SAKILA MOVIE RENTAL

SALES REPORT

YEARLY REPORT 2005

Presented by

Siddhant Chandekar

SAKILA MOVIE RENTAL: SALES REPORT

This report presents a comprehensive overview of the Sakila Movie Rental dataset, meticulously detailing each CSV file, which functions as a distinct table within a relational database structure. For each table, its primary purpose, columnar schema, and inferred data types are precisely delineated.

1. `actor.csv`

- **Purpose:** This file contains information about the actors involved in the films.

Column Name	Inferred Data Type	Description
actor_id	INTEGER	Unique identifier for each actor.
first_name	TEXT	The first name of the actor.
last_name	TEXT	The last name of the actor.
last_update	DATETIME	Timestamp of the last update to the record.

2. `address.csv`

- **Purpose:** This file stores address details for customers, staff members, and stores.

Column Name	Inferred Data Type	Description
address_id	INTEGER	Unique identifier for each address.
address	TEXT	The primary street address.
address2	TEXT	Secondary address information (e.g., apartment, suite number).
district	TEXT	The district or region within the city.
city_id	INTEGER	Foreign key linking to the `city` table.
postal_code	TEXT	The postal code for the address.
phone	TEXT	The phone number associated with the address.
location	TEXT	Geographical coordinates of the address (e.g., POINT (longitude latitude)).
last_update	DATETIME	Timestamp of the last update to the record.

3. `category.csv`

- **Purpose:** This file defines and categorizes the various genres or types of films available.

Column Name	Inferred Data Type	Description
category_id	INTEGER	Unique identifier for each film category.
name	TEXT	The name of the film category (e.g., Action, Comedy).
last_update	DATETIME	Timestamp of the last update to the record.

4. `city.csv`

- **Purpose:** This file lists city names and their corresponding countries.

Column Name	Inferred Data Type	Description
city_id	INTEGER	Unique identifier for each city.
city	TEXT	The name of the city.
country_id	INTEGER	Foreign key linking to the `country` table.
last_update	DATETIME	Timestamp of the last update to the record.

5. `country.csv`

- **Purpose:** This file lists the countries associated with cities in the database.

Column Name	Inferred Data Type	Description
country_id	INTEGER	Unique identifier for each country.
country	TEXT	The name of the country.
last_update	DATETIME	Timestamp of the last update to the record.

6. `customer.csv`

- **Purpose:** This file contains detailed information about each customer of the rental service.

Column Name	Inferred Data Type	Description
customer_id	INTEGER	Unique identifier for each customer.
store_id	INTEGER	Foreign key linking to the `store` table, indicating the customer's primary store.
first_name	TEXT	The first name of the customer.
last_name	TEXT	The last name of the customer.
email	TEXT	The email address of the customer.
address_id	INTEGER	Foreign key linking to the `address` table.
active	INTEGER	Indicates if the customer account is active (1 for active, 0 for inactive).
create_date	DATETIME	The date and time the customer account was created.
last_update	DATETIME	Timestamp of the last update to the record.

7. `film.csv`

- **Purpose:** This file provides comprehensive details about each movie in the rental catalog.

Column Name	Inferred Data Type	Description
film_id	INTEGER	Unique identifier for each film.
title	TEXT	The title of the film.
description	TEXT	A brief summary or synopsis of the film.
release_year	INTEGER	The year the film was released.
language_id	INTEGER	Foreign key linking to the `language` table.
original_language_id	INTEGER	Original language ID (if different from `language_id`).
rental_duration	INTEGER	The number of days a film can be rented for.
rental_rate	DECIMAL	The cost to rent the film.

length	INTEGER	The running length of the film in minutes.
replacement_cost	DECIMAL	The cost to replace the film if lost or damaged.
rating	TEXT	The MPAA rating of the film (e.g., G, PG, NC-17).
special_features	TEXT	A list of special features included (e.g., Trailers, Deleted Scenes).
last_update	DATETIME	Timestamp of the last update to the record.

8. `film_actor.csv`

- **Purpose:** This file acts as a junction table, linking actors to the films they have appeared in (many-to-many relationship).

Column Name	Inferred Data Type	Description
actor_id	INTEGER	Foreign key linking to the `actor` table.
film_id	INTEGER	Foreign key linking to the `film` table.
last_update	DATETIME	Timestamp of the last update to the record.

9. `film_category.csv`

- **Purpose:** This file acts as a junction table, linking films to their respective categories or genres (many-to-many relationship).

Column Name	Inferred Data Type	Description
film_id	INTEGER	Foreign key linking to the `film` table.
category_id	INTEGER	Foreign key linking to the `category` table.
last_update	DATETIME	Timestamp of the last update to the record.

10. `film_text.csv`

- **Purpose:** This file provides additional textual information for films, potentially for full-text search or display purposes.

Column Name	Inferred Data Type	Description

film_id	INTEGER	Unique identifier for each film.
title	TEXT	The title of the film.
description	TEXT	A brief summary or synopsis of the film.

11. `inventory.csv`

- **Purpose:** This file details each physical copy of a film available at a specific store location.

Column Name	Inferred Data Type	Description
inventory_id	INTEGER	Unique identifier for each physical film copy.
film_id	INTEGER	Foreign key linking to the `film` table.
store_id	INTEGER	Foreign key linking to the `store` table.
last_update	DATETIME	Timestamp of the last update to the record.

12. `language.csv`

- **Purpose:** This file lists the various languages in which films are available.

Column Name	Inferred Data Type	Description
language_id	INTEGER	Unique identifier for each language.
name	TEXT	The name of the language (e.g., English, Italian).
last_update	DATETIME	Timestamp of the last update to the record.

13. `payment.csv`

- **Purpose:** This file records all payment transactions made by customers for rentals.

Column Name	Inferred Data Type	Description
payment_id	INTEGER	Unique identifier for each payment transaction.
customer_id	INTEGER	Foreign key linking to the `customer` table.

staff_id	INTEGER	Foreign key linking to the `staff` table (who processed the payment).
rental_id	INTEGER	Foreign key linking to the `rental` table.
amount	DECIMAL	The amount of the payment.
payment_date	DATETIME	The date and time the payment was made.
last_update	DATETIME	Timestamp of the last update to the record.

14. `rentat.csv` (likely `rental.csv`)

- **Purpose:** This file records each film rental transaction.

Column Name	Inferred Data Type	Description
rental_id	INTEGER	Unique identifier for each rental transaction.
rental_date	DATETIME	The date and time the film was rented.
inventory_id	INTEGER	Foreign key linking to the `inventory` table (the specific copy rented).
customer_id	INTEGER	Foreign key linking to the `customer` table.
return_date	DATETIME	The date and time the film was returned (can be NULL if not yet returned).
staff_id	INTEGER	Foreign key linking to the `staff` table (who processed the rental).
last_update	DATETIME	Timestamp of the last update to the record.

15. `staff.csv`

- **Purpose:** This file contains information about the staff members working at the rental stores.

Column Name	Inferred Data Type	Description
staff_id	INTEGER	Unique identifier for each staff member.
first_name	TEXT	The first name of the staff member.
last_name	TEXT	The last name of the staff member.
address_id	INTEGER	Foreign key linking to the `address` table.

picture	TEXT	A reference or path to the staff member's picture.
email	TEXT	The email address of the staff member.
store_id	INTEGER	Foreign key linking to the `store` table (where the staff member works).
active	INTEGER	Indicates if the staff account is active (1 for active, 0 for inactive).
username	TEXT	The username for staff login.
password	TEXT	The hashed password for staff login.
last_update	DATETIME	Timestamp of the last update to the record.

16. `store.csv`

- **Purpose:** This file details the different rental store locations.

Column Name	Inferred Data Type	Description
store_id	INTEGER	Unique identifier for each store.
manager_staff_id	INTEGER	Foreign key linking to the `staff` table (the manager of the store).
address_id	INTEGER	Foreign key linking to the `address` table.
last_update	DATETIME	Timestamp of the last update to the record.

EDA PROBLEM:

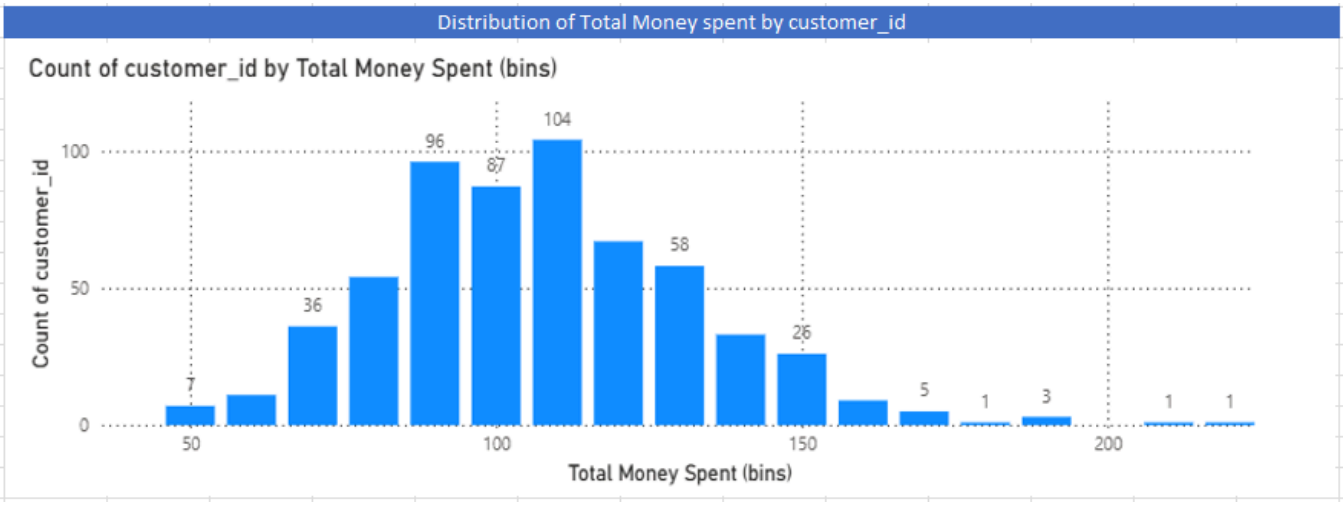
Q1. What are the purchasing patterns of new customers versus repeat customers?

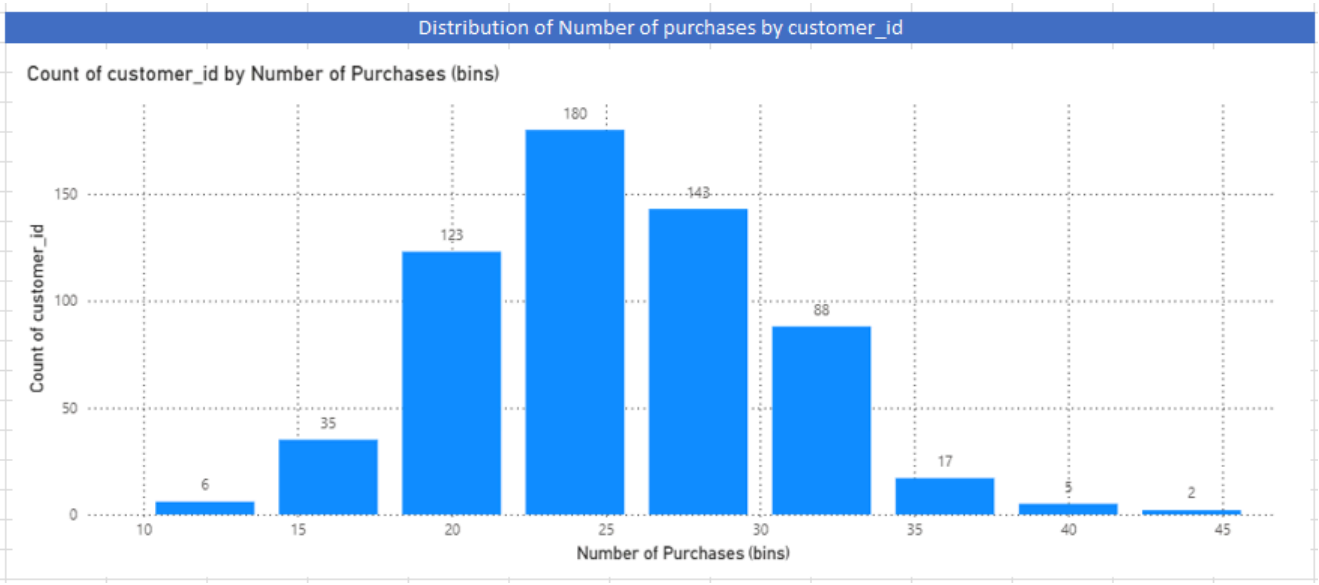
```
SELECT
  customer_type,
  COUNT(customer_id) AS number_of_customers
FROM (
  SELECT
    customer_id,
    CASE
      WHEN COUNT(customer_id) > 1 THEN 'Repeat Customer'
      ELSE 'New Customer'
    END AS customer_type
  FROM
    payment
  GROUP BY
    customer_id
) AS CustomerSegmentation
GROUP BY
  customer_type;
```

customer_type	NoOfCustomers
Repeat Customer	599

```
SELECT
  customer_id,
  SUM(amount) AS total_spending,
  COUNT(rental_id) AS no_of_purchases,
  AVG(amount) AS avg_amomunt_spent
FROM
  payment
GROUP BY
  customer_id
ORDER BY
  customer_id;
```

customer_id	total_spending	no_of_purchases	avg_amomunt_spent
1	118.68	32	3.71
2	128.73	27	4.77
3	135.74	26	5.22
4	81.78	22	3.72
5	144.62	38	3.81
6	93.72	28	3.35
7	151.67	33	4.60
8	92.76	24	3.86
9	89.77	23	3.90
10	99.75	25	3.99
11	106.76	24	4.45
12	103.72	28	3.70
13	131.73	27	4.88
14	117.72	28	4.20
15	134.68	32	4.21
16	118.72	28	4.24





Q2. Which films have the highest rental rates and are most in demand?

Movies with the highest rental rates

```
SELECT
   title,
   rental_rate
FROM
   film
ORDER BY
   rental_rate DESC
LIMIT 10;
```

title	rental_rate
CASUALTIES ENCINO	4.99
ACE GOLDFINGER	4.99
LEAGUE HELLFIGHTERS	4.99
FRISCO FORREST	4.99
PREJUDICE OLEANDER	4.99
FRONTIER CABIN	4.99
AIRPLANE SIERRA	4.99
AIRPORT POLLOCK	4.99
POSEIDON FOREVER	4.99
ALADDIN CALENDAR	4.99

Most high demand movies

```
SELECT
   f.title,
   COUNT(r.rental_id) AS rental_count,
   f.rental_rate
FROM
   film AS f
JOIN
   inventory AS i ON f.film_id = i.film_id
JOIN
   rentat AS r ON i.inventory_id = r.inventory_id
GROUP BY
   f.film_id, f.title, f.rental_rate
ORDER BY
   rental_count DESC
LIMIT 10;
```

title	rental_count	rental_rate
BUCKET BROTHERHOOD	34	4.99
ROCKETEER MOTHER	33	0.99
JUGGLER HARDLY	32	0.99
RIDGEMONT SUBMARINE	32	0.99
FORWARD TEMPLE	32	2.99
GRIT CLOCKWORK	32	0.99
SCALAWAG DUCK	32	4.99
HOBBIT ALIEN	31	0.99
TIMBERLAND SKY	31	0.99
GOODFELLAS SALUTE	31	4.99

Q3. Are there correlations between staff performance and customer satisfaction?

```
SELECT s.first_name, s.last_name,
   AVG(r.customer_satisfaction) AS avg_customer_satisfaction,
   SUM(p.amount) AS total_revenue,
   COUNT(r.rental_id) AS total_movies_sold
FROM staff AS s
JOIN rental AS r ON s.staff_id = r.staff_id
JOIN payment AS p ON r.rental_id = p.rental_id
GROUP BY s.staff_id, s.first_name, s.last_name
ORDER BY
   total_revenue DESC;
```

first_name	last_name	avg_customer_satisfaction	total_revenue	total_movies_sold
Jon	Stephens	3.0021	33881.94	8004
Mike	Hillyer	2.992	33524.62	8040

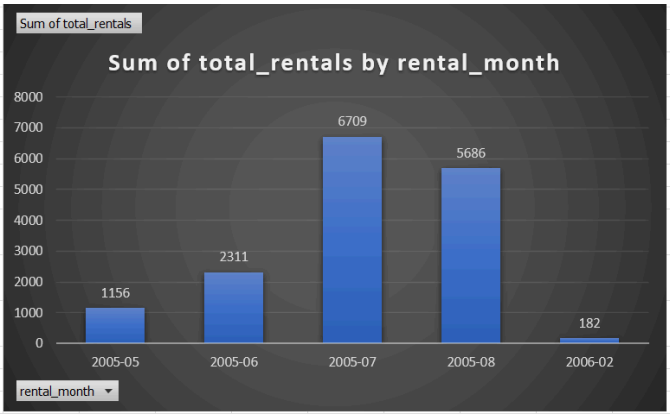
Here we see that the staff with the highest customer satisfaction is "Jon Stephens" as his Total revenue is also higher than the only other staff member even though he sold comparatively less movies than Mike

Q4. Are there seasonal trends in customer behavior across different locations?

```
SELECT c.country,
   DATE_FORMAT(STR_TO_DATE(r.rental_date, '%d-%m-%Y %H:%i:%s'), '%Y-%m') AS
rental_month,
   COUNT(r.rental_id) AS total_rentals
FROM rental AS r
JOIN customer AS cust ON r.customer_id = cust.customer_id
JOIN address AS a ON cust.address_id = a.address_id
JOIN city AS ci ON a.city_id = ci.city_id
JOIN country AS c ON ci.country_id = c.country_id
WHERE r.rental_date IS NOT NULL
GROUP BY c.country, rental_month
ORDER BY c.country, rental_month;
```

country	rental_month	total_rentals
Afghanistan	2005-06	3
Afghanistan	2005-07	8
Afghanistan	2005-08	7
Algeria	2005-05	8
Algeria	2005-06	12
Algeria	2005-07	40
Algeria	2005-08	28
Algeria	2006-02	2
American Samoa	2005-06	5
American Samoa	2005-07	11
American		

rental_month	Sum of total_rentals
2005-05	1156
2005-06	2311
2005-07	6709
2005-08	5686
2006-02	182



Q5.Are certain language films more popular among specific customer segments?

```
SELECT language_id, COUNT(*) AS film_count
FROM film
GROUP BY language_id;
```

language_id	film_count
1	1000

Based on the dataset provided, we observe that all the films are in Language English as language_id =1.

Q6. How does customer loyalty impact sales revenue over time?

```
WITH CustomerTotalSpent AS (
    SELECT
        p.customer_id,
        SUM(p.amount) AS total_spent
    FROM
        payment AS p
    GROUP BY
        p.customer_id
),
RankedCustomers AS (
    SELECT
        cts.customer_id,
        cts.total_spent,
        ROW_NUMBER() OVER (ORDER BY cts.total_spent DESC) AS
spend_rank,
        COUNT(*) OVER () AS total_customers
    FROM
        CustomerTotalSpent AS cts
),
LoyaltySegments AS (
    SELECT
        rc.customer_id,
        rc.total_spent,
        CASE
            WHEN rc.spend_rank <= rc.total_customers * 0.25 THEN
'Loyal'
            ELSE 'Non-Loyal'
        END AS loyalty_segment
    FROM
        RankedCustomers AS rc
)
SELECT
    DATE_FORMAT(p.payment_date, '%Y-%m') AS
payment_year_month,
    ls.loyalty_segment,
    SUM(p.amount) AS total_sales_revenue,
    COUNT(DISTINCT p.customer_id) AS no_of_customers
FROM
    payment AS p
JOIN
    LoyaltySegments AS ls ON p.customer_id = ls.customer_id
GROUP BY
    payment_year_month,
    ls.loyalty_segment
ORDER BY
    payment_year_month,
```

We Calculate the total money spent by each customer

Rank customers by their total spending in descending order and get the total count of distinct customers.

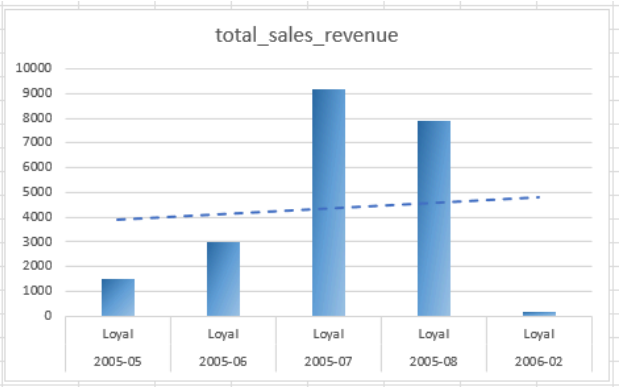
We Categorize customers into loyalty segments based on their rank.'Loyal' customers are defined as those whose rank falls within the top 25% of all customers.

The loyalty_segment is determined by checking if a customers spend_rank is within the top 25% (*made-up threshold) of total_customers (i.e., spend_rank <=

Joining payment data with loyalty segments and aggregate total sales revenue by month/year and loyalty segment.

payment_year_mo	loyalty_segm	total_sales_reve	no_of_custo
2005-05	Loyal	1497.51	136
2005-05	Non-Loyal	3325.93	384
2005-06	Loyal	2998.34	148
2005-06	Non-Loyal	6631.55	442
2005-07	Loyal	9198.63	149
2005-07	Non-Loyal	19170.28	450
2005-08	Loyal	7897.78	149
2005-08	Non-Loyal	16172.36	450
2006-02	Loyal	155.48	43
2006-02	Non-Loyal	358.7	115

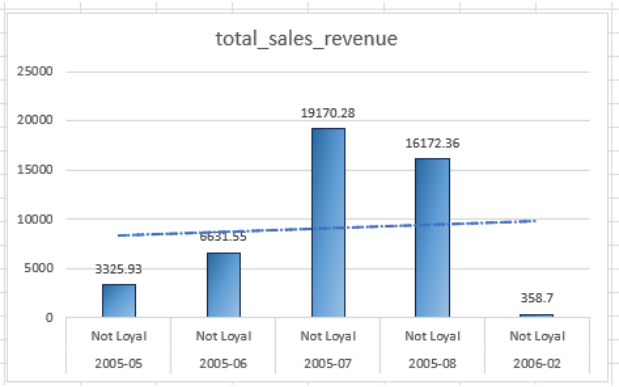
payment_year_month	loyalty_segment	total_sales_revenue
2005-05	Loyal	1497.51
2005-06	Loyal	2998.34
2005-07	Loyal	9198.63
2005-08	Loyal	7897.78
2006-02	Loyal	155.48



This segment, though smaller in number, consistently drives a significant portion of total sales revenue (around 30-33%).

They represent a highly valuable and stable revenue stream for the business, indicating strong engagement and repeat purchases.

payment_year_month	loyalty_segment	total_sales_revenue
2005-05	Not Loyal	3325.93
2005-06	Not Loyal	6631.55
2005-07	Not Loyal	19170.28
2005-08	Not Loyal	16172.36
2006-02	Not Loyal	358.7



Comprising the majority of the customer base, this segment individually contributes less to total sales.

They represent a significant opportunity for growth if strategies can successfully convert a portion into more loyal and higher-spending customers.

Q7. Are certain film categories more popular in specific locations?

```
WITH Top10Countries AS (
  SELECT
    co.country_id,
    co.country AS country_name,
    SUM(p.amount) AS total_country_revenue
  FROM payment AS p
  JOIN customer AS cu ON p.customer_id = cu.customer_id
  JOIN address AS a ON cu.address_id = a.address_id
  JOIN city AS ci ON a.city_id = ci.city_id
  JOIN country AS co ON ci.country_id = co.country_id
  GROUP BY co.country_id, co.country
  ORDER BY total_country_revenue DESC
  LIMIT 10
),
CategoryRevenuePerCountry AS (
  SELECT
    tpc.country_name,
    cat.name AS category_name,
    SUM(p.amount) AS category_total_revenue
  FROM payment AS p
  JOIN rental AS r ON p.rental_id = r.rental_id
  JOIN inventory AS i ON r.inventory_id = i.inventory_id
  JOIN film_category AS fc ON i.film_id = fc.film_id
  JOIN category AS cat ON fc.category_id = cat.category_id
  JOIN customer AS cu ON p.customer_id = cu.customer_id
  JOIN address AS a ON cu.address_id = a.address_id
  JOIN city AS ci ON a.city_id = ci.city_id
  JOIN Top10Countries AS tpc ON ci.country_id = tpc.country_id -- Filter for
top 10 countries
  GROUP BY tpc.country_name, cat.name
),
RankedCategories AS (
  SELECT
    crpc.country_name,
    crpc.category_name,
    crpc.category_total_revenue,
    ROW_NUMBER() OVER (PARTITION BY crpc.country_name ORDER
BY crpc.category_total_revenue DESC) AS rn
  FROM CategoryRevenuePerCountry AS crpc
)
SELECT rc.country_name,
  rc.category_name,
  rc.category_total_revenue AS amount
FROM RankedCategories AS rc
WHERE rc.rn <= 3
ORDER BY rc.country_name,
  rc.category_total_revenue DESC;
```

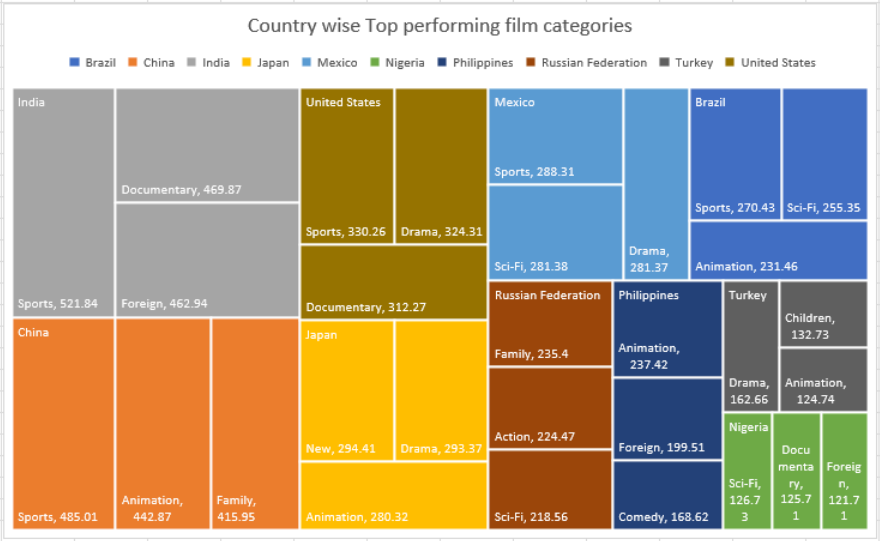
this CTE identifies the top 10 countries by the total amount of payments received from customers in those countries.

gathers all relevant payment data, associates it with both the film category and the customer's country, filters it to only include the previously identified top 10 countries, and then aggregates the total revenue for each film category within those specific countries.

Rank film categories by revenue within each country

Select the Top 3 film categories for each of the top 10 countries

country_name	category_name	amount
Brazil	Sports	270.43
Brazil	Sci-Fi	255.35
Brazil	Animation	231.46
China	Sports	485.01
China	Animation	442.87
China	Family	415.95
India	Sports	521.84
India	Documentary	469.87
India	Foreign	462.94
Japan	New	294.41
Japan	Drama	293.37
Japan	Animation	280.32
Mexico	Sports	288.31
Mexico	Sci-Fi	281.38
Mexico	Drama	281.37
Nigeria	Sci-Fi	126.73
Nigeria	Documentary	125.71
Nigeria	Foreign	121.71
Philippines	Animation	237.42
Philippines	Foreign	199.51
Philippines	Comedy	168.62
Russian Federation	Family	235.4
Russian Federation	Action	224.47
Russian Federation	Sci-Fi	218.56
Turkey	Drama	162.66
Turkey	Children	132.73
Turkey	Animation	124.74
United States	Sports	330.26
United States	Drama	324.31
United States	Documentary	312.27



Q8.How does the availability and knowledge of staff affect customer ratings?

```
SELECT s.first_name,
   s.last_name,
   COUNT(r.rental_id) AS total_rentals_processed,
   AVG(r.customer_satisfaction) AS average_customer_satisfaction
FROM staff AS s
JOIN rental AS r ON s.staff_id = r.staff_id
GROUP BY s.staff_id, s.first_name, s.last_name
ORDER BY total_rentals_processed DESC;
```

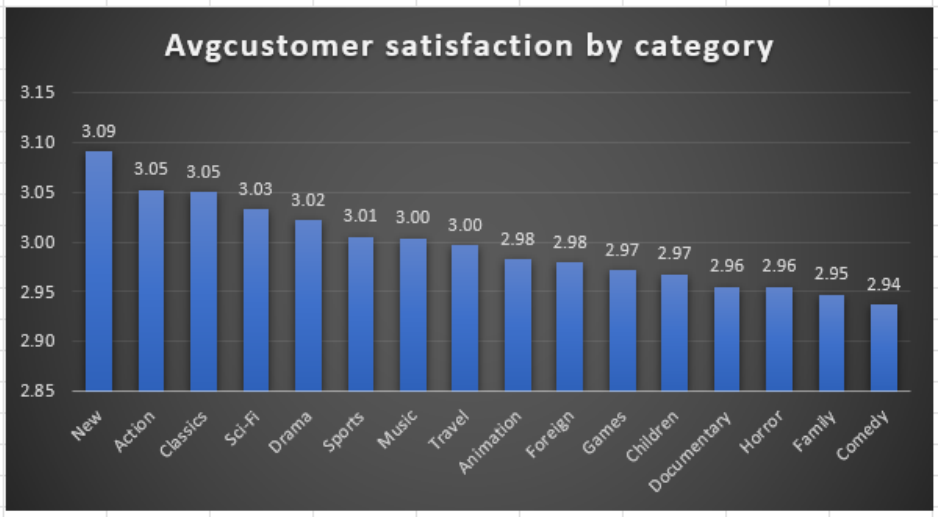
first_name	last_name	total_rentals_processe	Avg_customer_satisfactio
Mike	Hillyer	8040	2.992
Jon	Stephens	8004	3.0021

Based on the dataset provided, it's not possible to directly measure "staff knowledge" or "staff availability" as there are no columns in the staff or rental tables that contain the information.However, we can analyze the relationship between a staff member's rental volume and the average customer satisfaction ratings they receive. This can give us some insight into how active staff members are rated by customers.

Q9. How does the proximity of stores to customers impact rental frequency?

```
SELECT c.name AS category_name,
   AVG(r.customer_satisfaction) AS avg_customer_satisfaction
FROM category AS c
JOIN film_category AS fc ON c.category_id = fc.category_id
JOIN film AS f ON fc.film_id = f.film_id
JOIN inventory AS i ON f.film_id = i.film_id
JOIN rental AS r ON i.inventory_id = r.inventory_id
GROUP BY c.name
ORDER BY avg_customer_satisfaction DESC
```

category_name	avg_customer_satisfaction
New	3.09
Action	3.05
Classics	3.05
Sci-Fi	3.03
Drama	3.02
Sports	3.01
Music	3.00
Travel	3.00
Animation	2.98
Foreign	2.98
Games	2.97
Children	2.97
Documentary	2.96
Horror	2.96
Family	2.95
Comedy	2.94



Q10. How are the customer satisfaction ratings of top and least spending customers?

Top Spenders

```
WITH CustomerSpending AS (
   SELECT
      c.customer_id,
      c.first_name,
      c.last_name,
      SUM(p.amount) AS total_spending
   FROM customer AS c
   JOIN payment AS p ON c.customer_id = p.customer_id
   GROUP BY
      c.customer_id, c.first_name, c.last_name
   ORDER BY total_spending DESC
)
SELECT CONCAT(cs.first_name, ' ', cs.last_name) AS customer_name,
   COUNT(r.rental_id) AS total_rentals,
   cs.total_spending,
   AVG(r.customer_satisfaction) AS average_satisfaction
FROM CustomerSpending AS cs
JOIN rental AS r ON cs.customer_id = r.customer_id
GROUP BY
   cs.customer_id, cs.first_name, cs.last_name, cs.total_spending
ORDER BY cs.total_spending DESC
LIMIT 10;
```

customer_name	total_rentals	total_spending	average_satis
KARL SEAL	45	221.55	2.69
ELEANOR HUNT	46	216.54	3.00
CLARA SHAW	42	195.58	2.76
MARION SNYDER	39	194.61	2.85
RHONDA KENNEDY	39	194.61	2.82
TOMMY COLLAZO	38	186.62	2.79
WESLEY BULL	40	177.6	2.98
TIM CARY	39	175.61	3.49
MARCIA DEAN	42	175.58	2.86
ANA BRADLEY	34	174.66	3.06

Least Spenders

```
WITH CustomerSpending AS (
  SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    SUM(p.amount) AS total_spending
  FROM customer AS c
  JOIN payment AS p ON c.customer_id = p.customer_id
  GROUP BY
    c.customer_id, c.first_name, c.last_name
  ORDER BY total_spending ASC
)
SELECT CONCAT(cs.first_name, ' ', cs.last_name) AS customer_name,
  COUNT(r.rental_id) AS total_rentals, cs.total_spending,
  AVG(r.customer_satisfaction) AS average_satisfaction
FROM CustomerSpending AS cs
JOIN rental AS r ON cs.customer_id = r.customer_id
GROUP BY
  cs.customer_id, customer_name, cs.total_spending
ORDER BY cs.total_spending ASC
LIMIT 10;
```

customer_name	total_rentals	total_spending	average_satis
CAROLINE BOWMAN	15	50.85	3.33
LEONA OBRIEN	14	50.86	2.29
BRIAN WYMAN	12	52.88	2.58
JOHNNY TURPIN	19	57.81	2.68
ANNIE RUSSELL	18	58.82	2.94
KATHERINE RIVERA	14	58.86	3.00
TIFFANY JORDAN	14	59.86	3.71
ANITA MORALES	15	62.85	2.67
MATTIE HOFFMAN	22	64.78	2.77
KIRK STCLAIR	19	64.81	3.37

>> Despite the wide difference in spending, both groups have similar average satisfaction (~2.9).
>> High spenders rent more often, but not necessarily spend more per rental.
>> Both groups include customers with satisfaction scores below 3.0, which is subpar.
>>Mattie Hoffman and Kirk Stclair are low spenders but have high rentals and above-average satisfaction.These are loyal customers who may respond well to upselling or loyalty programs.
>> Customer Satisfaction leaders exist in both segments
    Tiffany Jordan (low spender) → 3.71 satisfaction
    Tim Cary (high spender) → 3.49 satisfaction

Q11. What are the demographics and preferences of the highest-spending customers?

```
WITH TopSpendingCustomers AS (
    SELECT c.customer_id, c.first_name, c.last_name,
        SUM(p.amount) AS total_spent,
        a.address, a.district,
        ci.city, co.country
    FROM payment AS p
    JOIN customer AS c ON p.customer_id = c.customer_id
    JOIN address AS a ON c.address_id = a.address_id
    JOIN city AS ci ON a.city_id = ci.city_id
    JOIN country AS co ON ci.country_id = co.country_id
    GROUP BY c.customer_id, c.first_name, c.last_name,
        a.address, a.district,
        ci.city, co.country
    ORDER BY total_spent DESC
    LIMIT 10
),
RankedCategoryPreferences AS (
    SELECT tpswd.customer_id, cat.name AS category_name,
        COUNT(*) AS rental_count,
        ROW_NUMBER() OVER (PARTITION BY tpswd.customer_id ORDER BY COUNT(*) DESC) AS rn
    FROM TopSpendingCustomers AS tpswd
    JOIN rentat AS r ON tpswd.customer_id = r.customer_id
    JOIN inventory AS i ON r.inventory_id = i.inventory_id
    JOIN film_category AS fc ON i.film_id = fc.film_id
    JOIN category AS cat ON fc.category_id = cat.category_id
    GROUP BY tpswd.customer_id, cat.name
)
SELECT CONCAT(tpswd.first_name, ' ', tpswd.last_name) AS Customer_Name, tpswd.total_spent,
    tpswd.address, tpswd.district, tpswd.city, tpswd.country,
    GROUP_CONCAT(rcp.category_name ORDER BY rcp.rn ASC SEPARATOR ', ') AS preferred_categories
FROM TopSpendingCustomers AS tpswd
LEFT JOIN RankedCategoryPreferences AS rcp ON tpswd.customer_id = rcp.customer_id AND rcp.rn <= 3
GROUP BY tpswd.customer_id, tpswd.first_name, tpswd.last_name,
    tpswd.total_spent, tpswd.address, tpswd.district,
    tpswd.city, tpswd.country
ORDER BY tpswd.total_spent DESC;
```

>> Combines calculating total spending per customer and identifying the top 10, along with gathering their demographic details.

>> Links rentals from top customers to film categories, counts rentals per category,and then ranks them directly within this CTE.

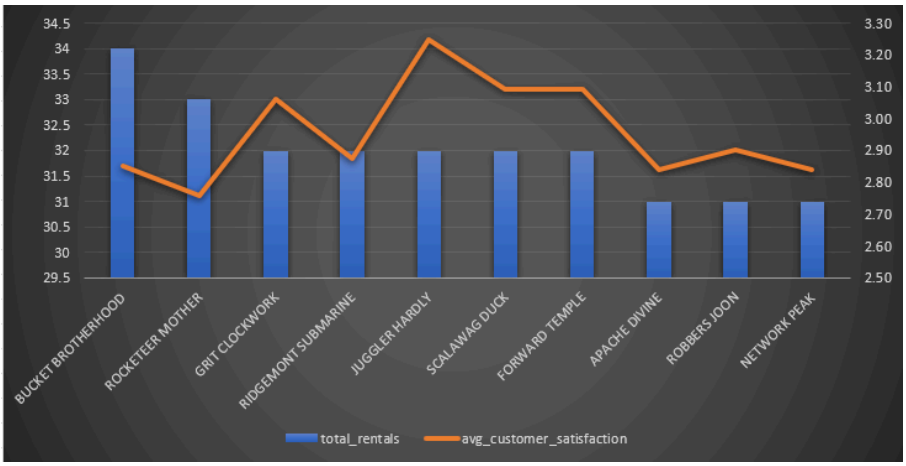
>> Select the requested columns, including concatenated preferred categories

Customer_name	total_spent	address	district	city	country	preferred_categories
KARL SEAL	221.55	1427 Tabuk Place	Florida	Cape Coral	United States	Animation, Family, Horror
ELEANOR HUNT	216.54	1952 Pune Lane	Saint-Denis	Saint-Denis	Réunion	Sci-Fi, Family, Travel
CLARA SHAW	195.58	1027 Songkhla	Minsk	Molodetšno	Belarus	Drama, New, Action
RHONDA KENNEDY	194.61	1749 Daxian Place	Gelderland	Apeldoorn	Netherlands	Games, Sports, Comedy
MARION SNYDER	194.61	1891 Rizhao Boulevard	São Paulo	Santa Brbara	Brazil	Travel, Drama, Foreign
TOMMY COLLAZO	186.62	76 Kermanshah Manor	Esfahan	Qomsheh	Iran	Family, Comedy, Horror
WESLEY BULL	177.6	1469 Plock Lane	Galicia	Ourense (Orense)	Spain	Games, Foreign, Family
TIM CARY	175.61	1257 Guadalajara Street	Karnataka	Bijapur	India	Animation, Sports, Family
MARCIA DEAN	175.58	1479 Rustenburg Boulevard	Southern Tagalog	Tanza	Philippines	Foreign, Games, Family
ANA BRADLEY	174.66	682 Garden Grove Place	Tennessee	Memphis	United States	Family, Horror, Foreign

Q12. How does the availability of inventory impact customer satisfaction and repeat business?

```
SELECT
  f.film_id,
  f.title,
  COUNT(r.rental_id) AS total_rentals,
  AVG(r.customer_satisfaction) AS avg_customer_satisfaction
FROM
  film AS f
JOIN
  inventory AS i ON f.film_id = i.film_id
JOIN
  rental AS r ON i.inventory_id = r.inventory_id
GROUP BY
  f.film_id, f.title
ORDER BY
  total_rentals DESC;
```

film_id	title	total_rentals	avg_customer_satisfaction
103	BUCKET BROTHERHOOD	34	2.85
738	ROCKETEER MOTHER	33	2.76
382	GRIT CLOCKWORK	32	3.06
730	RIDGEMONT SUBMARINE	32	2.88
489	JUGGLER HARDLY	32	3.25
767	SCALAWAG DUCK	32	3.09
331	FORWARD TEMPLE	32	3.09
31	APACHE DIVINE	31	2.84
735	ROBBERS JOON	31	2.90
621	NETWORK PEAK	31	2.84
973	WIFE TURN	31	2.81
418	HOBBIT ALIEN	31	3.35
753	RUSH GOODFELLAS	31	3.29
891	TIMBERLAND SKY	31	3.48
369	GOODFELLAS SALUTE	31	2.65
1000	ZORRO ARK	31	2.55
450	IDOLS SNATCHERS	30	3.10
979	WITCHES PANIC	30	3.13
403	HARRY IDAHO	30	3.27
748	RUGRATS SHAKESPEARE	30	3.33
869	SUSPECTS QUILLS	30	3.07
702	PULP BEVERLY	30	3.00
239	DOGMA FAMILY	30	3.03
563	MASSACRE USUAL	30	3.03



Based on the dataset provided, it's not possible to directly measure the impact of inventory on customer satisfaction and repeat business, as this is a conceptual question that would require more than a single SQL query to definitively answer.

However, a query can be used to analyze the relationship between a film's rental frequency (a proxy for its availability) and its average customer satisfaction rating. The following SQL query allows you to examine this relationship:

Q13. What are the busiest hours or days for each store location, and how does it impact staffing requirements?

```
WITH BusiestTimes AS (
   SELECT
      s.store_id,st.staff_id,
      DAYOFWEEK(STR_TO_DATE(r.rental_date, '%d-%m-%Y %H:%i:%s')) AS
day_of_week,
      HOUR(STR_TO_DATE(r.rental_date, '%d-%m-%Y %H:%i:%s')) AS hour_of_day,
      COUNT(r.rental_id) AS total_rentals,
      SUM(p.amount) AS total_sales
   FROM  rental AS r
   JOIN staff AS st ON r.staff_id = st.staff_id
   JOIN store AS s ON st.store_id = s.store_id
   JOIN payment AS p ON r.rental_id = p.rental_id
   GROUP BY
      s.store_id,st.staff_id, day_of_week, hour_of_day
)
SELECT
   store_id,staff_id,
   CASE
      WHEN day_of_week = 1 THEN 'Sunday'
      WHEN day_of_week = 2 THEN 'Monday'
      WHEN day_of_week = 3 THEN 'Tuesday'
      WHEN day_of_week = 4 THEN 'Wednesday'
      WHEN day_of_week = 5 THEN 'Thursday'
      WHEN day_of_week = 6 THEN 'Friday'
      WHEN day_of_week = 7 THEN 'Saturday'
   END AS day,
   hour_of_day,
   total_rentals,
   total_sales
FROM  BusiestTimes
ORDER BY store_id, total_rentals DESC;
```

store_id	staff_id	day	hour_of_day	total_rentals	total_sales
1	1	Tuesday	15	132	399.7
1	1	Thursday	7	72	278.28
1	1	Thursday	10	69	334.31
1	1	Wednesday	21	63	271.37
1	1	Monday	4	63	265.37
1	1	Monday	15	61	250.39
1	1	Tuesday	18	61	226.39
1	1	Tuesday	0	60	223.4
1	1	Sunday	3	60	230.4
1	1	Sunday	20	59	252.41
1	1	Saturday	8	59	255.41
1	1	Tuesday	4	57	251.43
1	1	Wednesday	16	57	229.43
1	1	Saturday	0	57	248.43
1	1	Sunday	9	56	236.44
1	1	Friday	6	56	249.44
1	1	Sunday	11	56	238.44
1	1	Wednesday	3	56	253.44
1	1	Thursday	14	55	242.45
1	1	Sunday	10	55	254.45
1	1	Friday	16	55	227.45
1	1	Monday	1	55	221.45
1	1	Saturday	10	54	206.46
1	1	Monday	0	54	240.46

After this step we make a pivot table from the output table and compare both the stores on the basis of Total_rentals by days and hours_of_day

We Observe:

Both stores have their peak day on Tuesday, indicating that staffing should be highest on this day to handle the increased customer traffic.
>> Store 1 is also very busy on Sunday and Wednesday, so these days would require a higher level of staffing than the slower days of Thursday and Friday.
>> Store 2 sees a significant increase in rentals on Friday and Sunday, in addition to Tuesday. Staffing should be adjusted accordingly to ensure adequate coverage on these three days.
Overall, both stores have their slowest days mid-week (Wednesday, Thursday, and Friday for Store 1; Wednesday and Thursday for Store 2), suggesting that staffing could be reduced on these days to optimize labor costs.

>> Store 1: Total Sales by Days
Busiest Day: Sunday, with total sales of $4,900.45.
Next Busiest Days: Tuesday ($4,855.72) and Monday ($4,848.54).

>> Store 2: Total Sales by Days
Busiest Day: Tuesday, with total sales of $5,361.65.
Next Busiest Days: Saturday ($5,116.43) and Friday ($4,939.22).

>> Store 1 experiences its highest sales on Sunday, followed closely by Tuesday and Monday. Staffing should be at its peak on these days to handle the increased business. Sales are lowest on Friday, suggesting that staffing levels could be reduced to optimize labor costs on that day.

>> Store 2 has a clear peak on Tuesday, which would require the highest level of staffing. Sales are also significantly high on Saturday and Friday, so staffing should be increased on those days as well. Monday and Wednesday are the slowest days for Store 2, indicating that staffing can be reduced on these two days.

The busiest days for each store are different. Store 1's peak is on Sunday, while Store 2's peak is on Tuesday. Staffing schedules should be tailored to these specific peak days for each location to ensure optimal customer service and operational efficiency.

Q14.How long does it take for a new customer to make their second rental, on average?

```
WITH RankedRentals AS (
    SELECT
        customer_id,
        rental_date,
        ROW_NUMBER() OVER (PARTITION BY customer_id ORDER BY
STR_TO_DATE(rental_date, '%d-%m-%Y %H:%i:%s')) AS rental_rank
    FROM
        rental
)
SELECT
    AVG(DATEDIFF(
        STR_TO_DATE(r2.rental_date, '%d-%m-%Y %H:%i:%s'),
        STR_TO_DATE(r1.rental_date, '%d-%m-%Y %H:%i:%s')
    )) AS average_days_to_second_rental
FROM
    RankedRentals AS r1
JOIN
    RankedRentals AS r2
    ON r1.customer_id = r2.customer_id
WHERE
    r1.rental_rank = 1 AND r2.rental_rank = 2;
```

average_days_to_second_rental
7.0284

Here we see that after acquiring a new customer it takes on average a week (7.02) days for the customer to get the second rental

Q15.How does the availability of films in different languages impact customer satisfaction and rental frequency?

```
SELECT
   l.name AS language,
   COUNT(r.rental_id) AS total_rentals,
   AVG(r.customer_satisfaction) AS avg_customer_satisfaction
FROM
   language AS l
JOIN
   film AS f ON l.language_id = f.language_id
JOIN
   inventory AS i ON f.film_id = i.film_id
JOIN
   rental AS r ON i.inventory_id = r.inventory_id
GROUP BY
   l.name
ORDER BY
   total_rentals DESC;
```

language	tota_rentals	avg_customer_satisfaction
English	16044	2.9971