**Q1.** What are the top and least rented (in-demand) genres and what are their total sales?

 I first identified with tables I would need to Join, which are:

Category>film\_category>film>inventory>rental>customer>payment

Below is the query used to extract to answer the question:

Query: With t1 AS (SELECT \* FROM

(SELECT category.name AS genre, count(customer\_csv.customer\_id) AS total\_rent\_demand

FROM (customer\_csv AS customer\_csv

JOIN (rental\_redshift AS rental\_redshift

JOIN (f\_inventory\_json AS f\_inventory\_json

JOIN (f\_film\_xml AS f\_film\_xml

JOIN (category AS category

JOIN filmcategory\_xml AS f\_film\_category\_xml

ON category.category\_id = f\_film\_category\_xml.category\_id)

ON f\_film\_xml.film\_id = f\_film\_category\_xml.film\_id)

ON f\_inventory\_json.film\_id = f\_film\_xml.film\_id)

ON rental\_redshift.inventory\_id = f\_inventory\_json.inventory\_id)

ON customer\_csv.customer\_id = rental\_redshift.customer\_id)

GROUP BY category.name ) ORDER BY total\_rent\_demand DESC),

t2 AS (SELECT \* FROM

(SELECT category.name AS genre, sum(f\_payment\_json.amount) AS total\_sales

FROM (f\_payment\_json AS f\_payment\_json

JOIN (rental\_redshift AS rental\_redshift

JOIN (f\_inventory\_json AS f\_inventory\_json

JOIN (f\_film\_xml AS f\_film\_xml

JOIN (category AS category

JOIN filmcategory\_xml AS f\_film\_category\_xml

ON category.category\_id = f\_film\_category\_xml.category\_id)

ON f\_film\_xml.film\_id = f\_film\_category\_xml.film\_id)

ON f\_inventory\_json.film\_id = f\_film\_xml.film\_id)

ON rental\_redshift.inventory\_id = f\_inventory\_json.inventory\_id)

ON f\_payment\_json.rental\_id = rental\_redshift.rental\_id)

GROUP BY category.name ) ORDER BY total\_sales DESC)

SELECT \* FROM

(SELECT t1.genre AS genre, t1.total\_rent\_demand AS total\_rent\_demand, t2.total\_sales AS total\_sales

FROM (t1 AS t1 JOIN t2 AS t2 ON t1.genre = t2.genre)

GROUP BY t1.genre, t1.total\_rent\_demand, t2.total\_sales)

**Q2.** Can we know how many distinct users have rented each genre?

The tables to join are as follows:

Category > film\_Category > film > inventory > rental > customer

Below is my query for this question:

SELECT \* FROM

(SELECT DISTINCT name AS genre, count (DISTINCT customer\_id) AS total\_rent\_demand

FROM category\_j\_f\_film\_category\_xml\_j\_f\_film\_xml\_j\_f\_inventory\_json\_j\_rental\_redshift\_j\_customer\_csv\_ AS category\_j\_f\_film\_category\_xml\_j\_f\_film\_xml\_j\_f\_inventory\_json\_j\_rental\_redshift\_j\_customer\_csv\_

GROUP BY category\_j\_f\_film\_category\_xml\_j\_f\_film\_xml\_j\_f\_inventory\_json\_j\_rental\_redshift\_j\_customer\_csv\_.name ) ORDER BY total\_rent\_demand DESC

Insights:

I want to know how many distinct customers rented each of the genres. One fascinating thing from the query is that although the music genre has the least total rented record, it does not have the least number of distinct customers who rented the genre. The travel genre holds that record.

By taking a step back and connecting the insights derived from question 1 and 2, we can say that the travel genre was re-rented more times than the music genre.

And of course, the sports genre has the highest number of distinct customers who rented the genre.

**Q3.** What is the average rental rate for each genre? (from the highest to the lowest)

The tables to join are as follows:

Category > film\_Category **>**; film

Below is my query for this question:

SELECT \* FROM

(SELECT category.name AS genre, round(avg(f\_film\_xml.rental\_rate), 2) AS avg rental rate

FROM (f\_film\_xml AS f\_film\_xml

JOIN (category AS category

JOIN filmcategory\_xml AS f\_film\_category\_xml

ON category.category\_id = f\_film\_category\_xml.category\_id)

ON f\_film\_xml.film\_id = f\_film\_category\_xml.film\_id)

GROUP BY category.name ) ORDER BY avg rental rate DESC

**Q4.** In which countries do Rent A Film have a presence in and what is the customer base in each country? What are the total sales in each country? (From most to least)

The tables to join are as follows:

Country > City > Address > customer> payment

See query below:

SELECT \* FROM

(SELECT DISTINCT country\_csv.country AS country, sum(f\_payment\_json.amount) AS total\_sales, count(DISTINCT customer\_csv.customer\_id) AS customer\_base

FROM (f\_payment\_json AS f\_payment\_json

JOIN (customer\_csv AS customer\_csv

JOIN (address AS address

JOIN (country\_csv AS country\_csv

JOIN city\_csv AS city\_csv

ON country\_csv.country\_id = city\_csv.country\_id)

ON address.city\_id = city\_csv.city\_id)

ON customer\_csv.address\_id = address.address\_id)

ON f\_payment\_json.customer\_id = customer\_csv.customer\_id)

GROUP BY country\_csv.country ) ORDER BY customer\_base DESC

**Q5.** Who are the top 5 customers per total sales and can we get their detail just in case Rent A Film wants to reward them?

The tables to join are as follows:

Country > City > Address > customer> payment

See query below:

SELECT \* FROM

(SELECT (customer\_csv.first\_name||customer\_csv.last\_name) AS full\_name, customer\_csv.email AS email, address.address AS address, address.phone AS phone, city\_csv.city AS city, country\_csv.country AS country, sum(f\_payment\_json.amount) AS total\_purchase\_in\_currency

FROM (f\_payment\_json AS f\_payment\_json

JOIN (country\_csv AS country\_csv

JOIN (city\_csv AS city\_csv

JOIN (customer\_csv AS customer\_csv

JOIN address AS address

ON customer\_csv.address\_id = address.address\_id)

ON city\_csv.city\_id = address.city\_id)

ON country\_csv.country\_id = city\_csv.country\_id)

ON f\_payment\_json.customer\_id = customer\_csv.customer\_id)

GROUP BY customer\_csv.email,address.address,address.phone,city\_csv.city,country\_csv.country,customer\_csv.first\_name,customer\_csv.last\_name ) ORDER BY total\_purchase\_in\_currency DESC LIMIT 5;