For this Project I am assuming the role as an owner of the DVD rental shop. Since this is a business the profitability and sales of the business provides an upper-level view of the business. However due to pending financial data, I am limiting to reviewing business performance based on the following 2:

* ***Sales made by location/Store***: This is to evaluate which location is profitable and can lead to business decisions such as deciding if the location is worth keeping or moving to another location
* ***Sales made by film genre***: Certain films attract certain level of customers and having a stock of films fitting that criteria will keep customers from going to our competitor

**Sales made by location/Store**

As a business owner, one of the most crucial decisions that an owner can make is to decide if the location provides a necessary stream of revenue. Another use case scenario is to see which area and which date the rental is being carried out. Dates provide an idea on which days are rental more frequent as well as the Genre usually being picked on.

Two views are being used here:

* ***sales\_by\_location\_detailed***: which provides the sales data in detailed by Genre and location where the individual sales by location which uses two fields “city.city” and “country.country” columns concatenated with spaces. The columns are taken from the City table and the Country table.

The table is made from 11 tables using a JOIN statement with the main table being the “Payment” Table, the other tables that are joined are from the “Rental”, “Inventory”, ”Film”, ”Film\_Category”, “Category”, “Staff”, “Store”, “Address”, “City” and “Country” Tables which have been joined using the respective Primary and Foreign keys (Please see code).

The main fields that are being utilized are the following:

* City (City Table) Concatenated with “, “ and Country (Country table) columns
* Payment\_ID (Payment Table)
* Rental\_ID (Payment Table)
* Payment\_Date (Payment Table) that using CAST() has been converted to a DATE Format making it easier to read what date the payment was made instead of a TIMESTAMP format
* Inventory\_ID (Rental Table)
* Name (Category Table) which was renamed as “Genre” for it to be easier to identify respective film category
* Title (Film Table) which is the name of the Films
* Amount (Payment Table) which was using the CAST() function converted to MONEY as currency format is a financial view that is easy to understand when presenting to view at a high level view

1. CREATE OR REPLACE VIEW sales\_by\_location\_detailed AS

2. SELECT

3. city.city||', '||country.country AS city\_address,

4. payment.payment\_id,

5. payment.rental\_id,

6. CAST(payment.payment\_date as DATE),

7. rental.inventory\_id,

8. category.name AS Genre,

9. film.title,

10. CAST(payment.amount AS MONEY)

11. FROM payment

12. INNER JOIN rental ON rental.rental\_id = payment.rental\_id

13. INNER JOIN inventory ON rental.inventory\_id= inventory.inventory\_id

14. INNER JOIN film ON inventory.film\_id = film.film\_id

15. INNER JOIN film\_category ON film\_category.film\_id = film.film\_id

16. INNER JOIN category ON category.category\_id = film\_category.category\_id

17. INNER JOIN staff ON payment.staff\_id = staff.staff\_id

18. INNER JOIN store ON payment.staff\_id = store.manager\_staff\_id

19. INNER JOIN address ON store.address\_id=address.address\_id

20. INNER JOIN city ON city.city\_id =address.city\_id

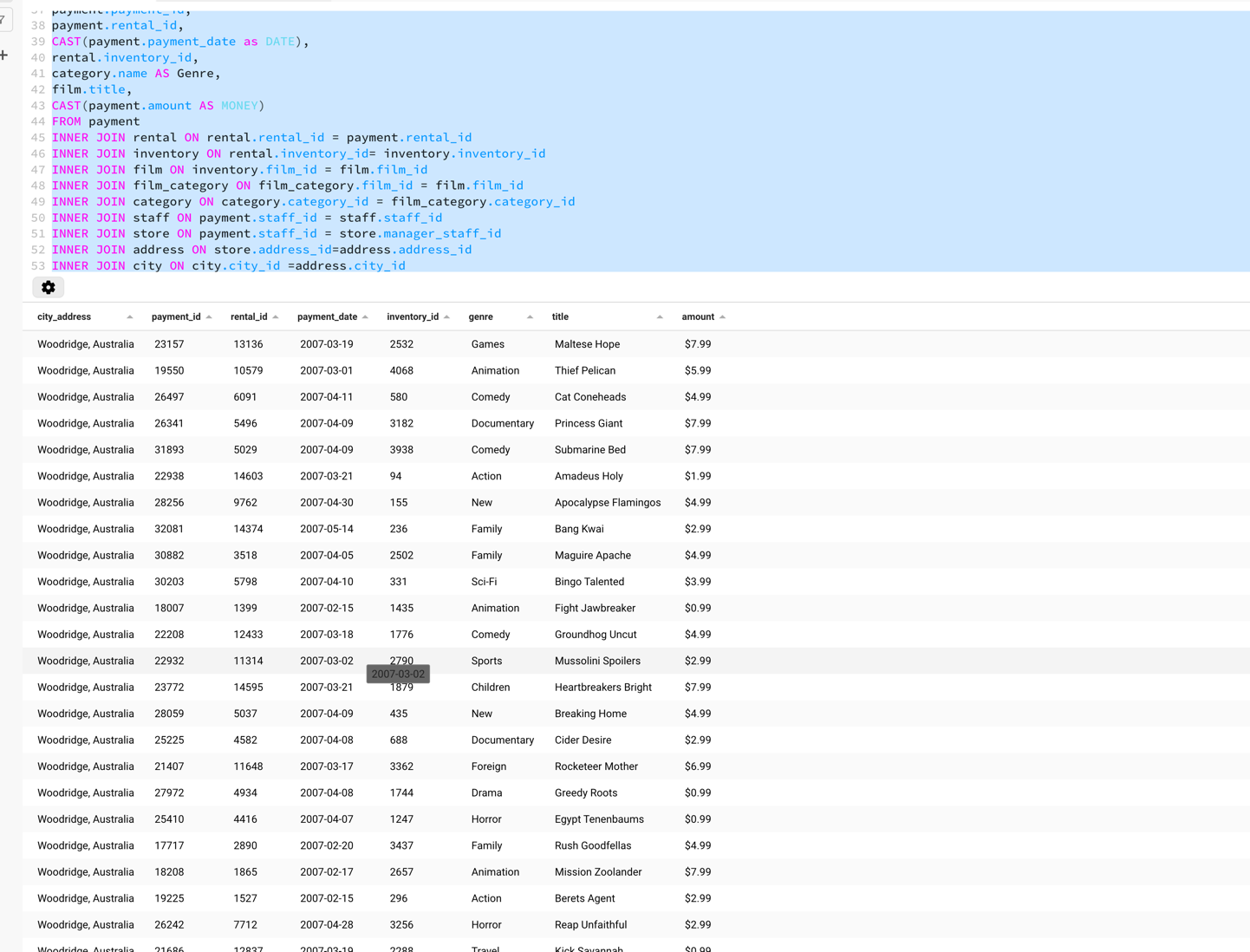
21. INNER JOIN country ON city.country\_id = country.country\_id

22. GROUP BY category.name, payment.payment\_id, rental.inventory\_id, city.city, country.country, film.title

23. ORDER BY city\_address DESC;

24.

Screenshot also can be seen in the screenshot folder



* ***sales\_by\_location\_summary*** *:* The Summary view provides the business owners a quick glance of how much revenue each location made. As a business owner I would use such a report first and compare location revenue to see under or over performing and based on that next decision to review further can be taken. This view only requires to Fields
* City (City Table Column) Concatenated with “, “ and Country (Country table) columns
* Amount (Payment Table Column) using the SUM() function location was summed up and then using the CAST() was converted to MONEY Format as the “$” makes it easier financially to identify money related transactions

Two Tables are necessary for this view to work; The Payment Table and the Staff Table. The Payment provides the Revenue details whereas the Staff table provides the Staff id which can identify the location of the where the rental revenue was made.

The Table utilizes the same JOINS and GROUPING as from the summary table however, the only difference are only 2 columns being used mentioned above.

1. CREATE OR REPLACE VIEW sales\_by\_location\_summary AS

2. SELECT

3. city.city||', '||country.country AS city\_address,

4. CAST(SUM(payment.amount) as money)

5. FROM payment

6. INNER JOIN staff

7. ON payment.staff\_id = staff.staff\_id

8. INNER JOIN store

9. ON payment.staff\_id = store.manager\_staff\_id

10. INNER JOIN address

11. ON store.address\_id=address.address\_id

12. INNER JOIN city

13. ON city.city\_id =address.city\_id

14. INNER JOIN country

15. ON city.country\_id = country.country\_id

16. GROUP by payment.staff\_id, staff.staff\_id,address.address\_id, city.city,country.country\_id

17. ORDER BY city\_address, SUM(payment.amount);

18.

19. SELECT \* FROM sales\_by\_location\_summary;

20.

Screenshot of Summary also found in the Screenshot folder

