

# RDBMS FUNDAMENTALS

## ASSIGNMENT 1

**Analyze a given business scenario and create an ER diagram that includes entities, relationships, attributes, and cardinality. Ensure that the diagram reflects proper normalization up to the third normal form.**

### Business Scenario :

An online shopping platform needs a system to manage its operations. The platform sells products to customers who can browse and place orders. Each order can contain multiple products, and each product can be part of multiple orders. The platform also manages suppliers who provide products, with each supplier being able to supply multiple products. Customers can leave reviews for products they've purchased. Additionally, the platform needs to keep track of its employees who manage orders and customer service.

### Entities :

- **Customer** : Represents customers of the platform.
- **Product** : Represents products available for sale.
- **Order** : Represents customer orders.
- **Employee** : Represents employees managing orders.
- **Supplier** : Represents suppliers providing products
- **Review** : Represents customer reviews for products.

### Relationship :

- **Place** : Relationship between Customer and Order.
- **Contain** : Relationship between Order and Product.
- **Handle** : Relationship between Employee and Order.
- **Supply** : Relationship between Supplier and Product.
- **Write** : Relationship between Customer and Review.
- **ReviewFor** : Relationship between Review and Product.

### Attributes :

- **Customer** : CustomerID (PK), Name, Address, PhoneNumber, Email.
- **Product** : ProductID (PK), Name, Description, Price, StockQuantity.
- **Order** : OrderID (PK), OrderDate, TotalAmount, CustomerID (FK), EmployeeID (FK).
- **Employee** : EmployeeID (PK), Name, Position, Email.
- **Supplier** : SupplierID (PK), Name, ContactName, ContactPhone
- **Review** : ReviewID (PK), Rating, Comment, ReviewDate, CustomerID (FK), ProductID (FK).

- **Order Product** : OrderID (FK), ProductID (FK), Quantity (Composite Key).

### Cardinality :

- **Customer-Order** : One customer can place multiple orders, but each order is placed by one customer.
- **Order-Product** : Each order can contain multiple products, and each product can be part of multiple orders (many-to-many relationship).
- **Employee-Order** : One employee can handle multiple orders, but each order is handled by one employee.
- **Supplier-Product**: Each supplier can supply multiple products, and each product can be supplied by multiple suppliers (many-to-many relationship).
- **Customer-Review** : One customer can write multiple reviews, but each review is written by one customer.
- **Review-Product** : One product can have multiple reviews, but each review is for one product.

### Normalization :

**1NF** : All tables have atomic columns.

**2NF** : All non-key attributes are fully functional dependent on the primary key.

**3NF** : No transitive dependencies between non-key attributes.

The data is normalized up to the third normal form (3NF). Each entity's attributes are atomic, and there are no partial dependencies or transitive dependencies.

### Explanation :

- **Customer** and **Order** have a one-to-many relationship, indicating that each customer can place multiple orders.
- **Order** and **Product** have a many-to-many relationship, with the **OrderProduct** junction table managing the relationship. This table contains attributes OrderID, ProductID, and Quantity.
- **Order** and **Employee** have a many-to-one relationship, indicating that each order is handled by one employee, but each employee can handle multiple orders.
- **Supplier** and **Product** have a many-to-many relationship, indicating that each supplier can supply multiple products, and each product can be supplied by multiple suppliers.
- **Customer** and **Review** have a one-to-many relationship, indicating that each customer can write multiple reviews.
- **Review** and **Product** have a many-to-one relationship, indicating that each product can have multiple reviews, but each review is for one product.

