**Group No 13**

**Name:**

**Dhruv Patel(202301095)**

**Prasham Shah(202301252)**

**Manthan Gajera(202301488)**

**SQL Queries**

1. **Top Performing Restaurants by Average Rating and Order Count**

* Identifies restaurants with high average ratings (from reviews) and significant order volumes.
* Ranks by average rating and order count, filtering for restaurants with at least 5 reviews.
* Query: Joins Restaurant, Order, and Review, using HAVING COUNT(ReviewID) >= 5.

SELECT r.RestaurantID, r.Name, r.AverageRating, COUNT(o.OrderID) AS TotalOrders

FROM Restaurant r

JOIN Orders o ON r.RestaurantID = o.RestaurantID

GROUP BY r.RestaurantID, r.Name, r.AverageRating

ORDER BY r.AverageRating DESC, TotalOrders DESC

LIMIT 10;

1. **Monthly Revenue Breakdown for Each Restaurant**

* Calculates the monthly gross and net revenue for each restaurant from completed payments.
* Groups results by restaurant and month, showing transaction volume and earnings.
* Query: Uses Payment, Order, and Restaurant tables, with PaymentStatus = 'Completed'.

SELECT

r.RestaurantID,

r.Name AS RestaurantName,

TO\_CHAR(p.PaymentDate, 'YYYY-MM') AS Month,

SUM(p.TotalAmount) AS GrossRevenue,

SUM(p.RestaurantEarnings) AS NetRevenue

FROM Payment p

JOIN Orders o ON p.OrderID = o.OrderID

JOIN Restaurant r ON o.RestaurantID = r.RestaurantID

WHERE p.PaymentStatus = 'Completed'

GROUP BY r.RestaurantID, r.Name, TO\_CHAR(p.PaymentDate, 'YYYY-MM')

ORDER BY Month DESC, r.RestaurantID;

1. **Delivery Agent Performance Overview**

* Summarizes delivery agent performance, including total orders delivered, earnings, and average ratings.
* Focuses on completed deliveries and payouts, sorted by earnings for top performers.
* Query: Uses DeliveryAgent, User, Order, PayoutHistory, and Review, with Status = 'Delivered'.

SELECT

da.DeliveryAgentID,

u.Name AS AgentName,

COUNT(DISTINCT o.OrderID) AS OrdersDelivered,

COALESCE(SUM(ph.TotalPayout), 0) AS TotalEarnings,

COALESCE(AVG(r.Rating), 0) AS AverageRating

FROM DeliveryAgent da

JOIN User u ON da.UserID = u.UserID

LEFT JOIN `Order` o ON da.DeliveryAgentID = o.DeliveryAgentID

AND o.Status = 'Delivered'

LEFT JOIN PayoutHistory ph ON o.OrderID = ph.OrderID

AND ph.PaymentStatus = 'Completed'

LEFT JOIN Review r ON o.OrderID = r.OrderID

GROUP BY da.DeliveryAgentID, u.Name

ORDER BY TotalEarnings DESC;

1. **Customers Who Use the Most Reward Coins**

* Lists the top 10 customers by total reward coins used in successful transactions.
* Displays customer ID, name, and coins used, ordered by highest coin usage.
* Query: Joins RewardCoinUsage, Customer, User, and Payment, filtering for PaymentStatus = 'Completed'.

SELECT

c.CustomerID,

u.Name AS CustomerName,

SUM(rcu.CoinsUsed) AS TotalCoinsUsed

FROM RewardCoinUsage rcu

JOIN Customer c ON rcu.CustomerID = c.CustomerID

JOIN Users u ON c.UserID = u.UserID

JOIN Payment p ON rcu.TransactionID = p.TransactionID

AND p.PaymentStatus = 'Completed'

GROUP BY c.CustomerID, u.Name

ORDER BY TotalCoinsUsed DESC

LIMIT 10;

1. **Review Summary: Top Reviewed Restaurants**

* Shows the top 10 restaurants with 1+ reviews, ranked by average rating and review count.
* Calculates ratings directly from reviews for accuracy, excluding low-review restaurants.
* Query: Joins Restaurant, Order, and Review, with HAVING TotalReviews >= 1.

SELECT

r.RestaurantID,

r.Name AS RestaurantName,

COUNT(rv.ReviewID) AS TotalReviews,

ROUND(AVG(rv.Rating), 2) AS AvgRating

FROM Restaurant r

JOIN Orders o ON r.RestaurantID = o.RestaurantID

JOIN Review rv ON o.OrderID = rv.OrderID

GROUP BY r.RestaurantID, r.Name

HAVING COUNT(rv.ReviewID) >= 1

ORDER BY AvgRating DESC, TotalReviews DESC

LIMIT 10;

1. **Platform-Wide Monthly Profitability (Commission)**

* Calculates monthly platform commission revenue from completed payments.
* Groups by month, showing transaction counts and total commissions earned.
* Query: Uses Payment, filtering for PaymentStatus = 'Completed' and summing PlatformCommission.

SELECT

TO\_CHAR(PaymentDate, 'YYYY-MM') AS Month,

COUNT(TransactionID) AS TotalPayments,

ROUND(SUM(PlatformCommission), 2) AS TotalCommission

FROM Payment

WHERE PaymentStatus = 'Completed'

GROUP BY Month

ORDER BY Month DESC;

1. **Orders with High Reward Coin Utilization**

* Identifies orders with >100 reward coins used in completed, delivered transactions.
* Includes order details, customer name, coins used, and coin value percentage.
* Query: Joins RewardCoinUsage, Payment, Order, Customer, and User, with status filters.

SELECT

o.OrderID,

u.Name AS CustomerName,

rc.CoinsUsed,

rc.CoinsValue,

p.TotalAmount,

ROUND((rc.CoinsValue / p.TotalAmount \* 100), 2) AS CoinValuePercentage

FROM RewardCoinUsage rc

JOIN Payment p ON rc.TransactionID = p.TransactionID

JOIN Orders o ON p.OrderID = o.OrderID

JOIN Customer c ON rc.CustomerID = c.CustomerID

JOIN Users u ON c.UserID = u.UserID

WHERE

rc.CoinsUsed > 25

AND p.PaymentStatus = 'Completed'

AND o.Status = 'Delivered'

ORDER BY rc.CoinsUsed DESC

LIMIT 50;

1. **Most Popular Menu Items Across the Platform**

* Identifies the top 10 menu items by units ordered, limited to delivered orders.
* Groups by unique MenuID, item, and restaurant for precise, reliable results.
* Query: Joins OrderDetails, Menu, Restaurant, and Order, with Status = 'Delivered'.

SELECT

m.MenuID,

m.ItemName,

r.Name AS RestaurantName,

SUM(od.Quantity) AS TotalOrdered

FROM OrderDetails od

JOIN Menu m ON od.MenuID = m.MenuID

JOIN Restaurant r ON m.RestaurantID = r.RestaurantID

JOIN Orders o ON od.OrderID = o.OrderID

WHERE o.Status = 'Delivered'

GROUP BY m.MenuID, m.ItemName, r.Name

ORDER BY TotalOrdered DESC

LIMIT 10;

1. **Customer's Complete Order History With Ratings**

* Shows completed, delivered orders with customer names, payment amounts, and review details.
* Ensures accurate data with status filters and caps output for manageability.
* Query: Joins Order, Customer, User, Payment, and Review, with Delivered and Completed filters.

SELECT

u.Name AS CustomerName,

o.OrderID,

o.OrderDate,

p.TotalAmount,

r.Rating,

r.Comment

FROM `Order` o

JOIN Customer c ON o.CustomerID = c.CustomerID

JOIN User u ON c.UserID = u.UserID

JOIN Payment p ON o.OrderID = p.OrderID

LEFT JOIN Review r ON o.OrderID = r.OrderID

WHERE

p.PaymentStatus = 'Completed'

AND o.Status = 'Delivered'

ORDER BY

o.OrderDate DESC

LIMIT 1000;

1. **Average Spending per Customer**

* Shows average spending per customer for delivered, completed orders, ranked by amount.
* Groups by unique CustomerID and name, limiting to top 100 for clarity.
* Query: Joins Order, Customer, User, and Payment, with status filters.

SELECT

c.CustomerID,

u.Name AS CustomerName,

ROUND(AVG(p.TotalAmount), 2) AS AvgSpending

FROM Orders o

JOIN Customer c ON o.CustomerID = c.CustomerID

JOIN Users u ON c.UserID = u.UserID

JOIN Payment p ON o.OrderID = p.OrderID

WHERE

p.PaymentStatus = 'Completed'

AND o.Status = 'Delivered'

GROUP BY

c.CustomerID, u.Name

ORDER BY

AvgSpending DESC

LIMIT 100;

1. **Top 5 Earning Restaurants**

* Retrieves the top 5 restaurants by total earnings from completed payments, including those with no orders.
* Joins Restaurant, Order, and Payment tables, summing RestaurantEarnings with nulls as 0, rounded to two decimals.
* Groups by RestaurantID and Name, sorted by earnings in descending order, limited to 5 results.

SELECT

r.RestaurantID,

r.Name,

ROUND(SUM(COALESCE(p.RestaurantEarnings, 0)), 2) AS TotalEarnings

FROM Restaurant r

LEFT JOIN Orders o ON o.RestaurantID = r.RestaurantID

LEFT JOIN Payment p ON p.OrderID = o.OrderID

WHERE

p.PaymentStatus = 'Completed' OR p.PaymentStatus IS NULL

GROUP BY

r.RestaurantID, r.Name

ORDER BY

TotalEarnings DESC

LIMIT 5;

1. **Restaurants with the Most Menu Items**

* Identifies the top 10 restaurants with the highest number of active menu items.
* Joins Restaurant and Menu tables, counting available MenuIDs per restaurant.
* Groups by RestaurantID and Name, sorted by item count in descending order.

SELECT

r.RestaurantID,

r.Name,

COUNT(m.MenuID) AS MenuItems

FROM Restaurant r

LEFT JOIN Menu m ON r.RestaurantID = m.RestaurantID

WHERE m.Availability = TRUE

GROUP BY r.RestaurantID, r.Name

ORDER BY MenuItems DESC

LIMIT 10;

1. **Orders That Were Cancelled**

* Lists all cancelled orders with customer names, order dates, and rounded total amounts.
* Joins Order, Customer, and User tables, filtering for Status = 'Cancelled'.
* Orders results by OrderDate in descending order to show recent cancellations first.

SELECT

o.OrderID,

u.Name AS CustomerName,

o.OrderDate,

ROUND(o.TotalAmount, 2) AS TotalAmount

FROM "Order" o

JOIN Customer c ON o.CustomerID = c.CustomerID

JOIN User u ON c.UserID = u.UserID

WHERE o.Status = 'Cancelled'

ORDER BY o.OrderDate DESC;

1. **Breakdown of Payment Methods Used**

* Summarizes completed payment methods by transaction count and total amount paid.
* Filters Payment table for PaymentStatus = 'Completed', grouping by PaymentMethod.
* Orders results by rounded TotalAmount in descending order for priority ranking.

SELECT

p.PaymentMethod,

COUNT(\*) AS NumTransactions,

ROUND(SUM(p.AmountPaid), 2) AS TotalAmount

FROM Payment p

WHERE p.PaymentStatus = 'Completed'

GROUP BY p.PaymentMethod

ORDER BY TotalAmount DESC;

1. **Delivery Agent Payout Summary**

* Summarizes total payouts and delivery counts for each agent, including those with no payouts.
* Joins DeliveryAgent, User, and PayoutHistory, filtering for completed payouts, rounded to two decimals.
* Groups by DeliveryAgentID and Name, sorted by TotalPayout in descending order.

SELECT

da.DeliveryAgentID,

u.Name AS AgentName,

ROUND(SUM(COALESCE(ph.TotalPayout, 0)), 2) AS TotalPayout,

COUNT(ph.OrderID) AS DeliveriesMade

FROM DeliveryAgent da

JOIN Users u ON da.UserID = u.UserID

LEFT JOIN PayoutHistory ph ON da.DeliveryAgentID = ph.DeliveryAgentID

WHERE ph.PaymentStatus = 'Completed' OR ph.PaymentStatus IS NULL

GROUP BY da.DeliveryAgentID, u.Name

ORDER BY TotalPayout DESC;

1. **Customer Lifetime Value Estimation**

* Estimates customer lifetime value by summing paid amounts for delivered orders with completed payments.
* Joins Customer, User, Order, and Payment, aggregating AmountPaid (nulls as 0), rounded to two decimals.
* Groups by CustomerID and Name, sorted by LifetimeSpending in descending order, including customers with no orders.

SELECT

c.CustomerID,

u.Name AS CustomerName,

ROUND(SUM(COALESCE(p.AmountPaid, 0)), 2) AS LifetimeSpending

FROM Customer c

JOIN Users u ON c.UserID = u.UserID

LEFT JOIN Orders o ON o.CustomerID = c.CustomerID

LEFT JOIN Payment p ON o.OrderID = p.OrderID

WHERE (o.Status = 'Delivered' AND p.PaymentStatus = 'Completed') OR o.OrderID IS NULL

GROUP BY c.CustomerID, u.Name

ORDER BY LifetimeSpending DESC;

1. **Restaurant Admin View: All Pending Orders**

* Lists all pending orders (statuses: Placed, Preparing, Ready, On the way) for restaurant admin review.
* Joins Order, Restaurant, Customer, and User tables to show order details and customer names.
* Orders results by OrderDate in ascending order to prioritize older pending orders.

SELECT

o.OrderID,

r.Name AS RestaurantName,

o.Status,

o.OrderDate,

u.Name AS CustomerName

FROM "Order" o

JOIN Restaurant r ON o.RestaurantID = r.RestaurantID

JOIN Customer c ON o.CustomerID = c.CustomerID

JOIN User u ON c.UserID = u.UserID

WHERE o.Status IN ('Placed', 'Preparing', 'Ready', 'On the way')

ORDER BY o.OrderDate ASC;

1. **Commission Summary per Restaurant**

* Summarizes total platform commissions per restaurant from completed payments, including those with no orders.
* Joins Restaurant, Order, and Payment tables, aggregating PlatformCommission (nulls as 0), rounded to two decimals.
* Groups by RestaurantID and Name, sorted by TotalCommission in descending order.

SELECT

r.RestaurantID,

r.Name AS RestaurantName,

ROUND(SUM(COALESCE(p.PlatformCommission, 0)), 2) AS TotalCommission

FROM Restaurant r

LEFT JOIN Orders o ON o.RestaurantID = r.RestaurantID

LEFT JOIN Payment p ON p.OrderID = o.OrderID

WHERE p.PaymentStatus = 'Completed' OR p.PaymentStatus IS NULL

GROUP BY r.RestaurantID, r.Name

ORDER BY TotalCommission DESC;

1. **Total Orders Placed by Each Customer**

* Counts the total number of orders placed by each customer, including those with zero orders, regardless of order status.
* Joins Customer, User, and Order tables, grouping by CustomerID and Name.
* Orders results by TotalOrders in descending order to highlight the most active customers.

SELECT

c.CustomerID,

u.Name AS CustomerName,

COUNT(o.OrderID) AS TotalOrders

FROM Customer c

JOIN Users u ON c.UserID = u.UserID

LEFT JOIN Orders o ON o.CustomerID = c.CustomerID

GROUP BY c.CustomerID, u.Name

ORDER BY TotalOrders DESC;

1. **Pending Payments Summary**

* Lists all pending payments with transaction IDs, order IDs, customer names, rounded total amounts, and payment dates.
* Joins Payment, Order, Customer, and User tables, filtering for PaymentStatus = 'Pending'.
* Orders results by PaymentDate in descending order to show the most recent pending payments first.

SELECT

p.TransactionID,

o.OrderID,

u.Name AS CustomerName,

ROUND(p.TotalAmount, 2) AS TotalAmount,

p.PaymentDate

FROM Payment p

JOIN Orders o ON p.OrderID = o.OrderID

JOIN Customer c ON o.CustomerID = c.CustomerID

JOIN Users u ON c.UserID = u.UserID

WHERE p.PaymentStatus = 'Pending'

ORDER BY p.PaymentDate DESC;