Inventory Management Using SQL

By Ashraf Ahmad Khan

Total Number of Products in Inventory:

```
SELECT COUNT(*) AS Total_Number_Of_Products

FROM Inventory;
```

Unique Product Categories Offered by the Company:

```
SELECT DISTINCT Category

FROM Products;
```

Average Unit Price of All Products:

```
SELECT AVG(Unit_Price) AS Average_Unit_Price

FROM Products;
```

Total Number of Delivered Orders:

```
SELECT COUNT(*) AS Delivered_Orders

FROM Orders

WHERE Order_Status = 'Delivered';
```

Total Number of Suppliers:

```
SELECT COUNT(*) AS Number_Of_Suppliers

FROM Supplier;
```

Products with Unit Price Above ₹350,000:

```sql
SELECT Product_Name

FROM Products

WHERE Unit_Price > 350000;
```

Highest and Lowest Unit Price:

```sql
SELECT MAX(Unit_Price) AS Highest_Unit_Price, MIN(Unit_Price) AS

Lowest_Unit_Price

FROM Products;
```

Total Number of Pending Orders:

```sql
SELECT COUNT(*) AS Pending_Orders

FROM Orders

WHERE Order_Status = 'Pending';
```

Number of Products in the 'Electronics' Category:

```sql
SELECT COUNT(*) AS Number_Of_Electronics_Category_Product

FROM Products

WHERE Category = 'Electronics';
```

List of Supplier Names and Contact Details:

```sql
SELECT Supplier_Name, Contact_Email, Contact_Phone

FROM Supplier;
```

Products with Reorder Points Between 2 and 8:

```
SELECT A.Product_Name, A.Category, B.Inventory_ID, B.Reorder_Point

FROM Products AS A

JOIN Inventory AS B ON A.Product_ID = B.Product_ID

WHERE B.Reorder_Point BETWEEN 2 AND 8;
```

Total Revenue Generated from All Orders:

```
SELECT A.Product_ID, A.Product_Name, B.Quantity * B.Unit_Price AS

Total_Amount

FROM Products AS A

JOIN Order_Items AS B ON A.Product_ID = B.Product_ID;
```

Product Category with the Highest Revenue:

```
SELECT B.Category, SUM(A.Quantity * A.Unit_Price) AS Total_Revenue

FROM Order_Items AS A

JOIN Products AS B ON A.Product_ID = B.Product_ID

GROUP BY B.Category;
```

Losses Incurred on Cancelled Orders:

```
SELECT C.Product_Name, C.Category, B.Quantity * B.Unit_Price AS

Total_Amount

FROM Orders AS A

JOIN Order_Items AS B ON A.Order_ID = B.Order_ID

JOIN Products AS C ON B.Product_ID = C.Product_ID

WHERE A.Order_Status = 'Cancelled';
```

Number of Orders Placed in August:

```
SELECT COUNT(*) AS Orders_Count

FROM Orders

WHERE MONTH(Order_Date) = 8;
```

Classify Products Based on Unit Price:

```
SELECT Product_ID, Product_Name, Unit_Price,
    CASE
        WHEN Unit_Price > 45000 THEN 'Premium'
        WHEN Unit_Price > 30000 THEN 'High'
        WHEN Unit_Price > 15000 THEN 'Medium'
        ELSE 'Low'
    END AS Price_Category
FROM Products;
```

Supplier Distribution Across All Products:

```
SELECT A.Product_ID, A.Product_Name, A.Category, B.*

FROM Products AS A

LEFT JOIN Supplier AS B ON A.Product_ID = B.Product_ID;
```

Total Sales Revenue by Category Above ₹200,000:

```
SELECT A.Category, SUM(B.Quantity * B.Unit_Price) AS Total_Sales

FROM Products AS A

JOIN Order_Items AS B ON A.Product_ID = B.Product_ID

GROUP BY A.Category

HAVING SUM(B.Quantity * B.Unit_Price) > 200000;
```