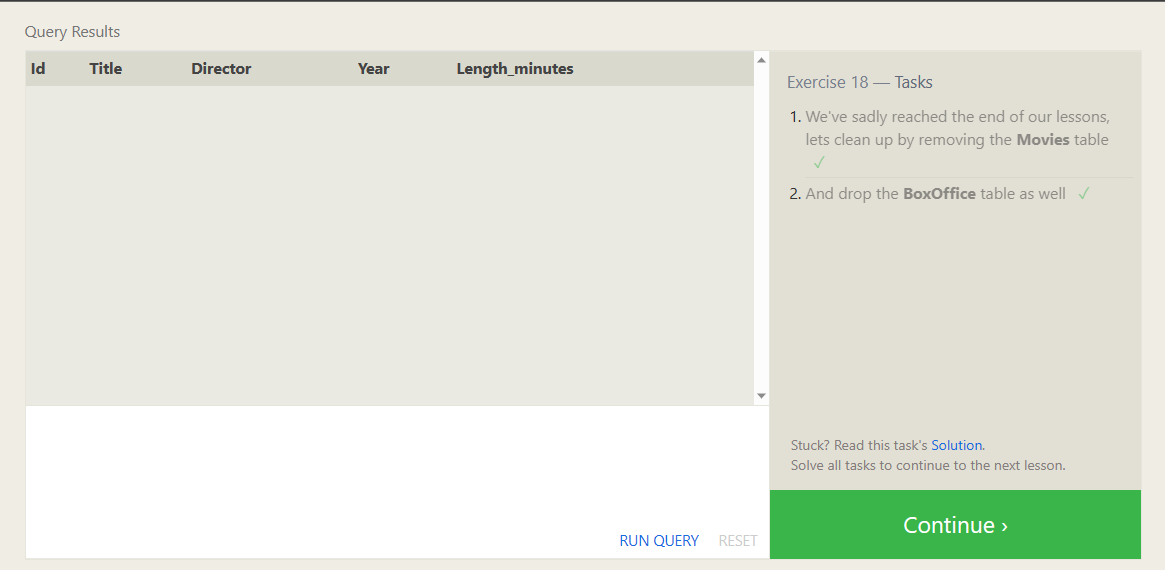
1. CREATE TABLE Database (Name TEXT, Version FLOAT, Download\_Count INTEGER);

**SQL Lesson 17: Altering tables**

****

1. ALTER TABLE Movies ADD COLUMN Aspect\_ratio FLOAT DEFAULT 3;
2. ALTER TABLE Movies ADD COLUMN Language TEXT DEFAULT "English";

**SQL Lesson 18: Dropping tables**

****

1. DROP TABLE Movies;
2. DROP TABLE BoxOffice;

