SQL DATA ANALYSIS PROJECT

MAVEN MOVIES

Dheeraj S Kulkarni

06th July 2023

Introduction

THE SITUATION

You and your business partner were recently approached by another local business owner who is interested in purchasing Maven Movies. He primarily owns restaurants and bars, so he has lots of questions for you about your business and the rental business in general. His offer seems very generous, so you are going to entertain his questions.

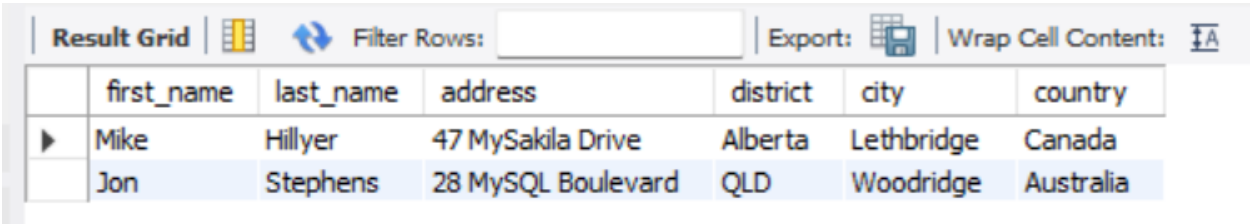
THE OBJECTIVE

Use MySQL to:

Leverage your SQL skills to extract and analyze data from various tables in the Maven Movies database to answer your potential Acquirer's questions. Each question will require you to write a multi-table SQL query, joining at least two tables.

1) My partner and I want to come by each of the stores in person and meet the managers. Please send over the managers' names at each store, with the full address of each property (street address, district, city, and country).

```sql
SELECT
    staff.first_name,
    staff.last_name,
    address.address,
    address.district,
    city.city,
    country.country
FROM
    store
        LEFT JOIN staff ON store.store_id = staff.store_id
        LEFT JOIN address ON store.address_id = address.address_id
        LEFT JOIN city ON address.city_id = city.city_id
        LEFT JOIN country ON city.country_id = country.country_id;
```

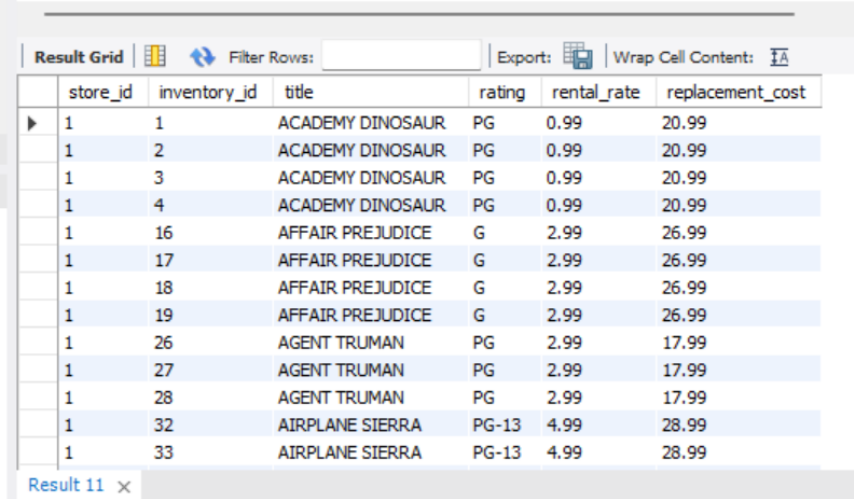

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

first_name	last_name	address	district	city	country
Mike	Hillyer	47 MySakila Drive	Alberta	Lethbridge	Canada
Jon	Stephens	28 MySQL Boulevard	QLD	Woodridge	Australia

2) I would like to get a better understanding of all of the inventory that would come along with the business. Please pull together a list of each inventory item you have stocked, including the store_id number, the inventory_id, the name of the film, the film's rating, its rental rate and replacement cost.

```sql
SELECT
    inventory.store_id,
    inventory.inventory_id,
    film.title,
    film.rating,
    film.rental_rate,
    film.replacement_cost

FROM inventory
LEFT JOIN film ON inventory.film_id = film.film_id;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

store_id	inventory_id	title	rating	rental_rate	replacement_cost
1	1	ACADEMY DINOSAUR	PG	0.99	20.99
1	2	ACADEMY DINOSAUR	PG	0.99	20.99
1	3	ACADEMY DINOSAUR	PG	0.99	20.99
1	4	ACADEMY DINOSAUR	PG	0.99	20.99
1	16	AFFAIR PREJUDICE	G	2.99	26.99
1	17	AFFAIR PREJUDICE	G	2.99	26.99
1	18	AFFAIR PREJUDICE	G	2.99	26.99
1	19	AFFAIR PREJUDICE	G	2.99	26.99
1	26	AGENT TRUMAN	PG	2.99	17.99
1	27	AGENT TRUMAN	PG	2.99	17.99
1	28	AGENT TRUMAN	PG	2.99	17.99
1	32	AIRPLANE SIERRA	PG-13	4.99	28.99
1	33	AIRPLANE SIERRA	PG-13	4.99	28.99

Result 11

3) From the same list of films you just pulled, please roll that data up and provide a summary-level overview of your inventory. We would like to know how many inventory items you have with each rating at each store.

```sql
SELECT
    inventory.store_id,
    film.rating,
    COUNT(inventory_id) AS inventory_items
FROM inventory
    LEFT JOIN film ON inventory.film_id = film.film_id
GROUP BY
    inventory.store_id, film.rating;
```

Result Grid | Filter Rows:

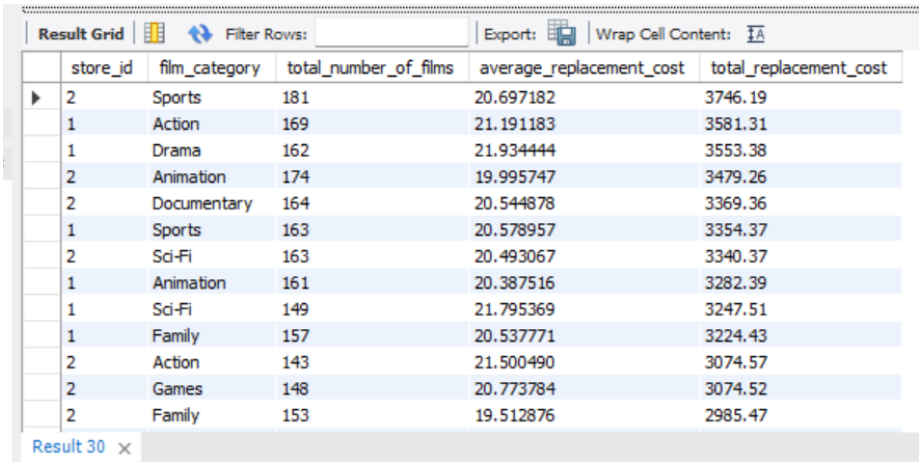
store_id	rating	inventory_items
1	PG	444
1	G	394
1	PG-13	525
1	NC-17	465
1	R	442
2	PG	480
2	G	397
2	NC-17	479
2	PG-13	493
2	R	462

4) Similarly, we want to understand how diversified the inventory is in terms of replacement costs. We want to see how big of a hit it would be if a certain category of film became unpopular at a certain store. We would like to see the number of films, as well as the average replacement cost, and total replacement cost, sliced by store and film category.

```sql
SELECT
    inventory.store_id,
    category.name AS film_category,
    COUNT(inventory.inventory_id) AS total_number_of_films,
    AVG(film.replacement_cost) AS average_replacement_cost,
    SUM(film.replacement_cost) AS total_replacement_cost

FROM inventory
    LEFT JOIN film ON inventory.film_id = film.film_id
    LEFT JOIN film_category ON film.film_id = film_category.film_id
    LEFT JOIN category ON film_category.category_id = category.category_id

GROUP BY inventory.store_id, category.name
ORDER BY total_replacement_cost DESC;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

store_id	film_category	total_number_of_films	average_replacement_cost	total_replacement_cost
2	Sports	181	20.697182	3746.19
1	Action	169	21.191183	3581.31
1	Drama	162	21.934444	3553.38
2	Animation	174	19.995747	3479.26
2	Documentary	164	20.544878	3369.36
1	Sports	163	20.578957	3354.37
2	Sci-Fi	163	20.493067	3340.37
1	Animation	161	20.387516	3282.39
1	Sci-Fi	149	21.795369	3247.51
1	Family	157	20.537771	3224.43
2	Action	143	21.500490	3074.57
2	Games	148	20.773784	3074.52
2	Family	153	19.512876	2985.47

Result 30

5) We want to make sure you folks have a good handle on who your customers are. Please provide a list of all customer names, which store they go to, whether or not they are currently active, and their full addresses – street address, city, and country.

```sql
SELECT
    CONCAT(customer.first_name,' ',customer.last_name) AS customer_name,
    customer.store_id,
    customer.active,
    address.address,
    city.city,
    country.country

FROM customer
    LEFT JOIN address ON customer.address_id = address.address_id
    LEFT JOIN city ON address.city_id = city.city_id
    LEFT JOIN country ON city.country_id = country.country_id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_name	store_id	active	address	city	country
MARY SMITH	1	1	1913 Hanoi Way	Sasebo	Japan
PATRICIA JOHNSON	1	1	1121 Loja Avenue	San Bernardino	United States
LINDA WILLIAMS	1	1	692 Joliet Street	Athenai	Greece
BARBARA JONES	2	1	1566 Inegl Manor	Myingyan	Myanmar
ELIZABETH BROWN	1	1	53 Idfu Parkway	Nantou	Taiwan
JENNIFER DAVIS	2	1	1795 Santiago de Compostela Way	Laredo	United States
MARIA MILLER	1	1	900 Santiago de Compostela Parkway	Kragujevac	Yugoslavia
SUSAN WILSON	2	1	478 Joliet Way	Hamilton	New Zealand
MARGARET MOORE	2	1	613 Korolev Drive	Masqat	Oman
DOROTHY TAYLOR	1	1	1531 Sal Drive	Esfahan	Iran
LISA ANDERSON	2	1	1542 Tarlac Parkway	Sagamihara	Japan
NANCY THOMAS	1	1	808 Bhopal Manor	Yamuna Nagar	India
KAREN JACKSON	2	1	270 Amroha Parkway	Osmaniye	Turkey

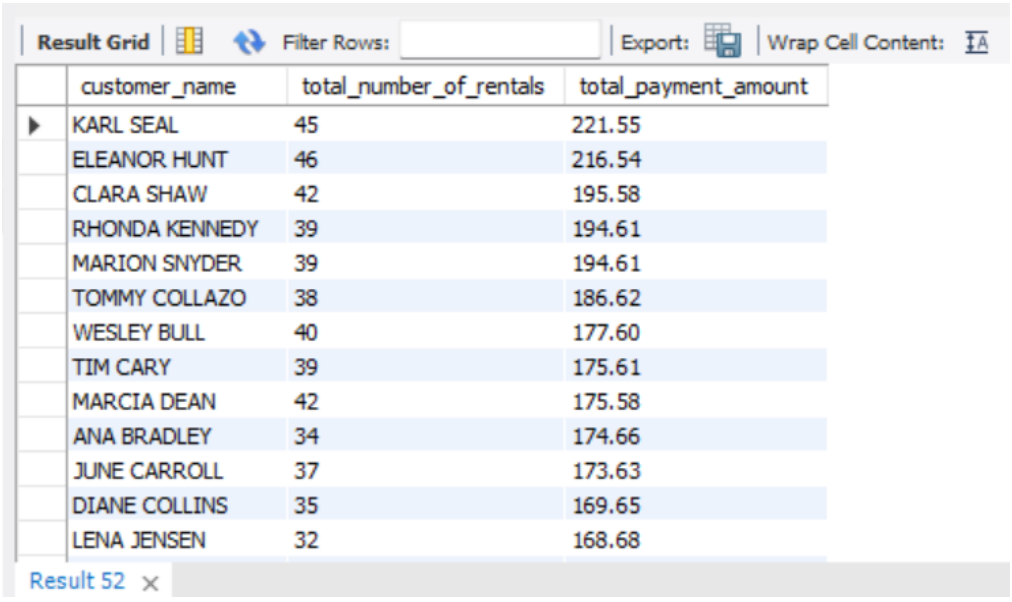
Result 44

6) We would like to understand how much your customers are spending with you, and also to know who your most valuable customers are. Please pull together a list of customer names, their total lifetime rentals, and the sum of all payments you have collected from them. It would be great to see this ordered on total lifetime value, with the most valuable customers at the top of the list.

```sql
SELECT
    CONCAT(customer.first_name,' ',customer.last_name) AS customer_name,
    COUNT(rental.rental_id) AS total_number_of_rentals,
    SUM(payment.amount) AS total_payment_amount

FROM customer
    LEFT JOIN rental ON customer.customer_id = rental.customer_id
    LEFT JOIN payment ON rental.rental_id = payment.rental_id

GROUP BY customer.first_name, customer.last_name
ORDER BY total_payment_amount DESC;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

customer_name	total_number_of_rentals	total_payment_amount
KARL SEAL	45	221.55
ELEANOR HUNT	46	216.54
CLARA SHAW	42	195.58
RHONDA KENNEDY	39	194.61
MARION SNYDER	39	194.61
TOMMY COLLAZO	38	186.62
WESLEY BULL	40	177.60
TIM CARY	39	175.61
MARCIA DEAN	42	175.58
ANA BRADLEY	34	174.66
JUNE CARROLL	37	173.63
DIANE COLLINS	35	169.65
LENA JENSEN	32	168.68

Result 52

7) My partner and I would like to get to know your board of advisors and any current investors. Could you please provide a list of advisor and investor names in one table? Could you please note whether they are an investor or an advisor, and for the investors, it would be good to include which company they work with.

```sql
SELECT
    'investor' AS type,
    CONCAT(investor.first_name,' ',investor.last_name) AS full_name,
    company_name
FROM investor

UNION

SELECT
    'advisor' AS type,
   CONCAT(advisor.first_name,' ',advisor.last_name) AS full_name,
    NULL
FROM advisor;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

type	full_name	company_name
investor	Montgomery Burns	Springfield Syndicators
investor	Anthony Stark	Iron Investors
investor	William Wonka	Chocolate Ventures
advisor	Barry Beenthere	NULL
advisor	Cindy Smartypants	NULL
advisor	Mary Moneybags	NULL
advisor	Walter White	NULL

8) We're interested in how well you have covered the most-awarded actors. Of all the actors with three types of awards, for what % of them do we carry a film? And how about actors with two types of awards? Same questions. Finally, how about actors with just one award?

```sql
SELECT
    CASE
        WHEN actor_award.awards = 'Emmy, Oscar, Tony ' THEN '3 awards'
        WHEN actor_award.awards IN ('Emmy, Oscar','Emmy, Tony', 'Oscar, Tony') THEN '2 awards'
        ELSE '1 award'
    END AS number_of_total_awards,
    AVG(CASE WHEN actor_award.actor_id IS NULL THEN 0 ELSE 1 END)  AS percent_with_atleast_one_film
FROM actor_award

GROUP BY
    CASE
        WHEN actor_award.awards = 'Emmy, Oscar, Tony ' THEN '3 awards'
        WHEN actor_award.awards IN ('Emmy, Oscar','Emmy, Tony', 'Oscar, Tony') THEN '2 awards'
        ELSE '1 award'
    END
```

Result Grid | Filter Rows: | Export:

number_of_total_awards	percent_with_atleast_one_film
3 awards	0.5714
2 awards	0.9242
1 award	0.8333

Learning

During this SQL project, I gained a firm grasp of intermediate SQL concepts such as JOINS, CASE statements, UNION, GROUP BY, and AGGREGATION, along with STRING functions. I successfully applied multiple JOINS to connect tables and utilized multiple CASE statements to analyze and manipulate the data effectively. These experiences enhanced my understanding of SQL and allowed me to perform comprehensive data analysis.

THANK YOU