MAVEN MOVIES SQL ANALYSIS

by
Wahyu Novitasari

INTRODUCING THE FINAL PROJECT

THE SITUATION

I was recently approached by a local business owner who showed interest in acquiring Maven Movies. While he mainly owns restaurants and bars, he had many questions about how the DVD rental business works and how Maven Movies operates. His offer seemed promising, so I decided to explore the business further and prepare insights to answer his inquiries.

THE OBJECTIVES

Use MySQL to:
Leverage my SQL skills to extract and analyze data from various tables in the Maven Movies database in order to answer the potential buyer's questions. Each insight required me to write multi-table SQL queries, joining at least two tables to generate relevant and actionable findings.

INTRODUCING THE FINAL PROJECT

THE LETTER

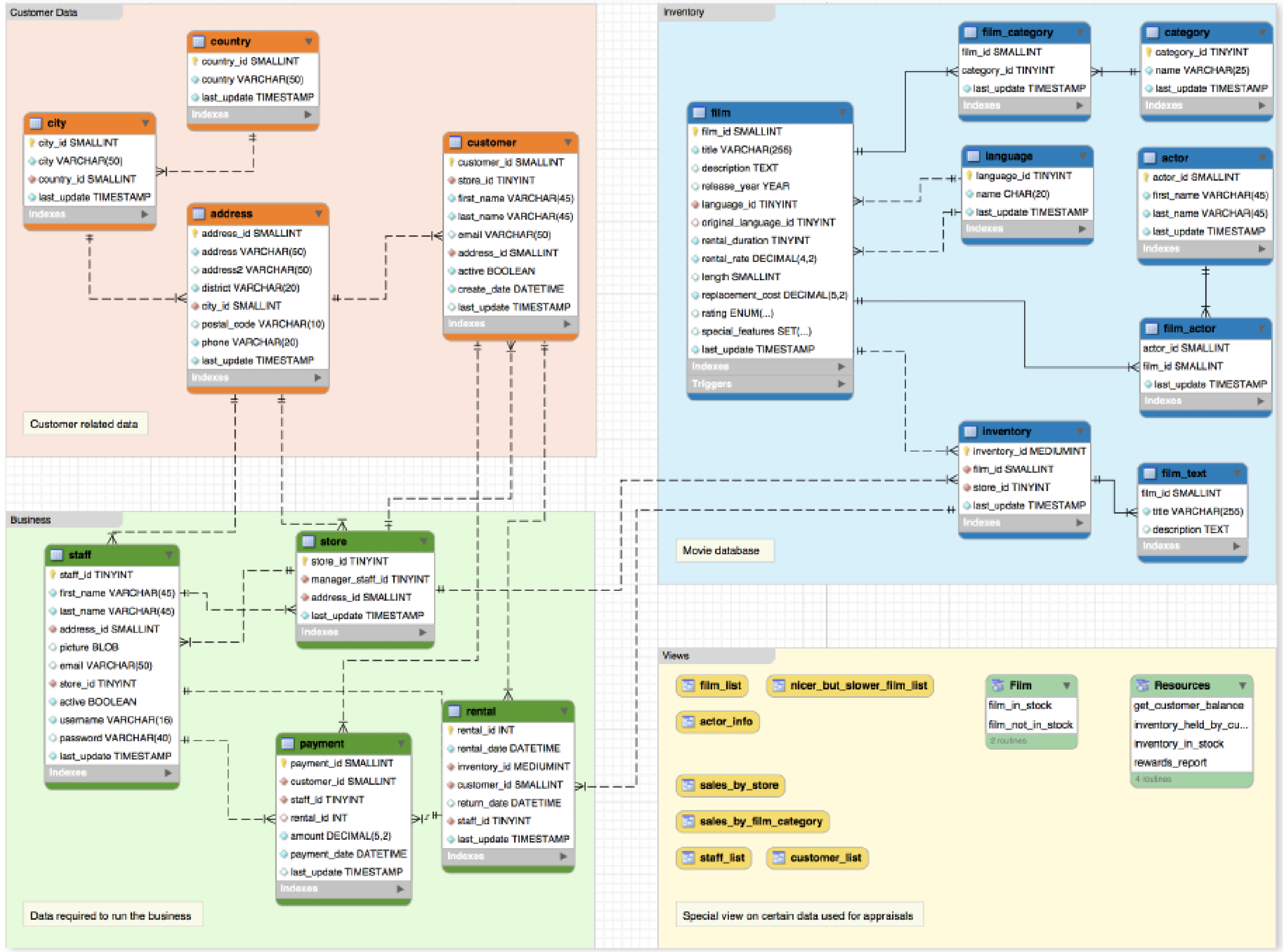
Dear Maven Movies Management,

I am excited about the potential acquisition and learning more about your rental business.

Please bear with me as I am new to the industry, but I have a number of questions for you. Assuming you can answer them all, and that there are no major surprises, we should be able to move forward with the purchase.

Best, Martin Moneybags

GET TO KNOW THE DATABASE



Maven Movies is a traditional DVD rental business with operations stored in a structured MySQL database.

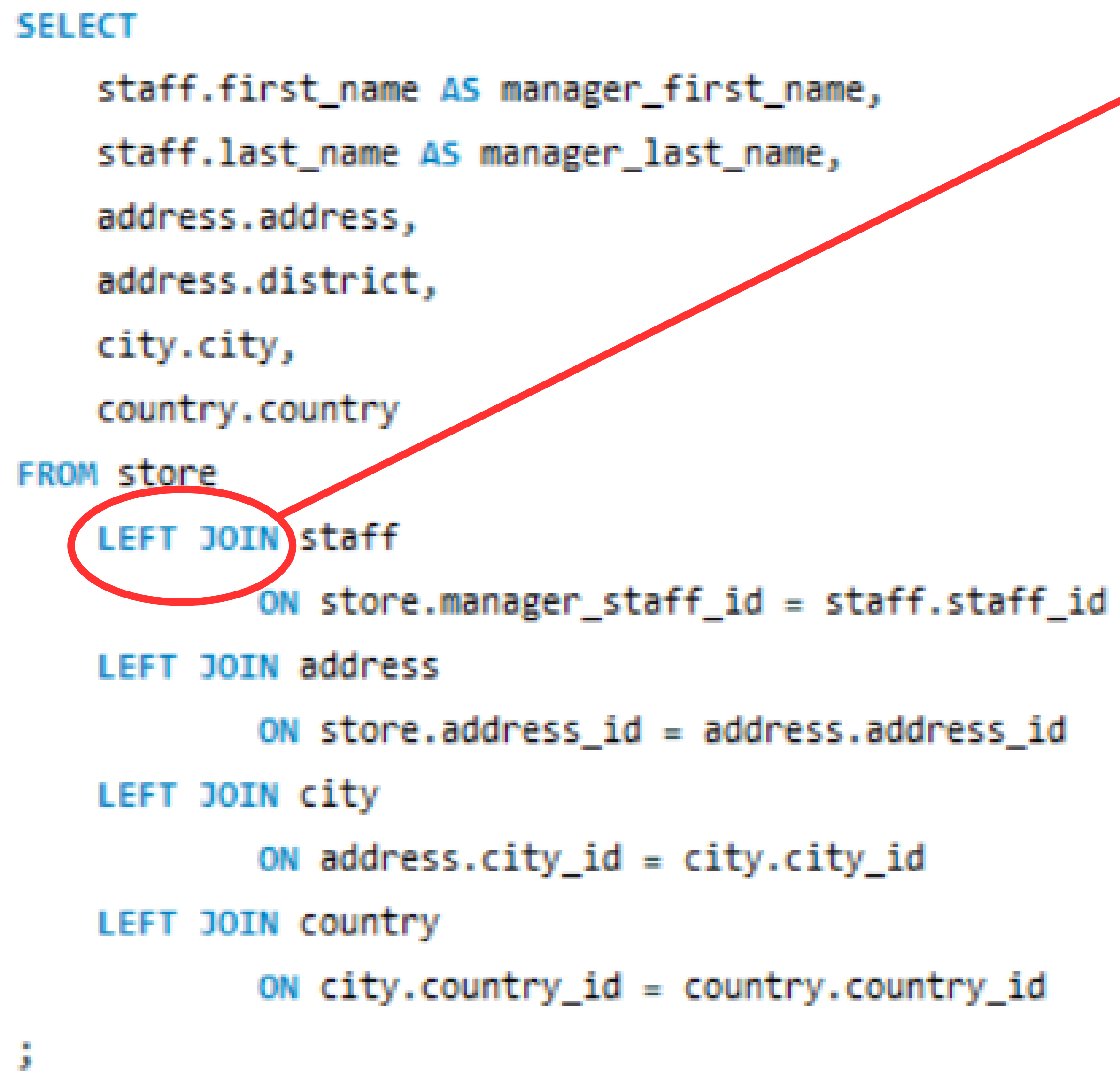
In this case our database includes **16 related tables,** containing information about:

- **Customers** (*Name, Address, etc.*)
- **Business** (*Staff, Rentals, etc.*)
- **Inventory** (*Films, Categories, etc.*)

FINAL PROJECT QUESTIONS

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 please).

MY SQL QUERY IN ACTION



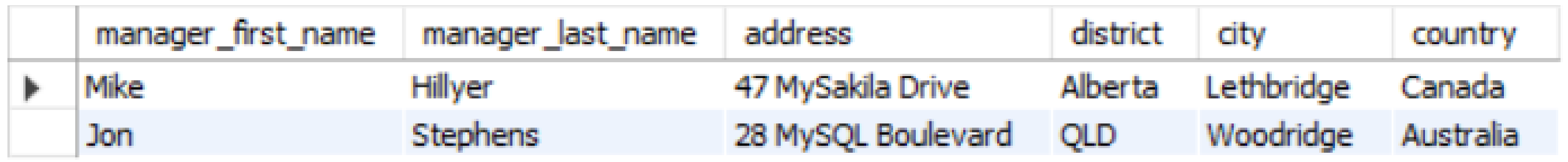
```
SELECT
    staff.first_name AS manager_first_name,
    staff.last_name AS manager_last_name,
    address.address,
    address.district,
    city.city,
    country.country
FROM store
    LEFT JOIN staff
            ON store.manager_staff_id = staff.staff_id
    LEFT JOIN address
            ON store.address_id = address.address_id
    LEFT JOIN city
            ON address.city_id = city.city_id
    LEFT JOIN country
            ON city.country_id = country.country_id
;
```

Why LEFT JOIN Is Used in the Query

The LEFT JOIN ensures that all stores from the store table appear in the result, even if some related data is missing from the staff, address, city, or country tables.

Store → staff : Ensures all stores are included, even if no staff is currently assigned as their manager.
Staff → address : Ensures staff records are included even if their address information is missing.
Address → city : Ensures address records are included even if the city information is missing.
City → country : Ensures cities are included even if no country data is available.

QUERY RESULTS

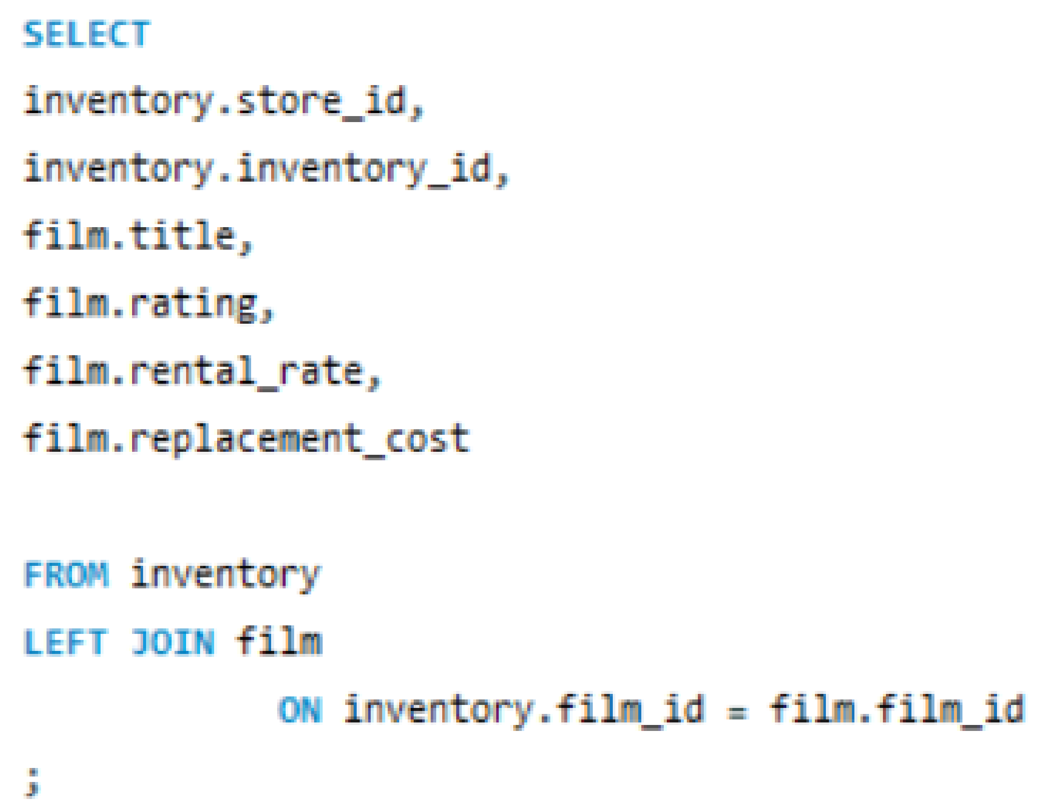


	manager_first_name	manager_last_name	address	district	city	country
▶	Mike	Hillyer	47 MySakila Drive	Alberta	Lethbridge	Canada
	Jon	Stephens	28 MySQL Boulevard	QLD	Woodridge	Australia

FINAL PROJECT QUESTIONS

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.

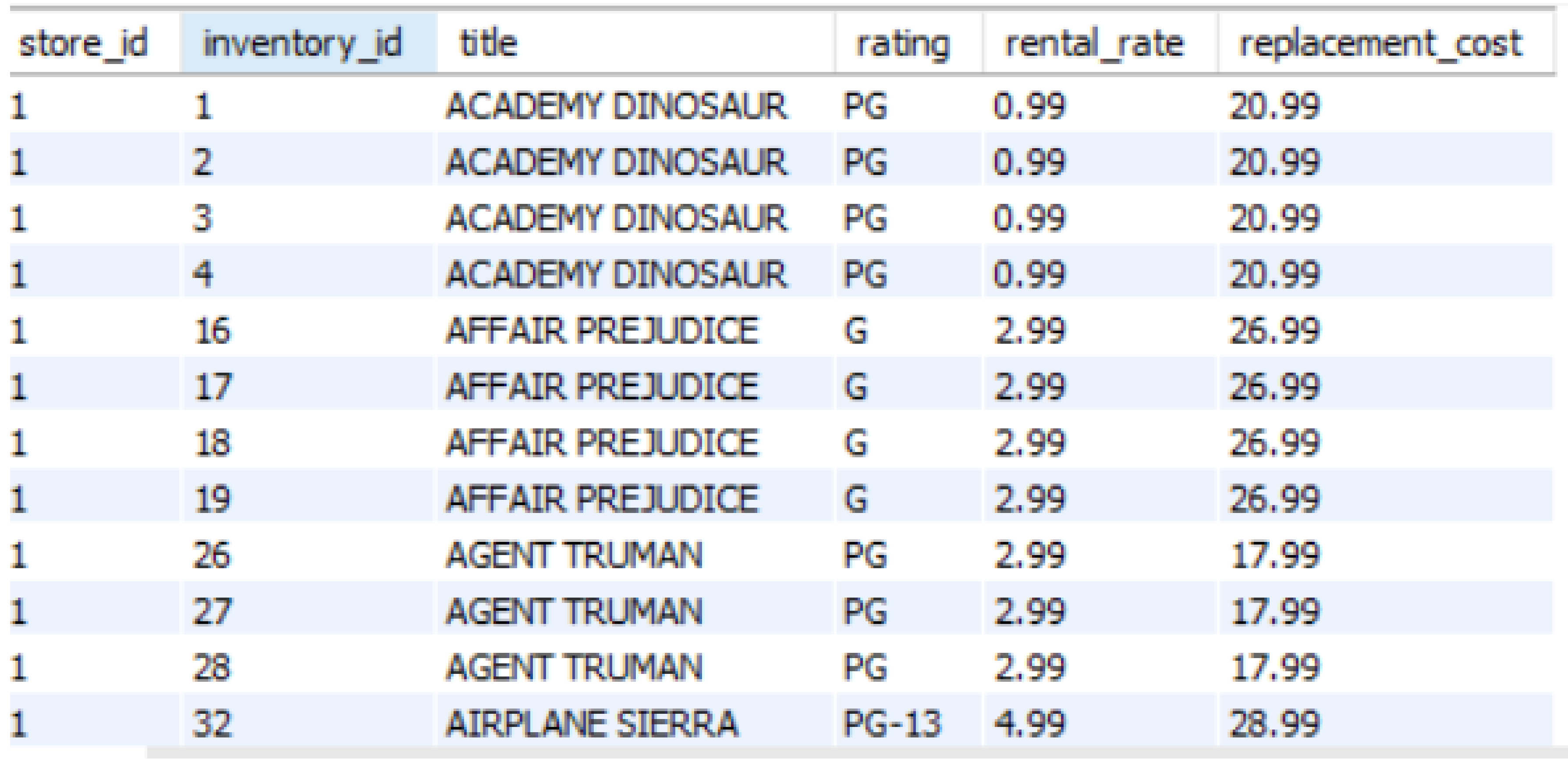
MY SQL QUERY IN ACTION



```
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
;
```

QUERY RESULTS

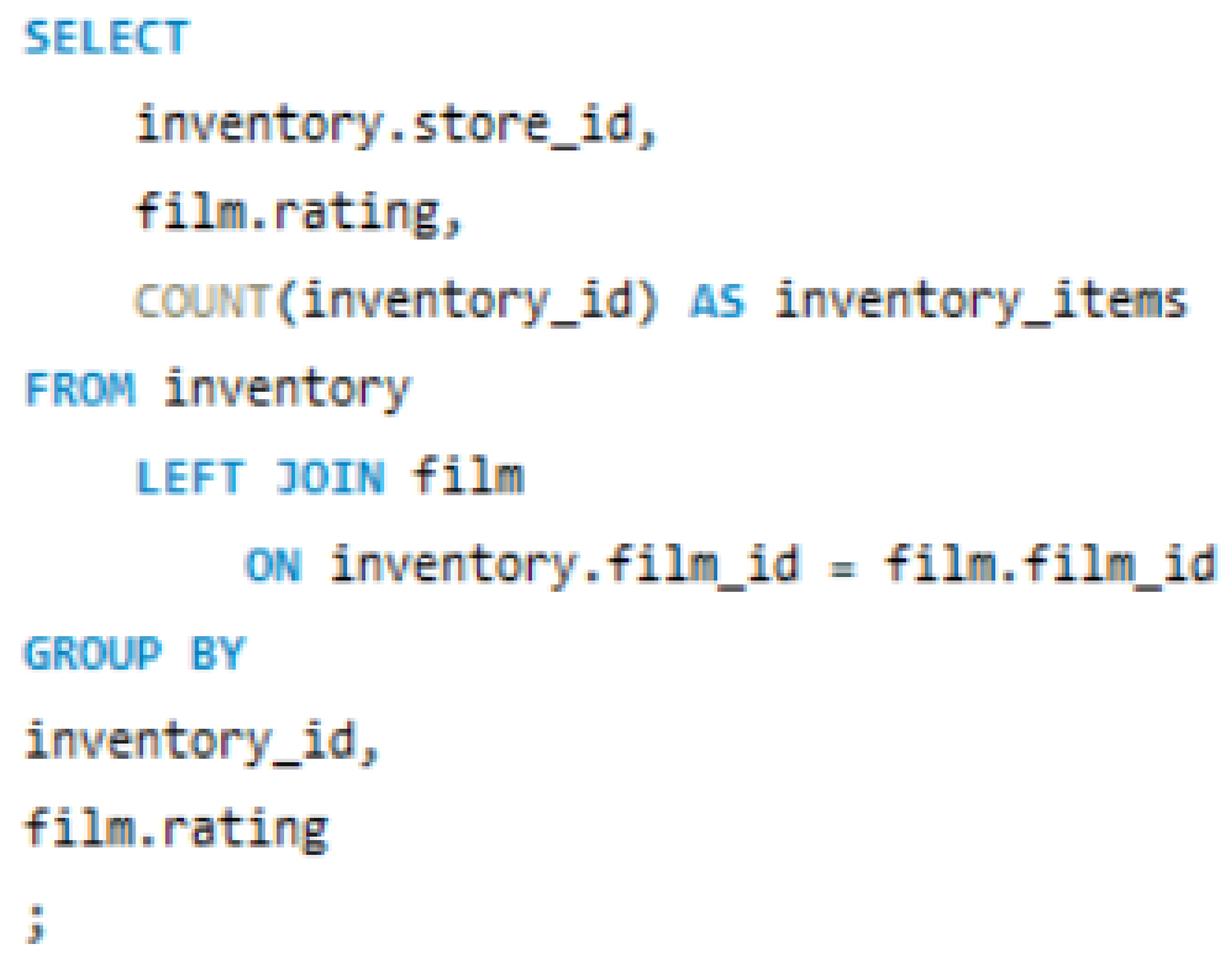


store_id	inventory_id	title	rating	rental_rate	replacement_cost
1	1	ACADEMY DINOSAUR	PG	0.99	20.99
1	2	ACADEMY DINOSAUR	PG	0.99	20.99
1	3	ACADEMY DINOSAUR	PG	0.99	20.99
1	4	ACADEMY DINOSAUR	PG	0.99	20.99
1	16	AFFAIR PREJUDICE	G	2.99	26.99
1	17	AFFAIR PREJUDICE	G	2.99	26.99
1	18	AFFAIR PREJUDICE	G	2.99	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

FINAL PROJECT QUESTIONS

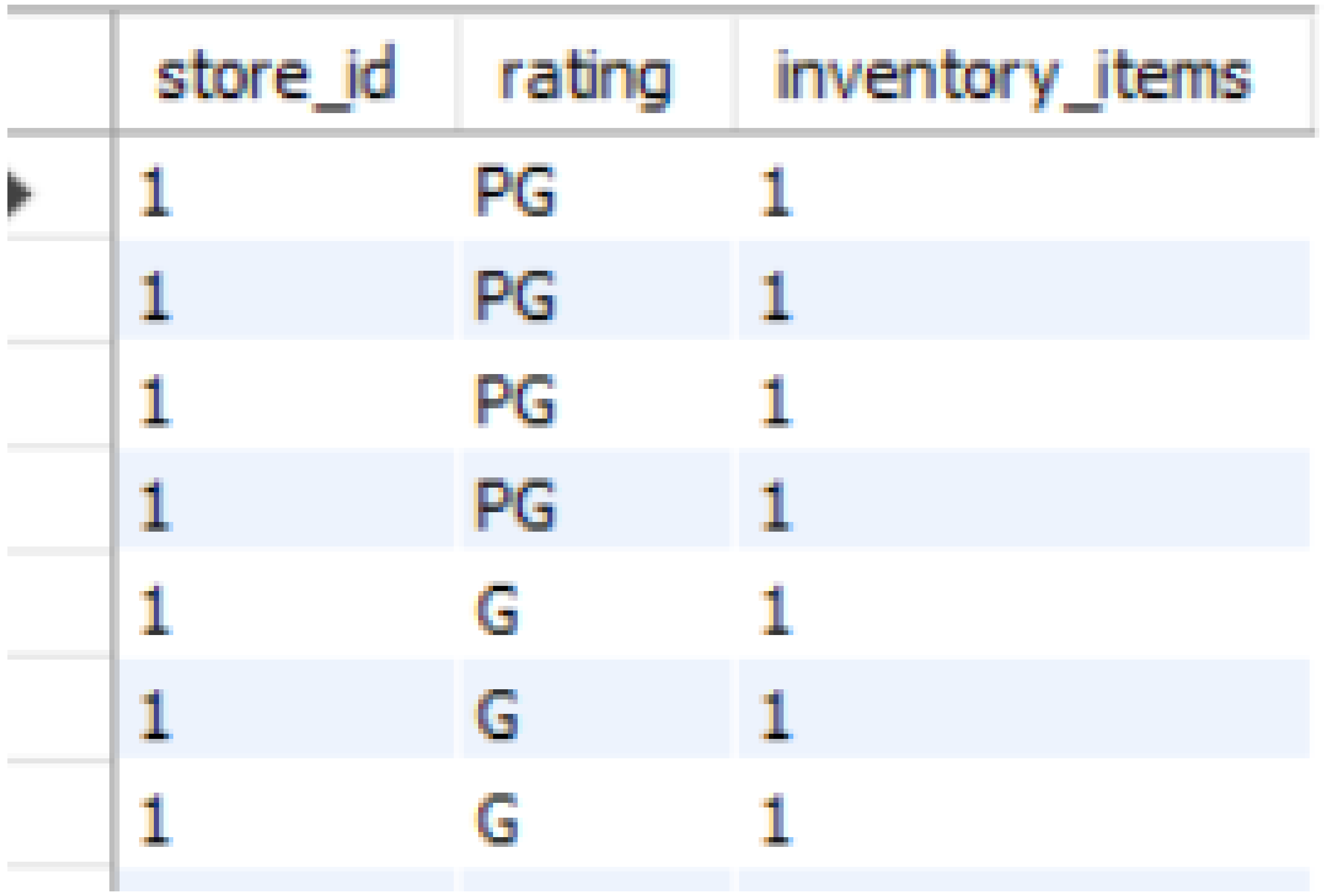
From the same list of films you just pulled, please roll that data up and provide a summary level overview of your inventory. We would like to know how many inventory items you have with each rating at each store.

MY SQL QUERY IN ACTION



```
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_id,
film.rating
;
```

QUERY RESULTS

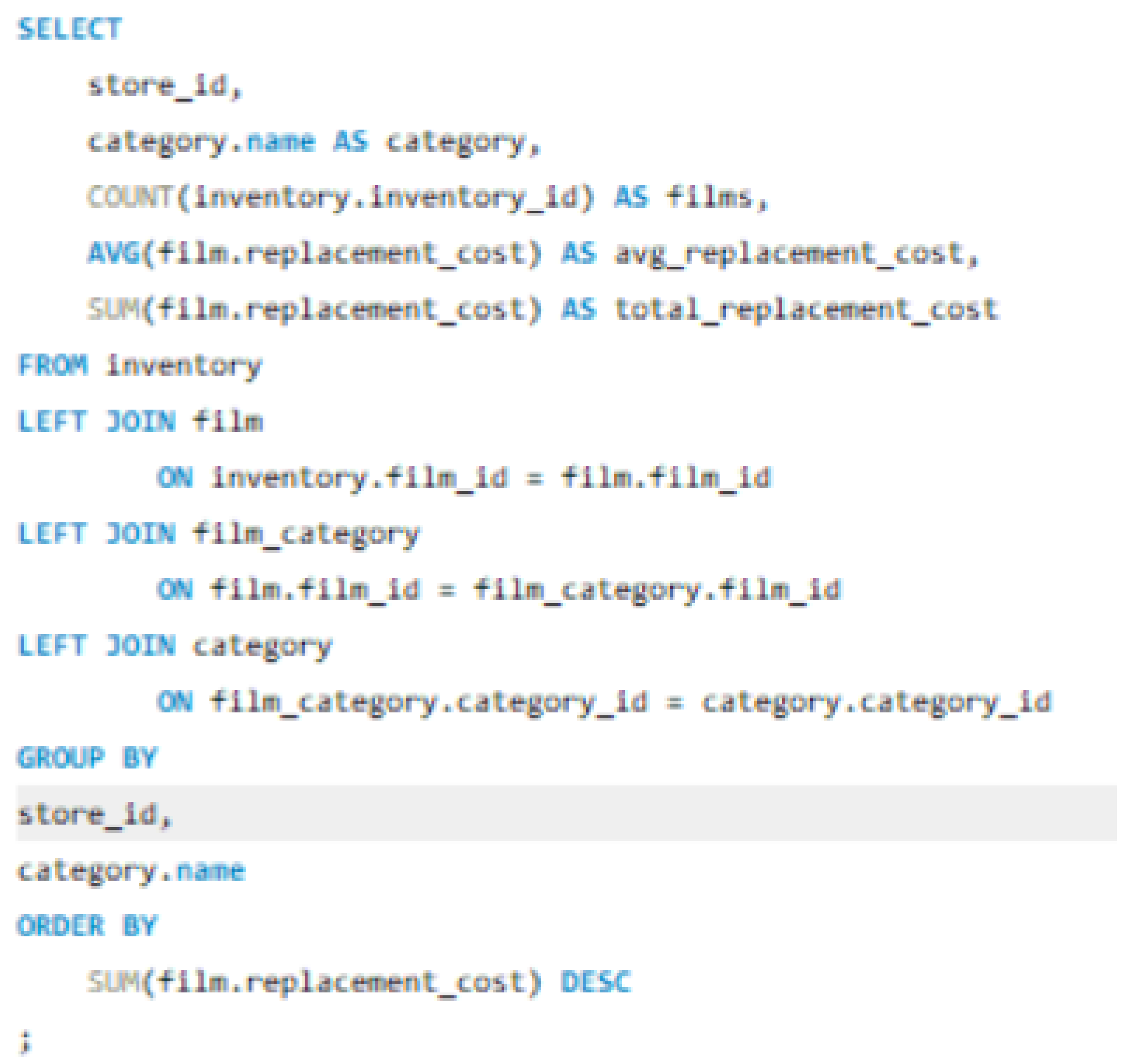


store_id	rating	inventory_items
1	PG	1
1	PG	1
1	PG	1
1	PG	1
1	G	1
1	G	1
1	G	1

FINAL PROJECT QUESTIONS

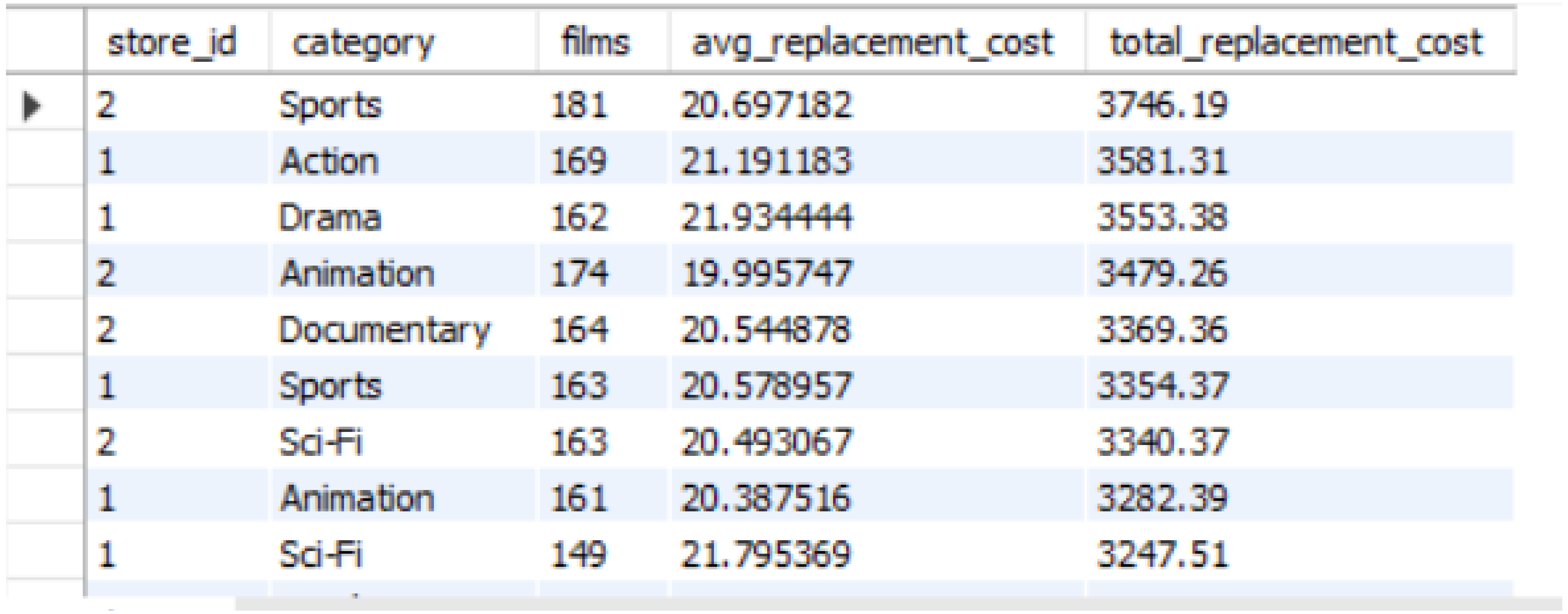
Similarly, we want to understand how diversified the inventory is in terms of replacement cost. We want to see how big of a hit it would be if a certain category of film became unpopular at a certain store. We would like to see the number of films, as well as the average replacement cost, and total replacement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

MY SQL QUERY IN ACTION



```
SELECT
    store_id,
    category.name AS category,
    COUNT(inventory.inventory_id) AS films,
    AVG(film.replacement_cost) AS avg_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
store_id,
category.name
ORDER BY
    SUM(film.replacement_cost) DESC
;
```

QUERY RESULTS

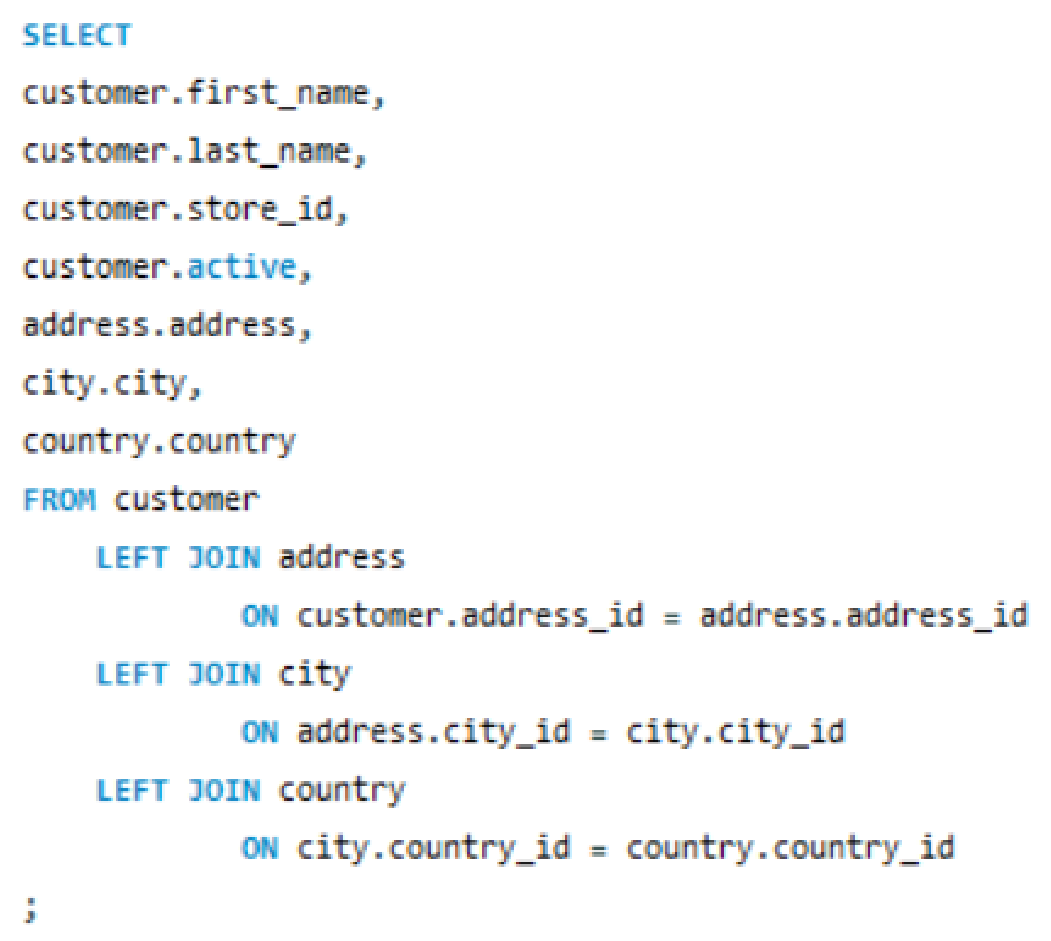


store_id	category	films	avg_replacement_cost	total_replacement_cost
2	Sports	181	20.697182	3746.19
1	Action	169	21.191183	3581.31
1	Drama	162	21.934444	3553.38
2	Animation	174	19.995747	3479.26
2	Documentary	164	20.544878	3369.36
1	Sports	163	20.578957	3354.37
2	Sci-Fi	163	20.493067	3340.37
1	Animation	161	20.387516	3282.39
1	Sci-Fi	149	21.795369	3247.51

FINAL PROJECT QUESTIONS

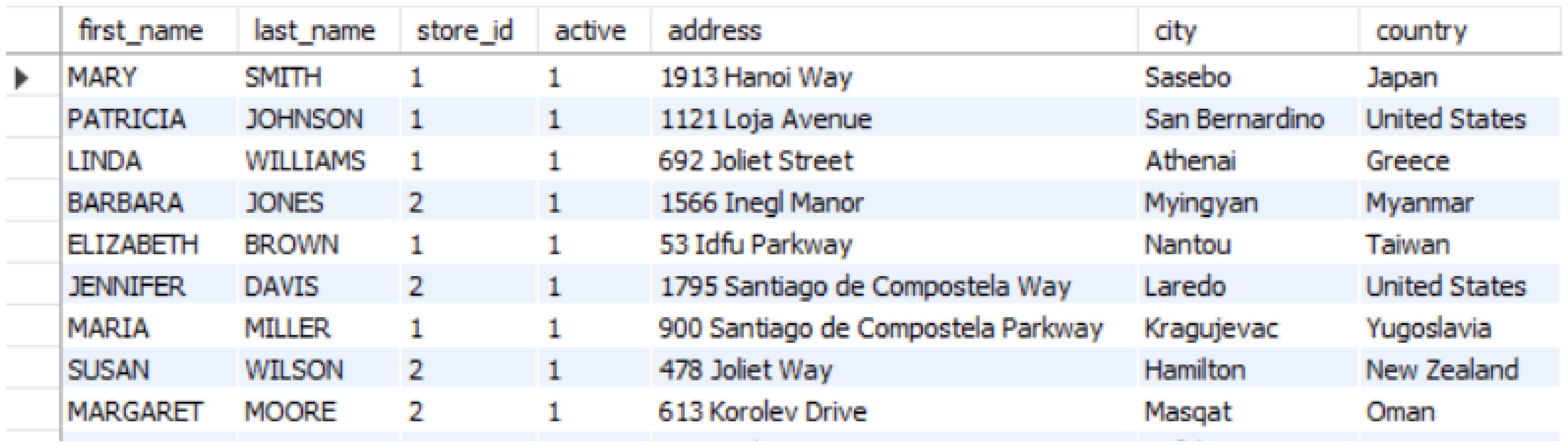
We want to make sure you folks have a good handle on who your customers are. Please provide a list of all customer names, which store they go to, whether or not they are currently active, and their full addresses – street address, city, and country.

MY SQL QUERY IN ACTION



```
SELECT
customer.first_name,
customer.last_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
;
```

QUERY RESULTS

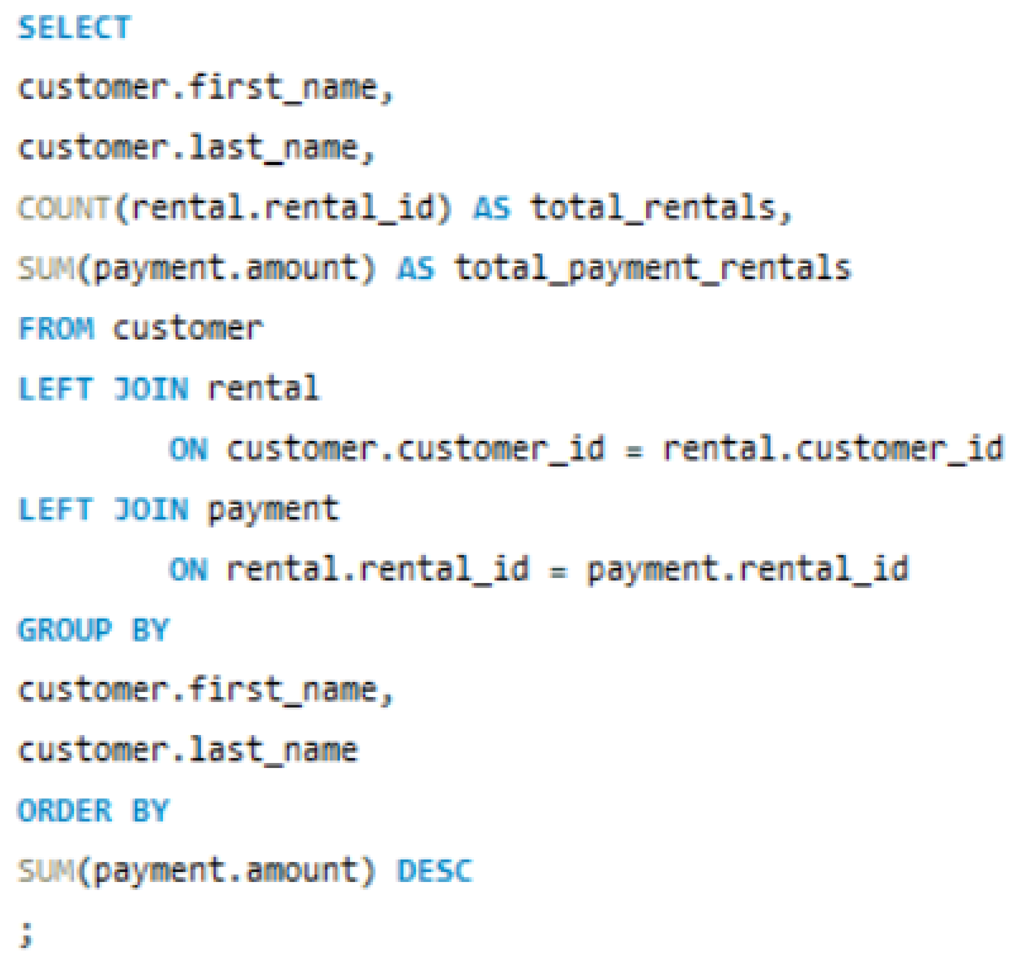


first_name	last_name	store_id	active	address	city	country
MARY	SMITH	1	1	1913 Hanoi Way	Sasebo	Japan
PATRICIA	JOHNSON	1	1	1121 Loja Avenue	San Bernardino	United States
LINDA	WILLIAMS	1	1	692 Joliet Street	Athenai	Greece
BARBARA	JONES	2	1	1566 Inegl Manor	Myingyan	Myanmar
ELIZABETH	BROWN	1	1	53 Idfu Parkway	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

FINAL PROJECT QUESTIONS

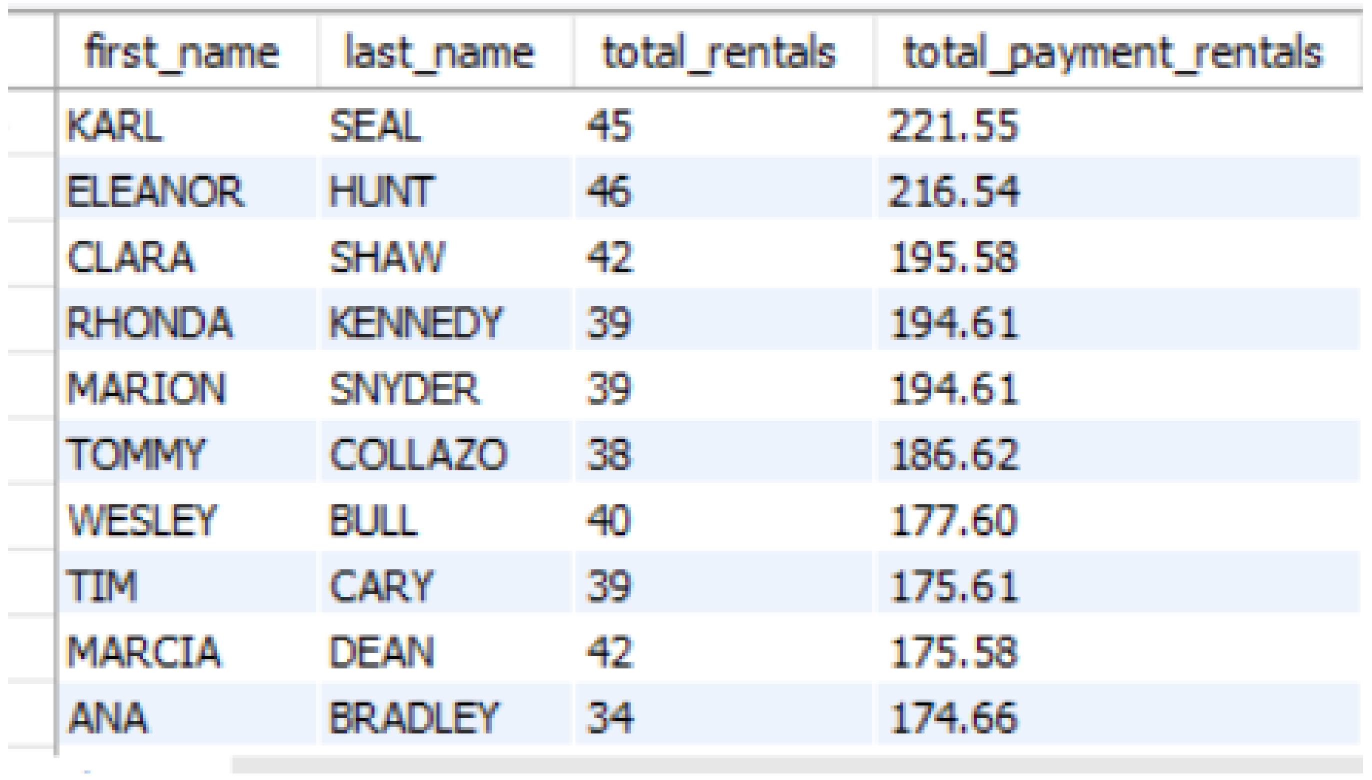
We would like to understand how much your customers are spending with you, and also to know who your most valuable customers are. Please pull together a list of customer names, their total lifetime rentals, and the sum of all payments you have collected from them. It would be great to see this ordered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

MY SQL QUERY IN ACTION



```sql
SELECT
customer.first_name,
customer.last_name,
COUNT(rental.rental_id) AS total_rentals,
SUM(payment.amount) AS total_payment_rentals
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
SUM(payment.amount) DESC
;
```

QUERY RESULTS

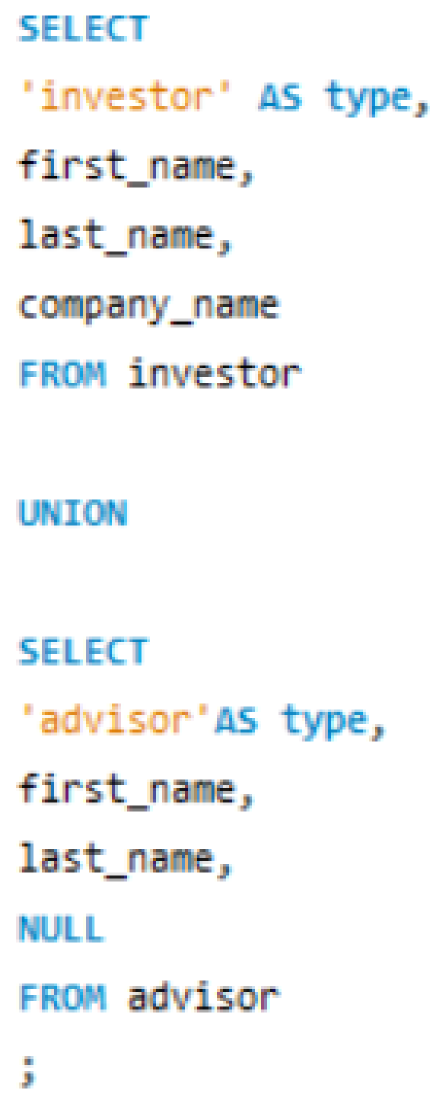


first_name	last_name	total_rentals	total_payment_rentals
KARL	SEAL	45	221.55
ELEANOR	HUNT	46	216.54
CLARA	SHAW	42	195.58
RHONDA	KENNEDY	39	194.61
MARION	SNYDER	39	194.61
TOMMY	COLLAZO	38	186.62
WESLEY	BULL	40	177.60
TIM	CARY	39	175.61
MARCIA	DEAN	42	175.58
ANA	BRADLEY	34	174.66

FINAL PROJECT QUESTIONS

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.

MY SQL QUERY IN ACTION

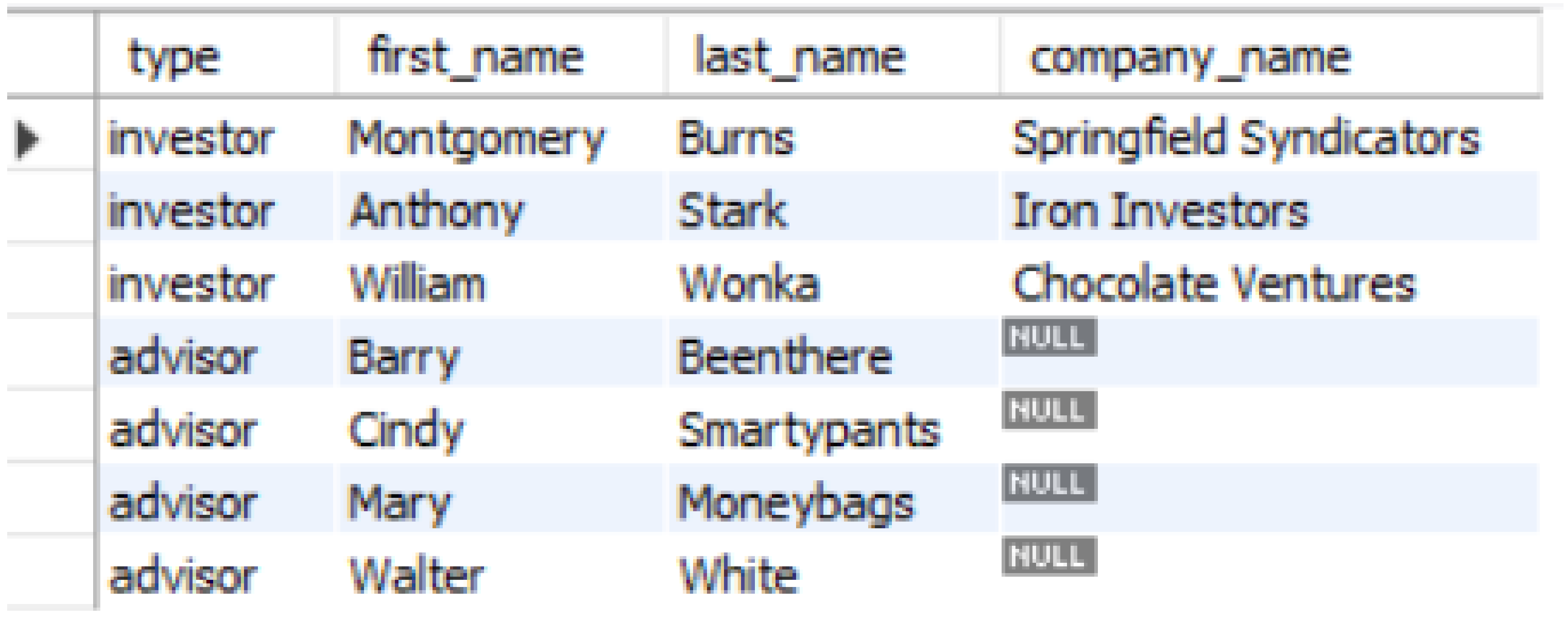


```
SELECT
'investor' AS type,
first_name,
last_name,
company_name
FROM investor

UNION

SELECT
'advisor'AS type,
first_name,
last_name,
NULL
FROM advisor
;
```

QUERY RESULTS

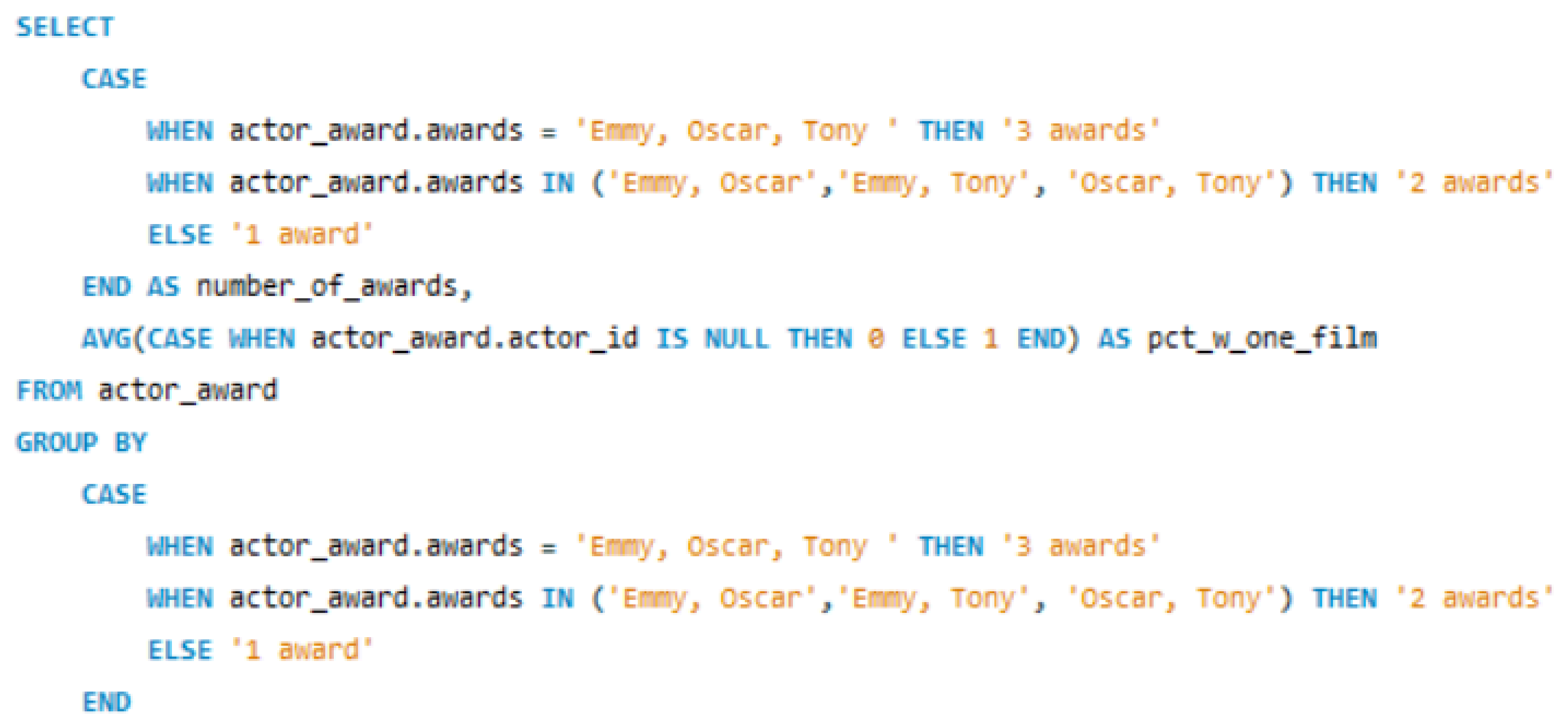


type	first_name	last_name	company_name
investor	Montgomery	Burns	Springfield Syndicators
investor	Anthony	Stark	Iron Investors
investor	William	Wonka	Chocolate Ventures
advisor	Barry	Beenthere	NULL
advisor	Cindy	Smartypants	NULL
advisor	Mary	Moneybags	NULL
advisor	Walter	White	NULL

FINAL PROJECT QUESTIONS

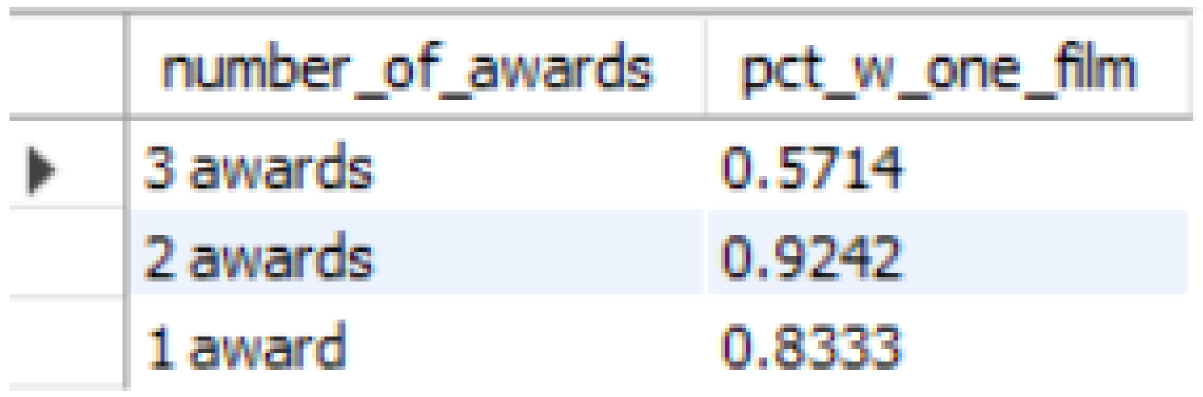
We're interested in how well you have covered the most-awarded actors. Of all the actors with three types of awards, for what % of them do we carry a film? And how about for actors with two types of awards? Same questions. Finally, how about actors with just one award?

MY SQL QUERY IN ACTION



```
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_awards,
    AVG(CASE WHEN actor_award.actor_id IS NULL THEN 0 ELSE 1 END) AS pct_w_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
```

QUERY RESULTS



number_of_awards	pct_w_one_film
3 awards	0.5714
2 awards	0.9242
1 award	0.8333

THANK
YOU

+62 85749999556

wnovitasari34@gmail.com

github.com/noviiwnn

linkedin.com/wahyunovitasari