SQL DATABASE FOR SEPHORA

INSY 661 – DATA & DIST SYS FOR ANALYTICS
CAPSTONE PROJECT

Ayda, Felix, Palvi, Raine, Ricardo, Scarlett

AGENDA

1. Business Overview
2. Business Functionality
3. Entity Relationship Diagram
4. Relational Schema
5. SQL Queries & KPI’s

BUSINESS OVERVIEW

BUSINESS OVERVIEW

- Leading global beauty retailer
- Founded in France in 1969
- Known for wide product offering
    - Cosmetics, Skincare, Haircare, & Fragrance
- Strong online presence.
- Growing e-commerce platform.

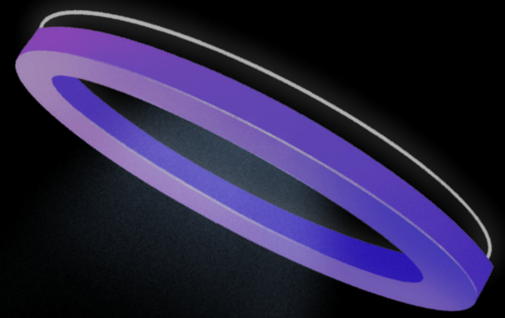
MISSION STATEMENT

The purpose of Sephora's online database system is to ***ensure accurate and up-to-date information,*** *offering a comprehensive data management* ***solution that facilitates detailed analysis and informed decision-making*** *in areas such as customer management, order processing, employee supervision, and product inventory.*

MISSION OBJECTIVES

- Maintain and update customer data: memberships, orders, payments, reviews, and preferences.
- Manage orders, linking each to a single invoice and shipment for seamless processing.
- Oversee product data, including reviews, promotions, and accurate inventory tracking.
- Coordinate service schedules and deliveries, ensuring proper task assignment and accurate shipment tracking.

BUSINESS PROCESS



1
Customer Management

2
Order Placement

3
Payment Processing

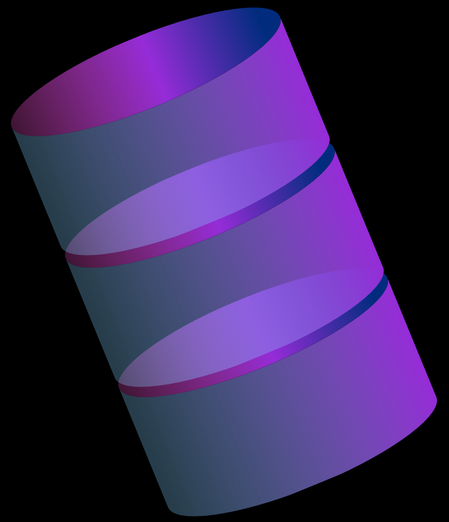
4
Product and Inventory Management

5
Order Fulfillment and Shipping

6
Refunds and Returns

7
Employee and Department Management

DATABASE GOALS



- Enhance operational efficiency and strategic decision-making.
- Real-time access for employees to manage deliveries, services, and inventory.
- Analyze sales to identify product performance.
- Insights into customer behaviour and preferences.
- Monitor effectiveness of marketing efforts.
    - Sales
    - Promotions

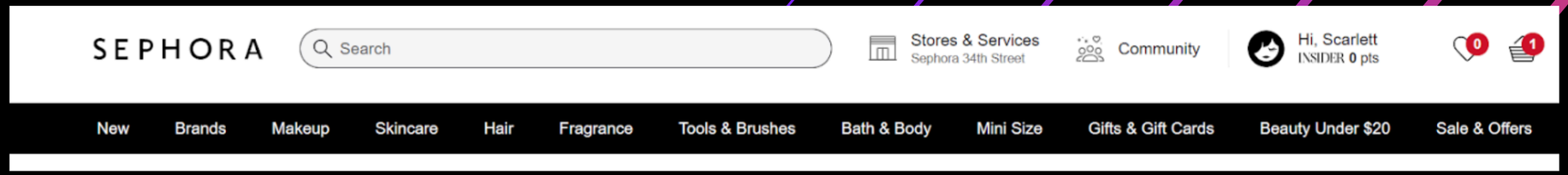
BUSINESS FUNCTIONALITY

MAIN USERS

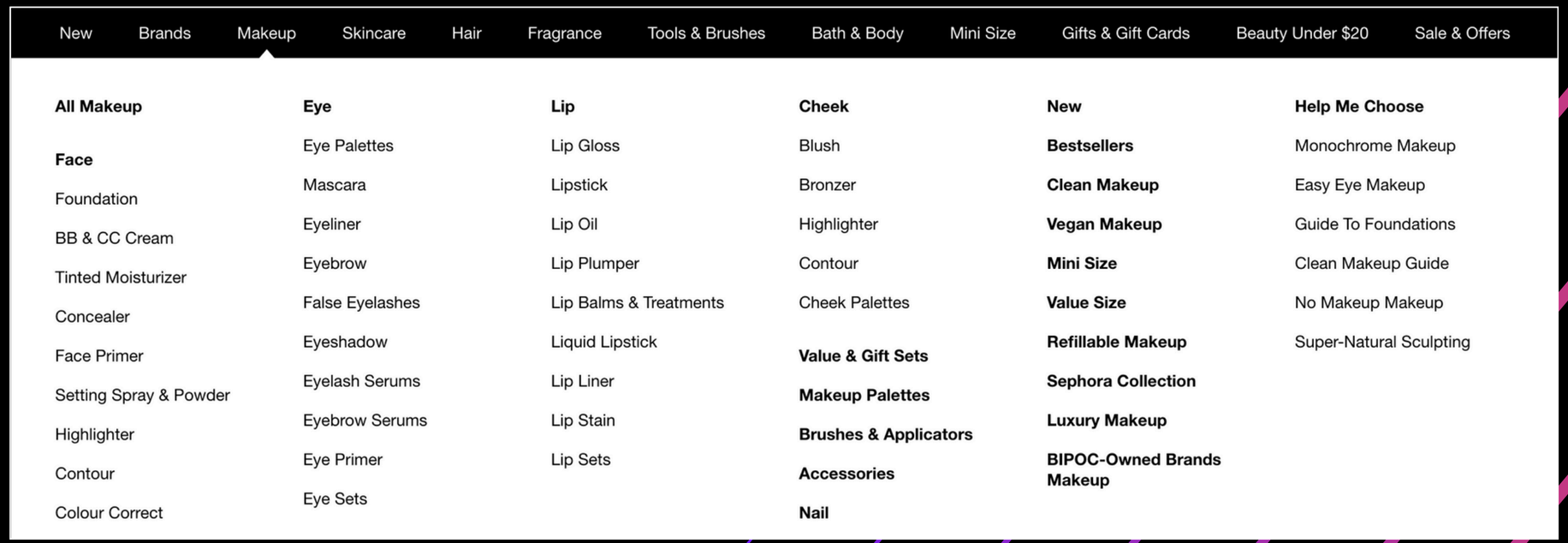
1. Business Analysts
2. Supply Chain Analysts
3. Marketing Analysts
4. In-store Emlployees
5. Customer Service Representatives

WEBSITE SAMPLES

Categories

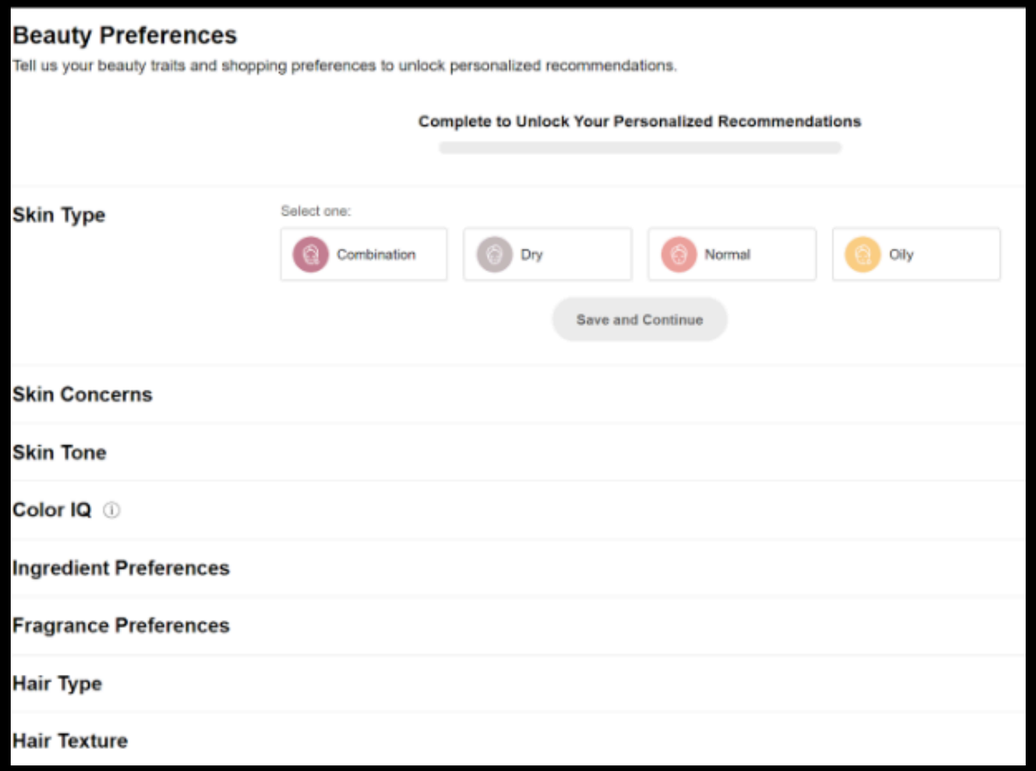


Sub-Categories

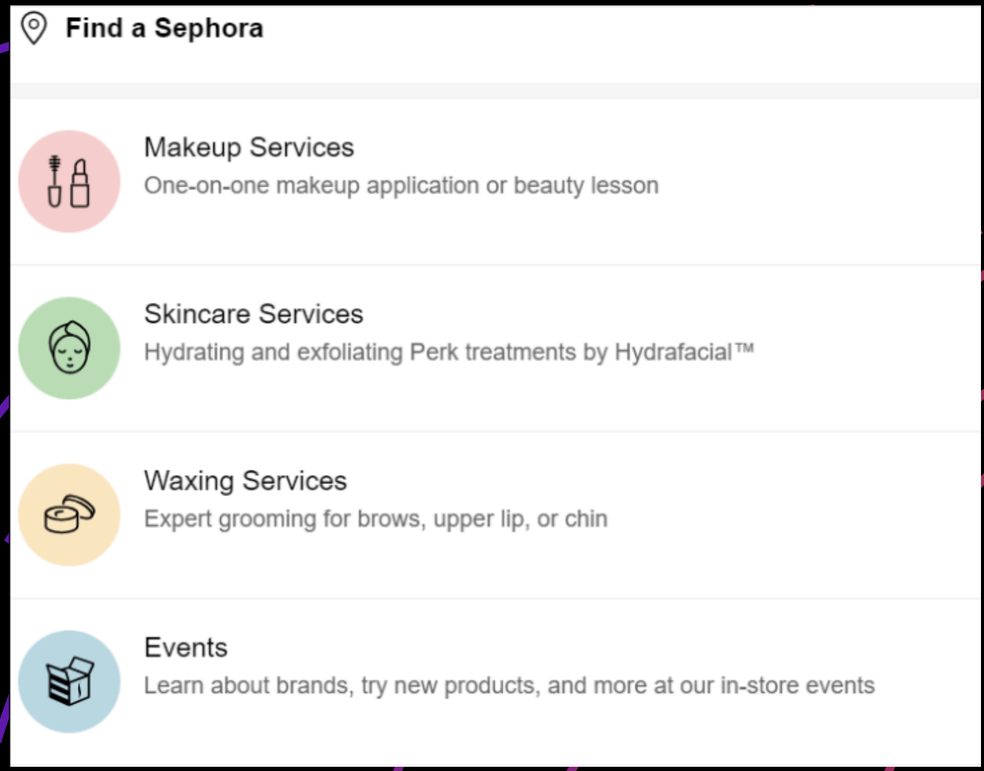


WEBSITE SAMPLES

Beauty Preferences

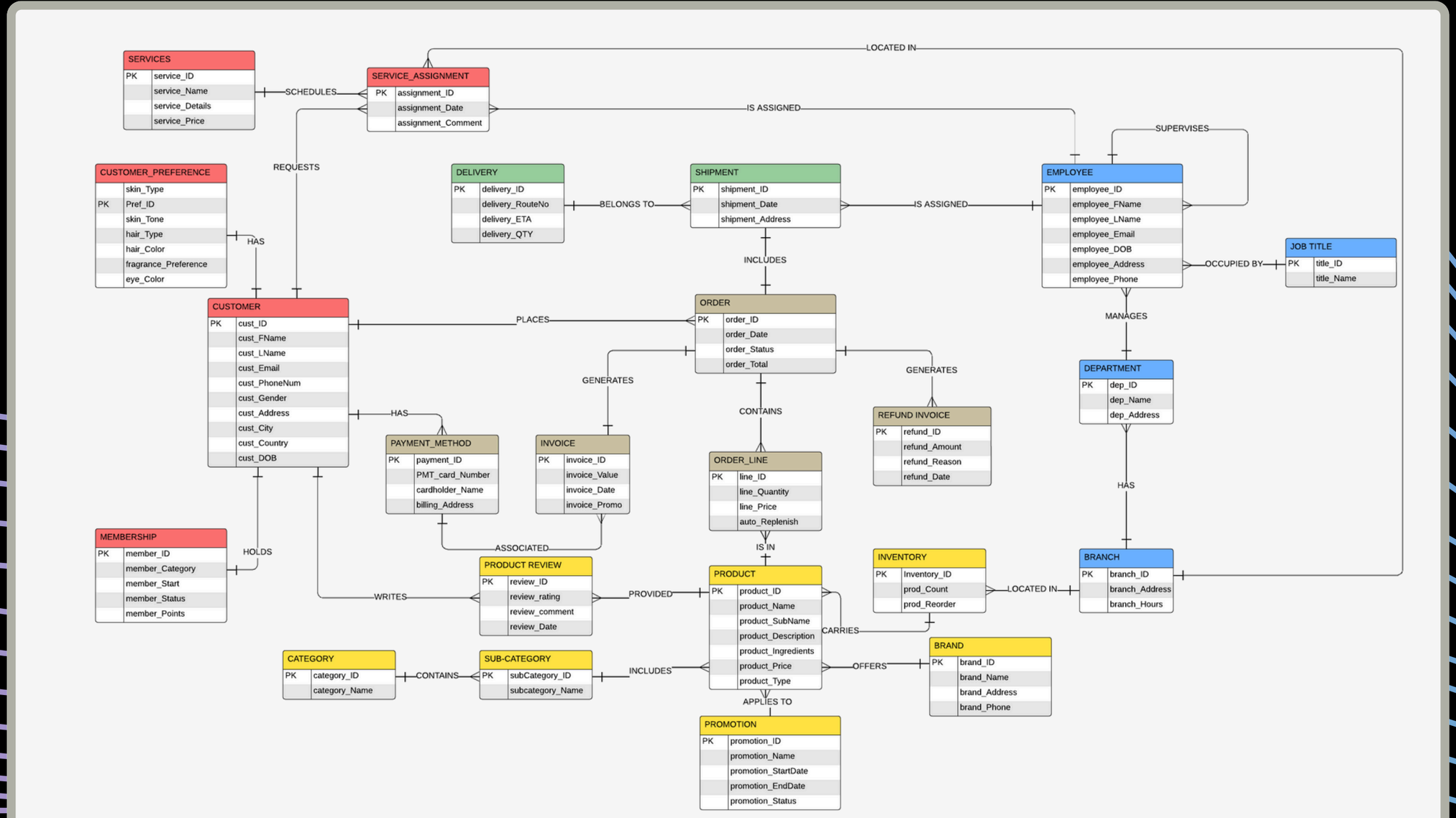


Beauty Services

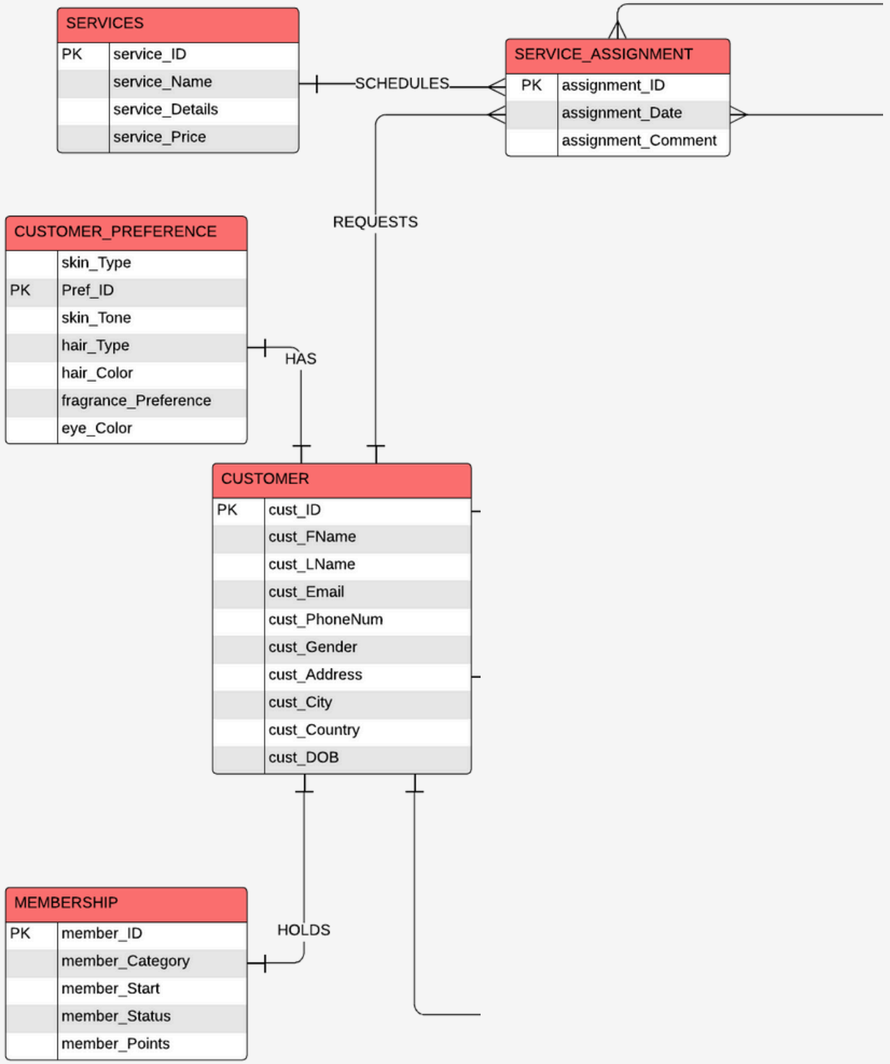


ENTITY RELATIONSHIP DIAGRAM

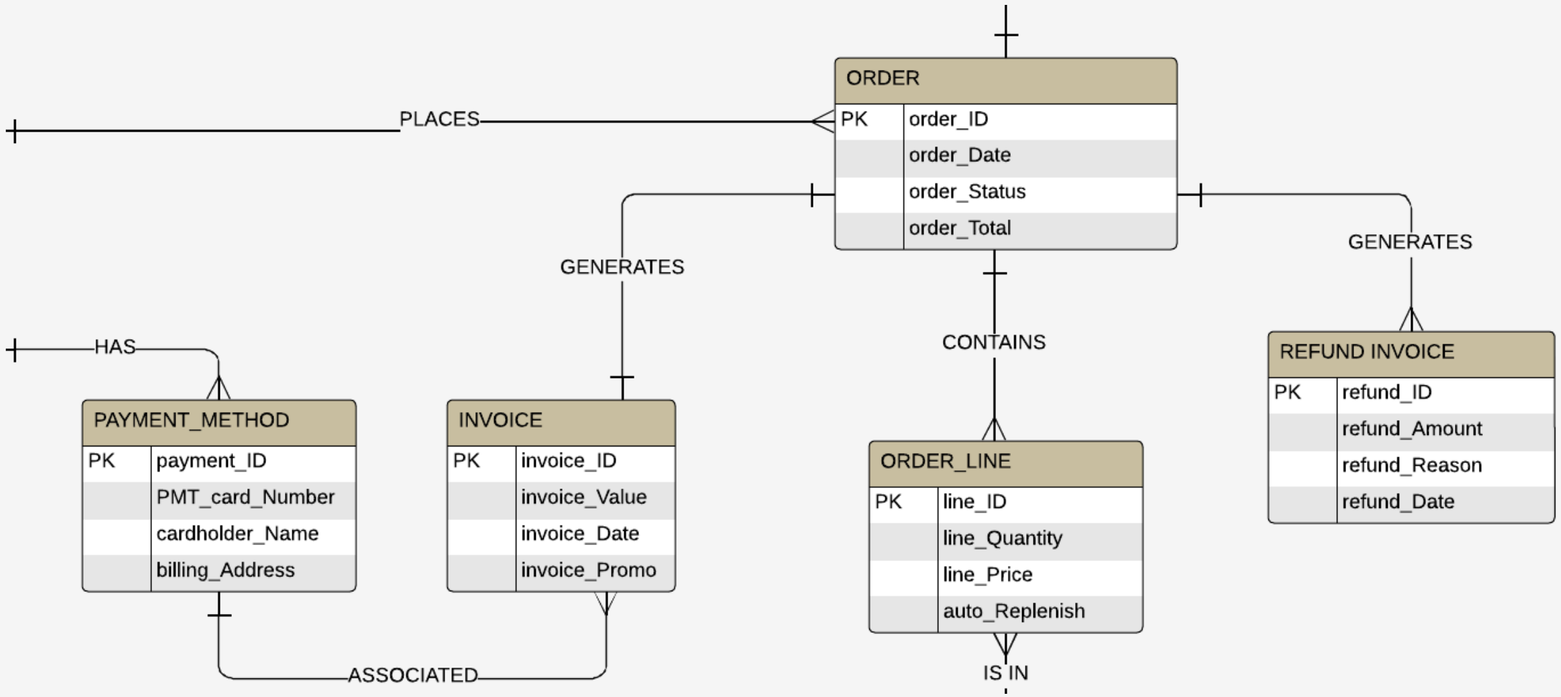
ERD OVERVIEW



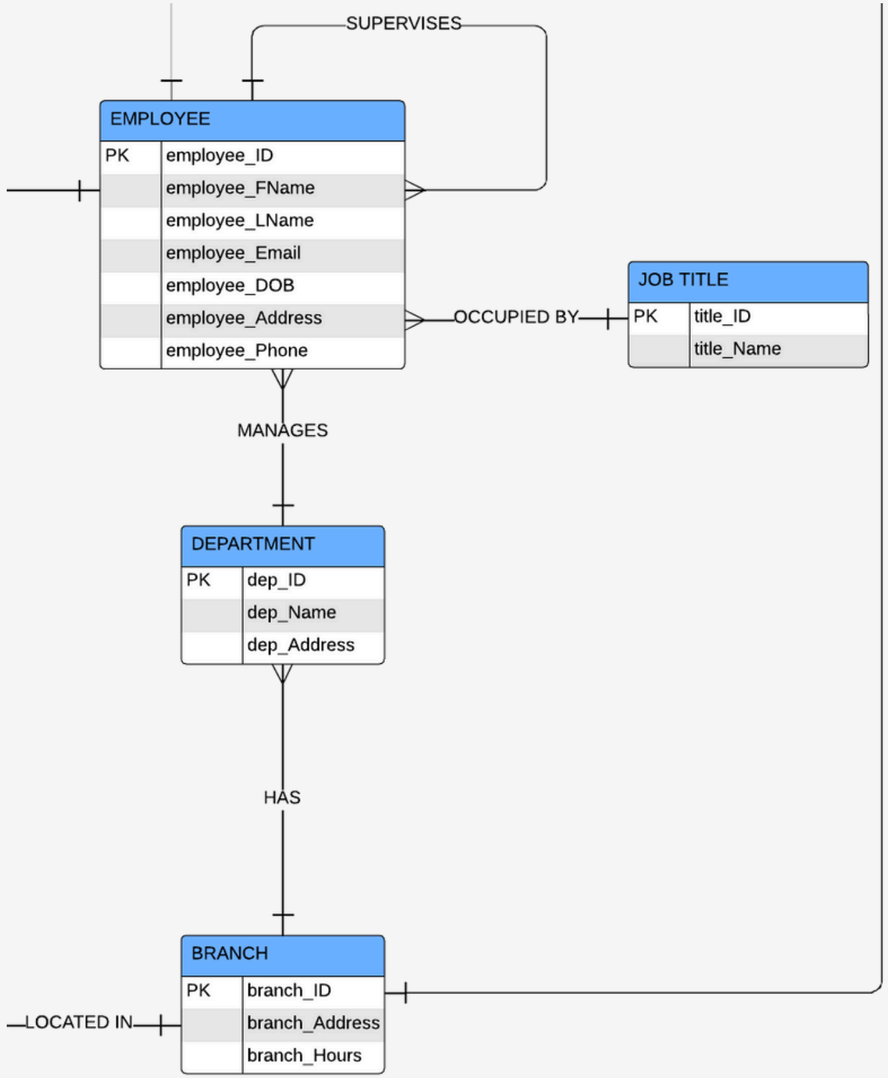
CUSTOMER MANAGEMENT



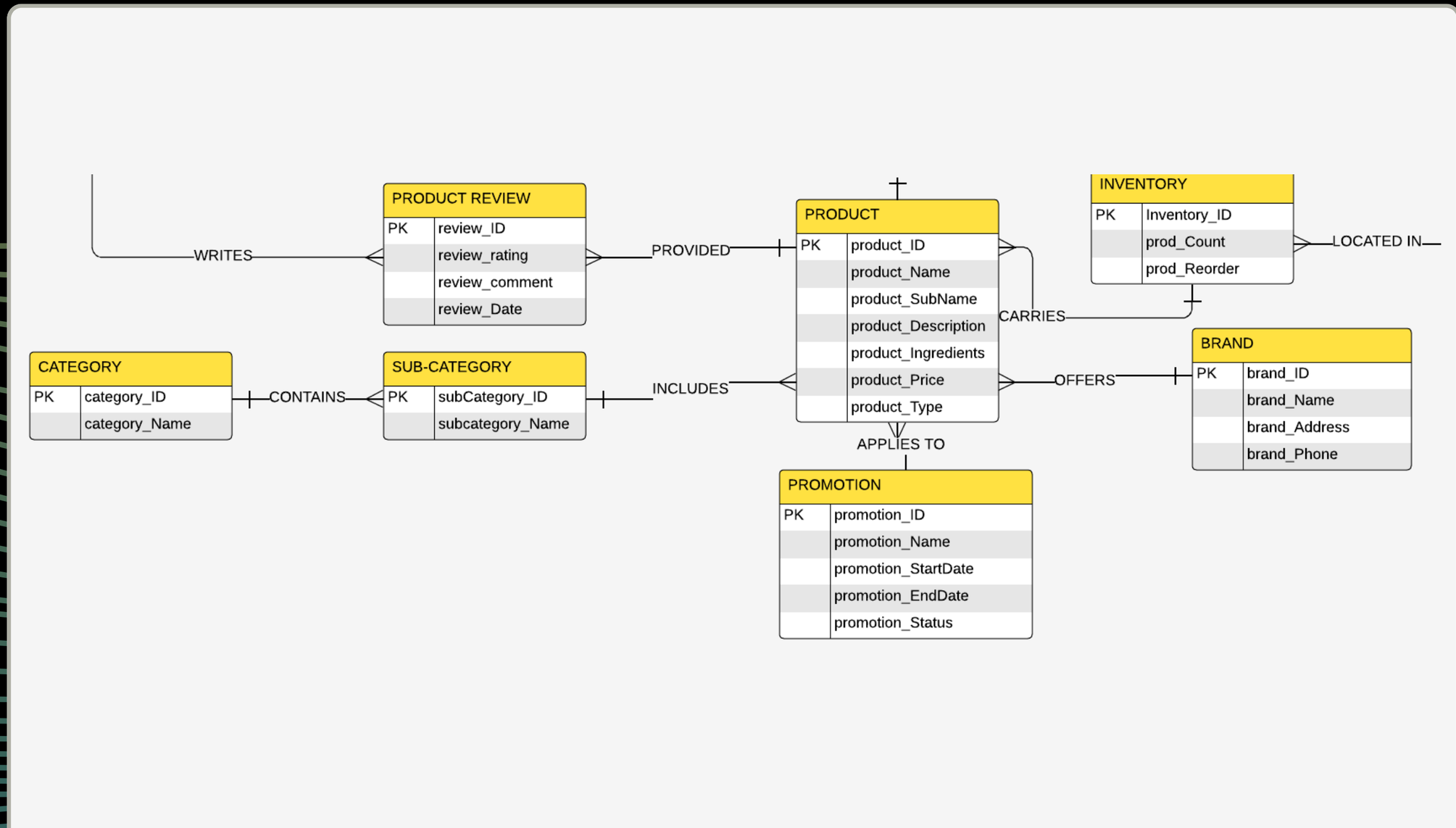
ORDER MANAGEMENT



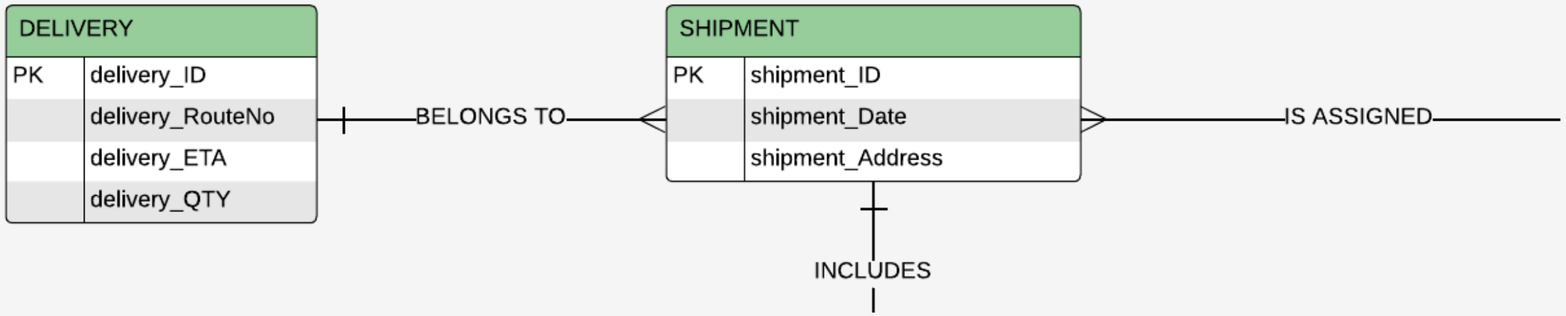
EMPLOYEE MANAGEMENT



PRODUCT MANAGEMENT



SHIPMENT MANAGEMENT



RELATIONAL SCHEMA

CUSTOMER MANAGEMENT

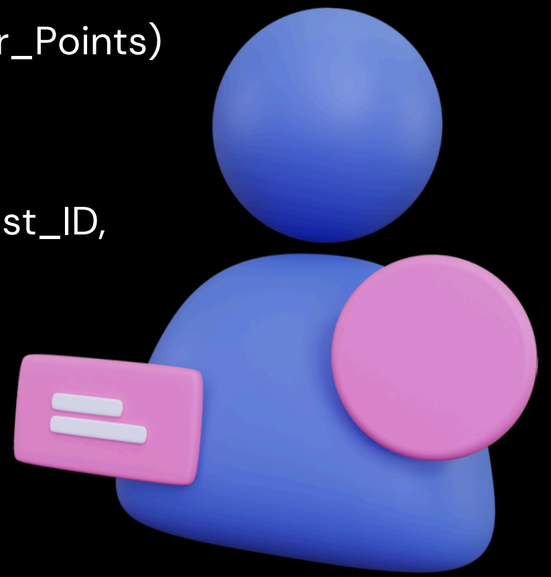
CUSTOMER (cust_ID (PK), cust_FName, cust_LName, cust_Email, cust_PhoneNum, cust_Gender, cust_Address, cust_City, cust_Country, cust_DOB)
PK: cust_ID

CUSTOMER_PREFERENCE(cust_ID, skin_Type, skin_Tone, hair_Type, hair_Color, fragrance_Preference, eye_Color)
PK :cust_ID
FK: cust_ID references CUSTOMER(cust_ID)

MEMBERSHIP(member_ID, cust_ID, member_Category, member_Start, member_Status, member_Points)
PK: member_ID
FK: cust_ID references CUSTOMER(cust_ID)

SERVICE_ASSIGNMENT(assignment_ID, assignment_Date, assignment_Comment, branch_ID, cust_ID, service_ID, employee_id)
PK: assignment_ID
FK: branch_ID references BRANCH(branch_ID)
FK: cust_ID references CUSTOMER(cust_ID)
FK: service_ID references SERVICES(service_ID)
FK: employee_ID references employee(employee_ID)

SERVICES(service_ID, service_Name, service_Details, service_Price)
PK: service_ID



ORDER MANAGEMENT

ORDER(order_ID, order_Date, order_Status, order_Total, cust_ID, shipment_ID, invoice_ID) PK: order_ID
FK: cust_ID references CUSTOMER(cust_ID)
FK: shipment_ID references SHIPMENT(shipment_ID)
FK: invoice_ID references INVOICE(invoice_ID)

REFUND_INVOICE(refund_ID, refund_Amount, refund_Reason, refund_Date, order_ID)
PK: refund_ID
FK: order_ID references ORDER(order_ID)

INVOICE(invoice_ID, invoice_Value, invoice_Date, invoice_Promo, order_ID, payment_ID)
PK: invoice_ID
FK: order_ID references ORDER(order_ID)
FK: payment_ID references PAYMENT_METHOD(payment_ID)

PAYMENT_METHOD(payment_ID, PMT_card_Number, cardholder_Name, billing_Address, cust_ID)
PK: payment_ID
FK: cust_ID references CUSTOMER(cust_ID)

ORDER_LINE(line_ID, line_Quantity, line_Price, auto_Replenish, order_ID, product_ID)
PK: line_ID
FK: order_ID references ORDER(order_ID)
FK: product_ID references PRODUCT(product_ID)

EMPLOYEE MANAGEMENT

EMPLOYEE(employee_ID, employee_FName, employee_LName, employee_Email, employee_DOB, employee_Address, employee_Phone, manager_ID, dep_ID, title_ID)
PK: employee_ID
FK: manager_ID references EMPLOYEE(employee_ID)
FK: dep_ID references DEPARTMENT(dep_ID)
FK: title_ID references Job Title(title_ID)

JOB_TITLE(title_ID, title_Name)
PK: title_ID

DEPARTMENT(dep_ID, dep_Name, dep_Address, branch_ID)
PK: dep_ID
FK: branch_ID references BRANCH(branch_ID)

BRANCH(branch_ID, branch_Address, branch_Hours)
PK: branch_ID

PRODUCT MANAGEMENT

PRODUCT(Product_ID, Product_Name, Product_SubName, Product_Description, Product_Ingredients, Product_Price, Product_Type, Brand_ID, Sub_Category_ID, Brand_ID, Promotion_ID, Inventory_ID)
PK: Product_ID
FK: SubCategory_ID reference Sub-category(SubCategory_ID)
FK: Brand_ID reference Brand(brand_ID)
FK: Promotion_ID reference Promotion(Promotion_ID)
FK: Inventory_ID reference Inventory(Inventory_ID)

BRAND(brand_ID, brand_Address, brand_Phone)
PK: brand_ID

PROMOTION(promotion_ID, promotion_Name, promotion_StartDate, promotion_EndDate, promotion_Status)
PK: promotion_ID

CATEGORY(category_ID, category_Name)
PK: category_ID
PRODUCT_REVIEW(review_ID, review_rating, review_comment, review_Date, cust_ID, product_ID)
PK: review_ID
FK: cust_ID reference custome(cust_ID)
FK: product_ID reference product(product_ID)
INVENTORY(prod_Count, prod_Reorder, branch_ID)
FK: branch_ID reference Brach(Brach_ID)

SUB_CATEGORY (subcatgory_ID, subcategory_Name, Category_ID)
PK: subCategory_ID
FK: Category_ID reference Category(category_ID)

SHIPMENT MANAGEMENT

SHIPMENT(shipment_ID, order_ID, employee_ID, shipment_Date, shipment_Address, delivery_ID)
PK: shipment_ID
FK: order_ID references ORDER(order_ID)
FK: employee_ID references EMPLOYEE(employee_ID)
FK: delivery_ID references Delivery(delivery_id)

DELIVERY(delivery_ID， delivery_RouteNo, delivery_ETA, delivery_QTY)
PK: delivery_ID

SAMPLE QUERIES...

CUSTOMER RETENTION AND ORDER FREQUENCY ANALYSIS

QUERY #1

- **Purpose:**
  - Analyze customer retention and order frequency
  - Track how often customers place orders and how this correlates with their membership status.

- **Business Insight:**
  - Tracks how often customers place orders
  - Duration of their engagement with the company.
  - Improves customer retention strategies and targets frequent buyers with loyalty programs.

QUERY #1

```
SELECT
 CUST_ID,  CUST_FNAME, CUST_LNAME, MEMBER_CATEGORY, TOTAL_ORDERS,
 DENSE_RANK() OVER (PARTITION BY CUST_ID ORDER BY ORDER_DATE) AS
ORDER_FREQUENCY_RANK,
 LAST_ORDER_DATE, FIRST_ORDER_DATE,
 DATEDIFF(LAST_ORDER_DATE, FIRST_ORDER_DATE) AS
DAYS_BETWEEN_FIRST_AND_LAST_ORDER
FROM (
 SELECT
  C.CUST_ID, C.CUST_FNAME, C.CUST_LNAME, M.MEMBER_CATEGORY,
  COUNT(O.ORDER_ID) AS TOTAL_ORDERS,
  MAX(O.ORDER_DATE) AS LAST_ORDER_DATE,
  MIN(O.ORDER_DATE) AS FIRST_ORDER_DATE,
  O.ORDER_DATE
FROM CUSTOMER C
 JOIN MEMBERSHIP M ON C.CUST_ID = M.CUST_ID
 JOIN ORDERS O ON C.CUST_ID = O.CUST_ID
 JOIN ORDER_LINE OL ON O.ORDER_ID = OL.ORDER_ID
 JOIN PRODUCT P ON OL.PRODUCT_ID = P.PRODUCT_ID
 JOIN PROMOTION PR ON P.PROMOTION_ID = PR.PROMOTION_ID
 JOIN BRAND B ON P.BRAND_ID = B.BRAND_ID
 JOIN INVOICE I ON O.ORDER_ID = I.ORDER_ID
 GROUP BY C.CUST_ID, C.CUST_FNAME, C.CUST_LNAME, M.MEMBER_CATEGORY, O.ORDER_DATE
) AS SUBQUERY
ORDER BY TOTAL_ORDERS DESC;
```

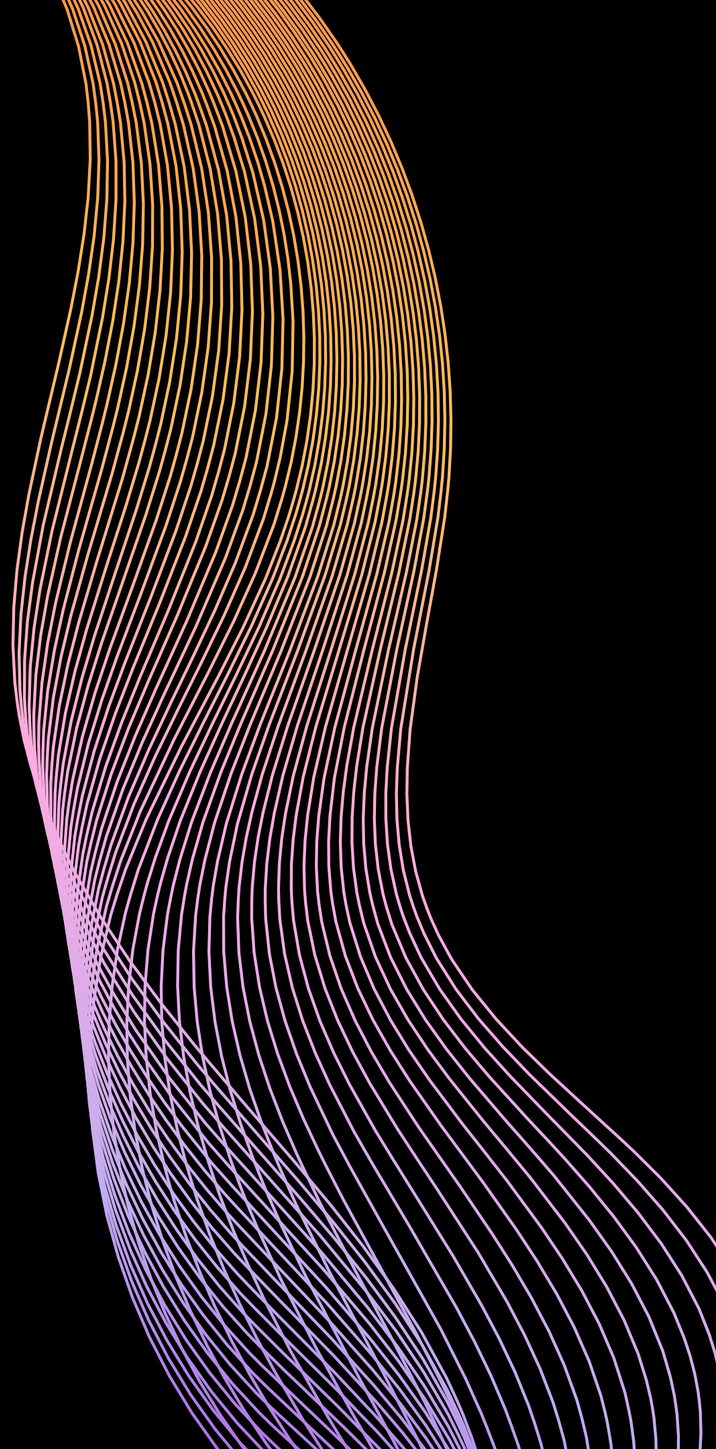
QUERY #1

Output:

cust_ID	cust_FName	cust_LName	member_Category	Total_Orders	Last_Order_Date	First_Order_Date	Order_Frequency_Rank	Days_Between_First_And_Last_Order
1	John	Doe	Gold	1	2024-02-15	2024-02-15	1	0
2	Jane	Smith	Silver	1	2024-02-16	2024-02-16	1	0
3	Alice	Brown	Bronze	1	2024-02-17	2024-02-17	1	0
4	Bob	Johnon	Gold	1	2024-02-18	2024-02-18	1	0
5	Carol	Davis	Silver	1	2024-02-19	2024-02-19	1	0
6	Eve	Miller	Bronze	1	2024-02-20	2024-02-20	1	0
7	Frank	Wilson	Gold	1	2024-02-21	2024-02-21	1	0
8	Grace	Taylor	Silver	1	2024-02-22	2024-02-22	1	0
9	Henry	White	Bronze	1	2024-02-23	2024-02-23	1	0
10	Isabel	Green	Gold	1	2024-02-24	2024-02-24	1	0

SAMPLE QUERIES...

CUSTOMER SEGMENTATION FOR PERSONALIZED PRODUCT RECOMMENDATIONS

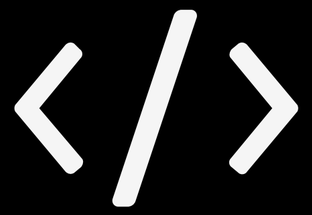


QUERY #2

- **Purpose**:
  - Segments customers based on their purchasing behaviour (frequency, recency, and monetary value)
  - Provides personalized product recommendations.
  - Increase customer engagement and sales.

- **Business Insight:**
  - Identify the most valuable and active customers
  - Recommend products based on similar customer purchasing behaviour.
  - Sephora can increase the likelihood of future sales.
  - Improve customer retention and satisfaction by offering relevant product suggestions

QUERY #2



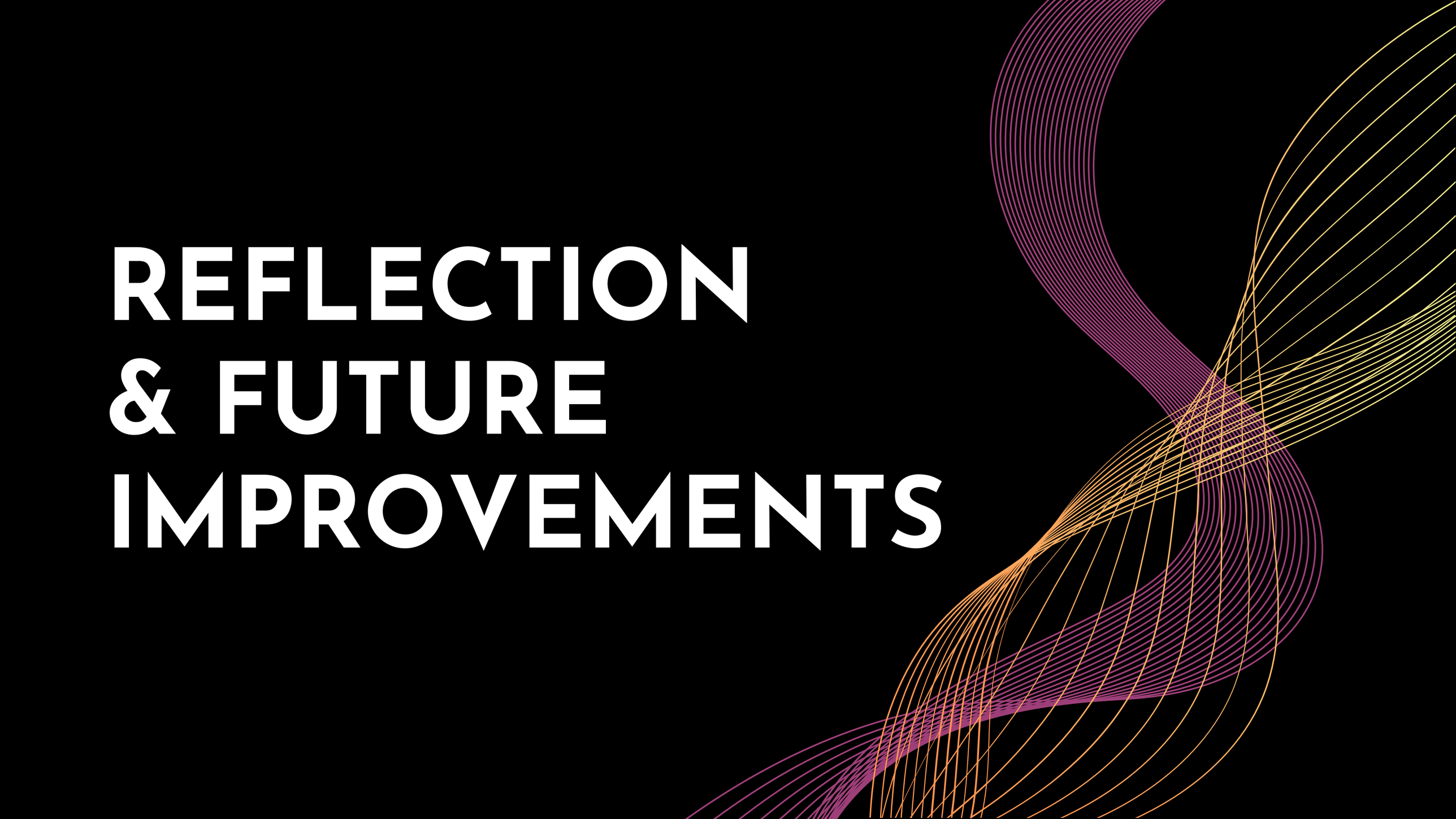
```
WITH CUSTOMER_SEGMENTATION AS (
 SELECT
  C.CUST_ID, C.CUST_FNAME, C.CUST_LNAME,
  COUNT(O.ORDER_ID) AS ORDER_FREQUENCY,
  MAX(O.ORDER_DATE) AS LAST_ORDER_DATE,
  SUM(I.INVOICE_VALUE) AS TOTAL_SPENT,
  DENSE_RANK() OVER (ORDER BY COUNT(O.ORDER_ID) DESC) AS
FREQUENCY_RANK,
  DENSE_RANK() OVER (ORDER BY MAX(O.ORDER_DATE) DESC) AS
RECENCY_RANK,
  DENSE_RANK() OVER (ORDER BY SUM(I.INVOICE_VALUE) DESC)
AS MONETARY_RANK
 FROM CUSTOMER C
 JOIN ORDERS O ON C.CUST_ID = O.CUST_ID
 JOIN INVOICE I ON O.ORDER_ID = I.ORDER_ID
 JOIN ORDER_LINE OL ON O.ORDER_ID = OL.ORDER_ID
 JOIN PRODUCT P ON OL.PRODUCT_ID = P.PRODUCT_ID
 GROUP BY C.CUST_ID, C.CUST_FNAME, C.CUST_LNAME ),

TOP_PRODUCTS_PER_SEGMENT AS (
 SELECT
  CS.CUST_ID, P.PRODUCT_NAME,
    SUM(OL.LINE_QUANTITY) AS TOTAL_QUANTITY,
  RANK() OVER (PARTITION BY CS.CUST_ID ORDER BY
SUM(OL.LINE_QUANTITY) DESC) AS PRODUCT_RANK
 FROM CUSTOMER_SEGMENTATION CS
 JOIN ORDERS O ON CS.CUST_ID = O.CUST_ID
 JOIN ORDER_LINE OL ON O.ORDER_ID = OL.ORDER_ID
 JOIN PRODUCT P ON OL.PRODUCT_ID = P.PRODUCT_ID GROUP BY
CS.CUST_ID, P.PRODUCT_NAME ),

Product_Recommendations AS (
 SELECT
  cs.cust_FName,
  cs.cust_LName,
  cs.Frequency_Rank,
  cs.Recency_Rank,
  cs.Monetary_Rank,
  tp.product_Name AS Recommended_Product
FROM Customer_Segmentation cs
 JOIN Top_Products_Per_Segment tp ON cs.cust_ID =
tp.cust_ID
 WHERE tp.Product_Rank = 1
  AND cs.Monetary_Rank <= 5 -- Target top 5 spenders
  AND cs.Recency_Rank <= 5 -- Target customers with recent
activity )


SELECT
 pr.cust_FName,
 pr.cust_LName,
 pr.Recommended_Product,
 pr.Frequency_Rank,
 pr.Recency_Rank, pr.Monetary_Rank
FROM
 Product_Recommendations pr
ORDER BY pr.Monetary_Rank, pr.Recency_Rank,
pr.Frequency_Rank;
```

QUERY #2

Output:

cust_FName	cust_LName	Recommended_Product	Frequency_Rank	Recency_Rank	Monetary_Rank
Henry	Reed	Perfume	1	4	4

REFLECTION & FUTURE IMPROVEMENTS



REFLECTION

- **Create Independent Tables First**
  - Always create tables without foreign key dependencies before those that reference them. This ensures data consistency and avoids errors where foreign key values lack corresponding entries in the referenced table.

- **Carefully Plan Relationships and Use ER Diagrams**
  - Before implementation, carefully design and validate relationships between tables using ER diagrams. This helps maintain data integrity and prevents redundancy by ensuring accuracy in the conceptual design phase.

IMPROVEMENT

Use Triggers for automation:

- Consider implementing triggers such as auto_increment for tasks that need to happen automatically, such as updating customer id for new customers.

Enforce constraints:

- We can apply appropriate constraints (e.g., UNIQUE, NOT NULL) to maintain data integrity and prevent invalid data from being entered into the database if we have more information on the business context.

THANK YOU

WE APPRECIATE YOUR TIME AND ATTENTION.