EER to Relational Data Model WriteUp

- STRONG ENTITIES – **CUSTOMER** (Fname, Mname, Lname, Email_ID (Primary Key), Address, Cust_Type), **ORDER**(Order_ID(Primary Key), Order_Summary, Order_Amount, Order_DateTime), **PAYMENT** (P_ID (Primary Key), Paid_Amount, Payment_Type, Payment_Status), **RESTAURANT** (Res_ID (Primary Key), Name, City, Work_Phone, Address, Res_Type, Owner), **DELIVERY_MAN** (Emp_ID (Primary Key), Name, Work_Phone, Salary)
- WEAK ENTITY – FOOD is mapped as separate relation and primary key of owner class (Res_ID of RESTAURANT) is added as foreign key (Res_ID) in FOOD. **FOOD** (Res_ID, Item_Name). Primary Key is Res_ID and Item_Name.
- Mapping Binary 1: N relationship between CUSTOMER ("1" side) and ORDER ("N" side) – the primary key of CUSTOMER (Email_ID) is referenced in ORDER relation. Add attribute User_Email_ID as foreign key in ORDER.
- Mapping Binary 1: N relationship between RESTAURANT ("1" side) and ORDER ("N" side) – the primary key of RESTAURANT (Res_ID) is referenced in ORDER relation. Add attribute Res_ID as foreign key in ORDER. **ORDER** (Order_ID(Primary Key), Order_Summary, Order_Amount, Order_DateTime, Res_ID, User_Email_ID)
- Mapping Binary 1: N relationship between RESTAURANT ("1" side) and DELIVERY_MAN ("N" side) – the primary key of RESTAURANT (Res_ID) is referenced in DELIVERY_MAN relation. Add attribute Res_ID as foreign key in DELIVERY_MAN. **DELIVERY_MAN** (Emp_ID (Primary Key), Name, Work_Phone, Salary, Res_ID)
- Mapping Binary 1: N relationship between RESTAURANT ("1" side) and PAYMENT ("N" side) – the primary key of RESTAURANT (Res_ID) is referenced in PAYMENT relation. Add attribute Res_ID as foreign key in PAYMENT. **PAYMENT** (P_ID (Primary Key), Paid_Amount, Payment_Type, Payment_Status, Res_ID)
- Mapping Ternary relationship among CUSTOMER, ORDER and PAYMENT – add new relation ORDER_PAYMENT and add primary key attributes of CUSTOMER, ORDER and PAYMENT as foreign key attributes in ORDER_PAYMENT. ORDER_PAYMENT (User_Email_ID, P_ID, Order_ID). Primary Key is User_Email_ID, P_ID and Order_ID.

- Mapping Ternary relationship among PAYMENT, ORDER and CUSTOMER – add new relation ORDER_PAYMENT and add primary key attributes of PAYMENT, ORDER and CUSTOMER as foreign key attributes in ORDER_PAYMENT. **ORDER_PAYMENT** (P_ID, Order_ID, User_Email_ID). Primary Key is P_ID, Order_ID and User_Email_ID.
- Mapping Ternary relationship among DELIVERY_MAN, ORDER and CUSTOMER – add new relation ORDER_DELIVERY and add primary key attributes of DELIVERY_MAN, ORDER and CUSTOMER as foreign key attributes in ORDER_DELIVERY. **ORDER_DELIVERY** (DM_ID, Order_ID, User_Email_ID). Primary Key is DM_ID, Order_ID and User_Email_ID.
- Mapping multivalued attribute – Add new relation for multivalued Phone attribute of CUSTOMER with primary key of CUSTOMER is added to new relation as foreign key. **CUST_PHONE** (Email_ID, PhoneNo). Primary Key is Email_ID and PhoneNo.
- Here, superclass is CUSTOMER and its disjoint sub classes are PREMIUM_MEMBER and REGULAR_MEMBER. Convert each specialization with subclasses (PREMIUM_MEMBER and REGULAR_MEMBER) and generalized superclass (CUSTOMER) into single relation (CUSTOMER). Add attributes of subclasses as attributes to super class. (Mapping specialization based on Step 8.c). **CUSTOMER** (Fname, Mname, Lname, Email_ID (Primary Key), Address, Cust_Type, Reward_Points, Standard_Delivery_Charge)