Project_2| Simple Inventory Management System

An inventory management system is a software application used to track inventory levels, orders, sales, and deliveries. The system helps businesses to manage their inventory efficiently and avoid product overstock and outages. In this project, we will create a simple inventory management system using MySQL to store data related to products and orders.

The system will allow users to add, update, and delete products, as well as place orders and keep track of the inventory status. This project can serve as a starting point for developing a more comprehensive inventory management system tailored to specific business needs.

Step-1

Create a new database named "inventory_db" to store all the tables related to the inventory management system.

```
CREATE DATABASE inventory_db;
```

Step-2

Create a new table named "products" with the following columns

```
CREATE TABLE inventory_db.products (

   id INT PRIMARY KEY AUTO_INCREMENT,

   product_name VARCHAR(50),

   description VARCHAR(100),

   category VARCHAR(50),

   quantity INT,

   price DECIMAL(10,2));
```

Step-3

Insert some sample data into the "products" table. Here's some sample data:

```
INSERT INTO inventory_db.products (product_name, description, category, quantity, price)

VALUES

   ('Laptop', 'HP Envy x360', 'Electronics', 10, 8000.00),

   ('Chair', 'Office chair', 'Furniture', 20, 1200.00),

   ('Mouse', 'Wireless mouse', 'Electronics', 30, 2500.00),

   ('Table', 'Coffee table', 'Furniture', 5, 3000.00);
```

Step-4 / Retrieving Data

- Select Quantity and price from the Products Table
  ```
  SELECT quantity, price FROM products;
  ```
- Select Product_name, Quantity,Price, and Total_gain from the Products Table
  ```
  SELECT product_name, quantity, price, (quantity * price) as Total_gain FROM products;
  ```
- Select Product_name, Quantity,Price, and Total_gain>5000 from the Products Table
  ```
  SELECT product_name, quantity, price, (quantity * price) as Total_gain FROM products
  where Total_gain>5000;
  ```
- Select Product_name, Quantity,Price and order by price from the Products Table
  ```
  SELECT product_name, quantity, price,   FROM products where price>500 order by
  price;
  ```
- Select Product_name, Quantity,Price group by Catogary from the Products Table
  ```
  SELECT product_name, quantity, price,   FROM products where price>500 group by
  catogary;
  ```

Step-4 | Retrieving Data

- COUNT: To count the number of rows in a table, you can use the COUNT function with the asterisk symbol (*). For example, to count the number of products in the "products" table, you can use the following query:
  SELECT COUNT(product_name) FROM products;

- AVG: To calculate the average of a column in a table, you can use the AVG function. For example, to calculate the average price of products in the "products" table, you can use the following query:
  SELECT AVG(price) FROM products;

- SUM: To calculate the sum of a column in a table, you can use the SUM function. For example, to calculate the total quantity of products in the "products" table, you can use the following query:
  SELECT SUM(quantity) FROM products;

- MIN: To find the minimum value of a column in a table, you can use the MIN function. For example, to find the minimum price of products in the "products" table, you can use the following query:
  SELECT MIN(price) FROM products;

- MAX: To find the maximum value of a column in a table, you can use the MAX function. For example, to find the maximum price of products in the "products" table, you can use the following query:
  SELECT MAX(price) FROM products;