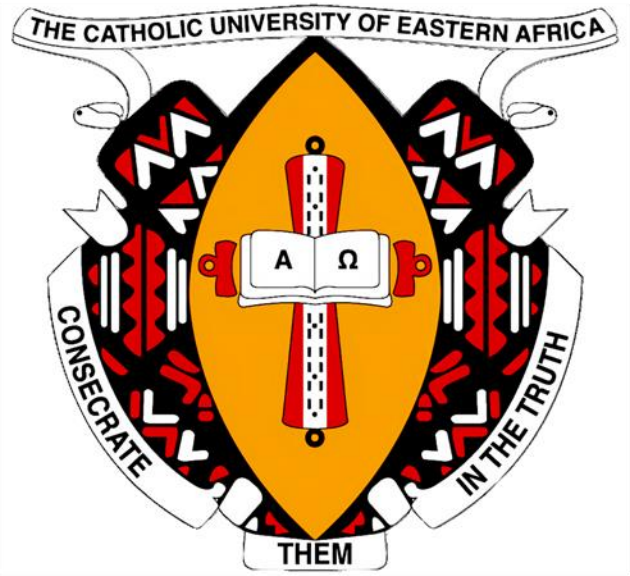
CATHOLIC UNIVERSITY OF EASTERN AFRICA

FACULTY OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE



UNIT CODE: CMT 302

UNIT TITLE: ADVANCED DATABASE SYSTEMS

MILESTONE 1: ER DIAGRAMS, SQL SCHEMA AND TABLE CREATION SCRIPTS

GROUP 17: SALES AND INVENTORY SYSTEM

GROUP MEMBERS:

1. **1049078**
2. **1049433**
3. **1048879**
4. **1049412**

Database Schema and Tables

1. categories Table

- **Purpose**: Stores information about different product categories, allowing products to be organized by type.
- **Columns**:
    - category_id - INT: Primary key for the table, uniquely identifies each category. It's set to auto-increment to ensure uniqueness.
    - name - VARCHAR(50): The name of the category (e.g., "Electronics" or "Furniture"). It is set to NOT NULL, meaning a name must be provided.

2. suppliers Table

- **Purpose**: Stores details of suppliers who provide products, enabling tracking of product origins.
- **Columns**:
    - supplier_id - INT: Primary key, auto-incremented for unique identification of each supplier.
    - name - VARCHAR(100): The name of the supplier, such as "ABC Electronics." This field is NOT NULL.
    - contact_info - VARCHAR(255): Optional contact information for the supplier, such as a phone number, email, or address.

3. products Table

- **Purpose**: Stores information about each product available in the inventory, along with category and supplier links.
- **Columns**:
    - product_id - INT: Primary key, auto-incremented to uniquely identify each product.
    - name - VARCHAR(100): The name of the product (e.g., "Laptop"). It is NOT NULL.
    - category_id - INT: Foreign key referencing categories(category_id), linking each product to a category. When the category is deleted, this value is set to NULL (ON DELETE SET NULL).
    - supplier_id - INT: Foreign key referencing suppliers(supplier_id), linking each product to a supplier. When the supplier is deleted, this value is set to NULL (ON DELETE SET NULL).

- price - DECIMAL(10, 2): Price of the product, with up to 10 digits and 2 decimal places (e.g., “49.99”). This is NOT NULL.
- created_at - TIMESTAMP: Automatically records when each product record was created, defaulting to the current timestamp.

4. inventory Table

- **Purpose**: Tracks the quantity of each product in stock, enabling inventory management.
- **Columns**:
  - inventory_id - INT: Primary key, auto-incremented to uniquely identify each inventory record.
  - product_id - INT: Foreign key referencing products(product_id), linking each inventory entry to a specific product. If a product is deleted, related inventory entries are also deleted (ON DELETE CASCADE).
  - quantity - INT: Quantity of the product available in stock, with a NOT NULL constraint to prevent undefined values.
  - last_updated - TIMESTAMP: Records the last time the inventory was updated. Automatically updates whenever the record is modified, allowing for tracking of inventory adjustments.

5. employees Table

- **Purpose**: Stores details of employees, providing information for tracking who handled each sale.
- **Columns**:
  - employee_id - INT: Primary key, auto-incremented to uniquely identify each employee.
  - first_name - VARCHAR(50): First name of the employee, with a NOT NULL constraint.
  - last_name - VARCHAR(50): Last name of the employee, also NOT NULL.
  - position - VARCHAR(50): Position or job title of the employee (e.g., “Sales Representative”), which is optional.

6. customers Table

- **Purpose**: Stores customer details, enabling sales records to be linked to specific customers.
- **Columns**:
    - customer_id - INT: Primary key, auto-incremented to uniquely identify each customer.
    - name - VARCHAR(100): Name of the customer, NOT NULL.
    - contact_info - VARCHAR(255): Optional contact information for the customer, such as a phone number or email address.

7. sales Table

- **Purpose**: Records each sale transaction, including details about the employee handling the sale, the customer, and the total amount.
- **Columns**:
    - sale_id - INT: Primary key, auto-incremented to uniquely identify each sale.
    - employee_id - INT: Foreign key referencing employees(employee_id), linking each sale to the employee who handled it. If an employee is deleted, this value is set to NULL (ON DELETE SET NULL).
    - customer_id - INT: Foreign key referencing customers(customer_id), linking each sale to a customer. If the customer is deleted, this value is set to NULL (ON DELETE SET NULL).
    - sale_date - TIMESTAMP: Automatically records the date and time of the sale, defaulting to the current timestamp.
    - total_amount - DECIMAL(10, 2): Total monetary amount for the sale, with up to 10 digits and 2 decimal places (e.g., “150.75”), NOT NULL.

8. sales_details Table

- **Purpose**: Stores detailed information for each item in a sale, including quantity and price per unit, linking to both the sales and products tables.
- **Columns**:
    - sale_detail_id - INT: Primary key, auto-incremented to uniquely identify each line item in the sale.
    - sale_id - INT: Foreign key referencing sales(sale_id), linking each detail line to a specific sale transaction. If the sale is deleted, related details are also deleted (ON DELETE CASCADE).
    - product_id - INT: Foreign key referencing products(product_id), linking each line item to a specific product. If the product is deleted, this value is set to NULL (ON DELETE SET NULL).
    - quantity - INT: Quantity of the product sold in the specific transaction, NOT NULL.
    - unit_price - DECIMAL(10, 2): Price per unit of the product for the sale, allowing precise pricing for each item within a sale, NOT NULL.

Table Name	Column Name	Data Type
categories	category_id	INT
	name	VARCHAR(50)
suppliers	supplier_id	INT
	name	VARCHAR(100)
	contact_info	VARCHAR(255)
products	product_id	INT
	name	VARCHAR(100)
	category_id	INT
	supplier_id	INT
	price	DECIMAL(10, 2)
	created_at	TIMESTAMP
inventory	inventory_id	INT
	product_id	INT
	quantity	INT
	last_updated	TIMESTAMP
employees	employee_id	INT
	first_name	VARCHAR(50)
	last_name	VARCHAR(50)
	position	VARCHAR(50)
customers	customer_id	INT
	name	VARCHAR(100)
	contact_info	VARCHAR(255)

sales	sale_id	INT
	employee_id	INT
	customer_id	INT
	sale_date	TIMESTAMP
	total_amount	DECIMAL(10, 2)
sales_details	sale_detail_id	INT
	sale_id	INT
	product_id	INT
	quantity	INT
	unit_price	DECIMAL(10, 2)

ERD DIAGRAM

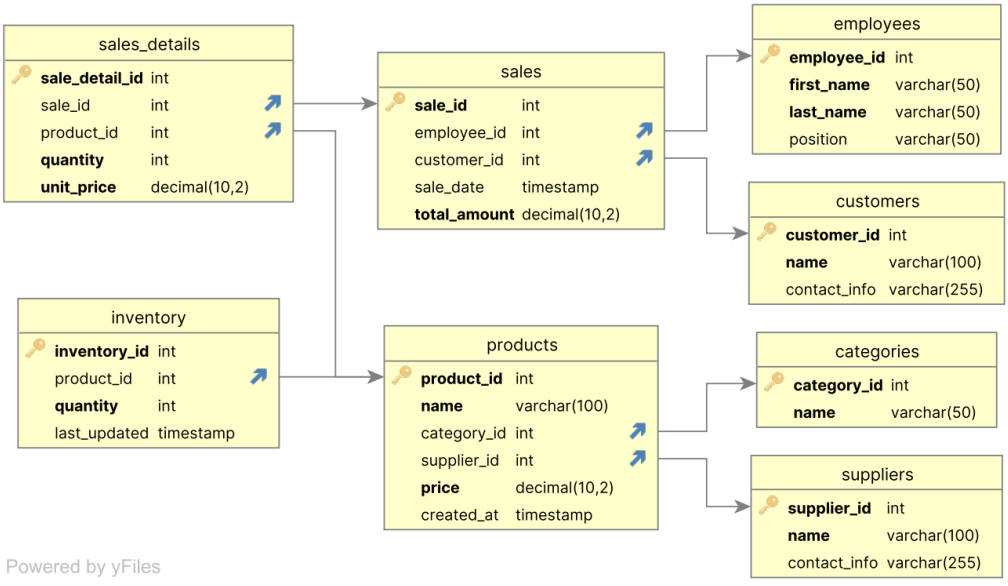


Table Creation Scripts

```
-- 1. Categories Table
CREATE TABLE categories (
    category_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(50) NOT NULL
);

-- 2. Suppliers Table
CREATE TABLE suppliers (
    supplier_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    contact_info VARCHAR(255)
);

-- 3. Products Table
CREATE TABLE products (
    product_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    category_id INT,
    supplier_id INT,
    price DECIMAL(10, 2) NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (category_id) REFERENCES categories(category_id) ON DELETE SET NULL,
    FOREIGN KEY (supplier_id) REFERENCES suppliers(supplier_id) ON DELETE SET NULL
);
```

```
-- 4. Inventory Table
CREATE TABLE inventory (
    inventory_id INT PRIMARY KEY AUTO_INCREMENT,
    product_id INT,
    quantity INT NOT NULL,
    last_updated TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
    FOREIGN KEY (product_id) REFERENCES products(product_id) ON DELETE CASCADE
);

-- 5. Employees Table
CREATE TABLE employees (
    employee_id INT PRIMARY KEY AUTO_INCREMENT,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    position VARCHAR(50)
);

-- 6. Customers Table
CREATE TABLE customers (
    customer_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    contact_info VARCHAR(255)
);
```

```
-- 7. Sales Table
CREATE TABLE sales (
    sale_id INT PRIMARY KEY AUTO_INCREMENT,
    employee_id INT,
    customer_id INT,
    sale_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    total_amount DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (employee_id) REFERENCES employees(employee_id) ON DELETE SET NULL,
    FOREIGN KEY (customer_id) REFERENCES customers(customer_id) ON DELETE SET NULL
);

-- 8. Sales Details Table
CREATE TABLE sales_details (
    sale_detail_id INT PRIMARY KEY AUTO_INCREMENT,
    sale_id INT,
    product_id INT,
    quantity INT NOT NULL,
    unit_price DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (sale_id) REFERENCES sales(sale_id) ON DELETE CASCADE,
    FOREIGN KEY (product_id) REFERENCES products(product_id) ON DELETE SET NULL
);
```