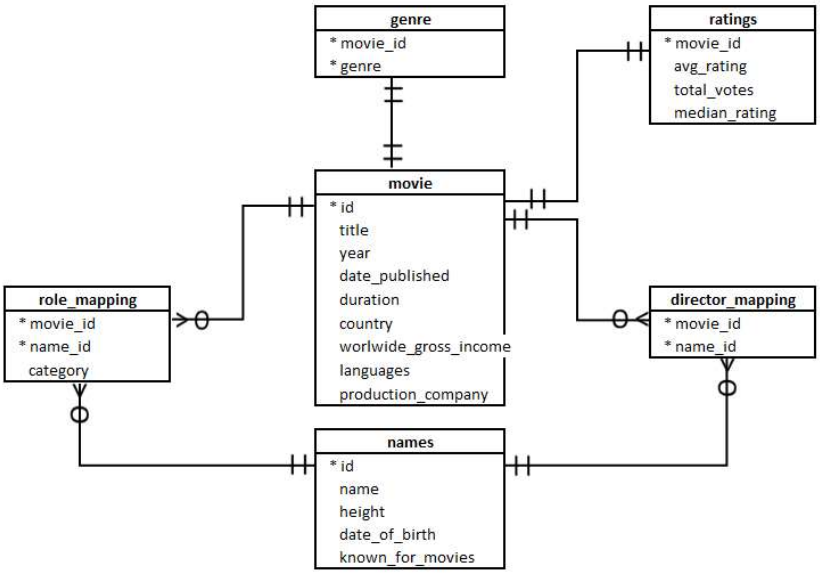
Sql case Study(on Movie data)

27 February 2024 00:27 AM

Below are the ERD- Diagram and Table columns

table	column
movie	id
movie	title
movie	year
movie	date_published
movie	duration
movie	country
movie	worlwide_gross_income
movie	languages
movie	production_company
genre	movie_id
genre	genre
director_mapping	movie_id
director_mapping	name_id
role_mapping	movie_id
role_mapping	name_id
role_mapping	category
names	id
names	name
names	height
names	date_of_birth
names	known_for_movies
ratings	movie_id
ratings	avg_rating
ratings	total_votes
ratings	median_rating



I am using Microsoft SQL Server for analysis. Below is the step-by-step process

I took this data from Kaggle, where I uploaded all the files into our database and performed some analysis based on the given data. I have written queries to understand the data, and below are all remarks and analyses performed

Database init
Database name -- sql_case_study

- **Importing 6 tables into this database manually.**
- **I am performing this task using the import method.**
- **The data has been successfully imported.**

- **Now, checking each table one by one and examining the rows/columns for each table.**

```
select count(*)as total_cnt from movie;
select count(*)as total_cnt from genre;
select count(*)as total_cnt from names;
select count(*)as total_cnt from ratings;
select count(*)as total_cnt from role_mapping;
```

Printing all columns name

```
SELECT COLUMN_NAME
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'movie';
```

	COLUMN_NAME
1	id
2	title
3	year
4	date_published
5	duration

```
select count(1) from movie -- row_count
```

- **There are 9 columns in the movie dataset.**
- **The total shape of the data is (rows: 7997, columns: 9).**

- **Checking head(10)**

```
SELECT *FROM movie LIMIT 100;-- I don't know why this command is not working. I am trying to find another command to display the top 10 rows of the dataset.
```

- **--this is working in SQL**

```
SELECT TOP 10 * FROM movie;
```

	id	title	year	date_published	duration	country	worlwide_gross_income	languages	production_company
1	tt0012494	Der m❖de Tod	2017	2017-06-09	97	Germany	12156.00	German	Decla-Bioscop AG
2	tt0038733	A Matter of Life and Death	2017	2017-12-08	104	UK	124241.00	English, French, Russian	The Archers
3	tt0361953	The Nest of the Cuckoo Birds	2017	2017-10-16	81	USA	NULL	English	Bert Williams Motion Pictures and Distributor
4	tt0235166	Against All Hope	2017	2017-10-20	90	USA	NULL	English	NULL
5	tt0337383	Vaikai is Amerikos viesbucio	2017	2017-03-09	88	Soviet Union	NULL	Lithuanian, Russian	Lietuvos Kinostudija

- **Check total movie= 7997**

```
select count(*)as total_movie from movie;
```

- **I need to check the total number of movies until this date, based on the dataset.**
- **I can see the following data points: in 2017, the total movie contribution is 38%; in 2018, it is 36%; and in 2019, it is 25%."**

```
SELECT
 year, COUNT(1) AS Total_movie_cnt,
 COUNT(1)as Total_movie,ROUND((COUNT(1) / CAST((SELECT COUNT(1) FROM movie) AS DECIMAL)) * 100, 2) AS Total_pcnt
FROM movie
GROUP BY year
ORDER BY total_movie_cnt DESC;
```

Output

	year	Total_movie_cnt	Total_movie	Total_pcnt
1	2017	3052	3052	38.160000000000000000
2	2018	2944	2944	36.810000000000000000
3	2019	2001	2001	25.020000000000000000

- **I need to print the month-to-month distribution of movies. - For plotting i will use python**
- **Since the monthname function is not available in SQL, I am using the CASE function to display the month names.**
- **I am interested in identifying the top 5 months where I observe a larger number of movie releases.**
- **In March, September, January, and October, 10% of the movies were released, while in April, 8% were released.**

```
SELECT TOP 5
 MONTH(date_published) AS Month,
 CASE MONTH(date_published)
       WHEN 1 THEN 'January'
   WHEN 2 THEN 'February'
   WHEN 3 THEN 'March'
   WHEN 4 THEN 'April'
   WHEN 5 THEN 'May'
   WHEN 6 THEN 'June'
   WHEN 7 THEN 'July'
   WHEN 8 THEN 'August'
   WHEN 9 THEN 'September'
   WHEN 10 THEN 'October'
   WHEN 11 THEN 'November'
   WHEN 12 THEN 'December'
 END AS Month_name,count(1)as Total_movie,
 ROUND((COUNT(1) / CAST((SELECT COUNT(1) FROM movie) AS DECIMAL)) * 100, 2)as Total_pcnt
FROM movie
group by   MONTH(date_published),
 CASE MONTH(date_published)
       WHEN 1 THEN 'January'
   WHEN 2 THEN 'February'
   WHEN 3 THEN 'March'
   WHEN 4 THEN 'April'
   WHEN 5 THEN 'May'
   WHEN 6 THEN 'June'
   WHEN 7 THEN 'July'
   WHEN 8 THEN 'August'
   WHEN 9 THEN 'September'
   WHEN 10 THEN 'October'
   WHEN 11 THEN 'November'
   WHEN 12 THEN 'December'
 END
order by  count(1) desc;
```

	Month	Month_name	Total_movie	Total_pcnt
1	3	March	824	10.300000000000000000
2	9	September	809	10.120000000000000000
3	1	January	804	10.050000000000000000
4	10	October	801	10.020000000000000000
5	4	April	680	8.500000000000000000

There are some questions to check:

1. **First, title-wise.**
2. **Identify the top 5 movies from each country based on their average rating, with only one movie selected from each country.**
3. **Duration-wise.**
4. **Income-wise.**
5. **Language-wise.**

1- top 5 movie title wise
Output should (movie name,rating)

```
select top 10 title,avg_rating from movie t1
left join ratings t2
on t1.id=t2.movie_id
order by avg_rating desc
```

	title	avg_rating
1	Love in Kilnerry	10
2	Kirket	10
3	Gini Helida Kathe	9.80000019073486
4	Runam	9.69999980926514
5	Android Kunjappan Version 5.25	9.60000038146973

2- Identify the top 5 movies from each country based on their highest average rating, with only one movie selected from each country.
Output >>>-- Title,Country,Avg_rating

```
with main as(
	select title,country,avg_rating,ROW_NUMBER()over(partition by country order by avg_rating desc)as rn
	from movie t1
	left join ratings t2
	on t1.id=t2.movie_id
	where country is not null
)
select top 5 title,country,avg_rating from main
where rn=1
order by avg_rating desc
```

	title	country	avg_rating
1	Kirket	India	10
2	Love in Kilnerry	USA	10
3	The Brighton Miracle	Australia	9.5
4	Zana	Kosovo, Albania	9.39999961853027
5	Our Little Haven	UK	9.39999961853027

3- duration wise --top10 Movie where rating is >= to the total average rating, and the votes are >= the total average votes, duration should be in desc order
Output >>>-- title ,rating ,votes, duration in hour

There are 135 movie list-

```
with a as(
	select title,duration,country,avg_rating,total_votes from movie t1
	left join ratings t2
	on t1.id=t2.movie_id
	where country='india'
		)
select top 10 title,avg_rating as rating,
			total_votes as vote,round(cast(duration as decimal)/60,2) as Duration
from a
where a.avg_rating>=(select avg(a.avg_rating) from a)
 and a.total_votes>=(select avg(total_votes) from a)
 order by a.duration desc
```

	title	rating	vote	Duration
1	Arjun Reddy	8.19999980926514	17529	3.030000
2	Bigil	7.59999990463257	6925	2.980000
3	Mahanati	8.5	7698	2.950000
4	Super Deluxe	8.39999961853027	4535	2.930000
5	Maharshi	7.40000009536743	4325	2.930000

4- Income wise >>>>-- top 5 movies for each country based on movie income, with the condition that the year is 2017.
Output >>>-- country ,title ,total_income

```
with main as(
select country,title,worlwide_gross_income,
	ROW_NUMBER()over(partition by country order by worlwide_gross_income)as rn
from movie t1
left join ratings t2
on t1.id=t2.movie_id
where year='2017'
 and worlwide_gross_income is not null
 )
select main.country,main.title,main.worlwide_gross_income from main
where rn <=5
```

	country	title	worlwide_gross_income
1	Argentina	El futuro perfecto	115.00
2	Argentina	Jess & James	514.00
3	Argentina	K◆kszak◆ll◆	1289.00
4	Argentina	Mater	5519.00
5	Argentina	Madraza	33048.00

5- languge wise >>>-- Need to find the top 1 movies with the highest ratings and highest votes (1 movie for every language).
Output >>>-- Title,Language,total_vote,Rating

```
with main as(
select title,languages,total_votes,avg_rating
	,ROW_NUMBER()over(partition by languages order by avg_rating desc,total_votes desc) as rn
from movie t1
left join ratings t2
on t1.id=t2.movie_id
where languages is not null
```

```
)
select title as Title,languages as Language,total_votes as Vote,avg_rating as rating from main
where rn =1
order by avg_rating desc
```

	Title	Language	Vote	rating
1	Love in Kilnerry	English	2360	10
2	Kirket	Hindi	587	10
3	Gini Helida Kathe	Kannada	425	9.80000019073486
4	Runam	Telugu	133	9.69999980926514
5	Android Kunjappan Version 5.25	Malayalam	1176	9.60000038146973

- **Need to find out which actor/actress has done the most films in their career. top 10**
  Output >>-- Nmae, Movie count

```
with t1 as(
select id,name,movie_id,category from names a
join role_mapping b
on a.id=b.name_id)

select top 10 name, count(distinct t1.id)as Movie_count from movie t2
	join t1
	on t2.id=t1.movie_id
group by name
order by count(distinct t1.id) desc
```

	name	Movie_count
1	Aaron Davis	2
2	Anand	2
3	Anjali	2
4	Anusha	2
5	B.N. Sharma	2

- **descriptive statistics for movie dataset**
  There is one movie which is aound 13 hr >>>'la flor '

```
SELECT
	COUNT(*) AS Total_count,
	AVG(duration) AS duration_mean,
	SUM(duration) AS duration_sum,
	stdev(duration)as duration_std,
	min(duration)as duration_min,
	max(duration)as duration_max,
	AVG(worlwide_gross_income) AS gorss_income_mean,
	SUM(worlwide_gross_income) AS gorss_income_sum,
	STDEV(worlwide_gross_income)as gorss_income_std,
	min(worlwide_gross_income)as gorss_income_min,
	max(worlwide_gross_income)as gorss_income_max

FROM
movie
```

	Total_count	duration_mean	duration_sum	duration_std	duration_min	duration_max	gorss_income_mean	gorss_income_sum	gorss_income_std	gorss_income_min	gorss_income_max
1	7997	103	830827	22.928304034675...	41	808	24168379.6974	103271486447.00	116372608.297115	37.00	2797800564.00

Click to select the whole column