

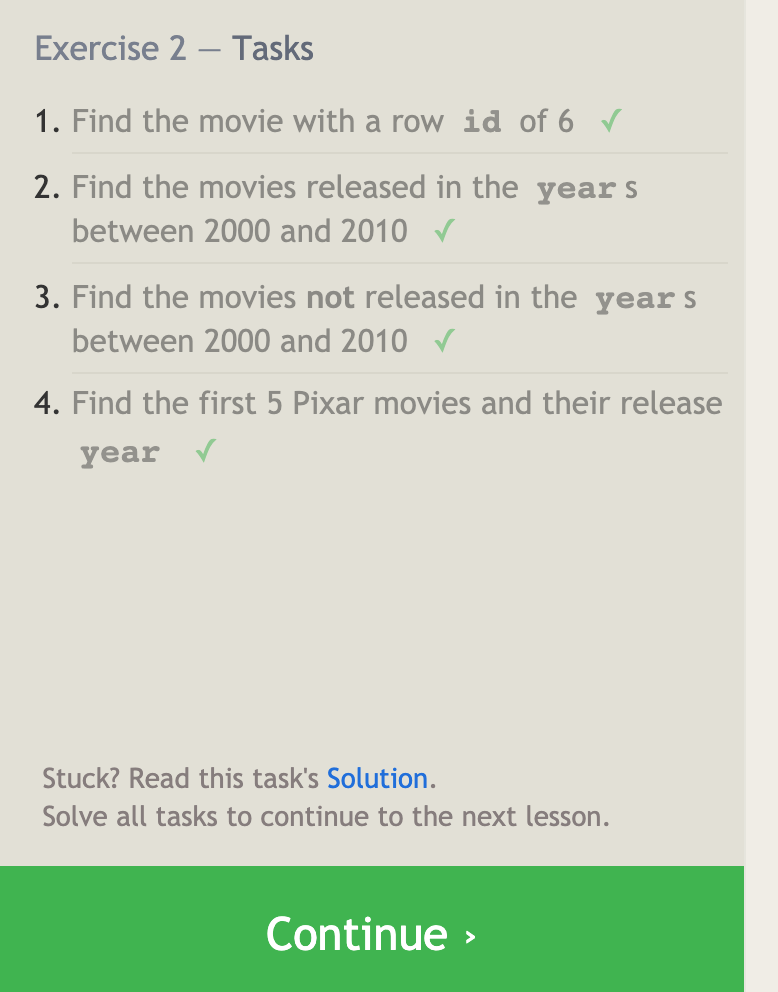
SELECT title FROM movies;

SELECT director FROM movies;

SELECT title,director FROM movies;

SELECT title,year FROM movies;

SELECT \* FROM movies;



SELECT \* FROM movies where id==6;

SELECT \* FROM movies where (year >=2000 AND year<=2010) ;

SELECT \* FROM movies where (year<2000 OR year>2010) ;

SELECT \* FROM movies WHERE year <= 2003;

Graphical user interface, text, application, email

Description automatically generated

SELECT \* FROM movies where title LIKE "%toy%";

SELECT \* FROM movies where director=='John Lasseter';

SELECT \* FROM movies where director!='John Lasseter';

SELECT \* FROM movies where title LIKE 'WALL-%';

Graphical user interface, text

Description automatically generated

SELECT DISTINCT director FROM movies

order by director asc;

select \* from movies order by year desc limit 4;

select \* from movies order by title asc limit 5;

select \* from movies order by title asc limit 5 offset 5;

Graphical user interface, text, application, email

Description automatically generated

SELECT city,population FROM north\_american\_cities

where country=='Canada';

SELECT \* FROM north\_american\_cities

where country=='United States'

order by latitude desc

SELECT city, longitude FROM north\_american\_cities

WHERE longitude < -87.629798

ORDER BY longitude ASC;

SELECT city FROM north\_american\_cities

where country=='Mexico'

order by population desc limit 2

SELECT \* FROM north\_american\_cities

where country=='United States'

order by population desc limit 2 offset 2

Graphical user interface, text, application, email

Description automatically generated

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 International\_sales>Domestic\_sales;

SELECT title,Domestic\_sales,International\_sales

FROM movies

INNER JOIN Boxoffice

ON movies.id == Boxoffice.Movie\_id

order by Rating desc;

Graphical user interface, text, application, email

Description automatically generated

SELECT DISTINCT Building

FROM Buildings

LEFT JOIN Employees

ON Buildings.Building\_name == Employees.Building

where Building IS NOT NULL;

SELECT Building\_name,capacity

FROM Buildings

SELECT DISTINCT Building\_name,ROLE

FROM Buildings

LEFT JOIN Employees

ON Buildings.Building\_name == Employees.Building ;

Graphical user interface, text, application, email

Description automatically generated

SELECT ROLE,NAME,BUILDING FROM EMPLOYEES

WHERE BUILDING IS NULL

SELECT \*

from Buildings

LEFT JOIN Employees

ON Buildings.Building\_name == Employees.Building

WHERE Building is NULL;

Graphical user interface, text, application, email

Description automatically generated

SELECT title,

(Domestic\_sales+International\_sales)/1000000 as Total\_sales FROM movies

INNER JOIN Boxoffice

ON movies.Id == Boxoffice.Movie\_id;

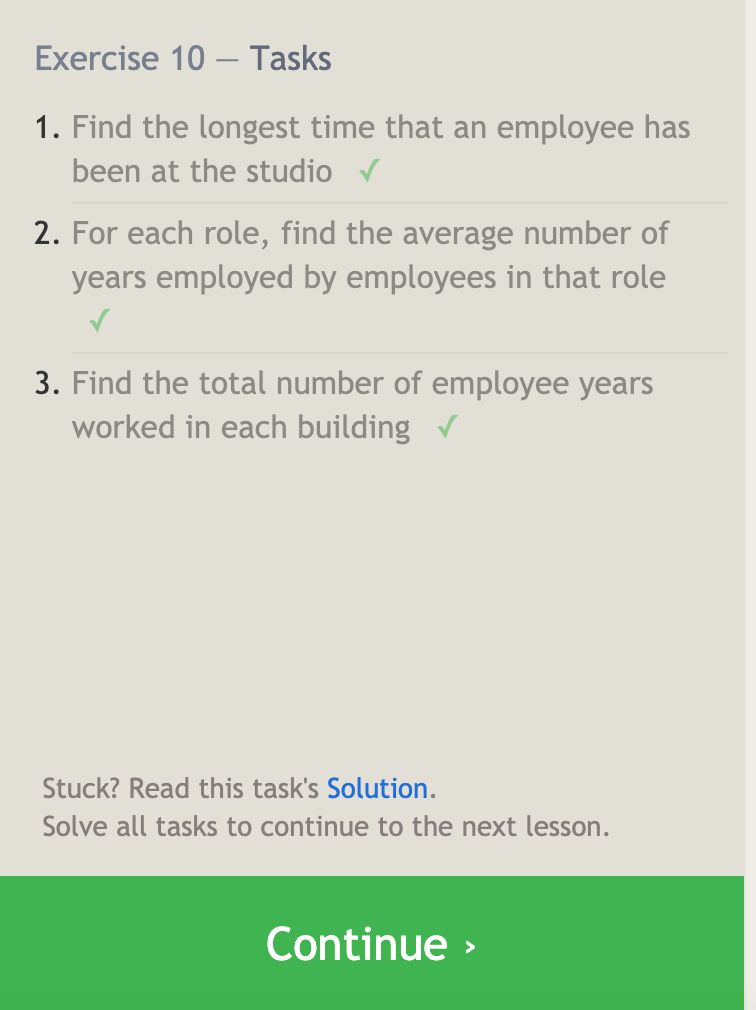
SELECT title,

rating\*10 as Ratings FROM movies

INNER JOIN Boxoffice

ON movies.Id == Boxoffice.Movie\_id;

SELECT title, year as EVEN\_Years FROM movies where year%2 == 0



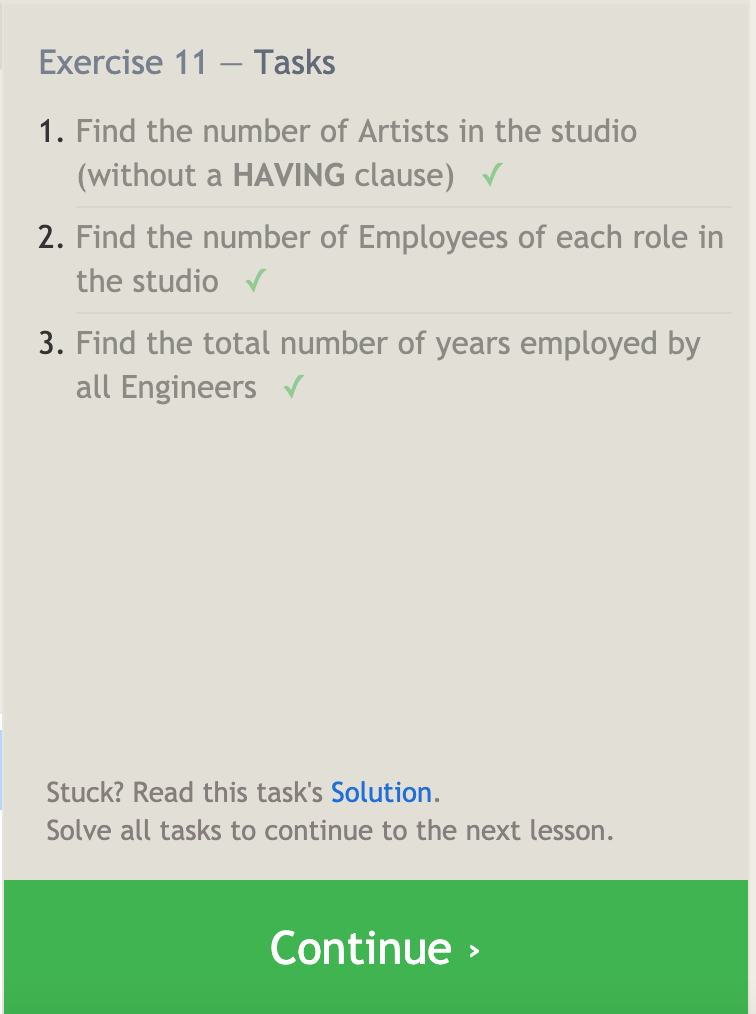
SELECT MAX(Years\_employed) FROM employees;

SELECT role, AVG(Years\_employed) from Employees

GROUP BY role

SELECT Building, SUM(Years\_employed) from Employees

GROUP BY Building



SELECT COUNT() FROM employees

WHERE Role =='Artist';

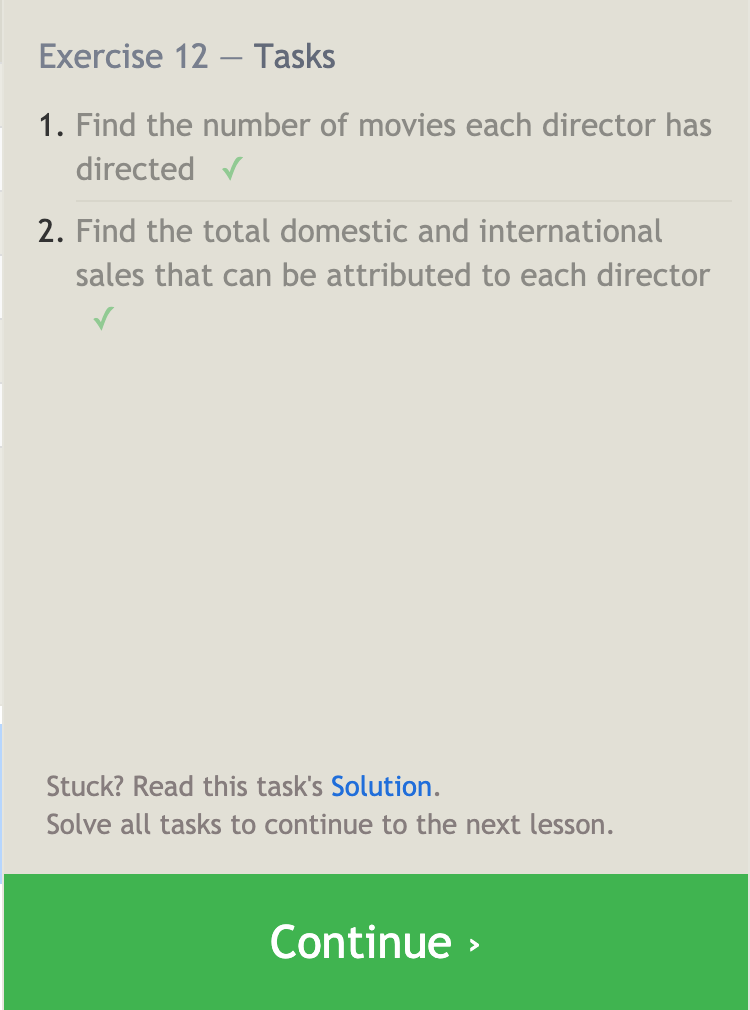
SELECT Role, COUNT() as Count from Employees

GROUP BY Role

SELECT Role, SUM(Years\_employed) as Total\_years from Employees

GROUP BY Role

HAVING Role=='Engineer'



SELECT Director,COUNT() as COUNT FROM movies

GROUP BY Director;

SELECT Director,SUM(Domestic\_sales+International\_sales) as Total\_sales

FROM movies

INNER JOIN Boxoffice

ON movies.id==Boxoffice.Movie\_id

GROUP BY Director;

Text, letter

Description automatically generated

INSERT INTO movies

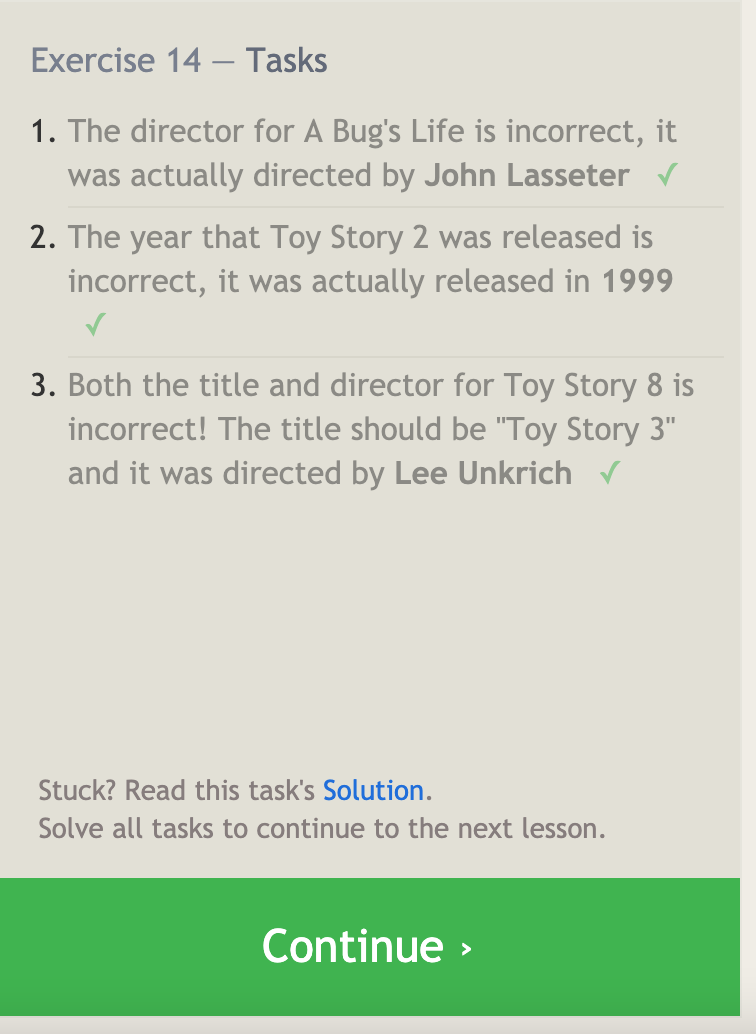
(Title,Director,Year,Length\_minutes)

VALUES('Toy Story 4','John Lasseter',2021, 100);

INSERT INTO Boxoffice

(Movie\_id,Rating,Domestic\_sales,International\_sales)

VALUES((select Id from movies where title=='Toy Story 4'),8.7,340\*1000000,279\*1000000);



UPDATE Movies

SET Director='John Lasseter'

WHERE Title IS "A Bug's Life"

UPDATE Movies

SET Year='1999'

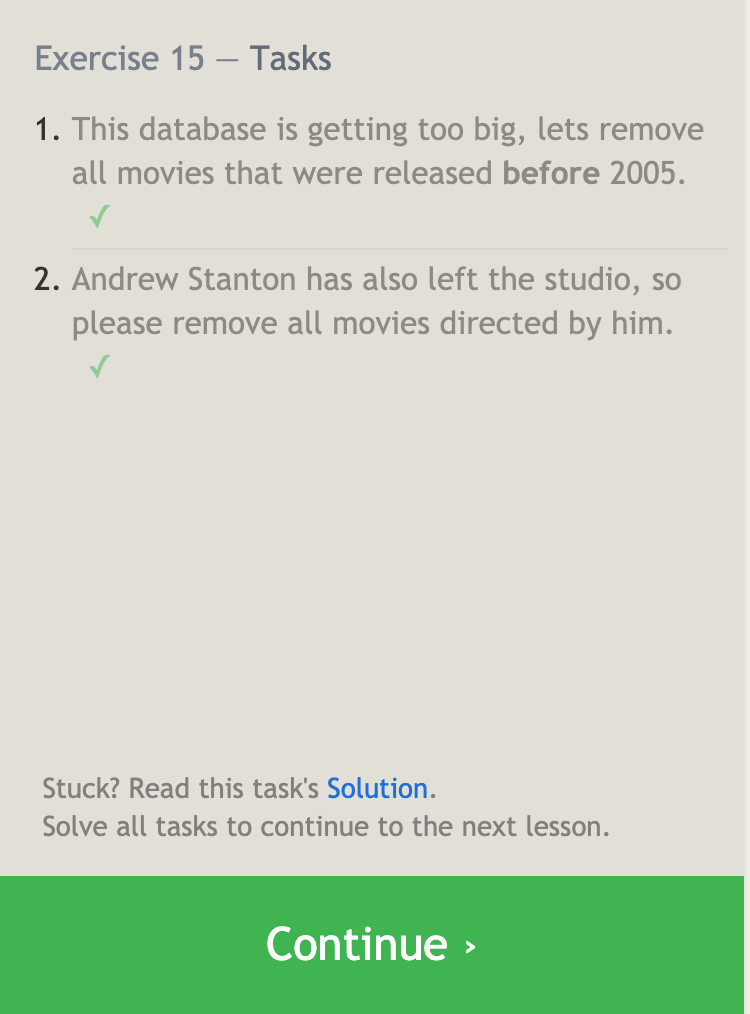
WHERE Title IS "Toy Story 2"

UPDATE Movies

SET Title='Toy Story 3',

Director='Lee Unkrich'

WHERE Title IS "Toy Story 8"

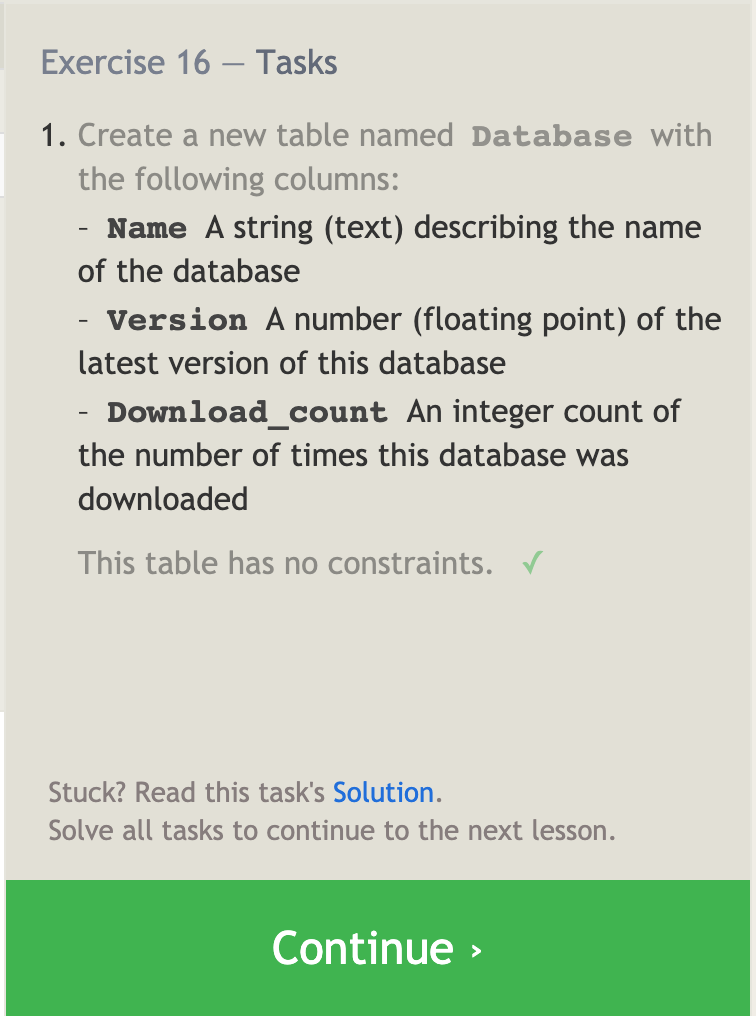


DELETE FROM movies

WHERE Year<2005;

DELETE FROM movies

WHERE Director IS 'Andrew Stanton';



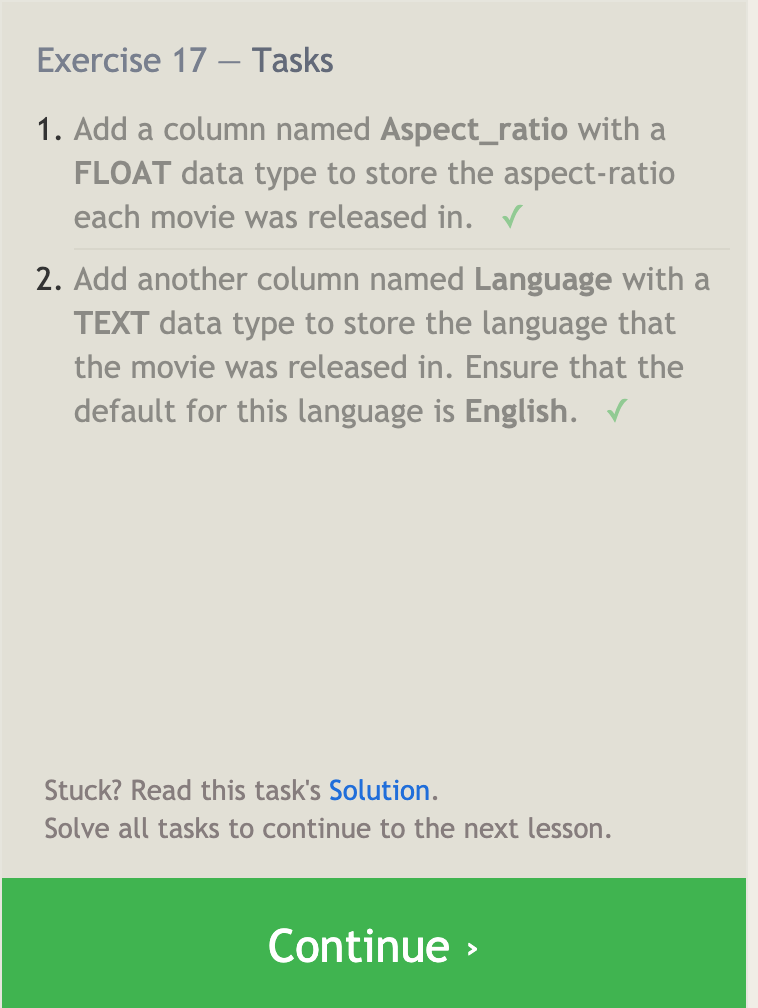
CREATE TABLE IF NOT EXISTS Database(

Name TEXT,

Version FLOAT,

Download\_count INTEGER

)



ALTER Table Movies

ADD Aspect\_ratio FLOAT;

ALTER Table Movies

ADD Language TEXT DEFAULT 'English';

Graphical user interface, text, application, email

Description automatically generated

DROP Table IF EXISTS Movies

DROP Table IF EXISTS Boxoffice