**Database Lab Project Report**

**Names:** Muhammad Yamin Tariq, Shaheer Ahsan

**Roll no:** 241840, 241882

**Github link:** <https://github.com/Susheer-hehe/Database-systems-project.git>

**Phase 1:** Database Schema Design

Develop an ER Diagram for the database.

A diagram of a diagram

AI-generated content may be incorrect.

# **Entities:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Customers** | **Employees** | **Stores** | **Product** | **Games** | **Orders** | **PC Parts** |
| **Inventory** | **Membership** | **Shipment** | **Sales** | **Rentals** | **Consoles** |  |

# **Attributes:**

**Customers:**

* customer\_id (PK)
* first\_name
* last\_name
* email
* phone\_number
* address
* membership\_id (FK to Membership)

**Employees:**

 employee\_id (PK)

 first\_name

 last\_name

 email

 phone\_number

 position

Salary

 store\_id (FK)

Stores:

 store\_id (PK)

 Location

 email

 established\_at

**Games:**

* game\_id (PK)
* product\_id (FK to Products)
* platform
* genre
* publisher
* release\_date

Orders:

 order\_id (PK)

 customer\_id (FK)

 order\_date

 payment\_method

 total\_amount

* product\_id (FK)
* quantity
* unit\_price

**Inveentory:**

 inventory\_id (PK)

 store\_id (FK)

 product\_id (FK)

 stock\_quantity

**Membership:**

 membership\_id (PK)

 membership\_name

 discount\_percent

 benefits\_description

**Shipment:**

 restock\_id (PK)

 store\_id (FK)

 product\_id (FK)

 supplier\_name

 supplier\_contact

 restock\_date

 quantity

 status

**Rentals:**

 rental\_id (PK)

 customer\_id (FK)

 product\_id (FK)

 rental\_date

 due\_date

 return\_date (nullable)

 rental\_fee

 is\_returned (boolean)

**Console:**

 console\_id (PK)

 product\_id (FK)

 brand

 model

 storage\_capacity

Price

**Product:**

 product\_id (PK)

 product\_name

 product\_type *(e.g., Game, Console, PC Part)*

 price

 digital (boolean)

 release\_date

**Sales:**

 sale\_id (PK)

 product\_id (FK)

 store\_id (FK)

 sale\_date

 quantity\_sold

 unit\_price

 discount\_applied (boolean)

 total\_sale\_amount

**PC Parts:**

 part\_id (PK)

 product\_id (FK)

 part\_type *(e.g., GPU, CPU, RAM)*

 brand

 model

 warranty\_period

|  |  |
| --- | --- |
| Customer ↔ Membership | Many-to-One |

|  |  |  |
| --- | --- | --- |
| Employee ↔ Store |  | Many-to-One |

|  |  |
| --- | --- |
| Orders ↔ Customers | Many-to-One |

|  |  |
| --- | --- |
| Orders ↔ OrderDetails (or directly Products) | One-to-Many |

|  |  |
| --- | --- |
| OrderDetails ↔ Products | Many-to-One |

|  |  |
| --- | --- |
| Inventory ↔ Store, Product | Many-to-One (both) |

|  |  |
| --- | --- |
| Games, Consoles, PC Parts ↔ Product | One-to-One |

|  |  |
| --- | --- |
| Rentals ↔ Customers, Products | Many-to-One (both) |

|  |  |
| --- | --- |
| Shipment ↔ Store, Product | Many-to-One (both) |

|  |  |
| --- | --- |
| Sales ↔ Product, Store | Many-to-One (both) |

**Explain Schema Design Decisions and Assumptions.**

**1. Entity Design & Specialization**

* **Products as a Central Entity**:
  + A base Product table stores common attributes (name, type, price, release date).
  + Specialized tables (Games, Consoles, PC\_Parts) extend Product with unique attributes (e.g., genre for games, storage\_capacity for consoles).
  + **Assumption**: All sellable items (physical/digital) share core attributes but require type-specific details.

**2. Store & Employee Management**

* **Stores as Independent Entities**:
  + Each Store has a unique ID, location, and manager (linked to Employees).
  + **Assumption**: Employees work at a single store (Many-to-One relationship).

**3. Customer & Membership Model**

* **Optional Memberships**:
  + Customers may have a Membership (discounts, rewards).
  + **Assumption**: Not all customers are members; memberships provide fixed benefits.

**4. Inventory & Supply Chain**

* **Per-Store Inventory Tracking**:
  + Inventory links Store and Product to track stock levels.
  + **Assumption**: Products can be restocked via Shipments from suppliers.

**5. Sales vs. Rentals**

* **Dual Business Model**:
  + Sales records permanent purchases.
  + Rentals track temporary loans (with due/return dates and fees).
  + **Assumption**: Products can be both sold and rented (e.g., games).

**6. Order Management**

* **Separate Orders & Order Items**:
  + Orders capture transaction summaries (customer, date, payment).
  + Order\_Items list individual products purchased.
  + **Assumption**: An order can include multiple products (One-to-Many).

**7. Pricing Strategy**

* **Dynamic Pricing in Transactions**:
  + Base prices are stored in Product, but actual sale/rental prices may vary (e.g., discounts).
  + **Assumption**: Final prices can differ from listed prices (e.g., promotions).

**8. Digital vs. Physical Products**

* **Digital Flag in Products**:
  + The digital attribute distinguishes downloadable items (no inventory needed).
  + **Assumption**: Digital products skip inventory/shipment tracking.

**9. Supplier Relationships**

* **Dedicated Suppliers Table**:
  + Normalized supplier data (instead of repeating in Shipment).
  + **Assumption**: Suppliers provide multiple products over time.

**10. Normalization Compliance**

* **3NF Enforcement**:
  + No transitive dependencies (e.g., Membership discounts are not duplicated).
  + Atomic attributes (e.g., splitting address into components was considered but deemed unnecessary for scope).

**Key Assumptions Summary**

1. Employees are assigned to one store.
2. Products can be sold, rented, or both.
3. Memberships provide predefined discounts.
4. Inventory is store-specific.
5. Prices at sale/rental may differ from base prices.
6. Digital products bypass physical logistics.