

SQL Bolt Tutorial

<https://sqlbolt.com/>

Table: Movies (Read-Only)

<u>Id</u>	<u>Title</u>	<u>Director</u>	<u>Year</u>	<u>Length_minutes</u>
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
...

Table: Boxoffice (Read-Only)

<u>Movie_id</u>	<u>Rating</u>	<u>Domestic_sales</u>	<u>International_sales</u>
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
...

Find the domestic and international sales for each movie

Title	Domestic_sales	International_sales
Toy Story	191796233	170162503
A Bug's Life	162798565	200600000
Toy Story 2	245852179	239163000
Monsters, Inc.	289916256	272900000
Finding Nemo	380843261	555900000
The Incredibles	261441092	370001000
Cars	244082982	217900167
Ratatouille	206445654	417277164
WALL-E	223808164	297503696
Up	293004164	438338580

```
select Movies.title, boxoffice.domestic_sales, boxoffice.international_sales  
from Movies left join Boxoffice  
on Movies.Id = boxoffice.movie_id;
```

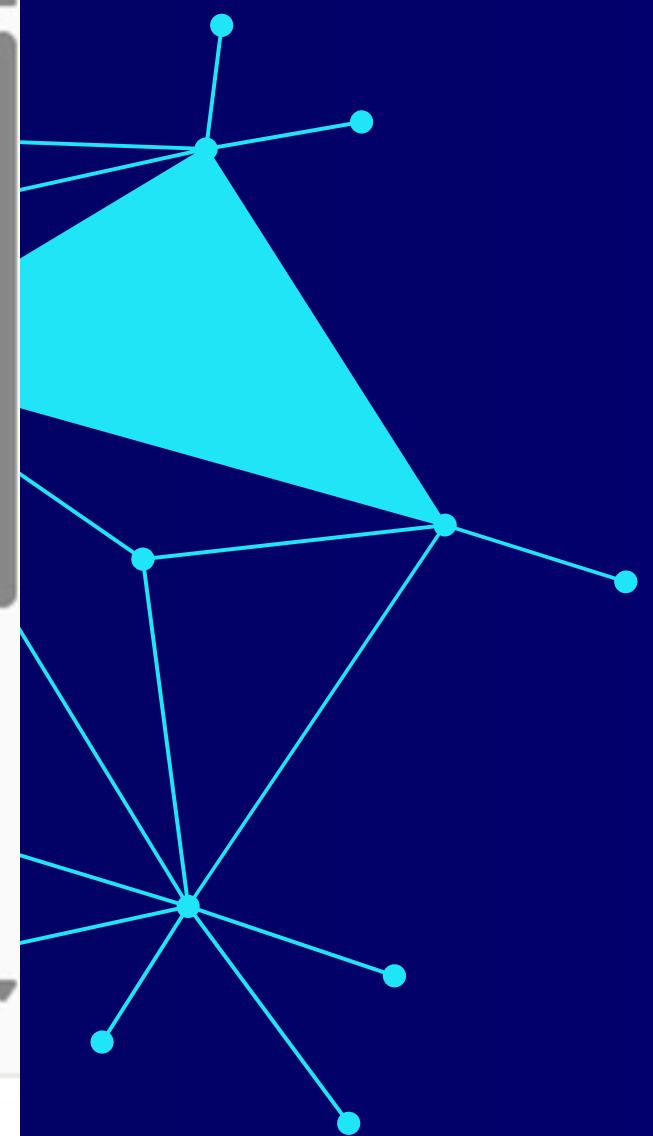
Show the sales numbers for each movie that did better internationally rather than domestically

Query Results

Id	Title	Director	Year	Length_minutes	Movie_id	Rating	Domestic_sales	International_sales
5	Finding Nemo	Andrew Stanton	2003	107	5	8.2	380843261	555900000
14	Monsters University	Dan Scanlon	2013	110	14	7.4	268492764	475066843
8	Ratatouille	Brad Bird	2007	115	8	8	206445654	417277164
12	Cars 2	John Lasseter	2011	120	12	6.4	191452396	368400000
6	The Incredibles	Brad Bird	2004	116	6	8	261441092	370001000
9	WALL-E	Andrew Stanton	2008	104	9	8.5	223808164	297503696

```
select * from Movies join Boxoffice  
on Movies.Id = boxoffice.movie_id  
where international_sales > domestic_sales  
;
```

RESET



List all the movies by their ratings in descending order

Query Results

Id	Title	Director	Year	Length_minutes	Movie_id	Rating	Domestic_sales	International_sales
9	WALL-E	Andrew Stanton	2008	104	9	8.5	223808164	297503696
11	Toy Story 3	Lee Unkrich	2010	103	11	8.4	415004880	648167031
1	Toy Story	John Lasseter	1995	81	1	8.3	191796233	170162503
10	Up	Pete Docter	2009	101	10	8.3	293004164	438338580
5	Finding Nemo	Andrew Stanton	2003	107	5	8.2	380843261	555900000
4	Monsters, Inc.	Pete Docter	2001	92	4	8.1	289916256	272900000

```
select * from Movies join Boxoffice  
on Movies.Id = boxoffice.movie_id  
order by rating desc  
;
```

RESET

Table: Buildings (Read-Only)

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

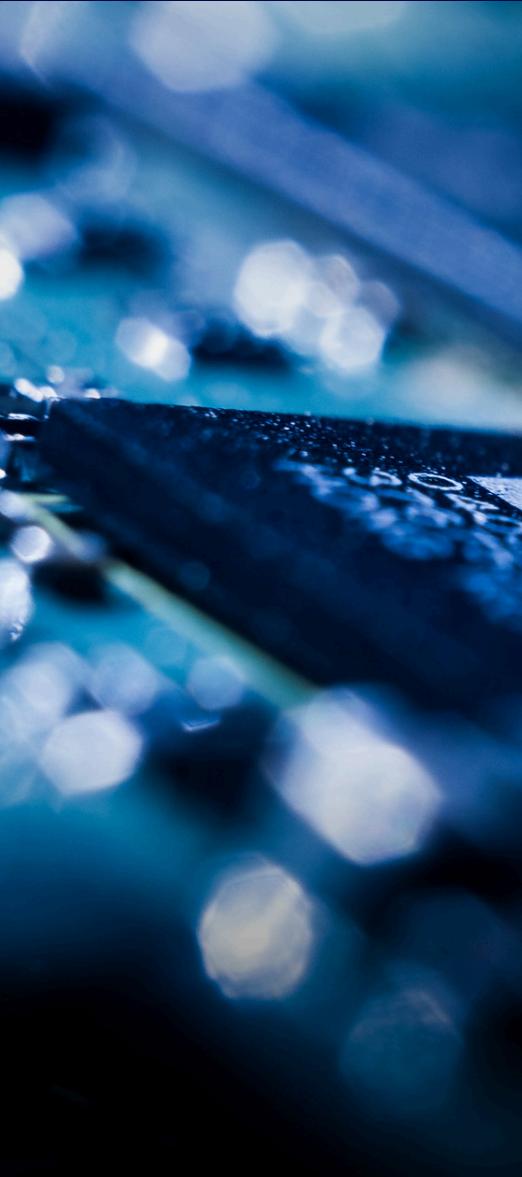
Query Results

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

Find the list of all buildings that have employees

Page 07



Query Results

Building
1e
2w

```
select distinct building from Employees; |
```

RESET

Find the list of all buildings and their capacity

Query Results

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

```
select * from Buildings;
```



List all buildings and the distinct employee roles in each building (including empty buildings)

Query Results

Building_name	Role
1e	Engineer
1e	Manager
1w	
2e	
2w	Artist
2w	Manager

```
SELECT DISTINCT building_name, role FROM buildings LEFT JOIN employees ON building_name = building;
```

RESET

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

Name	Role
Yancy I.	Engineer
Oliver P.	Artist

```
SELECT Name, Role from Employees where Building is null
```

Find the names of the buildings that hold no employees

Query Results

Building_name

1w

2e

```
SELECT Building_name from  
Buildings left join Employees  
on Buildings.Building_name = Employees.Building  
where Building is null;
```

RESET

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
-	-	-	-	-

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
-	-	-	-

List all movies and their combined sales in millions of dollars

Title	Sales
Finding Nemo	936.743261
Monsters University	743.559607
Ratatouille	623.722818
Cars 2	559.852396
Toy Story 2	485.015179
The Incredibles	631.442092
WALL-E	521.31186
Toy Story 3	1063.171911
Toy Story	361.958736
Cars	461.983149

```
select distinct title, (boxoffice.domestic_sales + boxoffice  
 .international_sales)/1000000 as Sales from movies join boxoffice on  
 Movies.Id = boxoffice.Movie_id;
```

List all movies and their ratings in percent

Title	Rate_percent
Finding Nemo	82
Monsters University	74
Ratatouille	80
Cars 2	64
Toy Story 2	79
The Incredibles	80
WALL-E	85
Toy Story 3	84
Toy Story	83
Cars	72

```
select distinct title, (rating*10) as Rate_percent from movies inner join
boxoffice on
movies.id = boxoffice.movie_id;
```

List all movies that were released on even number years

Title

A Bug's Life

The Incredibles

Cars

WALL-E

Toy Story 3

Brave

```
select distinct title from movies where year%2=0;
```

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

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

Max(Years_employed)

9

```
select max(years_employed) from Employees;
```

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

Table: Employees

Role	Avg_Yr
Artist	6
Engineer	3.4
Manager	6

```
select Distinct role, avg(years_employed) as Avg_Yr from Employees group by role;
```

Find the total number of employee years worked in each building

Building	Avg_Yr
1e	29
2w	36

```
select Distinct building, sum(years_employed) as Avg_Yr from Employees group by building;
```

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

Count(Name)

5

```
|  
| SELECT count(name) FROM employees where role = "Artist";
```

Find the number of Employees of each role in the studio

Table: Employees

Role	Count(Name)
Artist	5
Engineer	5
Manager	3

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

Find the total number of years employed by all Engineers

Sum(Years_employed)

17

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

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	Lee Unkrich	2006	117

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
8	8.5	222000000	207000000

Find the number of movies each director has directed

Query Results

Director	Count (Title)
Andrew Stanton	2
Brad Bird	2
Brenda Chapman	1
Dan Scanlon	1
John Lasseter	5
Lee Unkrich	1
Pete Docter	2

```
SELECT director, count (title) FROM movies group by director;
```

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

Query Results

Director	Sales
Andrew Stanton	1458055121
Brad Bird	1255164910
Brenda Chapman	538983207
Dan Scanlon	743559607
John Lasseter	2232208025
Lee Unkrich	1063171911
Pete Docter	1294159000

```
SELECT director,
sum(boxoffice.domestic_sales + boxoffice.international_sales) as Sales
FROM movies left join boxoffice on movies.id = boxoffice.movie_id
group by director;
```

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

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
15	Toy Story 4	Sarthak Patel	2025	85

Insert into Movies values (15,"Toy Story 4", "Sarthak Patel", 2025, 85);

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

Query Results

Movie_id	Rating	Domestic_sales	International_sales
3	7.9	245852179	239163000
1	8.3	191796233	170162503
2	7.2	162798565	200600000
15	8.7	340000000	270000000

Insert into Boxoffice values (15, 8.7, 340000000, 270000000);

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

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

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	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

```
update Movies set director = "John Lasseter" where title = "A Bug's Life";
```

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

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

```
update Movies set year = "1999" where title = "Toy Story 2";|
```

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

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

```
UPDATE movies
SET title = "Toy Story 3", director = "Lee Unkrich"
WHERE id = 11;
```

[RUN QUERY](#) [RESET](#)

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

Table: Movies

Id	Title	Director	Year	Length_minutes
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

```
delete from Movies where Year <2005;
```

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

Table: Movies

Id	Title	Director	Year	Length_minutes
7	Cars	John Lasseter	2006	117
8	Ratatouille	Brad Bird	2007	115
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

```
delete from Movies where director = "Andrew Stanton";|
```

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.

Table: Database		
Name	Version	Download_Count
SQLite	3.9	92000000
MySQL	5.5	512000000
Postgres	9.4	384000000

```
create table Database (Name text, Version float, Download_Count int);
```

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

Table: Movies

Id	Title	Director	Year	Length_minutes	Aspect_ratio
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	

```
Alter table Movies add Aspect_ratio float;
```

RUN QUERY RESET

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

Table: Movies

Id	Title	Director	Year	Length_minutes	Aspect_ratio	Language
1	Toy Story	John Lasseter	1995	81		English
2	A Bug's Life	John Lasseter	1998	95		English
3	Toy Story 2	John Lasseter	1999	93		English
4	Monsters, Inc.	Pete Docter	2001	92		English
5	Finding Nemo	Andrew Stanton	2003	107		English
6	The Incredibles	Brad Bird	2004	116		English
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

```
Alter table Movies add Language text default "English";
```

[RUN QUERY](#) [RESET](#)

Let's clean up by removing the Movies table

```
No such table: Movies
```

```
drop table Movies;
```

And drop the BoxOffice table as well

```
drop table BoxOffice;
```