Online Retail Store

Group: 66 (Hemant Latiyan, Sahil Sahu, Parmesh Yadav, Mehul Goel)

Problem Statement: Create a relational schema for an online retail store covering all aspects from suppliers to customers.

Stakeholders:

1. Customers
2. Suppliers
3. Employees

Entities:

- Customers
- Supplier
- Warehouse
- Employees
- Product
- Account
- Reviews
- Query

Attributes of entities along with primary keys:

- Customers (Cid, Full Name, Mobile no., email,City,State,PINCODE,DOB,age)
- Supplier(Sid, Company Name, Mobile No., City, State, Pincode)
- Warehouse(Wid, City, State, Pincode)
- Employees(Eid,,Full Name, Mobile No, email, DOJ, City, State, Pincode, salary, designation)
- Product(Pid, Name, Sid,Category,Price, stock, brand,Rating, offer)
- Account (Email-ID, password, premium subscription,Aid)

- Reviews (Pid, Star) {WEAK}
- Query (Cid, message, status, forum, time stamp) {WEAK}

Relationship between entities (with entity participation types):

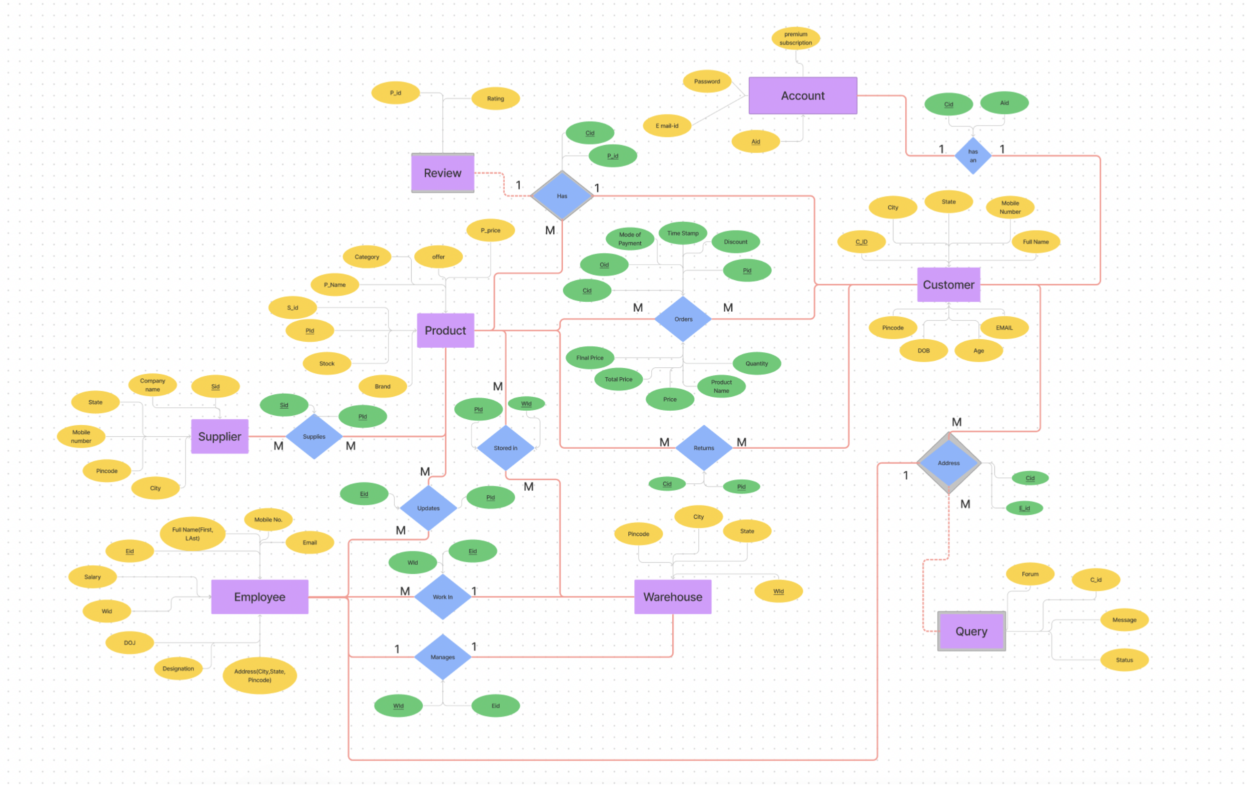
1. Supplier -**SUPPLIES**- Product **(M-M)**
2. Product - **STORED IN** - Warehouse **(M-M)**
3. Employees - **WORK IN** - Warehouse **(M-1)**
4. Customer - **ORDERS -** Product **(M-M)**
5. Employee - **UPDATES** - Product (**M-M)**
6. Employee - **DEALS WITH - QUERIES - BY** - Customers **(1-M-M)**
7. Customer - **HAS AN** - Account **(1-1)**
8. Employee(Admin) - **MANAGES** - Warehouse **(1-1)**
9. Customer - **HAS A - REVIEW - OF** - Product **(1-1-M)**
10. Customer - **RETURNS** - Product **(M-M)**

Relations:

Supplies(Sid, Pid)
Stored in(Wid, Pid)
Returns(Cid, Pid)
Orders(Oid, Cid, Pid, Quantity, Product Name.Price, Total Price, Discount, Final Price,Mode of Payment,Time Stamp)
Updates(Pid, Eid)
Has(Cid, Pid, rating)
Address(Cid, Eid, message, status, forum, time stamp(date_posted))

ER Diagram:

https://www.figma.com/file/XRgdcpYDpcuoY5IgGF3kRG/PROJECT_ER?node-id=0%3A1



Relational Schema:

- Customers (Cid, Full Name, Mobile no., email,City,State,PINCODE,DOB,age)
- Supplier(Sid, Company Name, Mobile No., City, State, Pincode)
- Warehouse(Wid, City, State, Pincode,Eid)
- Employees(Eid,,Full Name, Mobile No, email, DOJ, City, State, Pincode, salary, designation,Wid)
- Product(Pid, Name, Sid,Category,Price, stock, brand,Rating, offer)
- Account (Email-ID, password, premium subscription,Aid,Cid)
- Reviews (Pid, Star) {WEAK}
- Query (Cid, message, status, forum, time stamp) {WEAK}
- supplies(Sid, Pid)
- Stored in(Wid, Pid)
- Returns (Cid, Pid)
- Orders (Oid, Cid, Pid, Quantity, Product Name.Price, Total Price, Discount, Final Price,Mode of Payment,Time Stamp)
- Updates(Pid, Eid)
- Has (Cid, Pid, rating)
- Address (Cid, Eid, message, status, forum, time stamp(date_posted))

Weak Entity and why:

1. Query
2. Reviews

These are the weak entities because they cannot be uniquely identified based on their own attributes. Thus we have used cid as the foreign key in Query and Pid as the foreign key in reviews.

Ternary relationships and why:

1. Has
2. Address

Has is a ternary relationship because it connects customer, product and reviews.

Address is a ternary relationship because it connects customer to employee and queries

Constraints

- Customers: Primary - Cid
- Supplier: Primary - Sid
- Warehouse: Primary - Wid, Foreign - Eid
- Employee: Primary - Eid
- Product: Primary - Pid
- Account: Primary - Email-ID, Foreign Key - Aid
- Reviews: Primary - Pid
- Query: Primary - Cid
- Supplies: Primary - Sid, Foreign - Pid
- Stored in: Primary - Wid, Foreign - Pid
- Returns: Primary - Cid, Foreign - Pid
- Orders: Primary - Oid, Foreign - Cid, Pid
- Updates: Primary - Pid, Foreign - Eid
- Has: Primary - Cid, Foreign - Pid
- Address: Primary - Cid, Foreign - Eid

Queries:

1. **List all the products returned by Customer with id: "C1"**
   Select name from product where pid in (Select pid from returns where cid = c1)
2. **List all the customers who has Premium subscription**
   Select name from Customer where cid in (select cid from account where premium subscription =1)
3. **List all the Customers that ordered the product "P1"**
   Select name from customers where cid in (select cid from order where pid = p1)
4. **List the closest warehouse to a customer with given customerid (c1)**
   select W.wid from customers C, Warehouse W where C.cid = 100 and W.pincode-C.pincode = (select min(W1.pincode-C'.pincode) from customers C' , warehouse W1 where C'.cid = 100);

5. **List products with highest rating which belong to the same category(xyz)**
   Select p.pid from product p, orders o where p.rating = (select max(p1.rating) from product p1) and p.category = 'BOOKS';
6. **List all unsolved queries along with the employee that had to solve them**
   Select eid, cid from address where status = 0
7. **Find the average rating of all products**
   Select avg(reviews.star), pid from reviews groupby pid
8. **List the products with offering discount greater than 5%**
   Select pid, name from product where offer > 0.05
9. **List all products in the same order**
   Select pid from orders where oid in (select distinct oid from orders)
10. **List all products with product rating >= 4**
    Select pid from has where rating >= 4

Sufficient and Valid Data:

https://www.mockaroo.com

Responsibility of each member: All members contributed equally in the project