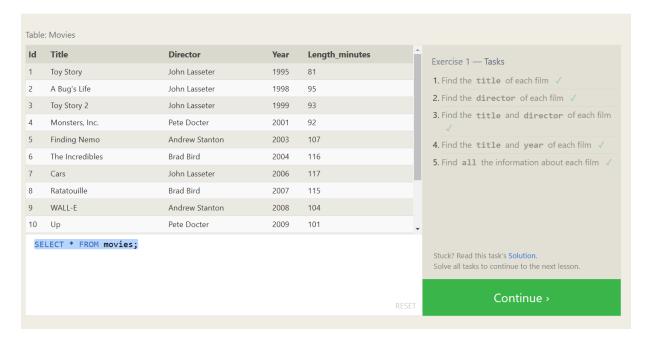
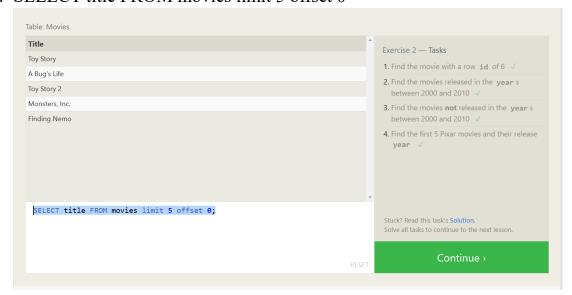
LESSION-1:

- 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;



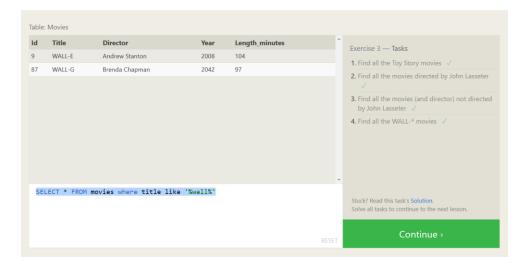
LESSION-2

- 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 FROM movies limit 5 offset 0



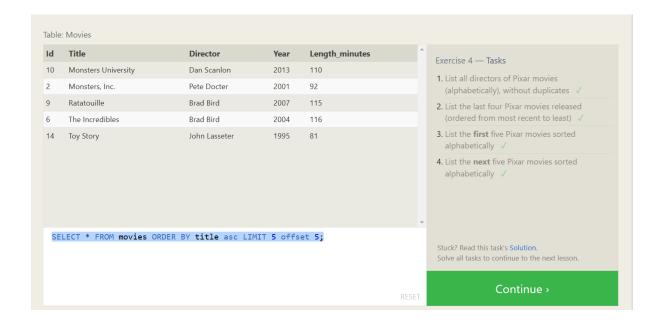
LESSION-3:

- 1. SELECT * FROM movies where title like '%Toy Story%';
- 2. SELECT * FROM movies where director like '%john lasseter%';
- 3. SELECT * FROM movies where director not like '%john lasseter%';
- 4. SELECT * FROM movies where title like '%wall%'



LESSION-4

- 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;



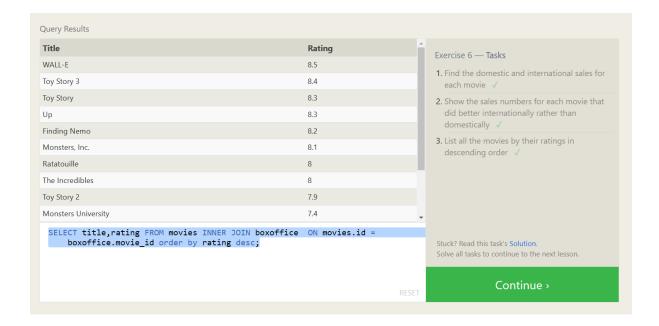
LESSION-5:

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



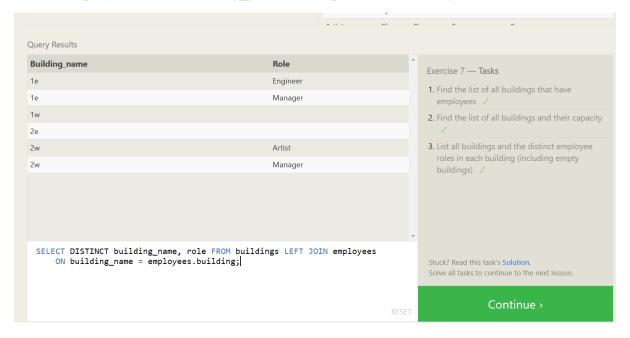
LESSION-6:

- 1. SELECT title, domestic_sales, international_sales FROM movies INNER JOIN boxoffice ON movies.id = boxoffice.movie id;
- SELECT title, domestic_sales, international_sales FROM movies INNER
 JOIN boxoffice ON movies.id = boxoffice.movie_id where
 domestic sales<international sales;
- 3. SELECT title,rating FROM movies INNER JOIN boxoffice ON movies.id = boxoffice.movie id order by rating desc;



LESSION-7:

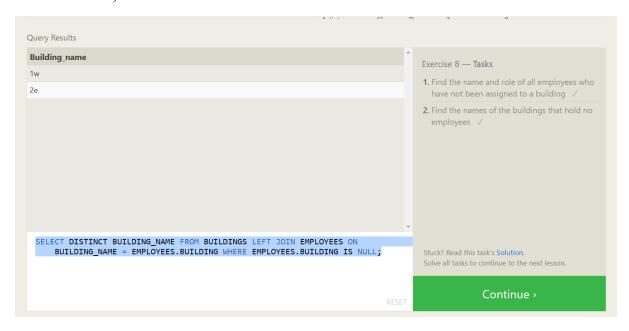
- 1. SELECT distinct building FROM employees;
- 2. SELECT * FROM buildings;
- 3. SELECT DISTINCT building_name, role FROM buildings LEFT JOIN employees ON building name = employees.building;



LESSION-8:

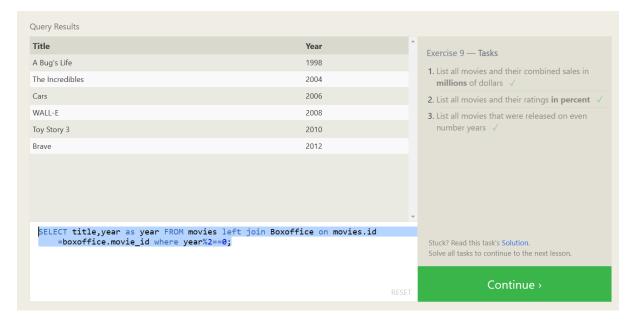
1. SELECT NAME, ROLE FROM employees WHERE BUILDING IS NULL;

2. SELECT DISTINCT BUILDING_NAME FROM BUILDINGS LEFT JOIN EMPLOYEES ON BUILDING_NAME = EMPLOYEES.BUILDING WHERE EMPLOYEES.BUILDING IS NULL;



LESSION-9:

- SELECT title,(domestic_sales+international_sales)/1000000 as combinedsales FROM movies left join Boxoffice on movies.id=boxoffice.movie id;
- 2. SELECT title,rating * 10 as present FROM movies left join Boxoffice on movies.id=boxoffice.movie id;
- 3. SELECT title, year as year FROM movies left join Boxoffice on movies.id=boxoffice.movie_id where year%2==0;



LESSION-10:

- 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;



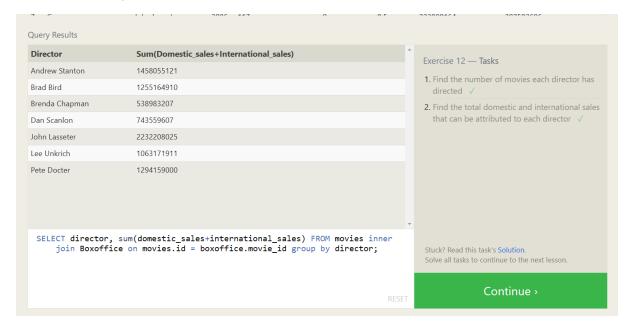
LESSION-11:

- 1. SELECT count(*) FROM employees where role = "Artist";
- 2. SELECT role,count(*) FROM employees group by role;
- 3. SELECT sum(years_employed) from employees where role ='Engineer';



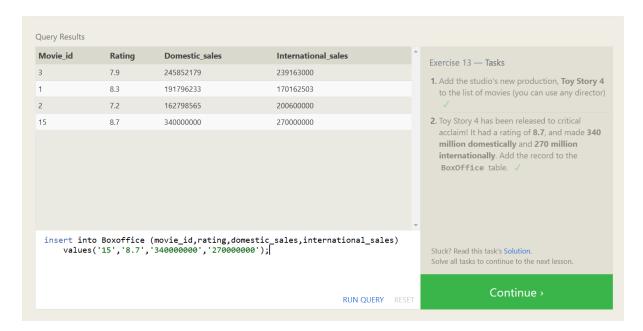
LESSION-12:

- 1. SELECT count(*)title, director FROM movies group by director;
- 2. SELECT director, sum(domestic_sales+international_sales) FROM movies inner join Boxoffice on movies.id = boxoffice.movie_id group by director;



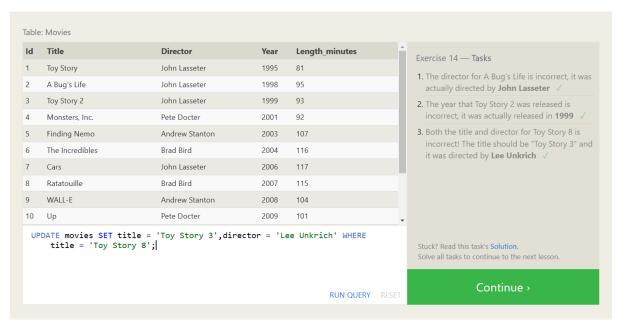
LESSION-13:

- 1. insert into Movies (title,director,year,length_minutes) values('Toy Story 4','John Lasseter',' 1999','81');
- 2. insert into Boxoffice (movie_id,rating,domestic_sales,international_sales) values('15','8.7','340000000','270000000')



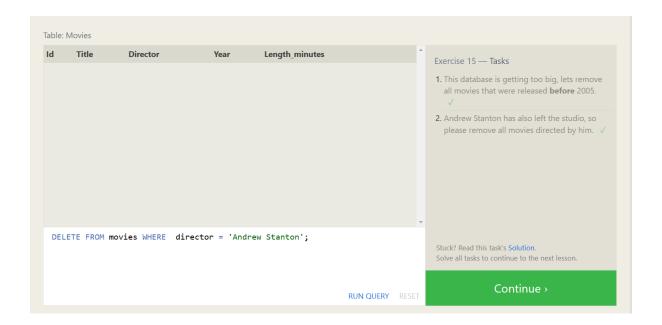
LESSION-14:

- 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';



LESSION-15:

- 1. Delete FROM movies where year < 2005;
- 2. DELETE FROM movies WHERE director = 'Andrew Stanton';



LESSION-16:

1. create table Database(

Name varchar(255),

Version int,

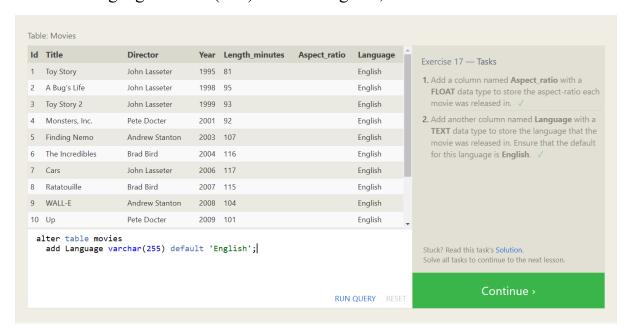
Download count int

)

lame	Version	Download_count		^	Exercise 16 — Tasks
QLite	3.9	92000000			1 Create a new table named Database with
lySQL	5.5	512000000			the following columns:
ostgres	9.4	38400000		*	 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. ✓
create tabl Name varcha Version int Download_co	,				Stuck? Read this task's Solution . Solve all tasks to continue to the next lesson.
)			RUN QUERY R		Continue >

LESSION-17:

- alter table movies
 add aspect_ratio float;
- 2. alter table moviesadd Language varchar(255) default 'English';



LESSION-18:

- 1. drop table movies;
- 2. drop table Boxoffice;

	Results Title	Director	Year	Laureth minutes		Α.	
d	Title	Director	Year	Length_minutes			Exercise 18 — Tasks
							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 \checkmark
dro	p table Bo	xoffice;				~	Stuck? Read this task's Solution. Solve all tasks to continue to the next lesson.
					RUN QUERY	RESET	Continue >