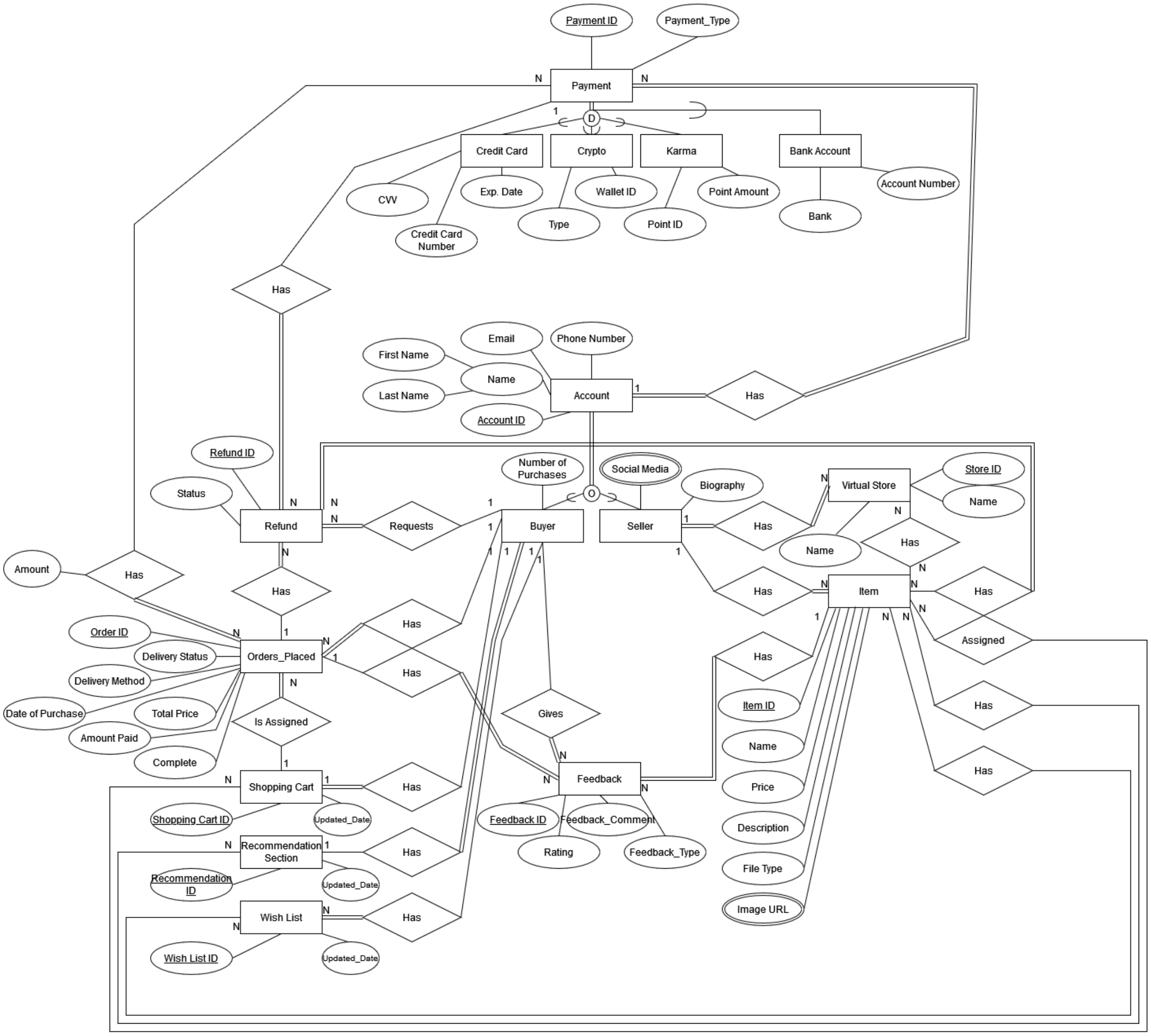
Rachelle Soh.26, Matthew Fong.131, Eddie Tassy.4, Irfan Fazdane.1

1)



2)

Payment (Payment_ID, Payment_Type, Account_ID)
Foreign Key **Account_ID** references Account

Credit_Card (Payment_ID, CVV, Credit_Card_Number, Exp_Date)
Foreign key **Payment_ID** references Payment

Crypto (Payment ID, Crypto_Type, Wallet_ID)
Foreign key **Payment_ID** references Payment

Karma (Payment_ID, Point_ID, Point_Amount)
Foreign key **Payment_ID** references Payment

Bank_Account (Payment_ID, Bank, Account_Number)
Foreign key **Payment_ID** references Payment

Account (Account_ID, First_Name, Last_Name, Email, Phone_Number, BFlag, Number_of_Purchases, Recommendation_ID, SFlag, Biography)
Foreign key **Recommendation_ID** references Recommendation_Section

Social_Media_Accounts (Seller_ID, Social_Media)
Foreign key **Seller_ID** references Account

Refund (Refund_ID, Payment_ID, Status, Order_ID, Buyer_ID)
Foreign key **Payment_ID** references Payment
Foreign key **Order_ID** references Orders_Placed
Foreign key **Buyer_ID** references Account

Orders_Placed (Order_ID, Delivery_Status, Delivery_Method, Date_of_Purchase, Total_Price, Amount_Paid, Completed, Buyer_ID, Shopping_Cart_ID)
Foreign key **Buyer_ID** references Account
Foreign key **Shopping_Cart ID** references Shopping_Cart

Shopping_Cart (Shopping_Cart_ID, Updated_Date, Buyer_ID)
Foreign key **Buyer_ID** references Account

Recommendation_Section (Recommendation_ID, Updated_Date, Buyer_ID)
Foreign key **Buyer_ID** references Account

Wish_List (Wish_List_ID, Updated_Date, Buyer_ID)
Foreign key **Buyer_ID** references Account

Feedback (Feedback_ID, Rating, Feedback_Comment, Buyer_ID, Item_ID, Order_ID)
Foreign Key **Buyer_ID** references Account
Foreign Key **Item_ID** references Item
Foreign Key **Order_ID** references Orders_Placed

Item (Item_ID, Item_Name, Price, Description, File_Type, Seller_ID)
Foreign key **Seller_ID** references Account

Image_Url_Links (Item_ID, Image_URL)
Foreign key **Item_ID** references Item

Virtual_Store (Store_ID, Name, Seller_ID)
Foreign key **Seller_ID** references Account

Payment_Order (Payment_ID, Order_ID, Amount)
Foreign Key **Payment_ID** references Payment
Foreign Key **Order_ID** references Orders_Placed

Item_Wishlist (Item_ID, Wish_List_ID)
Foreign Key **Item_ID** references Item
Foreign Key **Wish_List_ID** references Wish_List

Item_Recommend (Item_ID, Recommendation_ID)
Foreign Key **Item_ID** references Item
Foreign Key **Recommendation_List_ID** references Recommendation_Section

Item_Shopping_Cart (Item_ID, Shopping_Cart_ID)
Foreign Key **Item_ID** references Item
Foreign Key **Shopping_Cart_ID** references Shopping_Cart

Item_Refund (Item_ID, Refund_ID)
Foreign Key **Item_ID** references Item
Foreign Key **Refund_ID** references Refund

Item_VirtualStore (Item_ID, Store_ID)
Foreign Key **Item_ID** references Item
Foreign Key **Store_ID** references Virtual Store

3.

a.

Item_VirtualStore

b.

```
π Item_Name (
        σ Price < 10 (
                Item
        )
)
```

c.

```
π Item_Name, Date_of_Purchase (
        σ Buyer_ID = 1 (
                ( (Order * Shopping_Cart) * Item_Shopping_Cart) * Item
        )
)
```

**Buyer_ID = 1, 1 is a placeholder for the given Buyer_ID you want to search for*

d.

```
π First_Name, Last_Name, Item_Name (
    σ Store_ID = 1 (
            Account * (Order * (Shopping_Cart * (Item_Shopping_Cart * Item_VirtualStore)))
    )
)
```

**Store_ID = 1, 1 is a placeholder for the given Store_ID you want to search for*

E.

```
F MAX Number_of_Purchases (
        π Account_ID, Number_of_Purchases (
                Account
        )
)
```

F.

```
σ count ≤ 5 (
        Store_ID F COUNT Item_ID (
                VirtualStore_Item
        )
)
```

g.

```
π item_ID, max_itemID, count_itemID, add_itemID_price, FName, LName (
    item_ID F MAX item_ID, COUNT item_ID, SUM ((COUNT item_ID) * price) (
             Account * (Order * (Shopping_Cart * (Item_Shopping_Cart * Item)))
      )
 )
```

h.

```
Payment_Type F COUNT Payment_Type, SUM price (
        Order * (Payment_Order * Payment)
)
```

i.

```
π First_Name, Last_Name, Email (
        F MAX Point_Amount (
                Account * (Payment * Karma)
        )
)
```

4.

a. Create a list of people who requested refunds. The query should include all buyers, including those who haven't requested a refund.

```
Account ⟕ AccountID=BuyerID (Refund)
```

b. Find the buyer who has left the most feedback.

Count_Of_Buyer ← Buyer_ID F COUNT Feedback_ID (Feedback)
π First_Name, Last_Name (
    (Buyer_ID F MAX Feedback_ID (Count_Of_Buyer ⋈ Buyer_ID = Account_ID (Account)))
* Account
)

c. Find the wishlist with the most number of items.

Count_Of_Items ← Wish_List_ID F COUNT Wish_List_ID (Item_Wishlist)
Wish_List_ID F MAX Wish_List_ID (Count_Of_Items)