ABBOTTABAD UNIVERSITY OF SCIENCE & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE


# Project reports.

**Submitted By:**

**Names and roll numbers :**

**kinza sheraz 12124**

**Fatima Noor 12121**

**Quratulain sajid 12130**

**Mariam Noor 12127**

**Maham Nadeem 12126**

**Project Title : Inventory management system**

**Discipline : CS 4 A**

**Subject : Database**

**Submitted to: Ma’am Hadia**

**Introduction :**

**# Project Report: Inventory Management System Using SQL**

# Introduction

**The Inventory Management System is a crucial tool for tracking and managing products within a database. This system involves creating a database, defining tables for categories, products, and sales, and performing various operations such as inserting data, updating product information, recording sales, and generating reports. This report details the design and implementation of an SQL-based Inventory Management System.**

## Project Goals

- **Primary Goals**:
  - **Streamline Inventory Management**: Implement a structured approach to handle inventory data.
  - **Efficient Sales Tracking**: Record and manage sales transactions systematically.
  - **Maintain Stock Levels**: Ensure accurate tracking of stock availability and updates.
  - **Data Integrity**: Enforce consistency and reliability through referential integrity and validation checks.

## Advantages of the Project

- **Enhanced Efficiency**:
  - **Automated Processes**: Reduces manual intervention through automated updates and stock adjustments.
  - **Quick Data Retrieval**: Facilitates rapid access to inventory and sales data.
- **Improved Accuracy**:
  - **Consistency**: Ensures data consistency through foreign key constraints and validation rules.
  - **Accurate Reporting**: Provides precise and up-to-date reports on inventory and sales.

**Better Decision-Making**:

- **Informed Insights**: Enables data-driven decisions by offering detailed insights into inventory and sales trends.
- **Real-Time Updates**: Keeps inventory data current, aiding in timely decisions.

## Key Components of the Database

- **Categories Table**: Holds information about product categories.
  - **Fields**: Category_id, CategoryName
- **Product Table**: Stores details about individual products.
  - **Fields**: Product_id, Product_Name, Category_id, Unit_price, Units_instock
- **Sales Table**: Records sales transactions.
  - **Fields**: SaleID, Product_id, SaleDate, QuantitySold

# Database Creation

# Creating the Database

### The first step involves creating the database

### `InventoryDB`.

```sql

CREATE DATABASE InventoryDB;

USE InventoryDB;

```

# Creating Tables

Three tables are created to manage categories, products, and sales.

1. **Categories Table**: Stores category information.

2. **Product Table**: Stores product details, including a foreign key to the `Categories` table.

3. **Sales Table**: Records sales transactions, with a foreign key linking to the `Product` table.

```sql
CREATE TABLE Categories (
    Category_id INT PRIMARY KEY,
    CategoryName VARCHAR(50)
);

CREATE TABLE Product (
    Product_id INT PRIMARY KEY,
    Product_Name VARCHAR(55),
    Category_id INT,
    Unit_price DECIMAL(10,2),
    Units_instock INT,
    FOREIGN KEY (Category_id) REFERENCES
Categories(Category_id)
);

CREATE TABLE Sales (
    SaleID INT PRIMARY KEY,
```

Product_id INT,

SaleDate DATE,

QuantitySold INT,

FOREIGN KEY (Product_id) REFERENCES Product(Product_id)

);

```

# Inserting Data

## Inserting Categories

Insert initial data into the `Categories` table to define various product categories.

```sql

```
INSERT INTO Categories (Category_id,
CategoryName) VALUES
    (101, 'Electronics'), (102, 'Mobile Devices'), (103,
'Audio Accessories'),
    (104, 'Televisions'), (105, 'Cameras'), (106,
'Appliances'),
    (107, 'Home Cleaning'), (108, 'Kitchen Appliances'),
(109, 'Personal Care'),
    (110, 'Laundry Appliances'), (111, 'Furniture'), (112,
'Office Furniture'),
    (113, 'Home Decor'), (114, 'Wall Art'), (115,
'Outdoor Living'),
    (116, 'Sports Equipment'), (117, 'Sporting Goods'),
(118, 'Accessories'),
    (119, 'Stationery'), (120, 'Computer Accessories'),
(121, 'Storage Devices'),
    (122, 'Mobile Accessories'), (123, 'Kitchenware'),
(124, 'Dining & Serving'),
```

(125, 'Bathroom Accessories'), (126, 'Cleaning Supplies'), (127, 'Pet Supplies'),

(128, 'Gaming'), (129, 'Health & Fitness'), (130, 'First Aid & Medical'),

(131, 'Personal Care & Beauty'), (132, 'Weather Gear'), (133, 'Travel Accessories');

```

### Inserting Products

Insert product data into the `Product` table, linking each product to a category.

```sql

INSERT INTO Product (Product_id, Product_Name, Category_id, Unit_price, Units_instock) VALUES

(1, 'Laptop', 101, 1200.00, 50), (2, 'Smartphone', 102, 800.00, 100),

(3, 'Tablet', 102, 500.00, 75), (4, 'Desktop Computer', 101, 1500.00, 30),

(5, 'Headphones', 103, 100.00, 200), (6, 'Portable Speaker', 103, 150.00, 150),

(7, 'Television', 104, 2000.00, 40), (8, 'Digital Camera', 105, 600.00, 80),

(9, 'Microwave Oven', 106, 300.00, 120), (10, 'Refrigerator', 106, 1200.00, 60),

-- Additional products here

(100, 'Travel Pillow', 133, 15.00, 300);

```

# Updating Data

### Updating Product Information
```

**Various SQL queries are used to update product information, including updating unit prices and stock levels.**

# Updating Unit Price

**Reduce the unit price of electronics and mobile devices by 5%.**

**```sql**

**UPDATE Product**

**SET Unit_price = Unit_price * 0.95**

**WHERE Category_id = 102 OR Category_id = 101;**

**```**

**#### Updating Stock Levels**

**Decrease the stock levels by 5 units for categories 101 and 102 if the stock is sufficient.**

```sql

UPDATE Product

SET Units_instock = Units_instock - 5

WHERE (Category_id = 101 OR Category_id = 102)

AND Units_instock >= 5;

```

# Renaming a Product

Change the name of 'Desktop Computer' to 'PC'.

```sql

SET SQL_SAFE_UPDATES = 0;

UPDATE Product

SET Product_Name = "PC"

WHERE Product_Name = "Desktop Computer";

```

# Deleting a Product

**Delete the product 'Thermometer' from the `Product` table.**

**```sql**

**DELETE FROM Product**

**WHERE Product_Name = "Thermometer";**

**```**

# Recording Sales

# Inserting Sales Data

**Insert sample sales data into the `Sales` table.**

**```sql**

**INSERT INTO Sales (SaleID, Product_id, SaleDate, QuantitySold) VALUES**

**(1, 1, '2024-01-01', 3), (2, 5, '2024-01-02', 2), (3, 10, '2024-01-03', 1),**

**-- Additional sales data here**

**(50, 71, '2024-02-19', 1);**

**```**

# Managing Inventory

**### Updating Stock Based on Sales**

**Use a join operation to update the stock levels based on the sales data.**

**```sql**

**UPDATE Product**

```
LEFT JOIN Sales ON Product.Product_id =
Sales.Product_id
SET Product.Units_instock =
CASE
WHEN Sales.QuantitySold IS NOT NULL THEN
Product.Units_instock - Sales.QuantitySold
ELSE Product.Units_instock
END;
```

# Viewing Updated Inventory

Display the updated product information.

```sql
SELECT * FROM Product;
```

# Additional Tasks

## Data Integrity

**Ensure data integrity by implementing foreign key constraints between `Products.CategoryID` and `Categories.CategoryID`.**

## Data Validation

**Include basic data validation checks such as ensuring unit price is positive and quantity sold is non-negative.**

## Generating Reports

**Create SQL queries to generate reports such as total sales by product or category and current inventory status.**

# Documentation

**Document the database schema, explaining the purpose of each table and its columns. Include sample queries for basic operations and reports.**

# Testing

**Thoroughly test SQL queries to ensure they perform as expected and handle edge cases gracefully, such as attempting to sell more units than available in stock.**

# Conclusion

**The Inventory Management System designed using SQL provides a robust framework for managing product information, recording sales, and generating reports. The database schema and SQL queries ensure data integrity and facilitate efficient inventory management.**

```
CREATE DATABASE InventoryDB;
USE InventoryDB;

CREATE TABLE Categories(Category_id INT PRIMARY KEY,
CategoryName VARCHAR (50));

CREATE TABLE Product(Product_id INT PRIMARY KEY,
Product_Name VARCHAR(55), Category_id INT, Unit_price
DECIMAL(10,2), Units_instock INT,
FOREIGN KEY (Category_id) REFERENCES
Categories(Category_id));
```

```
CREATE TABLE Sales(SaleID INT PRIMARY KEY, Product_id INT,
SaleDate DATE, QuantitySold INT, FOREIGN KEY (Product_id)
REFERENCES Product(Product_id));

INSERT INTO Categories(Category_id, CategoryName) VALUES
(101, 'Electronics'),
(102, 'Mobile Devices'),
(103, 'Audio Accessories'),
(104, 'Televisions'),
(105, 'Cameras'),
(106, 'Appliances'),
(107, 'Home Cleaning'),
(108, 'Kitchen Appliances'),
(109, 'Personal Care'),
(110, 'Laundry Appliances'),
(111, 'Furniture'),
(112, 'Office Furniture'),
(113, 'Home Decor'),
(114, 'Wall Art'),
```

```
(115, 'Outdoor Living'),
(116, 'Sports Equipment'),
(117, 'Sporting Goods'),
(118, 'Accessories'),
(119, 'Stationery'),
(120, 'Computer Accessories'),
(121, 'Storage Devices'),
(122, 'Mobile Accessories'),
(123, 'Kitchenware'),
(124, 'Dining & Serving'),
(125, 'Bathroom Accessories'),
(126, 'Cleaning Supplies'),
(127, 'Pet Supplies'),
(128, 'Gaming'),
(129, 'Health & Fitness'),
(130, 'First Aid & Medical'),
(131, 'Personal Care & Beauty'),
(132, 'Weather Gear'),
(133, 'Travel Accessories');


INSERT INTO Product(Product_id, Product_Name, Category_id,
Unit_price, Units_instock) VALUES
(1, 'Laptop', 101, 1200.00, 50),
(2, 'Smartphone', 102, 800.00, 100),
(3, 'Tablet', 102, 500.00, 75),
(4, 'Desktop Computer', 101, 1500.00, 30),
(5, 'Headphones', 103, 100.00, 200),
(6, 'Portable Speaker', 103, 150.00, 150),
(7, 'Television', 104, 2000.00, 40),
(8, 'Digital Camera', 105, 600.00, 80),
(9, 'Microwave Oven', 106, 300.00, 120),
(10, 'Refrigerator', 106, 1200.00, 60),
(11, 'Vacuum Cleaner', 107, 250.00, 100),
(12, 'Toaster', 108, 50.00, 150),
```

(13, 'Coffee Maker', 108, 100.00, 100),
(14, 'Blender', 108, 80.00, 120),
(15, 'Electric Kettle', 108, 40.00, 200),
(16, 'Iron', 109, 30.00, 180),
(17, 'Hair Dryer', 109, 25.00, 250),
(18, 'Washing Machine', 110, 800.00, 70),
(19, 'Dishwasher', 110, 700.00, 60),
(20, 'Sofa', 111, 600.00, 25),
(21, 'Bed', 111, 800.00, 20),
(22, 'Dining Table', 111, 400.00, 40),
(23, 'Office Chair', 112, 150.00, 80),
(24, 'Bookshelf', 112, 200.00, 60),
(25, 'Curtains', 113, 50.00, 300),
(26, 'Carpet', 113, 80.00, 150),
(27, 'Wall Clock', 114, 20.00, 500),
(28, 'Painting', 114, 300.00, 30),
(29, 'Outdoor Grill', 115, 150.00, 70),
(30, 'Garden Hose', 115, 30.00, 200),
(31, 'Patio Furniture Set', 115, 800.00, 15),
(32, 'Bicycle', 116, 400.00, 50),
(33, 'Treadmill', 116, 1000.00, 20),
(34, 'Dumbbell Set', 116, 150.00, 100),
(35, 'Yoga Mat', 116, 20.00, 150),
(36, 'Basketball', 117, 30.00, 80),
(37, 'Football', 117, 25.00, 100),
(38, 'Golf Clubs', 117, 200.00, 40),
(39, 'Tennis Racket', 117, 80.00, 60),
(40, 'Running Shoes', 118, 100.00, 150),
(41, 'Backpack', 118, 50.00, 200),
(42, 'Sunglasses', 118, 30.00, 250),
(43, 'Watch', 118, 150.00, 100),
(44, 'Fountain Pen', 119, 20.00, 300),
(45, 'Notebook', 119, 5.00, 500),
(46, 'Desk Lamp', 119, 40.00, 150),
(47, 'Calculator', 119, 15.00, 200),

(48, 'Printer', 120, 200.00, 50),
(49, 'Scanner', 120, 150.00, 80),
(50, 'Projector', 120, 500.00, 25),
(51, 'External Hard Drive', 121, 100.00, 100),
(52, 'USB Flash Drive', 121, 20.00, 300),
(53, 'Keyboard', 121, 50.00, 150),
(54, 'Mouse', 121, 30.00, 200),
(55, 'Power Bank', 122, 40.00, 200),
(56, 'Charger Cable', 122, 10.00, 300),
(57, 'Wireless Earbuds', 122, 80.00, 150),
(58, 'Phone Case', 122, 15.00, 250),
(59, 'Kitchen Knife Set', 123, 50.00, 100),
(60, 'Cutting Board', 123, 20.00, 150),
(61, 'Cookware Set', 123, 100.00, 80),
(62, 'Blender', 123, 80.00, 120),
(63, 'Tea Set', 124, 30.00, 200),
(64, 'Coffee Mug', 124, 10.00, 300),
(65, 'Water Pitcher', 124, 20.00, 250),
(66, 'Dinnerware Set', 124, 150.00, 100),
(67, 'Bathroom Towel Set', 125, 40.00, 150),
(68, 'Shower Curtain', 125, 15.00, 250),
(69, 'Toothbrush Holder', 125, 10.00, 300),
(70, 'Soap Dispenser', 125, 12.00, 200),
(71, 'Laundry Detergent', 126, 20.00, 250),
(72, 'Fabric Softener', 126, 15.00, 200),
(73, 'Bleach', 126, 10.00, 300),
(74, 'Cleaning Wipes', 126, 8.00, 400),
(75, 'Dog Food', 127, 30.00, 150),
(76, 'Cat Litter', 127, 25.00, 200),
(77, 'Pet Bed', 127, 50.00, 100),
(78, 'Fish Tank', 127, 80.00, 75),
(79, 'Gaming Console', 128, 400.00, 50),
(80, 'Video Games', 128, 60.00, 200),
(81, 'Gaming Headset', 128, 80.00, 150),
(82, 'Gaming Chair', 128, 200.00, 80),

```
(83, 'Digital Watch', 129, 150.00, 100),
(84, 'Fitness Tracker', 129, 100.00, 120),
(85, 'Smart Scale', 129, 50.00, 150),
(86, 'Bluetooth Earphones', 129, 70.00, 180),
(87, 'First Aid Kit', 130, 30.00, 200),
(88, 'Thermometer', 130, 10.00, 300),
(89, 'Pain Reliever', 130, 15.00, 250),
(90, 'Bandages', 130, 8.00, 400),
(91, 'Sunscreen', 131, 12.00, 200),
(92, 'Moisturizer', 131, 15.00, 250),
(93, 'Facial Cleanser', 131, 10.00, 300),
(94, 'Shampoo', 131, 8.00, 400),
(95, 'Umbrella', 132, 20.00, 300),
(96, 'Raincoat', 132, 40.00, 150),
(97, 'Winter Jacket', 132, 80.00, 100),
(98, 'Gloves', 132, 10.00, 500),
(99, 'Backpack', 133, 50.00, 200),
(100, 'Travel Pillow', 133, 15.00, 300);

-- Update Product: SQL queries to update product information
(e.g., update unit price or units in stock).
        -- Update based on the Unit Price
UPDATE Product
SET Unit_price = Unit_price * 0.95 -- Unit_Price decrease of 5%
for Electronics and Mobile Devices due to New Year Offers.
WHERE Category_id = 102 OR Category_id = 101;

-- Update based on the Unit Stock after New Year
UPDATE Product
SET Units_instock = Units_instock - 5
WHERE Category_id = 101 OR Category_id = 102 AND
Units_instock >= 5;

-- Update the Product Name (Desktop Computer) to Personal
Computer
```

```
SET SQL_SAFE_UPDATES = 0;
UPDATE Product
SET Product_Name = "PC"
WHERE Product_Name = "Desktop Computer";

-- Delete Product: SQL queries to delete a product
(Thermometer) from the Products table.
DELETE FROM Product
WHERE Product_Name = "Thermometer";

-- Record Sales: SQL queries to record product sales in the
Sales table.
        -- Inserting sample sales data into the Sales table
INSERT INTO Sales(SaleID, Product_id, SaleDate, QuantitySold)
VALUES
(1, 1, '2024-01-01', 3), -- Laptop sold on January 1, 2024, 3
units
(2, 5, '2024-01-02', 2), -- Headphones sold on January 2, 2024,
2 units
(3, 10, '2024-01-03', 1), -- Refrigerator sold on January 3, 2024,
1 unit
(4, 17, '2024-01-04', 4), -- Hair Dryer sold on January 4, 2024, 4
units
(5, 22, '2024-01-05', 2), -- Dining Table sold on January 5, 2024,
2 units
(6, 31, '2024-01-06', 1), -- Patio Furniture Set sold on January 6,
2024, 1 unit
(7, 40, '2024-01-07', 3), -- Running Shoes sold on January 7,
2024, 3 units
(8, 48, '2024-01-08', 2), -- Printer sold on January 8, 2024, 2
units
(9, 55, '2024-01-09', 5), -- Power Bank sold on January 9, 2024,
5 units
(10, 63, '2024-01-10', 1), -- Tea Set sold on January 10, 2024, 1
unit
```

**(11, 71, '2024-01-11', 4), -- Laundry Detergent sold on January 11, 2024, 4 units**
**(12, 79, '2024-01-12', 2), -- Gaming Console sold on January 12, 2024, 2 units**
**(13, 86, '2024-01-13', 3), -- Bluetooth Earphones sold on January 13, 2024, 3 units**
**(14, 94, '2024-01-14', 1), -- Shampoo sold on January 14, 2024, 1 unit**
**(15, 100, '2024-01-15', 2), -- Travel Pillow sold on January 15, 2024, 2 units**
**(16, 5, '2024-01-16', 3), -- Headphones sold on January 16, 2024, 3 units**
**(17, 17, '2024-01-17', 2), -- Hair Dryer sold on January 17, 2024, 2 units**
**(18, 22, '2024-01-18', 1), -- Dining Table sold on January 18, 2024, 1 unit**
**(19, 31, '2024-01-19', 4), -- Patio Furniture Set sold on January 19, 2024, 4 units**
**(20, 40, '2024-01-20', 2), -- Running Shoes sold on January 20, 2024, 2 units**
**(21, 48, '2024-01-21', 1), -- Printer sold on January 21, 2024, 1 unit**
**(22, 55, '2024-01-22', 3), -- Power Bank sold on January 22, 2024, 3 units**
**(23, 63, '2024-01-23', 2), -- Tea Set sold on January 23, 2024, 2 units**
**(24, 71, '2024-01-24', 1), -- Laundry Detergent sold on January 24, 2024, 1 unit**
**(25, 79, '2024-01-25', 5), -- Gaming Console sold on January 25, 2024, 5 units**
**(26, 86, '2024-01-26', 2), -- Bluetooth Earphones sold on January 26, 2024, 2 units**
**(27, 94, '2024-01-27', 3), -- Shampoo sold on January 27, 2024, 3 units**

**(28, 100, '2024-01-28', 1), -- Travel Pillow sold on January 28, 2024, 1 unit**
**(29, 5, '2024-01-29', 2), -- Headphones sold on January 29, 2024, 2 units**
**(30, 17, '2024-01-30', 3), -- Hair Dryer sold on January 30, 2024, 3 units**
**(31, 22, '2024-01-31', 1), -- Dining Table sold on January 31, 2024, 1 unit**
**(32, 31, '2024-02-01', 4), -- Patio Furniture Set sold on February 1, 2024, 4 units**
**(33, 40, '2024-02-02', 2), -- Running Shoes sold on February 2, 2024, 2 units**
**(34, 48, '2024-02-03', 1), -- Printer sold on February 3, 2024, 1 unit**
**(35, 55, '2024-02-04', 5), -- Power Bank sold on February 4, 2024, 5 units**
**(36, 63, '2024-02-05', 2), -- Tea Set sold on February 5, 2024, 2 units**
**(37, 71, '2024-02-06', 3), -- Laundry Detergent sold on February 6, 2024, 3 units**
**(38, 79, '2024-02-07', 1), -- Gaming Console sold on February 7, 2024, 1 unit**
**(39, 86, '2024-02-08', 4), -- Bluetooth Earphones sold on February 8, 2024, 4 units**
**(40, 94, '2024-02-09', 2), -- Shampoo sold on February 9, 2024, 2 units**
**(41, 100, '2024-02-10', 1), -- Travel Pillow sold on February 10, 2024, 1 unit**
**(42, 5, '2024-02-11', 3), -- Headphones sold on February 11, 2024, 3 units**
**(43, 17, '2024-02-12', 2), -- Hair Dryer sold on February 12, 2024, 2 units**
**(44, 22, '2024-02-13', 1), -- Dining Table sold on February 13, 2024, 1 unit**

```
(45, 31, '2024-02-14', 4), -- Patio Furniture Set sold on February 14, 2024, 4 units
(46, 40, '2024-02-15', 2), -- Running Shoes sold on February 15, 2024, 2 units
(47, 48, '2024-02-16', 1), -- Printer sold on February 16, 2024, 1 unit
(48, 55, '2024-02-17', 3), -- Power Bank sold on February 17, 2024, 3 units
(49, 63, '2024-02-18', 2), -- Tea Set sold on February 18, 2024, 2 units
(50, 71, '2024-02-19', 1); -- Laundry Detergent sold on February 19, 2024, 1 unit


-- Using Join Operation and CASE(WHEN, ELSE) to update the Units_instock in the Products table based on the QuantitySold in the Sales table.
UPDATE Product
LEFT JOIN Sales ON Product.Product_id = Sales.Product_id
SET Product.Units_instock =
        CASE
                WHEN  Sales.QuantitySold IS NOT NULL THEN Product.Units_instock - Sales.QuantitySold
                ELSE Product.Units_instock
        END;

SELECT * FROM Product

/* Remaning Work to Complete
Data Integrity: Implement data integrity constraints such as foreign key constraints between Products.CategoryID and Categories.CategoryID.
Data Validation: Include basic data validation checks (e.g., ensuring unit price is positive, quantity sold is non-negative).
```

**Reports: Create SQL queries to generate basic reports such as total sales by product or category, current inventory status, etc.**
**Documentation: Provide documentation for your database schema, explaining the purpose of each table and its columns, along with sample queries for basic operations and reports.**
**Testing: Test your SQL queries thoroughly to ensure they perform as expected and handle edge cases gracefully (e.g., attempting to sell more units than available in stock).*/**