**Practical : 1**

**Aim:-Design a comprehensive data model for a business scenario**

* + **Objective:** Design a comprehensive data model for a business scenario.
  + **Tasks:**
    - **Conceptual Model:** Use Python to create a conceptual data model for a given business case (e.g., an e-commerce store).
    - **Physical Model:** Implement the physical model in MySQL.
  + **Tools:** Python (ERD libraries), MySQL Workbench
* **Level 1:**  
  Design a complete data model for an online retail business that includes customers, orders, products, inventory, suppliers, and payment transactions. Include normalization up to 3NF and consider relationships, constraints, and cardinality.
* **Level 2:**  
  Design a data model for a library management system including books, members, borrowing transactions, and fines. Focus on proper entity relationships and apply basic normalization.
* **Level 3:**  
  Create a simple data model to track students and courses in a small school. Include entities for students, courses, and enrolment with clear primary keys and foreign keys.

**Level :1**

Design a complete data model for an online retail business that includes customers, orders, products, inventory, suppliers, and payment transactions. Include normalization up to 3NF and consider relationships, constraints, and cardinality.

Design a complete data model for an online retail business:

* Customers, Orders, Products, Inventory, Suppliers, Payment Transactions

**Steps:**

1. **Identify Core Entities:**
   * Customers
   * Orders
   * Products
   * Inventory
   * Suppliers
   * Payments
2. **Define Relationships:**
   * A customer places many orders
   * Each order has multiple products (Order\_Items as bridge table)
   * Products are supplied by suppliers
   * Inventory tracks stock per product
   * Orders are linked to payment transactions
3. **Define Attributes (examples):**
   * **Customers:** Customer\_ID, Name, Email, Phone, Address
   * **Orders:** Order\_ID, Customer\_ID, Order\_Date, Status
   * **Order\_Items:** Order\_ID, Product\_ID, Quantity, Price
   * **Products:** Product\_ID, Name, Description, Supplier\_ID, Price
   * **Suppliers:** Supplier\_ID, Name, Contact\_Info
   * **Inventory:** Product\_ID, Quantity\_Available
   * **Payments:** Payment\_ID, Order\_ID, Amount, Payment\_Date, Method, Status
4. **Draw ERD** with cardinality, FK-PK relationships, and constraint details
5. **Normalize to 3NF:**
   * Remove derived and redundant attributes
   * Ensure atomic fields
   * Resolve transitive dependencies
6. **Implement using MySQL Workbench (DB + ERD sync)**
   * Create tables using **Forward Engineering**
7. **Add constraints:**
   * NOT NULL, UNIQUE, FOREIGN KEY, CHECK, DEFAULT where applicable
8. **Test with sample data entries**

Sql Workbench: <https://dev.mysql.com/downloads/installer/>