1)The business wants to run email campaigns for customers in store id 2, for that it needs the email ids, first name as well as last name of the customers. Write a query to fetch this data.

Syntax

```
SELECT first_name,last_name,email from customer where store_id = 2;
```

2)**While doing the audit of the business, one of the financial analyst found out that some dvd's are being rented out for $0.99. The finance department needs the count of such movies whose rental rate is $0.99. Can you help them?**

Syntax

```
SELECT count(film_id) from film where rental_rate = 0.99;
```

3)The accounts department, is thinking of comping up with a different way of accounting for the business costs. To be able to do that they need to find out the number of movies rented at different rental price points. Write an sql query so that this crucial input can be provided to the accounts department.

syntax

```
SELECT rental_rate,count(film_id) AS TOTAL_NO_FILMS from film GROUP BY rental_rate ;
```

4)The marketing team wants to understand how number of movies is spread across movie ratings. Can you help them?

syntax

```
SELECT rating,count(film_id) AS TOTAL_NO_FILMS from film GROUP BY rating ;
```

4)The marketing team now also wants to know how ratings are distributed across stores. The team needs to know the distribution of ratings for each store in the dataset. Write a sql query to help solve this problem.

syntax

```
SELECT b.store_id, a.rating, COUNT(a.film_id) AS TOTAL_NO_FILMS

FROM film a

JOIN inventory b ON a.film_id = b.film_id

GROUP BY b.store_id,a.rating;
```

5)The digital marketing team is studying what other movie rental businesses are doing. One of the analysis that they want to do is study what are the kind of movies that are being rented out by competitors as well as the current company. Your job is to provide the team with details on film name, category each film belongs to and the language in which the film is.

syntax

```
select f.title,c.name,l.name
from film f
join  film_category fc ON fC.film_id = f.film_id
join category c ON fc.category_id = c.category_id
join language l ON l.language_id = f.language_id;
```

6)One of the questions that the business is interested in is about the popularity of the movies in the current inventory also the stores and customers who bring in more revenue. Help the business in finding out

i)the number of times each movie is rented out

syntax

```
select f.title,count(r.rental_id) as no_of_rented from film f
join inventory i on f.film_id = i.film_id
join rental r on  r.inventory_id = i.inventory_id group by f.title order by no_of_rented DESC;
```

ii)Revenue pre movie

syntax

```
SELECT f.film_id, f.title, sum(p.amount) AS total_revenue
FROM film f
JOIN inventory i ON f.film_id = i.film_id
JOIN rental r ON i.inventory_id = r.inventory_id
JOIN payment p ON p.rental_id = r.rental_id
GROUP BY f.film_id,f.title
ORDER BY total_revenue DESC;
```

iii)**Most revenue earned by a store**

syntax

```
SELECT s.store_id, SUM(p.amount) AS "Total Spending"

FROM store s

JOIN inventory i ON i.store_id = s.store_id

JOIN rental r ON r.inventory_id = i.inventory_id

JOIN payment p ON p.rental_id = r.rental_id

GROUP BY 1

ORDER BY 2 DESC;
```

iv)Which customer has spent the most

syntax

```
select c.id,c.name,sum(p.amount)as Total_sales1 FROM customer_list c join payment p on p.customer_id = c.id group by c.id,c.name order by Total_sales1 desc LIMIT 1;
```

7)One important aspect of business is loyalty and reward programs for customers as well as internal stakeholder. The business is currently looking at launching some strategic initiatives for which they need to know the following information

i)Last Rental Date of every customer

syntax

```
select c.id,c.name,max(r.rental_date) as last_rental_date FROM customer_list c join rental r on r.customer_id = c.id group by c.id,c.name order by id;
```

ii)Total Revenue Per Month

syntax

```
select strftime("%m",rental_date) as month,sum(amount) as "total revenue"  from payment p join rental r on p.customer_id = r.customer_id group by month
```

iii)Number of distinct Renters per month

syntax

```
SELECT strftime("%m",rental_date) AS "Month",

            COUNT(DISTINCT(rental_id)) AS "Total Rentals",

            COUNT(DISTINCT(customer_id)) AS "Number Of Unique Renter",

FROM rental

GROUP BY 1;
```

iv)Number of Distinct Film Rented Each Month

syntax

```
SELECT i.film_id, f.title, strftime("%m",r.rental_date) AS "Month", COUNT(i.film_id) AS "Total Number Of Rental Times"

FROM rental r

JOIN inventory i ON r.inventory_id = i.inventory_id

JOIN film f ON f.film_id = i.film_id

GROUP BY i.film_id, strftime("%m",r.rental_date)

ORDER BY 1, 2, 3;
```

v)Number of Rentals in Comedy , Sports and Family

syntax

```
SELECT c.name, COUNT(c.name) AS "Number of Rentals"

FROM film f

JOIN film_category fc ON fc.film_id = f.film_id

JOIN category c ON c.category_id = fc.category_id

JOIN inventory i ON f.film_id = i.film_id

JOIN rental r ON r.inventory_id = i.inventory_id

WHERE c.name IN ("Comedy", "Sports", "Family")

GROUP BY 1;
```

vi)Users who have been rented at least 3 times

syntax

```
SELECT  c.id,c.name,COUNT(c.id) AS "Total Rentals"
FROM customer_list c
JOIN rental r ON c.id = r.customer_id
GROUP BY c.id
HAVING "Total Rentals" >= 3
ORDER BY 1;
```

vii)How much revenue has one single store made over PG13 and R rated films

syntax

```
SELECT s.store_id, f.rating, SUM(p.amount) AS "Total Revenue"
FROM store s
JOIN inventory i ON i.store_id = s.store_id
JOIN rental r ON r.inventory_id = i.inventory_id
JOIN payment p ON p.rental_id = r.rental_id
JOIN film f ON f.film_id = i.film_id
WHERE f.rating IN ("PG-13", "R")
GROUP BY 1,2;
```