Database Systems Project – Phase03 Version02
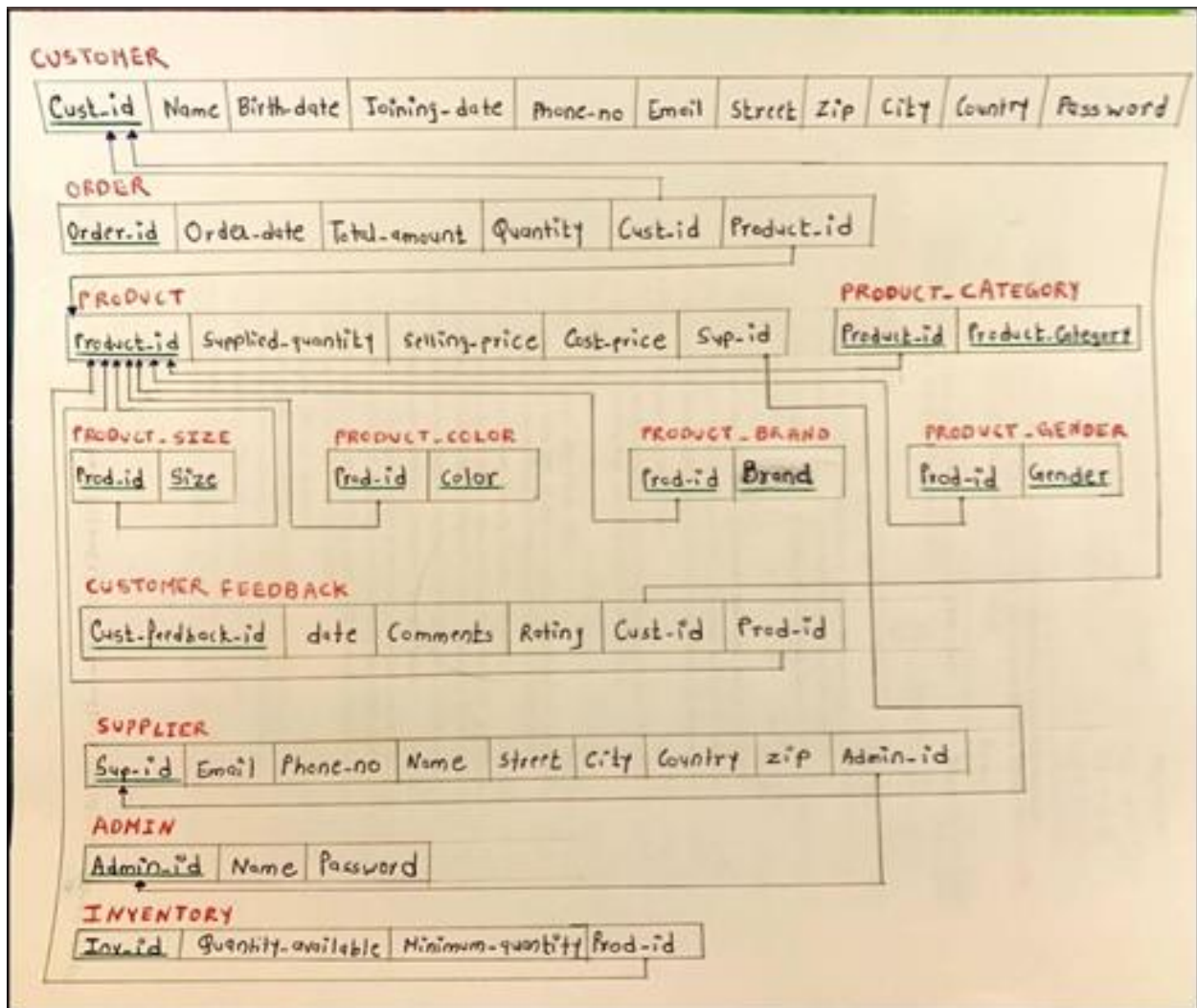
Mapping to Relations

Team Member Details:

Anu Sanjay Bhoyar Mav ID - 1001841237
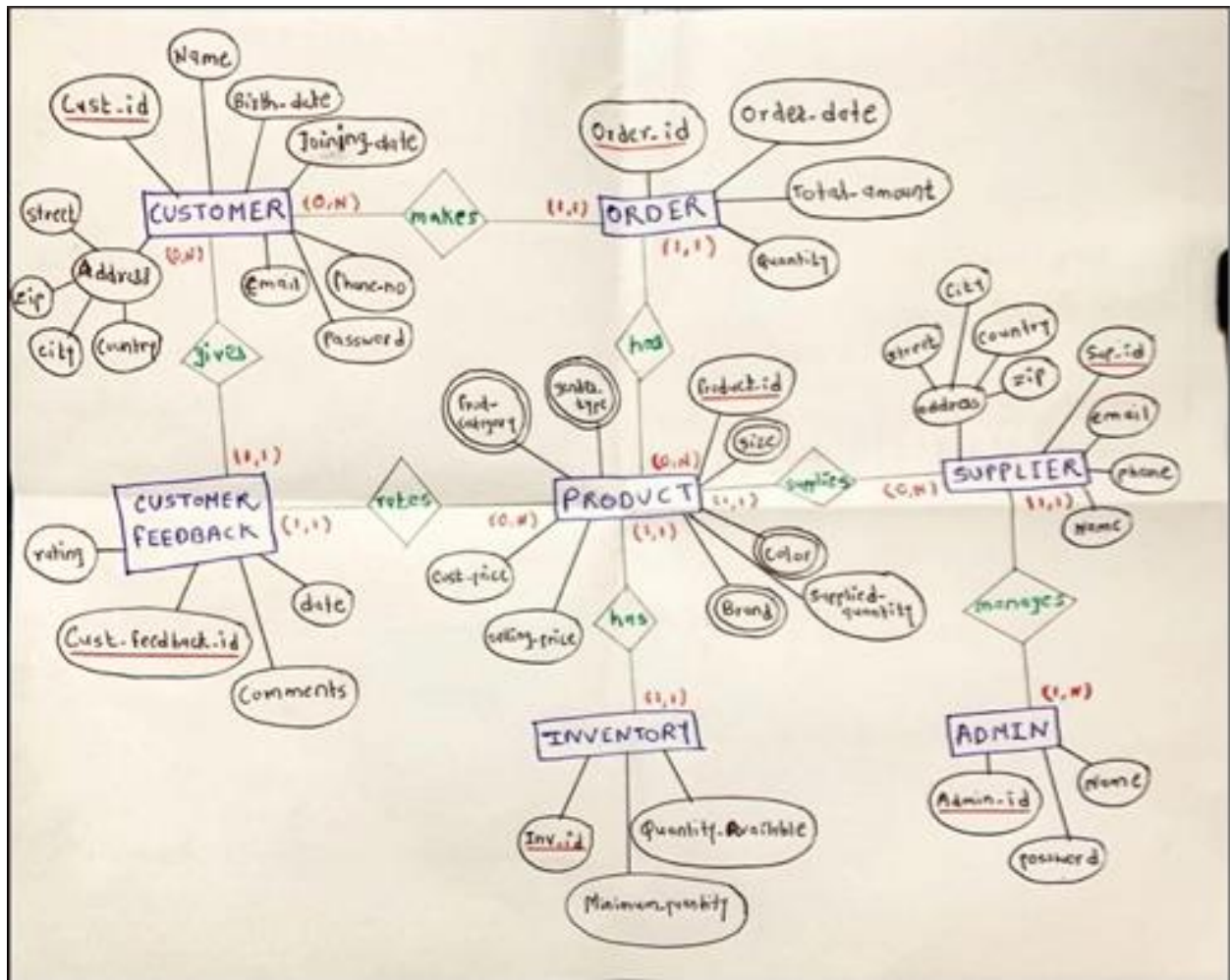Priya Kulkarni Mav ID - 1002088875
Sowjanya Sadashiva Mav ID - 1001898874

Mapped Relations:

CUSTOMER

Cust_id	Name	Birth-date	Joining-date	Phone-no	Email	Street	Zip	City	Country	Password

ORDER

Order_id	Order-date	Total-amount	Quantity	Cust_id	Product_id

PRODUCT

Product_id	Supplied-quantity	Selling-price	Cost-price	Sup-id

PRODUCT_CATEGORY

Product_id	Product-Category

PRODUCT_SIZE

Prod-id	Size

PRODUCT_COLOR

Prod-id	Color

PRODUCT_BRAND

Prod-id	Brand

PRODUCT_GENDER

Prod-id	Gender

CUSTOMER FEEDBACK

Cust-feedback-id	date	Comments	Rating	Cust-id	Prod-id

SUPPLIER

Sup-id	Email	Phone-no	Name	Street	City	Country	zip	Admin-id

ADMIN

Admin_id	Name	Password

INVENTORY

Inv_id	Quantity-available	Minimum-quantity	Prod-id

Updated ER diagram.

Entity relations, candidate keys and Functional dependency

Candidate keys:

1. Customer: *Cust_Id*
2. Order: *Order_id*
3. Product: *Prod_id*
4. Supplier: *Supplier_ID*
5. Customer Feedback: *CustFeed_ID*
6. Inventory: *Inventory_ID*
7. Admin: *Admin_id*

Functional dependency:

Customer:

1. CustId → { name, DOB, Joining_date, email, street, zip, city, country, phone no, password }
2. { CustId , name } → { DOB, Joining_date, email, street, zip, city, country, phone no }
3. { CustId , DOB } → { name, Joining_date, email, street, zip, city, country, phone no }
4. { CustId , Joining_date } → { name, email, street, zip, city, country, phone no }
5. { CustId , email } → { name, street, zip, city, country, phone no }
6. { CustId , Zip }→ { name, Joining_date, email, street, city, country, phone no }

Product:

1. ProdId → { qty_supplied, selling price, cost price, supplierId }
2. { ProdId , SupplierID }→ { qty supplied, selling price, cost price }
3. { ProdId , Cost price } → { qty supplied, supplier id, Selling price }

● Product has multi valued attributes: category, gender, size, color, brand.
● Each of which is represented as a separate table in relation with the primary key referencing the foreign key.

Order:

1. OrderId → { order date, total amt, qty, CustId, ProdId }
2. { OrderId, CustId, ProdId }→ { order date, total amt, qty }
3. { OrderId , ProdId } → { Order date, total amt, qty, CustId }
4. { OrderId , CustId }→ { Order date, total amt, qty, ProdId }

Customer Feedback:

1. CustFeedbackId→ { comments, date, rating, CustId, ProdId }
2. { CustFeedbackId, CustId, ProdId } → { comments, date, rating }
3. { CustFeedbackId , CustId } → { rating, comments, date, ProdId }
4. { CustFeedbackId , ProdId } → { rating, comments, date, CustId }

Supplier:

1. SupplierId → { name, email, phone, street, zip,city, country, adminId }
2. { SupplierId , adminId }→ { name, email, phone, street, zip, city, country }

Inventory

1. InventoryId → { ProdId, Qty_available, min qty }
2. { InventoryId , ProdId } → {qty available, min qty }

(Min_qty is used to generate a Min Quantity alert.)

Admin

1. Admin_id → { admin_name ,password}