

MetroMart

Requirement Analysis:

We need to create a database schema design based on the following requirements of the **SUPERMARKET Database**:

- The supermarket consists of multiple **DEPARTMENTS** such as Personal Care, Home Care, Grocery, Restaurant Supplies and Houseware, IT, Stationary & Office furniture, Luggage, Apparel, Health, Kitchen and Home appliances.
Each department has a unique name and number, and one employee is assigned as its supervisor.
We keep track of the department's location within the store.
- The supermarket sells a variety of **PRODUCTS**.
Each product has a unique product code, a name, price, stock quantity, and belongs to exactly one department.
Each product also has a reorder level, below which it should be restocked.
- The supermarket procures products from multiple **SUPPLIERS**.
Each supplier has a unique ID, name, contact details, and address.
Suppliers can supply multiple products, and each product can have multiple suppliers.
- Products are procured via **SUPPLY ORDERS**.
Each supply order has a unique number, a supplier, order date, and status (e.g., Pending, Delivered).
Each supply order may contain several products, along with the quantity and cost price.
- The supermarket serves multiple **CUSTOMERS**.
Each customer has a unique ID, name, contact information, and loyalty points.
Customers may or may not be registered (walk-in customers have no loyalty points).
- The supermarket generates **SALES INVOICES** for customer purchases.
Each invoice has a unique number, date & time, employee who handled the sale, total amount, discount applied, and final payable amount.
- Each sales invoice may include multiple **PRODUCTS**, with details of quantity and selling price for each.
- **EMPLOYEES** work in the supermarket with unique IDs, names, roles (Cashier, Manager, Stockkeeper, etc.), salary, and contact details.
Each sales transaction is handled by exactly one employee.
Each department has one employee assigned as its supervisor.
- Customers may return items via **RETURNS**.
Each return is linked to an existing sales invoice and specifies the product, quantity returned, and reason for return.
- The database should keep track of **loyalty points** for customers based on their purchases and update them automatically.

Key Relationships

- A **Department** has many **Products**.
- A **Product** can be supplied by many **Suppliers**, and a **Supplier** can supply many **Products** (many-to-many via Supply Order Details).
- A **Customer** can make many **Sales Invoices**.
- A **Sales Invoice** can have many **Products** (many-to-many via Sales Details).
- An **Employee** can handle many sales but only supervise one department.
- A **Return** belongs to exactly one sales invoice and one product.