

Q1. Movies (Not Boring)

SQL Schema:

Create table If Not Exists cinema (id int, movie varchar(255), description varchar(255), rating float(2, 1))

Truncate table cinema

insert into cinema (id, movie, description, rating) values ('1', 'War', 'great 3D', '8.9')

insert into cinema (id, movie, description, rating) values ('2', 'Science', 'fiction', '8.5')

insert into cinema (id, movie, description, rating) values ('3', 'irish', 'boring', '6.2')

insert into cinema (id, movie, description, rating) values ('4', 'Ice song', 'Fantacy', '8.6')

insert into cinema (id, movie, description, rating) values ('5', 'House card', 'Interesting', '9.1')

Problem Statement:

Write a query to report the movies with an **odd-numbered ID** and a description that is not "**boring**".

- Return the result table ordered by **rating** in **descending order**.

Sample Input:

Table: cinema

id	movie	description	rating
1	War	great 3D	8.9
2	Science	fiction	8.5
3	irish	boring	6.2
4	Ice song	Fantacy	8.6
5	House card	Interesting	9.1

Sample Output:

id	movie	description	rating
5	House card	Interesting	9.1
1	War	great 3D	8.9

Explanation:

- We have three movies with odd-numbered IDs: 1, 3, and 5.
- The movie with ID = 3 is boring so we do not include it in the answer.

Q2. OFFSET in SQL

What is the use of OFFSET in a SQL query?

- Specifies the number of rows of the result table to skip before any rows are retrieved.
- Specify the number of records to return.
- Eliminates the duplicate rows and display a unique list of values.
- Groups rows that have the same values into summary rows.

Q3. New Salary

SQL Schema:

```
create database new_salary;
use new_salary;
Create table If Not Exists employees (emp_id int, name varchar(255), salary int);
Truncate table employees;
insert into employees (emp_id, name, salary) values ('1', 'Louis', '6142');
insert into employees (emp_id, name, salary) values ('2', 'Den', '11259');
insert into employees (emp_id, name, salary) values ('3', 'Alexander', '5374');
insert into employees (emp_id, name, salary) values ('4', 'Shelli', '12572');
insert into employees (emp_id, name, salary) values ('5', 'Sigal', '6897');
```

Problem Description:

Write a query to calculate the salary of all employees after an increment of **20%**. Save the newly calculated salary column as '**New_salary**'.

Table: employees

Column Name	Type
emp_id	int
name	varchar
salary	int

Note:

- Return the columns **emp_id**, **name**, **salary**, and '**New_salary**'.
- Order the output by the **emp_id** in ascending order.

Steps to calculate the salary increment:

1. Multiply the current salary by the percentage of the increment.
2. Divide the result by 100.
3. Then add the result to the current salary.
4. Name the column as 'New_Salary'
5. Round off the 'New_salary'.

Sample Input:**Table:** employees

emp_id	name	salary
1	Luis	6142
2	Den	11259
3	Alexander	5374
4	Shelli	12572
5	Sigal	6897

Sample Output:

emp_id	name	salary	New_salary
1	Luis	6142	7370
2	Den	11259	13511
3	Alexander	5374	6449
4	Shelli	12572	15086
5	Sigal	6897	8276

Explanation: The New salary for Luis can be calculated as $6142 + (0.2 * 6142) = 7370$. In a similar manner, the New_salary is calculated for each employee.

Q4. Movies released after 2014

Problem Statement:

Write a query to display the titles of the movies that are released (i.e., release_year) after **2014** and have an average vote rating (i.e., vote_average) **greater than 7**.

- Return the column '**original_title**'.
- Return the result ordered by **original_title** in ascending order.

Dataset description for 'movies' table:

1. id - tmdb movie id
2. imdb_id - imdb movie id
3. popularity - A numeric quantity specifying the movie popularity.
4. budget - The budget in which the movie was made.
5. revenue - The worldwide revenue generated by the movie.
6. original_title - The title of the movie
7. cast - The name of the lead and supporting actors.
8. homepage - A link to the homepage of the movie.
9. director - The name of the director of the movie
10. tagline - Movie tagline.
11. keywords - The keywords or tags related to the movie.
12. overview - A brief description of the movie.
13. runtime - The running time of the movie in minutes.
14. genres - The genres of the movies
15. production_companies - The production house of the movie.
16. release_date - the date on which it was released.
17. vote_count - the count of votes received.
18. vote_average - average ratings the movie received.
19. release_year - the year in which it was released.

Sample Input:

Table: movies

id	imdb_id	popularity	budget	revenue	original_title	cast	homepage	director	tagline	keywords	overview	runtime	genres	production_companies	release_date	vote_count	vote_average	release_year	budget_adj	revenue_adj
135397	tt03696...	32.9858	15000...	15135...	Jurassic World	Chris...	http://ww...	Colin...	The p...	monster...	Twenty...	124	Actionl...	Universal StudiosAm...	6/9/2012	5562	6.5	2012	138000000	1392450000
262500	tt29084...	13.1125	11000...	29523...	Insurgent	Shail...	http://ww...	Rober...	One C...	based o...	Beatric...	119	Advent...	Summit Entertainment...	3/18/2017	2480	6.3	2017	101200000	271619000
168259	tt28208...	9.33501	19000...	15062...	Furious 7	Vin...	http://ww...	Jame...	Venge...	car race...	Deckar...	137	Actionl...	Universal PicturesOri...	4/1/2010	2947	7.3	2010	174800000	1385750000
167073	tt2381111	3.22733	11000...	62076...	Brooklyn	Saoir...	http://ww...	John...	Two c...	love tria...	In 1950...	111	Drama...	Wildgaze FilmsIrish Fi...	11/4/2015	754	7.3	2015	10120000	57110000
258480	tt24029...	2.88323	11800...	40272...	Carol	Cate...	http://car...	Todd...	Some...	gaylne...	In 1950...	118	Roman...	Killer FilmsFilm4Num...	11/20/2015	562	7.1	2015	10856000	37050300
331781	tt28706...	0.738404	0	8413144	Amy	Amy...	http://ww...	Asif K...	The gi...	jazzldru...	A docu...	128	Docum...	On The Corner Films...	7/3/2015	345	7.5	2015	0	7740090

Sample Output:

original_title
Amy
Brooklyn
Carol

Q5. Keywords

Problem Statement:

Write a query to list down all the movies along with their details that have **keywords** like 'sport' or 'sequel' or 'suspense'.

Note:

1. Return the columns '**original_title**', '**director**', '**genres**', '**cast**', '**budget**', '**revenue**', '**runtime**', and '**vote_average**'.
2. Return the columns ordered by **original_title** in ascending order.

Dataset description for 'movies' table is same as previous question

Sample Input:

Table: movies

id	original_title	keywords	director	genres	cast	budget	revenue	runtime	vote_average
168259	Furious 7	car race speed revenge suspense car	James Wan	Action Crime Thriller	Vin Diesel Paul Walker Dwayne Johnson Jason Statham Michelle Rodriguez	190000000	1506249360	137	7.3
76757	Jupiter Ascending	jupiter space woman director 3d interspecies romance	Lana Wachowski	Science Fiction Fantasy Romance	Mila Kunis Channing Tatum Josh Hartnett Scott Haze	176000003	183987723	124	5.2
99861	Avengers: Age of Ultron	marvel comic comic sequel superhero vision	Joss Whedon	Action Adventure Sci-Fi	Robert Downey Jr. Chris Evans Mark Ruffalo Chris Hemsworth Scarlett Johansson Anthony Mackie Sebastian Stan Liam Neeson Wendie Malick James Van Der Beek Toby Kebbell Josh Segarra Jude Plee Liam O'Brien Liam O'Brien	280000000	1405035767	141	7.4
131634	The Hunger Games: Mockingjay - Part 1	revolution strong woman dystopia game of death 3d	Francis Lawrence	War Adventure Sci-Fi	Jennifer Lawrence Josh Hutcherson Liam Hemsworth Dylan McDermott Elizabeth Banks Woody McClure Liam O'Brien Liam O'Brien	160000000	650523427	136	6.5
254128	San Andreas	california earthquake catastrophe disaster film 3d	Brad Peyton	Action Drama Thriller	Dwayne Johnson Keri Russell Alexandra Daddario Ben Foster Dennis Quaid Liam Neeson Liam O'Brien Liam O'Brien	110000000	470490832	114	6.1
207703	Kingsman: The Secret Service	spy great britain secret organization secret agent marvel comic	Matthew Vaughn	Crime Comedy Action	Taron Egerton Halle Berry Samuel L. Jackson Minnie Driver Ben Whishaw Tom Hollander Liam Neeson Liam O'Brien Liam O'Brien	81000000	403802136	130	7.6
296098	Bridge of Spies	spy cold war pilot lawyer	Steven Spielberg	Thriller Drama	Tom Hanks Meg Ryan Alan Rickman Liam Neeson Liam O'Brien Liam O'Brien	40000000	162610473	141	7.1
167073	Brooklyn	love triangle based on novel homesickness vorortelship	John Crowley	Drama Romance	Saoirse Ronan Dom Monaghan Liam Neeson Liam O'Brien Liam O'Brien	11000000	62076141	111	7.3

Sample output:

original_title	director	genres	cast	budget	revenue	runtime	vote_average
Avengers: Age of Ultron	Joss Whedon	Action Adventure Sci-Fi	Robert Downey Jr. Chris Evans Mark Ruffalo Chris Hemsworth Scarlett Johansson Anthony Mackie Sebastian Stan Liam Neeson Wendie Malick James Van Der Beek Toby Kebbell Josh Segarra Jude Plee Liam O'Brien Liam O'Brien	280000000	1405035767	141	7.4
Furious 7	James Wan	Action Crime Thriller	Vin Diesel Paul Walker Dwayne Johnson Jason Statham Michelle Rodriguez	190000000	1506249360	137	7.3

Q6. Horror genre

Problem Statement:

Display the details of the movies which belong to the '**Horror**' genre.

- Return the columns '**original_title**', and '**popularity**'.
- Return the result ordered by the **popularity** in descending order.

Dataset description for `movies` table is the same as previous example.

Sample Input:

Table: movies

id	imdb_id	popularity	budget	revenue	original_title	cast	homepage	director	tagline	keywords	overview	runtime	genres	production_companies	release_date	vote_count	vote_average	release_year	budget_adj	revenue_adj
135397	tt03696...	32.9858	15000...	15135...	Jurassic World	Chris...	http://ww...	Colin...	The p...	monster...	Twenty...	124	Action/Adventure/...	Universal Studios/Am...	6/9/2012	5562	6.5	2012	138000000	1392450000
76341	tt13921...	28.4199	15000...	37843...	Mad Max: Fury Road	Tom...	http://ww...	Georg...	What...	future/c...	An apo...	120	Action/Adventure/...	Village Roadshow Pict...	5/13/2010	6185	7.1	2010	138000000	348161000
262500	tt29084...	13.1125	11000...	29523...	Insurgent	Shaile...	http://ww...	Rober...	One C...	based o...	Beatric...	119	Adventure/Scienc...	Summit Entertainment...	3/18/2017	2480	6.3	2017	101200000	271619000
264660	tt04707...	6.11885	15000...	36869...	Ex Machina	Domh...	http://ex...	Alex...	There...	dancing...	Caleb...	108	Horror	DNA Films/Universal...	1/21/2010	2854	7.6	2010	138000000	33919800
260346	tt24460...	5.74976	48000...	32577...	Taken 3	Liam...	http://ww...	Olivier...	It End...	revenge...	Ex-gov...	109	Horror	Twentieth Century Fox...	1/1/2010	1578	6.1	2010	44160000	299710000
216015	tt23224...	4.7104	40000...	56965...	Fifty Shades of Grey	Dakot...	https://w...	Sam T...	Are yo...	based o...	When c...	125	Horror	Focus Features/Trigge...	2/11/2011	1865	5.3	2011	36800000	524079000

Sample Output:

original_title	popularity
Ex Machina	6.11885
Taken 3	5.74976
Fifty Shades of Grey	4.7104

Q7. 2012-2015

Problem Statement:

Find the details of the movies that are released **between** the years **2012-2015** i.e., (Including 2012 and 2015).

- Return the columns '**original_title**', '**genres**', '**vote_average**', and '**revenue**'.
- Return the result ordered by **original_title** in ascending order.

Dataset description for `movies` table is the same as previous example.

Sample Input:

Table: movies

id	imdb_id	popularity	budget	revenue	original_title	cast	homepage	director	tagline	keywords	overview	runtime	genres	production_companies	release_date	vote_count	vote_average	release_year	budget_adj	revenue_adj
135397	tt03696...	32.9858	15000...	15135...	Jurassic World	Chris Prattl...	http://ww...	Colin...	The p...	monster...	Twenty-...	124	Actionl...	Universal StudiosAmb...	6/9/2011	5562	6.5	2011	138000000	1392450000
76341	tt13921...	28.4199	15000...	37843...	Mad Max: Fury Road	Tom Hardyl...	http://ww...	Georg...	What...	futurelc...	An apo...	120	Actionl...	Village Roadshow Pict...	5/13/2010	6185	7.1	2010	138000000	348161000
262500	tt29084...	13.1125	11000...	29523...	Insurgent	Shailene W...	http://ww...	Rober...	One C...	based o...	Beatric...	119	Advent...	Summit Entertainment...	3/18/2011	2480	6.3	2011	101200000	271619000
140607	tt24884...	11.1731	20000...	20681...	Star Wars: The Forc...	Harrison F...	http://ww...	J.J. A...	Every...	androidl...	Thirty y...	136	Actionl...	LucasfilmTruenorth Pr...	12/15/2012	5292	7.5	2012	184000000	1902720000
168259	tt28208...	9.33501	19000...	15062...	Furious 7	Vin Diesell...	http://ww...	Jame...	Venge...	car race...	Deckar...	137	Action...	Universal PicturesOri...	4/1/2014	2947	7.3	2014	174800000	1385750000
281957	tt16632...	9.1107	13500...	53295...	The Revenant	Leonardo...	http://ww...	Alejan...	(n. On...	father-s...	In the 1...	156	Weste...	Regency EnterprisesI...	12/25/2016	3929	7.2	2016	124200000	490314000
87101	tt13401...	8.65436	15500...	44060...	Terminator Genisys	Arnold Sch...	http://ww...	Alan T...	Reset...	saving t...	The yea...	125	Scienc...	Paramount Pictures(S...	6/23/2015	2598	5.8	2015	142600000	405355000
286217	tt36593...	7.6674	10800...	59538...	The Martian	Matt Damo...	http://ww...	Ridley...	Bring...	based o...	During...	141	Drama...	Twentieth Century Fox...	9/30/2001	4572	7.6	2001	99360000	547750000

Sample Output:

original_title	genres	vote_average	revenue
Furious 7	Action Crime Thriller	7.3	1506249360
Star Wars: The Force Awakens	Action Adventure Sci...	7.5	2068178225
Terminator Genisys	Science Fiction Ac...	5.8	440603537

Q8. Weighted avg rating

Problem Statement:

Calculate the weighted average rating from the columns **vote_count** and **vote_average** and save the column as "**Weighted_avg_rating**".

Write a query to display the **top 10** movies and **their rating** up to **two** decimals based on the newly created column.

- Return the columns `original_title`, `Weighted_avg_rating`
- Return the output ordered by `Weighted_avg_rating` in descending order and `original_title` in ascending order.

Note: Use the given formula to calculate a weighted average rating:

$(v/(v+m) * R) + (m/(m+v) * C)$ Where,

- **v** is the number of votes for the movie - **vote_count**
- **m** is the minimum vote required, take **m as 104.0**
- **R** is the average rating of the movie - **vote_average**
- **C** is the mean vote across the whole report take **c as 5.97**

Dataset description for movies table is the same as previous example.

Sample Input:

Table: movies

id	original_title	vote_count	vote_average	popularity	tagline	director	budget	revenue
135397	Jurassic World	5562	6.5	32.985763	The pa...	Colin Trevorrow	150000000	1513528810
76341	Mad Max: Fury Road	6185	7.1	28.419936	What a...	George Miller	150000000	378436354
262500	Insurgent	2480	6.3	13.112507	One C...	Robert Schwentke	110000000	295238201
140607	Star Wars: The Force Awakens	5292	7.5	11.173104	Every...	J.J. Abrams	200000000	2068178225
168259	Furious 7	2947	7.3	9.335014	Venge...	James Wan	190000000	1506249360
76757	Jupiter Ascending	1937	5.2	6.189369	Expan...	Lana Wachowski...	176000003	183987723
99861	Avengers: Age of Ultron	4304	7.4	5.944927	A New...	Joss Whedon	280000000	1405035767
131634	The Hunger Games: Mockingjay...	2380	6.5	5.476958	The fir...	Francis Lawrence	160000000	650523427
254128	San Andreas	2060	6.1	4.907832	A rescu...	Brad Peyton	110000000	470490832
207703	Kingsman: The Secret Service	3833	7.6	4.503789	Manne...	Matthew Vaughn	81000000	403802136

Sample Output:

original_title	Weighted_avg_rating
Kingsman: The Secret Service	7.56
Star Wars: The Force Awakens	7.47
Avengers: Age of Ultron	7.37
Furious 7	7.25
Mad Max: Fury Road	7.08
Jurassic World	6.49
The Hunger Games: Mockingjay...	6.48
Insurgent	6.29
San Andreas	6.09
Jupiter Ascending	5.24