

**SVKM's NMIMS**  
**School of Technology Management & Engineering, Navi Mumbai**  
A.Y. 2023 - 24  
**Course: Database Management Systems**

**Project Report**

|                                |                             |                  |
|--------------------------------|-----------------------------|------------------|
| Program                        | B Tech Computer Engineering |                  |
| Semester                       | IV                          |                  |
| Name of the Project:           | Integrated Components       |                  |
|                                |                             |                  |
| Details of Project Members     |                             |                  |
| Batch                          | Roll No.                    | Name             |
| B2                             | A100                        | Nishant Chanchad |
| B2                             | A115                        | Hiral Gundecha   |
|                                |                             |                  |
| Date of Submission: 01.04.2024 |                             |                  |

**Contribution of each project Members:**

| Roll No. | Name:            | Contribution              |
|----------|------------------|---------------------------|
| A100     | Nishant Chanchad | Database, queries, report |
| A115     | Hiral Gundecha   | Database, queries, report |

# **Project Report**

## **Integrated Components**

**By**

**Nishant Chanchad, Roll no.: A100**

**Hiral Gundecha, Roll no.: A115**

**Course: DBMS**

**AY: 2023-24**

## Table of Contents

| <b>Sr no.</b> | <b>Topic</b>                   | <b>Page no.</b> |
|---------------|--------------------------------|-----------------|
| <b>1</b>      | Storyline                      | 4               |
| <b>2</b>      | Components of Database Design  | 5               |
| <b>3</b>      | Relational Model               | 10              |
| <b>4</b>      | Normalization                  | 11              |
| <b>5</b>      | SQL Queries                    | 12              |
| <b>6</b>      | Project Demonstration          | 23              |
| <b>7</b>      | Self-learning beyond classroom | 36              |
| <b>8</b>      | Learning from the project      | 37              |
| <b>9</b>      | Challenges faced               | 38              |
| <b>10</b>     | Conclusion                     | 39              |

## Storyline

A company named ChipBits, which specialized in electronics and gadgets, formerly existed in a busy city. They had trouble efficiently managing their inventories because of their constantly growing product line. In order to simplify their business processes. The CEO, realized they needed a strong database system.

Inventory management presented the first difficulty. Products, suppliers, and stock levels all required to be monitored. Name, price, and expiration date were among the distinctive features of each product. These products were supplied by suppliers, whose contact information was kept on file for convenience. Having inventory records that showed the amount in stock, location, and date of last update was essential. Every item that entered and exited the inventory was recorded in the transactions.

They added production lines to the system to improve production planning. These lines were necessary for the effective production of a variety of goods. Having a name and description for every line made the production process easier to manage.

Another essential component was order tracking. Consumers placed purchases for the devices of their choice, and order information was noted, including dates, total amounts, and payment statuses. Order items delineated the products included in each order, and order status monitored the order's journey from order placement to delivery.

For a business to run smoothly, employee management was essential. Employee information was kept in the database, including positions, departments, and pay. The division of departments into categories ensured that the business was well organized.

Maintaining customer satisfaction required strict adherence to quality control. Products underwent routine quality testing to make sure they adhered to requirements. These checks were the responsibility of the inspectors, and their specifics were documented for accountability.

Lastly, analytics and reporting offered insightful information for making decisions. The database produced reports with analytics on sales, inventories, and production effectiveness. These reports were useful for seeing patterns and streamlining processes.

With these interconnected parts in place, ChipBits was able to effectively manage personnel, uphold quality standards, expedite production, track orders with ease, manage inventory, and make judgements based on predictive analytics. Due to their inventive devices and electronics, they were able to maintain their success in the industry.

# **Components of Database Design**

All entities along with their attributes

## 1. Inventory Management:

- Products
  - Product ID (primary key)
  - Product name
  - Supplier ID (foreign key)
  - Unit price
  - Quantity in stock
  - Reorder level
  - Expiry date
  - Category
- Suppliers
  - Supplier ID (primary key)
  - Supplier name
  - Contact info
  - Address
  - Email
  - Phone number
- Inventory
  - Inventory ID (primary key)
  - Product ID (foreign key)
  - Location ID
  - Quantity on hand
  - Last updated
- Transactions
  - Transaction ID (primary key)
  - Transaction type
  - Product ID (foreign key)
  - Quantity
  - Transaction date
  - Employee ID (foreign key)
- Locations
  - Location ID (primary key)
  - Location name
  - Description

## 2. Production Planning:

- Production Lines
  - Line ID
  - Line name
  - Description

## 3. Order Tracking:

- Customers
  - Customer ID (primary key)
  - Customer Name
  - Email
  - Phone Number
  - Address
  -
- Orders
  - Order ID (primary key)
  - Customer ID (foreign key)
  - Order Date
  - Delivery Date
  - Total Amount
  - Payment Status
  - Shipping Method
  - Tracking Number
- Order Items
  - Order Item ID (primary key)
  - Order ID (foreign key)
  - Product ID (foreign key)
  - Quantity
  - Unit Price
- Order Status
  - Status ID (primary key)
  - Order ID (foreign key)
  - Status
  - Status Date

## 4. Employee Management:

- Employees
  - Employee ID (primary key)
  - First Name

- Last Name
- Date of Birth
- Gender
- Position
- Department
- Salary
- Hire date
- Supervisor ID (foreign key)

- Departments
  - Department ID
  - Department name

#### 5. Quality Control:

- Quality Checks
  - Check ID (primary key)
  - Product ID (foreign key)
  - Check date
  - Inspector (foreign key)
  - Defect found
  - Resolution
  - Status
  - Result
  - Remarks

- Inspectors
  - Inspector ID
  - Inspector name

#### 6. Reporting and Analytics:

- Reports
  - Report ID
  - Report type
  - Generation date
  - Parameters
  - Metrics
  - Analysis result
  - Conclusion
  - Recommendations
  - User ID
  - Department ID

All relationships among various entities and the cardinality and participation for all relationships.

#### Inventory Management:

1. Products – Suppliers
  - Relationship: One-to-Many
  - Cardinality: One product can have only one supplier, but one supplier can supply multiple products.
  - Participation: Mandatory on the product side (each product must have a supplier), optional on the supplier side (a supplier may not supply any product).
2. Products – Inventory
  - Relationship: One-to-One
  - Cardinality: Each product has one inventory record, and each inventory record corresponds to one product.
  - Participation: Mandatory on both sides (each product must have an inventory record, and each inventory record must correspond to a product).
3. Products – Transactions
  - Relationship: One-to-Many
  - Cardinality: One product can be involved in multiple transactions, but each transaction involves only one product.
  - Participation: Mandatory on the product side (each product must be involved in at least one transaction), optional on the transaction side (a transaction may not involve any product).
4. Inventory – Locations
  - Relationship: One-to-Many
  - Cardinality: One inventory record is associated with one location, but one location can have multiple inventory records.
  - Participation: Mandatory on the inventory side (each inventory record must be associated with a location), optional on the location side (a location may not have any inventory records).

#### Production Planning:

1. Production Orders - Production Lines
  - Relationship: Many-to-One
  - Cardinality: Multiple production orders can be assigned to one production line, but each production order is assigned to only one production line.
  - Participation: Mandatory on the production order side (each production order must be assigned to a production line), optional on the production line side (a production line may not have any production orders).

#### Order Tracking:

1. Customers – Orders
  - Relationship: One-to-Many
  - Cardinality: One customer can place multiple orders, but each order is placed by only one customer.
  - Participation: Mandatory on the customer side (each customer must place at least one order), optional on the order side (an order may not be placed by any customer).



## 2. Orders - Order Items

- Relationship: One-to-Many
- Cardinality: One order can have multiple order items, but each order item belongs to only one order.
- Participation: Mandatory on the order side (each order must have at least one order item), optional on the order item side (an order item may not belong to any order).

## 3. Orders - Order Status

- Relationship: One-to-Many
- Cardinality: One order can have multiple status updates, but each status update corresponds to only one order.
- Participation: Mandatory on the order side (each order must have at least one status update), optional on the status update side (a status update may not correspond to any order).

### Employee Management:

#### 1. Employees – Departments

- Relationship: Many-to-One
- Cardinality: Multiple employees can belong to one department, but each employee belongs to only one department.
- Participation: Mandatory on the employee side (each employee must belong to a department), optional on the department side (a department may not have any employees).

### Quality Control:

#### 1. Quality Checks - Inspectors

- Relationship: Many-to-One
- Cardinality: Multiple quality checks can be performed by one inspector, but each quality check is performed by only one inspector.
- Participation: Mandatory on the quality check side (each quality check must be performed by an inspector), optional on the inspector side (an inspector may not perform any quality checks).

### Reporting and Analytics:

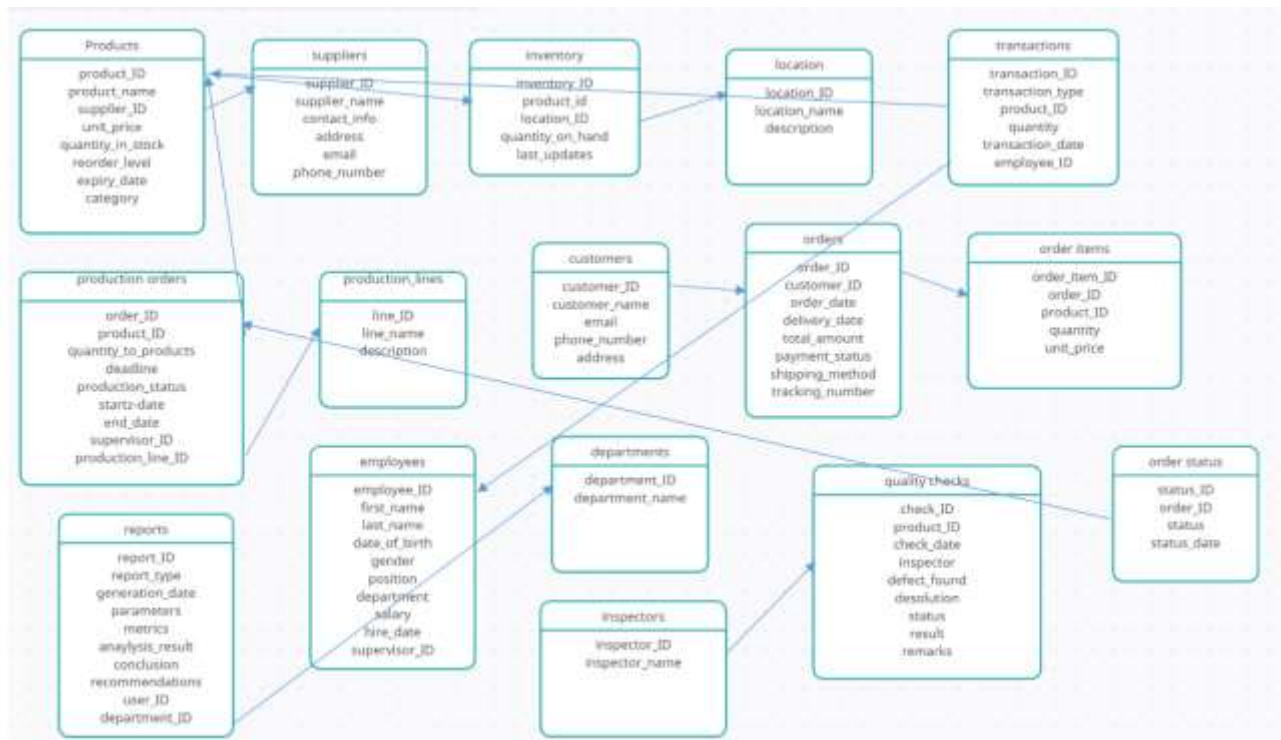
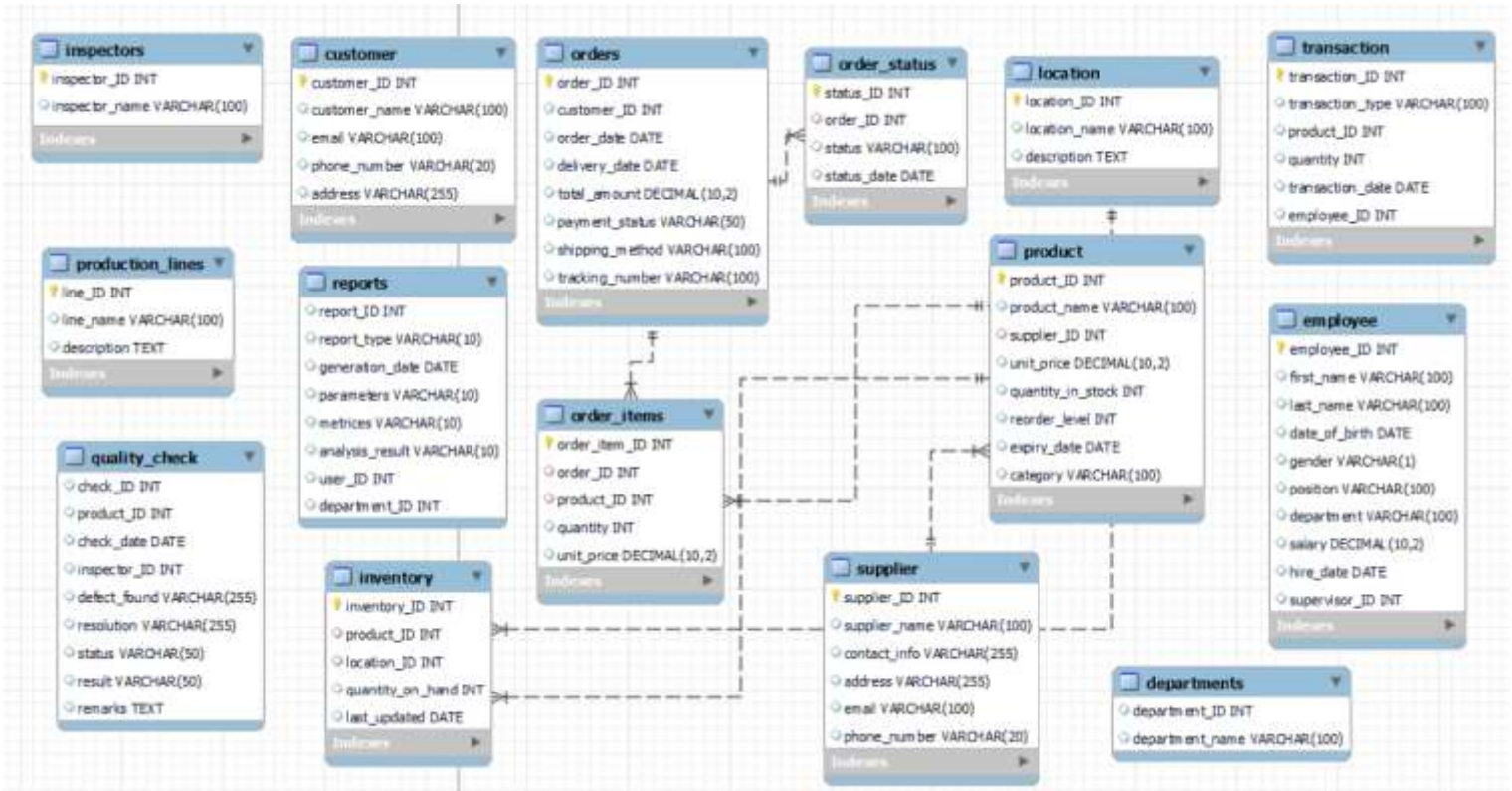
#### 1. Reports - Users

- Relationship: One-to-Many
- Cardinality: One report is generated by one user, but one user can generate multiple reports.
- Participation: Mandatory on the report side (each report must be generated by a user), optional on the user side (a user may not generate any reports).

#### 2. Reports – Departments

- Relationship: One-to-Many
- Cardinality: One report is associated with one department, but one department can have multiple reports.
- Participation: Mandatory on the report side (each report must be associated with a department), optional on the department side (a department may not have any reports).

# Relational Model



## **Normalization**

- Products: Table follows the rule of 1NF, 2NF, 3NF
- Suppliers: Table follows the rule of 1NF, 2NF, 3NF
- Inventory: Table follows the rule of 1NF, 2NF, 3NF
- Transactions: Table follows the rule of 1NF, 2NF, 3NF
- Locations: Table follows the rule of 1NF, 2NF, 3NF
- Production lines: Table follows the rule of 1NF, 2NF, 3NF
- Customers: Table follows the rule of 1NF, 2NF, 3NF
- Orders: Table follows the rule of 1NF, 2NF, 3NF
- Order items: Table follows the rule of 1NF, 2NF, 3NF
- Order status: Table follows the rule of 1NF, 2NF, 3NF
- Employees: Table follows the rule of 1NF, 2NF, 3NF
- Department: Table follows the rule of 1NF, 2NF, 3NF
- Quality checks: Table follows the rule of 1NF, 2NF, 3NF
- Inspectors: Table follows the rule of 1NF, 2NF, 3NF
- Reports: Table follows the rule of 1NF, 2NF, 3NF

# SQL Queries

```
/*01 product*/
```

```
create table product (product_ID int primary key,product_name varchar(100),supplier_ID int,
unit_price decimal(10, 2),quantity_in_stock int,reorder_level int,expiry_date date,
category varchar(100),foreign key (supplier_ID) references supplier(supplier_ID));
create index inventory_id on product(product_id);
insert into product values(101, 'Widget A', 1, 25.99, 100, 20, '2024-12-31', 'Electronics');
insert into product values(102, 'Gadget B', 2, 15.75, 200, 30, '2025-06-30', 'Tools');
insert into product values(103, 'Tool C', 3, 30.50, 150, 25, '2024-10-31', 'Electronics');
insert into product values(104, 'Appliance D', 4, 50.25, 80, 15, '2025-03-31', 'Appliances');
insert into product values(105, 'Device E', 5, 12.99, 120, 30, '2024-09-30', 'Electronics');
insert into product values(106, 'Gizmo F', 6, 35.50, 90, 20, '2025-01-31', 'Tools');
insert into product values(107, 'Equipment G', 7, 20.75, 130, 25, '2024-11-30', 'Electronics');
insert into product values(108, 'Machine H', 8, 18.00, 110, 30, '2025-02-28', 'Machinery');
insert into product values(109, 'Component I', 9, 40.25, 70, 20, '2024-08-31', 'Electronics');
insert into product values(110, 'Tool J', 10, 22.50, 140, 25, '2025-04-30', 'Tools');
select*from product;
```

|   | product_ID | product_name | supplier_ID | unit_price | quantity_in_stock | reorder_level | expiry_date | category    |
|---|------------|--------------|-------------|------------|-------------------|---------------|-------------|-------------|
| ▶ | 101        | Widget A     | 1           | 25.99      | 100               | 20            | 2024-12-31  | Electronics |
|   | 102        | Gadget B     | 2           | 15.75      | 200               | 30            | 2025-06-30  | Tools       |
|   | 103        | Tool C       | 3           | 30.50      | 150               | 25            | 2024-10-31  | Electronics |
|   | 104        | Appliance D  | 4           | 50.25      | 80                | 15            | 2025-03-31  | Appliances  |
|   | 105        | Device E     | 5           | 12.99      | 120               | 30            | 2024-09-30  | Electronics |
|   | 106        | Gizmo F      | 6           | 35.50      | 90                | 20            | 2025-01-31  | Tools       |
|   | 107        | Equipment G  | 7           | 20.75      | 130               | 25            | 2024-11-30  | Electronics |
|   | 108        | Machine H    | 8           | 18.00      | 110               | 30            | 2025-02-28  | Machinery   |
|   | 109        | Component I  | 9           | 40.25      | 70                | 20            | 2024-08-31  | Electronics |
|   | 110        | Tool J       | 10          | 22.50      | 140               | 25            | 2025-04-30  | Tools       |

```
/*02 supplier*/
```

```
create table supplier (supplier_ID int primary key,supplier_name varchar(100),contact_info varchar(255),
address varchar(255),email varchar(100),phone_number varchar(20));
insert into supplier values(1, 'ABC Electronics', 'John Doe', '123 Main St, City, Country', 'john.doe@abc.com', '1234567890');
insert into supplier values(2, 'XYZ Manufacturing', 'Alice Smith', '456 Elm St, City, Country', 'alice.smith@xyz.com', '1987654321');
insert into supplier values(3, 'UVW Distributors', 'Michael Johnson', '789 Oak St, City, Country', 'michael.johnson@uvw.com', '1122334455');
insert into supplier values(4, 'PQR Supplies', 'Emily Brown', '101 Pine St, City, Country', 'emily.brown@pqr.com', '1555666777');
insert into supplier values(5, 'RST Industries', 'David Wilson', '321 Maple St, City, Country', 'david.wilson@rst.com', '1444333222');
insert into supplier values(6, 'UVW Corporation', 'Sarah Martinez', '543 Cedar St, City, Country', 'sarah.martinez@uvw.com', '1777888999');
insert into supplier values(7, 'GHI Enterprises', 'Robert Taylor', '987 Birch St, City, Country', 'robert.taylor@ghi.com', '1999888777');
insert into supplier values(8, 'MNO Solutions', 'Jennifer Garcia', '654 Spruce St, City, Country', 'jennifer.garcia@mno.com', '1666777888');
insert into supplier values(9, 'DEF Supplies', 'William Anderson', '210 Walnut St, City, Country', 'william.anderson@def.com', '1888777666');
insert into supplier values(10, 'STU Technologies', 'Jessica Thomas', '876 Cherry St, City, Country', 'jessica.thomas@stu.com', '1223344555');
select*from supplier;
```



| supplier_ID | supplier_name     | contact_info     | address                      | email                    | phone_number |
|-------------|-------------------|------------------|------------------------------|--------------------------|--------------|
| 1           | ABC Electronics   | John Doe         | 123 Main St, City, Country   | john.doe@abc.com         | 1234567890   |
| 2           | XYZ Manufacturing | Alice Smith      | 456 Elm St, City, Country    | alice.smith@xyz.com      | 1987654321   |
| 3           | LMN Distributors  | Michael Johnson  | 789 Oak St, City, Country    | michael.johnson@lmn.com  | 1122334455   |
| 4           | PQR Supplies      | Emily Brown      | 101 Pine St, City, Country   | emily.brown@pqr.com      | 1555666777   |
| 5           | RST Industries    | David Wilson     | 321 Maple St, City, Country  | david.wilson@rst.com     | 1444333222   |
| 6           | UVW Corporation   | Sarah Martinez   | 543 Cedar St, City, Country  | sarah.martinez@uvw.com   | 1777888999   |
| 7           | GHI Enterprises   | Robert Taylor    | 987 Birch St, City, Country  | robert.taylor@ghi.com    | 1999888777   |
| 8           | MNO Solutions     | Jennifer Garcia  | 654 Spruce St, City, Country | jennifer.garcia@mno.com  | 1666777888   |
| 9           | DEF Supplies      | William Anderson | 210 Walnut St, City, Country | william.anderson@def.com | 1888777666   |
| 10          | STU Technologies  | Jessica Thomas   | 876 Cherry St, City, Country | jessica.thomas@stu.com   | 1223344556   |

```

/*03 inventory*/
create table inventory (inventory_ID int primary key,product_ID int,location_ID int,quantity_on_hand int,last_updated date,
    foreign key(product_ID) references product(product_ID),foreign key(location_ID) references location(location_ID));
insert into inventory values(11, 101, 31, 100, '2024-03-01');
insert into inventory values(12, 102, 32, 200, '2024-03-02');
insert into inventory values(13, 103, 33, 150, '2024-03-03');
insert into inventory values(14, 104, 34, 80, '2024-03-04');
insert into inventory values(15, 105, 35, 120, '2024-03-05');
insert into inventory values(16, 106, 36, 90, '2024-03-06');
insert into inventory values(17, 107, 37, 130, '2024-03-07');
insert into inventory values(18, 108, 38, 110, '2024-03-08');
insert into inventory values(19, 109, 39, 70, '2024-03-09');
insert into inventory values(20, 110, 40, 140, '2024-03-10');
select*from inventory;

```

|   | inventory_ID | product_ID | location_ID | quantity_on_hand | last_updated |
|---|--------------|------------|-------------|------------------|--------------|
| ► | 11           | 101        | 31          | 100              | 2024-03-01   |
|   | 12           | 102        | 32          | 200              | 2024-03-02   |
|   | 13           | 103        | 33          | 150              | 2024-03-03   |
|   | 14           | 104        | 34          | 80               | 2024-03-04   |
|   | 15           | 105        | 35          | 120              | 2024-03-05   |
|   | 16           | 106        | 36          | 90               | 2024-03-06   |
|   | 17           | 107        | 37          | 130              | 2024-03-07   |
|   | 18           | 108        | 38          | 110              | 2024-03-08   |
|   | 19           | 109        | 39          | 70               | 2024-03-09   |
|   | 20           | 110        | 40          | 140              | 2024-03-10   |

```

/*04 transaction*/
create table transaction (transaction_ID int primary key,transaction_type varchar(100),
product_ID int,quantity int,
transaction_date date,employee_ID int);
insert into transaction values(41, 'Sale', 101, 2, '2024-03-01', 401);
insert into transaction values(42, 'Purchase', 102, 5, '2024-03-02', 402);
insert into transaction values(43, 'Sale', 103, 1, '2024-03-03', 403);
insert into transaction values(44, 'Purchase', 104, 3, '2024-03-04', 404);
insert into transaction values(45, 'Sale', 105, 4, '2024-03-05', 405);
insert into transaction values(46, 'Purchase', 106, 2, '2024-03-06', 406);
insert into transaction values(47, 'Sale', 107, 3, '2024-03-07', 407);
insert into transaction values(48, 'Purchase', 108, 6, '2024-03-08', 408);
insert into transaction values(49, 'Sale', 109, 1, '2024-03-09', 409);
insert into transaction values(50, 'Purchase', 110, 2, '2024-03-10', 410);
select*from transaction;

```

|   | transaction_ID | transaction_type | product_ID | quantity | transaction_date | employee_ID |
|---|----------------|------------------|------------|----------|------------------|-------------|
| ► | 41             | Sale             | 101        | 2        | 2024-03-01       | 401         |
|   | 42             | Purchase         | 102        | 5        | 2024-03-02       | 402         |
|   | 43             | Sale             | 103        | 1        | 2024-03-03       | 403         |
|   | 44             | Purchase         | 104        | 3        | 2024-03-04       | 404         |
|   | 45             | Sale             | 105        | 4        | 2024-03-05       | 405         |
|   | 46             | Purchase         | 106        | 2        | 2024-03-06       | 2024-03-05  |
|   | 47             | Sale             | 107        | 3        | 2024-03-07       | 407         |
|   | 48             | Purchase         | 108        | 6        | 2024-03-08       | 408         |
|   | 49             | Sale             | 109        | 1        | 2024-03-09       | 409         |
|   | 50             | Purchase         | 110        | 2        | 2024-03-10       | 410         |



```

/*05 location*/
create table location (location_ID int primary key,location_name varchar(100),description text);
insert into location values(31, 'Warehouse A', 'Main warehouse for finished goods storage. ');
insert into location values(32, 'Warehouse B', 'Secondary warehouse for overflow storage. ');
insert into location values(33, 'Factory A', 'Main production facility for electronics. ');
insert into location values(34, 'Factory B', 'Secondary production facility for packaging. ');
insert into location values(35, 'Distribution Center A', 'Distribution center for local deliveries. ');
insert into location values(36, 'Retail Store A', 'Company-owned retail store in downtown. ');
insert into location values(37, 'Retail Store A', 'Company-owned retail store in downtown. ');
insert into location values(38, 'Retail Store B', 'Company-owned retail store in suburban area. ');
insert into location values(39, 'Office Building A', 'Headquarters office building. ');
insert into location values(40, 'Office Building B', 'Branch office building in another city. ');
select*from location;

```

|   | location_ID | location_name         | description                                  |
|---|-------------|-----------------------|--|
| ► | 31          | Warehouse A           | Main warehouse for finished goods storage.   |
|   | 32          | Warehouse B           | Secondary warehouse for overflow storage.    |
|   | 33          | Factory A             | Main production facility for electronics.    |
|   | 34          | Factory B             | Secondary production facility for packaging. |
|   | 35          | Distribution Center A | Distribution center for local deliveries.    |
|   | 36          | Retail Store A        | Company-owned retail store in downtown.      |
|   | 37          | Retail Store A        | Company-owned retail store in downtown.      |
|   | 38          | Retail Store B        | Company-owned retail store in suburban area. |
|   | 39          | Office Building A     | Headquarters office building.                |
|   | 40          | Office Building B     | Branch office building in another city.      |

```

/*06 production order*/
drop table production_order;

/*07 production lines*/
create table production_lines (line_ID int primary key,line_name varchar(100),description text);
insert into production_lines values(71, 'Assembly Line 1', 'Assembly line for small electronic devices. ');
insert into production_lines values(72, 'Packaging Line 1', 'Packaging line for food products. ');
insert into production_lines values(73, 'Manufacturing Line 1', 'Heavy-duty manufacturing line for automotive parts. ');
insert into production_lines values(74, 'Bottling Line 1', 'Bottling line for beverages. ');
insert into production_lines values(75, 'Assembly Line 2', 'Assembly line for consumer electronics. ');
insert into production_lines values(76, 'Packaging Line 2', 'Packaging line for pharmaceutical products. ');
insert into production_lines values(77, 'Manufacturing Line 2', 'Precision manufacturing line for medical devices. ');
insert into production_lines values(78, 'Bottling Line 2', 'Bottling line for cosmetics. ');
insert into production_lines values(79, 'Assembly Line 3', 'Assembly line for furniture components. ');
insert into production_lines values(80, 'Packaging Line 3', 'Packaging line for household goods. ');
select*from Production_lines;

```

|   | line_ID | line_name            | description                                       |
|---|---------|----------------------|---|
| ► | 71      | Assembly Line 1      | Assembly line for small electronic devices.       |
|   | 72      | Packaging Line 1     | Packaging line for food products.                 |
|   | 73      | Manufacturing Line 1 | Heavy-duty manufacturing line for automotive ...  |
|   | 74      | Bottling Line 1      | Bottling line for beverages.                      |
|   | 75      | Assembly Line 2      | Assembly line for consumer electronics.           |
|   | 76      | Packaging Line 2     | Packaging line for pharmaceutical products.       |
|   | 77      | Manufacturing Line 2 | Precision manufacturing line for medical devices. |
|   | 78      | Bottling Line 2      | Bottling line for cosmetics.                      |
|   | 79      | Assembly Line 3      | Assembly line for furniture components.           |
|   | 80      | Packaging Line 3     | Packaging line for household goods.               |

```

/*00 customer*/
create table customer (customer_ID int primary key,customer_name varchar(100),email varchar(100),phone_number varchar(20),address varchar(255));
insert into customer values(81, 'John Smith', 'john@example.com', '1234567890', '123 Main St, City, Country');
insert into customer values(82, 'Alice Johnson', 'alice@example.com', '1987654321', '456 Elm St, City, Country');
insert into customer values(83, 'Michael Brown', 'michael@example.com', '1122334455', '789 Oak St, City, Country');
insert into customer values(84, 'Emily Davis', 'emily@example.com', '1555666777', '101 Pine St, City, Country');
insert into customer values(85, 'David Wilson', 'david@example.com', '1444333222', '321 Maple St, City, Country');
insert into customer values(86, 'Sarah Martinez', 'sarah@example.com', '1777888999', '543 Cedar St, City, Country');
insert into customer values(87, 'Robert Taylor', 'robert@example.com', '1999888777', '987 Birch St, City, Country');
insert into customer values(88, 'Jennifer Garcia', 'jennifer@example.com', '1666777888', '654 Spruce St, City, Country');
insert into customer values(89, 'William Anderson', 'william@example.com', '1888777666', '210 Walnut St, City, Country');
insert into customer values(90, 'Jessica Thomas', 'jessica@example.com', '1223344556', '876 Cherry St, City, Country');
select*from Customer;

```

|   | customer_ID | customer_name    | email                | phone_number | address                      |
|---|-------------|------------------|----------------------|--------------|------------------------------|
| ► | 81          | John Smith       | john@example.com     | 1234567890   | 123 Main St, City, Country   |
|   | 82          | Alice Johnson    | alice@example.com    | 1987654321   | 456 Elm St, City, Country    |
|   | 83          | Michael Brown    | michael@example.com  | 1122334455   | 789 Oak St, City, Country    |
|   | 84          | Emily Davis      | emily@example.com    | 1555666777   | 101 Pine St, City, Country   |
|   | 85          | David Wilson     | david@example.com    | 1444333222   | 321 Maple St, City, Country  |
|   | 86          | Sarah Martinez   | sarah@example.com    | 1777888999   | 543 Cedar St, City, Country  |
|   | 87          | Robert Taylor    | robert@example.com   | 1999888777   | 987 Birch St, City, Country  |
|   | 88          | Jennifer Garcia  | jennifer@example.com | 1666777888   | 654 Spruce St, City, Country |
|   | 89          | William Anderson | william@example.com  | 1888777666   | 210 Walnut St, City, Country |
|   | 90          | Jessica Thomas   | jessica@example.com  | 1223344556   | 876 Cherry St, City, Country |



```

/*09 orders*/
create table orders (order_ID int primary key,customer_ID int,order_date date,delivery_date date,total_amount decimal(10, 2),
payment_status varchar(50),shipping_method varchar(100),tracking_number varchar(100));
insert into orders values(51, 81, '2024-03-01', '2024-03-05', 125.99, 'Paid', 'Express', 'TRK123456');
insert into orders values(52, 82, '2024-03-02', '2024-03-06', 182.50, 'Paid', 'Standard', 'TRK789012');
insert into orders values(53, 83, '2024-03-03', '2024-03-07', 75.25, 'Paid', 'Express', 'TRK345678');
insert into orders values(54, 84, '2024-03-04', '2024-03-08', 155.98, 'Paid', 'Standard', 'TRK901234');
insert into orders values(55, 85, '2024-03-05', '2024-03-09', 105.00, 'Paid', 'Express', 'TRK567890');
insert into orders values(56, 86, '2024-03-06', '2024-03-10', 320.75, 'Paid', 'Standard', 'TRK123456');
insert into orders values(57, 87, '2024-03-07', '2024-03-11', 220.00, 'Paid', 'Express', 'TRK789012');
insert into orders values(58, 88, '2024-03-08', '2024-03-12', 150.25, 'Paid', 'Standard', 'TRK345678');
insert into orders values(59, 89, '2024-03-09', '2024-03-13', 185.00, 'Paid', 'Express', 'TRK901234');
insert into orders values(60, 90, '2024-03-10', '2024-03-14', 75.50, 'Paid', 'Standard', 'TRK567890');
select*from orders;

```

|   | order_ID | customer_ID | order_date | delivery_date | total_amount | payment_status | shipping_method | tracking_number |
|---|----------|-------------|------------|---------------|--------------|----------------|-----------------|-----------------|
| ► | 51       | 81          | 2024-03-01 | 2024-03-05    | 125.99       | Paid           | Express         | TRK123456       |
|   | 52       | 82          | 2024-03-02 | 2024-03-06    | 182.50       | Paid           | Standard        | TRK789012       |
|   | 53       | 83          | 2024-03-03 | 2024-03-07    | 75.25        | Paid           | Express         | TRK345678       |
|   | 54       | 84          | 2024-03-04 | 2024-03-08    | 155.98       | Paid           | Standard        | TRK901234       |
|   | 55       | 85          | 2024-03-05 | 2024-03-09    | 105.00       | Paid           | Express         | TRK567890       |
|   | 56       | 86          | 2024-03-06 | 2024-03-10    | 320.75       | Paid           | Standard        | TRK123456       |
|   | 57       | 87          | 2024-03-07 | 2024-03-11    | 220.00       | Paid           | Express         | TRK789012       |
|   | 58       | 88          | 2024-03-08 | 2024-03-12    | 150.25       | Paid           | Standard        | TRK345678       |
|   | 59       | 89          | 2024-03-09 | 2024-03-13    | 185.00       | Paid           | Express         | TRK901234       |
|   | 60       | 90          | 2024-03-10 | 2024-03-14    | 75.50        | Paid           | Standard        | TRK567890       |

```

/*10 order items*/
create table order_items (order_item_ID int primary key,order_ID int,product_ID int,quantity int,unit_price decimal(10, 2),
foreign key (order_ID) references orders(order_ID),
foreign key (product_ID) references product(product_ID));
insert into order_items values(901, 51, 101, 2, 25.99);
insert into order_items values(902, 51, 102, 1, 10.50);
insert into order_items values(903, 52, 103, 3, 15.75);
insert into order_items values(904, 52, 104, 2, 30.00);
insert into order_items values(905, 53, 105, 1, 50.25);
insert into order_items values(906, 54, 106, 4, 12.99);
insert into order_items values(907, 55, 107, 2, 35.50);
insert into order_items values(908, 56, 108, 1, 20.75);
insert into order_items values(909, 57, 109, 3, 18.00);
insert into order_items values(910, 58, 110, 2, 40.25);
select*from order_items;

```

|   | order_item_ID | order_ID | product_ID | quantity | unit_price |
|---|---------------|----------|------------|----------|------------|
| ▶ | 901           | 51       | 101        | 2        | 25.99      |
|   | 902           | 51       | 102        | 1        | 10.50      |
|   | 903           | 52       | 103        | 3        | 15.75      |
|   | 904           | 52       | 104        | 2        | 30.00      |
|   | 905           | 53       | 105        | 1        | 50.25      |
|   | 906           | 54       | 106        | 4        | 12.99      |
|   | 907           | 55       | 107        | 2        | 35.50      |
|   | 908           | 56       | 108        | 1        | 20.75      |
|   | 909           | 57       | 109        | 3        | 18.00      |
|   | 910           | 58       | 110        | 2        | 40.25      |

```

/*11 order status*/
create table order_status (status_ID int primary key,order_ID int,status varchar(100),status_date date,
    foreign key (order_ID) references orders(order_ID));
insert into order_status values(21, 51, 'Processing', '2024-03-01');
insert into order_status values(22, 52, 'Shipped', '2024-03-02');
insert into order_status values(23, 53, 'Delivered', '2024-03-03');
insert into order_status values(24, 54, 'Processing', '2024-03-04');
insert into order_status values(25, 55, 'Processing', '2024-03-05');
insert into order_status values(26, 56, 'Shipped', '2024-03-06');
insert into order_status values(27, 57, 'Delivered', '2024-03-07');
insert into order_status values(28, 58, 'Processing', '2024-03-08');
insert into order_status values(29, 59, 'Shipped', '2024-03-09');
insert into order_status values(30, 60, 'Delivered', '2024-03-10');
select*from order_status;

```

|   | status_ID | order_ID | status     | status_date |
|---|-----------|----------|------------|-------------|
| ▶ | 21        | 51       | Processing | 2024-03-01  |
|   | 22        | 52       | Shipped    | 2024-03-02  |
|   | 23        | 53       | Delivered  | 2024-03-03  |
|   | 24        | 54       | Processing | 2024-03-04  |
|   | 25        | 55       | Processing | 2024-03-05  |
|   | 26        | 56       | Shipped    | 2024-03-06  |
|   | 27        | 57       | Delivered  | 2024-03-07  |
|   | 28        | 58       | Processing | 2024-03-08  |
|   | 29        | 59       | Shipped    | 2024-03-09  |
|   | 30        | 60       | Delivered  | 2024-03-10  |



```

/*12 employee*/
create table employee (employee_ID int primary key,first_name varchar(100),last_name varchar(100),
date_of_birth date,gender varchar(1),position varchar(100),department varchar(100),salary decimal(10,2),
hire_date date,supervisor_ID int);
insert into employee values(401, 'John', 'Doe', '1990-05-15', 'M', 'Manager', 'Operations', 60000.00, '2015-03-20', NULL);
insert into employee values(402, 'Jane', 'Smith', '1988-09-22', 'F', 'Senior Analyst', 'Finance', 55000.00, '2016-06-10', 501);
insert into employee values(403, 'Michael', 'Johnson', '1992-11-10', 'M', 'Software Engineer', 'Information Technology', 65000.00, '2017-02-15', 501);
insert into employee values(404, 'Emily', 'Brown', '1995-04-30', 'F', 'Marketing Specialist', 'Marketing', 50000.00, '2018-01-05', 502);
insert into employee values(405, 'David', 'Wilson', '1993-07-18', 'M', 'Customer Service Representative', 'Customer Service', 45000.00, '2019-04-22', 502);
insert into employee values(406, 'Sarah', 'Martinez', '1991-12-12', 'F', 'Sales Associate', 'Sales', 48000.00, '2019-08-15', 503);
insert into employee values(407, 'Robert', 'Taylor', '1987-03-25', 'M', 'HR Specialist', 'Human Resources', 52000.00, '2020-02-10', 501);
insert into employee values(408, 'Jennifer', 'Garcia', '1989-08-05', 'F', 'Legal Advisor', 'Legal', 60000.00, '2020-05-01', 503);
insert into employee values(409, 'William', 'Anderson', '1994-06-28', 'M', 'Quality Assurance Analyst', 'Quality Assurance', 55000.00, '2021-01-15', 504);
insert into employee values(410, 'Jessica', 'Thomas', '1990-02-14', 'F', 'Operations Coordinator', 'Operations', 52000.00, '2021-08-20', 505);
select*from employee;

```

|   | employee_ID | first_name | last_name | date_of_birth | gender | position                        | department             | salary   | hire_date  | supervisor_ID |
|---|-------------|------------|-----------|---------------|--------|---------------------------------|------------------------|----------|------------|---------------|
| ▶ | 401         | John       | Doe       | 1990-05-15    | M      | Manager                         | Operations             | 60000.00 | 2015-03-20 | NULL          |
|   | 402         | Jane       | Smith     | 1988-09-22    | F      | Senior Analyst                  | Finance                | 55000.00 | 2016-06-10 | 501           |
|   | 403         | Michael    | Johnson   | 1992-11-10    | M      | Software Engineer               | Information Technology | 65000.00 | 2017-02-15 | 501           |
|   | 404         | Emily      | Brown     | 1995-04-30    | F      | Marketing Specialist            | Marketing              | 50000.00 | 2018-01-05 | 502           |
|   | 405         | David      | Wilson    | 1993-07-18    | M      | Customer Service Representative | Customer Service       | 45000.00 | 2019-04-22 | 502           |
|   | 406         | Sarah      | Martinez  | 1991-12-12    | F      | Sales Associate                 | Sales                  | 48000.00 | 2019-08-15 | 503           |
|   | 407         | Robert     | Taylor    | 1987-03-25    | M      | HR Specialist                   | Human Resources        | 52000.00 | 2020-02-10 | 501           |
|   | 408         | Jennifer   | Garcia    | 1989-08-05    | F      | Legal Advisor                   | Legal                  | 60000.00 | 2020-05-01 | 503           |
|   | 409         | William    | Anderson  | 1994-06-28    | M      | Quality Assurance Analyst       | Quality Assurance      | 55000.00 | 2021-01-15 | 504           |
|   | 410         | Jessica    | Thomas    | 1990-02-14    | F      | Operations Coordinator          | Operations             | 52000.00 | 2021-08-20 | 505           |

```

/*13 departments*/
create table departments (department_ID int,department_name varchar(100));
insert into departments values(601, 'Human Resources');
insert into departments values(602, 'Finance');
insert into departments values(603, 'Marketing');
insert into departments values(604, 'Information Technology');
insert into departments values(605, 'Operations');
insert into departments values(606, 'Customer Service');
insert into departments values(607, 'Research and Development');
insert into departments values(608, 'Sales');
insert into departments values(609, 'Legal');
insert into departments values(610, 'Quality Assurance');
select*from departments;

```

|   | department_ID | department_name          |
|---|---------------|--------------------------|
| ▶ | 601           | Human Resources          |
|   | 602           | Finance                  |
|   | 603           | Marketing                |
|   | 604           | Information Technology   |
|   | 605           | Operations               |
|   | 606           | Customer Service         |
|   | 607           | Research and Development |
|   | 608           | Sales                    |
|   | 609           | Legal                    |
|   | 610           | Quality Assurance        |

```

/*14 quality check*/
create table quality_check (check_ID int,product_ID int,check_date date,inspector_ID int,
defect_found varchar(255),resolution varchar(255),status varchar(50),result varchar(50),remarks text);
insert into quality_check values (701, 101, '2024-03-19', 801, 'Scratched surface', 'Repaired', 'Completed', 'Pass', 'No further issues observed.
insert into quality_check values (702, 102, '2024-03-20', 802, 'Missing parts', 'Replaced missing parts', 'Completed', 'Pass', 'Product now meets specifications.
insert into quality_check values (703, 103, '2024-03-21', 803, 'Cracked casing', 'Repaired', 'Completed', 'Pass', 'Casing strengthened.
insert into quality_check values (704, 104, '2024-03-22', 804, 'Uneven finish', 'Refinished', 'Completed', 'Pass', 'Surface now smooth.
insert into quality_check values (705, 105, '2024-03-23', 805, 'Misaligned components', 'Adjusted alignment', 'Completed', 'Pass', 'Components now properly aligned.
insert into quality_check values (706, 106, '2024-03-24', 806, 'Faulty wiring', 'Rewired', 'Completed', 'Pass', 'Electrical connections restored.
insert into quality_check values (707, 107, '2024-03-25', 807, 'Loose screws', 'Tightened screws', 'Completed', 'Pass', 'Screws securely fastened.
insert into quality_check values (708, 108, '2024-03-26', 808, 'Inconsistent color', 'Recoloring', 'Completed', 'Pass', 'Color consistency achieved.
insert into quality_check values (709, 109, '2024-03-27', 809, 'Weak joints', 'Reinforced joints', 'Completed', 'Pass', 'Joints strengthened.
insert into quality_check values (710, 110, '2024-03-28', 810, 'Inaccurate measurements', 'Adjusted measurements', 'Completed', 'Pass', 'Measurements now accurate.
select*from Quality_check;

```

| check_ID | product_ID | check_date | inspector_ID | defect_found            | resolution             | status    | result | remarks                           |
|----------|------------|------------|--------------|-------------------------|------------------------|-----------|--------|-----------------------------------|
| 701      | 101        | 2024-03-19 | 801          | Scratched surface       | Repaired               | Completed | Pass   | No further issues observed.       |
| 702      | 102        | 2024-03-20 | 802          | Missing parts           | Replaced missing parts | Completed | Pass   | Product now meets specifications. |
| 703      | 103        | 2024-03-21 | 803          | Cracked casing          | Repaired               | Completed | Pass   | Casing strengthened.              |
| 704      | 104        | 2024-03-22 | 804          | Uneven finish           | Refinished             | Completed | Pass   | Surface now smooth.               |
| 705      | 105        | 2024-03-23 | 805          | Misaligned components   | Adjusted alignment     | Completed | Pass   | Components now properly aligned.  |
| 706      | 106        | 2024-03-24 | 806          | Faulty wiring           | Rewired                | Completed | Pass   | Electrical connections restored.  |
| 707      | 107        | 2024-03-25 | 807          | Loose screws            | Tightened screws       | Completed | Pass   | Screws securely fastened.         |
| 708      | 108        | 2024-03-26 | 808          | Inconsistent color      | Recoloring             | Completed | Pass   | Color consistency achieved.       |
| 709      | 109        | 2024-03-27 | 809          | Weak joints             | Reinforced joints      | Completed | Pass   | Joints strengthened.              |
| 710      | 110        | 2024-03-28 | 810          | Inaccurate measurements | Adjusted measurements  | Completed | Pass   | Measurements now accurate.        |



```
/*15. * Inspectors*/
```

```
create table inspectors (inspector_ID int primary key,inspector_name varchar(100));
insert into inspectors values (801, 'John Doe');
insert into inspectors values (802, 'Jane Smith');
insert into inspectors values (803, 'Michael Johnson');
insert into inspectors values (804, 'Emily Brown');
insert into inspectors values (805, 'David Wilson');
insert into inspectors values (806, 'Sarah Martinez');
insert into inspectors values (807, 'Robert Taylor');
insert into inspectors values (808, 'Jennifer Garcia');
insert into inspectors values (809, 'William Anderson');
insert into inspectors values (810, 'Jessica Thomas');
select*from Inspectors;
```

| inspector_ID | inspector_name   |
|--------------|------------------|
| 801          | John Doe         |
| 802          | Jane Smith       |
| 803          | Michael Johnson  |
| 804          | Emily Brown      |
| 805          | David Wilson     |
| 806          | Sarah Martinez   |
| 807          | Robert Taylor    |
| 808          | Jennifer Garcia  |
| 809          | William Anderson |
| 810          | Jessica Thomas   |

```
/*16. report*/
```

```
create table reports(report_ID int, report_type varchar(10),generation_date date, parameters varchar(10),
metrices varchar(10), analysis_result varchar(10), user_ID int, department_ID int);
insert into reports values('1','Inventory','2024-03-18','Stock','Sales','Sufficient','1001','1');
insert into reports values('2','Sales','2024-03-18','Products','Revenue','Good','1001','2');
insert into reports values('3','Production','2024-03-19','Orders','Efficiency','High','1002','2');
insert into reports values('4','Inventory','2024-03-20','Products','Sales','Stable','1002','1');
insert into reports values('5','Sales','2024-03-21','Customers','Revenue','Increasing','1003','2');
insert into reports values('6','Production','2024-03-22','Orders','Efficiency','Improving','1003','2');
insert into reports values('7','Inventory','2024-03-23','Products','Stock','Adequate','1004','1');
insert into reports values('8','Sales','2024-03-24','Products','Revenue','Decreasing','1004','2');
insert into reports values('9','Production','2024-03-25','Orders','Efficiency','Stable','1005','2');
insert into reports values('10','Inventory','2024-03-26','Products','Sales','Decreasing','1005','1');
select*from reports;
```

| report_ID | report_type | generation_date | parameters | metrices   | analysis_result | user_ID | department_ID |
|-----------|-------------|-----------------|------------|------------|-----------------|---------|---------------|
| 1         | Inventory   | 2024-03-18      | Stock      | Sales      | Sufficient      | 1001    | 1             |
| 2         | Sales       | 2024-03-18      | Products   | Revenue    | Good            | 1001    | 2             |
| 3         | Production  | 2024-03-19      | Orders     | Efficiency | High            | 1002    | 2             |
| 4         | Inventory   | 2024-03-20      | Products   | Sales      | Stable          | 1002    | 1             |
| 5         | Sales       | 2024-03-21      | Customers  | Revenue    | Increasing      | 1003    | 2             |
| 6         | Production  | 2024-03-22      | Orders     | Efficiency | Improving       | 1003    | 2             |
| 7         | Inventory   | 2024-03-23      | Products   | Stock      | Adequate        | 1004    | 1             |
| 8         | Sales       | 2024-03-24      | Products   | Revenue    | Decreasing      | 1004    | 2             |
| 9         | Production  | 2024-03-25      | Orders     | Efficiency | Stable          | 1005    | 2             |
| 10        | Inventory   | 2024-03-26      | Products   | Sales      | Decreasing      | 1005    | 1             |

## Project demonstration

```
/*1. Simple query to retrieve all products in the  
'Electronics' category:*/
```

```
SELECT * FROM product WHERE category = 'Electronics';
```

| product_ID | product_name | supplier_ID | unit_price | quantity_in_stock | reorder_level | expiry_date | category    |
|------------|--------------|-------------|------------|-------------------|---------------|-------------|-------------|
| 101        | Widget A     | 1           | 25.99      | 100               | 20            | 2024-12-31  | Electronics |
| 103        | Tool C       | 3           | 30.50      | 150               | 25            | 2024-10-31  | Electronics |
| 105        | Device E     | 5           | 12.99      | 120               | 30            | 2024-09-30  | Electronics |
| 107        | Equipment G  | 7           | 20.75      | 130               | 25            | 2024-11-30  | Electronics |
| 109        | Component I  | 9           | 40.25      | 70                | 20            | 2024-08-31  | Electronics |

```
/*2. Join to get the product name and supplier name for each product:*/
```

```
SELECT p.product_name, s.supplier_name
```

```
FROM product p
```

```
JOIN supplier s ON p.supplier_ID = s.supplier_ID;
```

| product_name | supplier_name     |
|--------------|-------------------|
| Widget A     | ABC Electronics   |
| Gadget B     | XYZ Manufacturing |
| Tool C       | LMN Distributors  |
| Appliance D  | PQR Supplies      |
| Device E     | RST Industries    |
| Gizmo F      | UVW Corporation   |
| Equipment G  | GHI Enterprises   |
| Machine H    | MNO Solutions     |
| Component I  | DEF Supplies      |
| Tool J       | STU Technologies  |

```
/*3 Group by category to count the number of products  
in each category:*/
```

```
SELECT category, COUNT(*) as num_products
```

```
FROM product
```

```
GROUP BY category;
```

| category    | num_products |
|-------------|--------------|
| Electronics | 5            |
| Tools       | 3            |
| Appliances  | 1            |
| Machinery   | 1            |

```
/*4. Subquery to find products with unit price greater
than the average unit price:*/
```

```
SELECT *
FROM product
WHERE unit_price > (SELECT AVG(unit_price) FROM product);
```

| product_ID | product_name | supplier_ID | unit_price | quantity_in_stock | reorder_level | expiry_date | category    |
|------------|--------------|-------------|------------|-------------------|---------------|-------------|-------------|
| 103        | Tool C       | 3           | 30.50      | 150               | 25            | 2024-10-31  | Electronics |
| 104        | Appliance D  | 4           | 50.25      | 80                | 15            | 2025-03-31  | Appliances  |
| 106        | Gizmo F      | 6           | 35.50      | 90                | 20            | 2025-01-31  | Tools       |
| 109        | Component I  | 9           | 40.25      | 70                | 20            | 2024-08-31  | Electronics |

```
/*5. Using LIKE to find products with names containing 'Gadget':*/
```

```
SELECT *
FROM product
WHERE product_name LIKE '%Gadget%';
```

| product_ID | product_name | supplier_ID | unit_price | quantity_in_stock | reorder_level | expiry_date | category |
|------------|--------------|-------------|------------|-------------------|---------------|-------------|----------|
| 102        | Gadget B     | 2           | 15.75      | 200               | 30            | 2025-06-30  | Tools    |

```
/*6. Using BETWEEN to find products with reorder level between
20 and 30:*/
```

```
SELECT *
FROM product
WHERE reorder_level BETWEEN 20 AND 30;
```



|   | product_ID | product_name | supplier_ID | unit_price | quantity_in_stock | reorder_level | expiry_date | category    |
|---|------------|--------------|-------------|------------|-------------------|---------------|-------------|-------------|
| ▶ | 101        | Widget A     | 1           | 25.99      | 100               | 20            | 2024-12-31  | Electronics |
|   | 102        | Gadget B     | 2           | 15.75      | 200               | 30            | 2025-06-30  | Tools       |
|   | 103        | Tool C       | 3           | 30.50      | 150               | 25            | 2024-10-31  | Electronics |
|   | 105        | Device E     | 5           | 12.99      | 120               | 30            | 2024-09-30  | Electronics |
|   | 106        | Gizmo F      | 6           | 35.50      | 90                | 20            | 2025-01-31  | Tools       |
|   | 107        | Equipment G  | 7           | 20.75      | 130               | 25            | 2024-11-30  | Electronics |
|   | 108        | Machine H    | 8           | 18.00      | 110               | 30            | 2025-02-28  | Machinery   |
|   | 109        | Component I  | 9           | 40.25      | 70                | 20            | 2024-08-31  | Electronics |
|   | 110        | Tool J       | 10          | 22.50      | 140               | 25            | 2025-04-30  | Tools       |

/\*7. Using IN to find products supplied by specific suppliers:\*/

```
SELECT *
FROM product
WHERE supplier_ID IN (1, 3, 5);
```

| product_ID | product_name | supplier_ID | unit_price | quantity_in_stock | reorder_level | expiry_date | category    |
|------------|--------------|-------------|------------|-------------------|---------------|-------------|-------------|
| 101        | Widget A     | 1           | 25.99      | 100               | 20            | 2024-12-31  | Electronics |
| 103        | Tool C       | 3           | 30.50      | 150               | 25            | 2024-10-31  | Electronics |
| 105        | Device E     | 5           | 12.99      | 120               | 30            | 2024-09-30  | Electronics |

/\*8. Join to get product details along with location descriptions:\*/

```
SELECT p.*, l.description
FROM product p
JOIN inventory i ON p.product_ID = i.product_ID
JOIN location l ON i.location_ID = l.location_ID;
```

| product_ID | product_name | supplier_ID | unit_price | quantity_in_stock | reorder_level | expiry_date | category    | description                                  |
|------------|--------------|-------------|------------|-------------------|---------------|-------------|-------------|--|
| 101        | Widget A     | 1           | 25.99      | 100               | 20            | 2024-12-31  | Electronics | Main warehouse for finished goods storage.   |
| 102        | Gadget B     | 2           | 15.75      | 200               | 30            | 2025-06-30  | Tools       | Secondary warehouse for overflow storage.    |
| 103        | Tool C       | 3           | 30.50      | 150               | 25            | 2024-10-31  | Electronics | Main production facility for electronics.    |
| 104        | Appliance D  | 4           | 50.25      | 80                | 15            | 2025-03-31  | Appliances  | Secondary production facility for packaging. |
| 105        | Device E     | 5           | 12.99      | 120               | 30            | 2024-09-30  | Electronics | Distribution center for local deliveries.    |
| 106        | Gizmo F      | 6           | 35.50      | 90                | 20            | 2025-01-31  | Tools       | Company-owned retail store in downtown.      |
| 107        | Equipment G  | 7           | 20.75      | 130               | 25            | 2024-11-30  | Electronics | Company-owned retail store in downtown.      |
| 108        | Machine H    | 8           | 18.00      | 110               | 30            | 2025-02-28  | Machinery   | Company-owned retail store in suburban area. |
| 109        | Component I  | 9           | 40.25      | 70                | 20            | 2024-08-31  | Electronics | Headquarters office building.                |
| 110        | Tool J       | 10          | 22.50      | 140               | 25            | 2025-04-30  | Tools       | Branch office building in another city.      |

```

/*9. Join to get the total quantity sold for each product:*/
SELECT p.product_name, SUM(quantity) as total_sold
FROM product p
JOIN transaction t ON p.product_ID = t.product_ID
WHERE t.transaction_type = 'Sale'
GROUP BY p.product_name;

```

| product_name | total_sold |
|--------------|------------|
| Widget A     | 2          |
| Tool C       | 1          |
| Device E     | 4          |
| Equipment G  | 3          |
| Component I  | 1          |

```

/*10. Using NOT to find products not in the 'Electronics' category:*/
SELECT *
FROM product
WHERE category NOT IN ('Electronics');

```

| product_ID | product_name | supplier_ID | unit_price | quantity_in_stock | reorder_level | expiry_date | category   |
|------------|--------------|-------------|------------|-------------------|---------------|-------------|------------|
| 102        | Gadget B     | 2           | 15.75      | 200               | 30            | 2025-06-30  | Tools      |
| 104        | Appliance D  | 4           | 50.25      | 80                | 15            | 2025-03-31  | Appliances |
| 106        | Gizmo F      | 6           | 35.50      | 90                | 20            | 2025-01-31  | Tools      |
| 108        | Machine H    | 8           | 18.00      | 110               | 30            | 2025-02-28  | Machinery  |
| 110        | Tool J       | 10          | 22.50      | 140               | 25            | 2025-04-30  | Tools      |

```

/*11. Using ORDER BY to sort products by unit price in
descending order:*/
SELECT *
FROM product
ORDER BY unit_price DESC;

```

| product_ID | product_name | supplier_ID | unit_price | quantity_in_stock | reorder_level | expiry_date | category    |
|------------|--------------|-------------|------------|-------------------|---------------|-------------|-------------|
| 104        | Appliance D  | 4           | 50.25      | 80                | 15            | 2025-03-31  | Appliances  |
| 109        | Component I  | 9           | 40.25      | 70                | 20            | 2024-08-31  | Electronics |
| 106        | Gizmo F      | 6           | 35.50      | 90                | 20            | 2025-01-31  | Tools       |
| 103        | Tool C       | 3           | 30.50      | 150               | 25            | 2024-10-31  | Electronics |
| 101        | Widget A     | 1           | 25.99      | 100               | 20            | 2024-12-31  | Electronics |
| 110        | Tool J       | 10          | 22.50      | 140               | 25            | 2025-04-30  | Tools       |
| 107        | Equipment G  | 7           | 20.75      | 130               | 25            | 2024-11-30  | Electronics |
| 108        | Machine H    | 8           | 18.00      | 110               | 30            | 2025-02-28  | Machinery   |
| 102        | Gadget B     | 2           | 15.75      | 200               | 30            | 2025-06-30  | Tools       |
| 105        | Device E     | 5           | 12.99      | 120               | 30            | 2024-09-30  | Electronics |

```

/*12. Using LIMIT to get the top 5 products with the highest
unit prices:*/
SELECT *
FROM product
ORDER BY unit_price DESC
LIMIT 5;

```

| product_ID | product_name | supplier_ID | unit_price | quantity_in_stock | reorder_level | expiry_date | category    |
|------------|--------------|-------------|------------|-------------------|---------------|-------------|-------------|
| 104        | Appliance D  | 4           | 50.25      | 80                | 15            | 2025-03-31  | Appliances  |
| 109        | Component I  | 9           | 40.25      | 70                | 20            | 2024-08-31  | Electronics |
| 106        | Gizmo F      | 6           | 35.50      | 90                | 20            | 2025-01-31  | Tools       |
| 103        | Tool C       | 3           | 30.50      | 150               | 25            | 2024-10-31  | Electronics |
| 101        | Widget A     | 1           | 25.99      | 100               | 20            | 2024-12-31  | Electronics |

```

/*13. Using UNION to combine results of two similar queries: */
(SELECT product_name FROM product WHERE category = 'Electronics')
UNION
(SELECT product_name FROM product WHERE category = 'Tools');

```

|   | product_name |
|---|--------------|
| ▶ | Widget A     |
|   | Tool C       |
|   | Device E     |
|   | Equipment G  |
|   | Component I  |
|   | Gadget B     |
|   | Gizmo F      |
|   | Tool J       |



```

/*14. Using JOIN to get product details and transaction details:*/
SELECT p.product_name, t.transaction_type, t.quantity
FROM product p
JOIN transaction t ON p.product_ID = t.product_ID;

```

| product_name | transaction_type | quantity |
|--------------|------------------|----------|
| Widget A     | Sale             | 2        |
| Gadget B     | Purchase         | 5        |
| Tool C       | Sale             | 1        |
| Appliance D  | Purchase         | 3        |
| Device E     | Sale             | 4        |
| Gizmo F      | Purchase         | 2        |
| Equipment G  | Sale             | 3        |
| Machine H    | Purchase         | 6        |
| Component I  | Sale             | 1        |
| Tool J       | Purchase         | 2        |

```

/*15. Using GROUP BY to find the total quantity of each
transaction type:*/
SELECT transaction_type, SUM(quantity) as total_quantity
FROM transaction
GROUP BY transaction_type;

```

| transaction_type | total_quantity |
|------------------|----------------|
| Sale             | 11             |
| Purchase         | 18             |

```

/*16. Using HAVING to filter out transaction types with total quantity
less than 10:*/
SELECT transaction_type, SUM(quantity) as total_quantity
FROM transaction
GROUP BY transaction_type
HAVING total_quantity >= 10;

```

| transaction_type | total_quantity |
|------------------|----------------|
| Sale             | 11             |
| Purchase         | 18             |

```

/*17. Using RANK() to rank products by unit price:*/
SELECT product_name, unit_price, RANK() OVER
(ORDER BY unit_price) as price_rank
FROM product;

```

| product_name | unit_price | price_rank |
|--------------|------------|------------|
| Device E     | 12.99      | 1          |
| Gadget B     | 15.75      | 2          |
| Machine H    | 18.00      | 3          |
| Equipment G  | 20.75      | 4          |
| Tool J       | 22.50      | 5          |
| Widget A     | 25.99      | 6          |
| Tool C       | 30.50      | 7          |
| Gizmo F      | 35.50      | 8          |
| Component I  | 40.25      | 9          |
| Appliance D  | 50.25      | 10         |

```

/*18. Using PARTITION BY to get the average unit price per category:*/
SELECT product_name, category, unit_price,
      AVG(unit_price) OVER (PARTITION BY category) as
      avg_price_per_category
FROM product;

```

| product_name | category    | unit_price | avg_price_per_category |
|--------------|-------------|------------|------------------------|
| Appliance D  | Appliances  | 50.25      | 50.250000              |
| Widget A     | Electronics | 25.99      | 26.096000              |
| Tool C       | Electronics | 30.50      | 26.096000              |
| Device E     | Electronics | 12.99      | 26.096000              |
| Equipment G  | Electronics | 20.75      | 26.096000              |
| Component I  | Electronics | 40.25      | 26.096000              |
| Machine H    | Machinery   | 18.00      | 18.000000              |
| Gadget B     | Tools       | 15.75      | 24.583333              |
| Gizmo F      | Tools       | 35.50      | 24.583333              |
| Tool J       | Tools       | 22.50      | 24.583333              |

```
/*19. Using TOP-N queries to get the top 3
products with the highest quantity in stock:*/
```

```
SELECT *
FROM product
ORDER BY quantity_in_stock DESC
LIMIT 3;
```

| product_ID | product_name | supplier_ID | unit_price | quantity_in_stock | reorder_level | expiry_date | category    |
|------------|--------------|-------------|------------|-------------------|---------------|-------------|-------------|
| 102        | Gadget B     | 2           | 15.75      | 200               | 30            | 2025-06-30  | Tools       |
| 103        | Tool C       | 3           | 30.50      | 150               | 25            | 2024-10-31  | Electronics |
| 110        | Tool J       | 10          | 22.50      | 140               | 25            | 2025-04-30  | Tools       |

```
/*20. Using EXISTS to find products with existing
inventory records:*/
```

```
SELECT * FROM product p
WHERE EXISTS (
    SELECT 1
    FROM inventory i
    WHERE i.product_ID = p.product_ID);
```

| product_ID | product_name | supplier_ID | unit_price | quantity_in_stock | reorder_level | expiry_date | category    |
|------------|--------------|-------------|------------|-------------------|---------------|-------------|-------------|
| 101        | Widget A     | 1           | 25.99      | 100               | 20            | 2024-12-31  | Electronics |
| 102        | Gadget B     | 2           | 15.75      | 200               | 30            | 2025-06-30  | Tools       |
| 103        | Tool C       | 3           | 30.50      | 150               | 25            | 2024-10-31  | Electronics |
| 104        | Appliance D  | 4           | 50.25      | 80                | 15            | 2025-03-31  | Appliances  |
| 105        | Device E     | 5           | 12.99      | 120               | 30            | 2024-09-30  | Electronics |
| 106        | Gizmo F      | 6           | 35.50      | 90                | 20            | 2025-01-31  | Tools       |
| 107        | Equipment G  | 7           | 20.75      | 130               | 25            | 2024-11-30  | Electronics |
| 108        | Machine H    | 8           | 18.00      | 110               | 30            | 2025-02-28  | Machinery   |
| 109        | Component I  | 9           | 40.25      | 70                | 20            | 2024-08-31  | Electronics |
| 110        | Tool J       | 10          | 22.50      | 140               | 25            | 2025-04-30  | Tools       |

```
/*21. Inner Join with Filtering:*/
```

```
SELECT p.product_name, t.transaction_type, t.quantity
FROM product p
INNER JOIN transaction t ON p.product_ID = t.product_ID
WHERE t.quantity > 2;
```

| product_name | transaction_type | quantity |
|--------------|------------------|----------|
| Gadget B     | Purchase         | 5        |
| Appliance D  | Purchase         | 3        |
| Device E     | Sale             | 4        |
| Equipment G  | Sale             | 3        |
| Machine H    | Purchase         | 6        |

*/\*22. Left Join with Null Values:\*/*

```
SELECT p.product_name, t.transaction_type, t.quantity
FROM product p
LEFT JOIN transaction t ON p.product_ID = t.product_ID;
```

| product_name | transaction_type | quantity |
|--------------|------------------|----------|
| Widget A     | Sale             | 2        |
| Gadget B     | Purchase         | 5        |
| Tool C       | Sale             | 1        |
| Appliance D  | Purchase         | 3        |
| Device E     | Sale             | 4        |
| Gizmo F      | Purchase         | 2        |
| Equipment G  | Sale             | 3        |
| Machine H    | Purchase         | 6        |
| Component I  | Sale             | 1        |
| Tool J       | Purchase         | 2        |

*/\*23. Right Join with Null Values:\*/*

```
SELECT p.product_name, t.transaction_type, t.quantity
FROM product p
RIGHT JOIN transaction t ON p.product_ID = t.product_ID;
```



| product_name | transaction_type | quantity |
|--------------|------------------|----------|
| Widget A     | Sale             | 2        |
| Gadget B     | Purchase         | 5        |
| Tool C       | Sale             | 1        |
| Appliance D  | Purchase         | 3        |
| Device E     | Sale             | 4        |
| Gizmo F      | Purchase         | 2        |
| Equipment G  | Sale             | 3        |
| Machine H    | Purchase         | 6        |
| Component I  | Sale             | 1        |
| Tool J       | Purchase         | 2        |

```
/*24. Self-Join to Find Related Products:*/
```

```
SELECT p1.product_name, p2.product_name as related_product
FROM product p1
JOIN product p2 ON p1.category = p2.category AND
p1.product_ID != p2.product_ID;
```

| product_name | related_product |
|--------------|-----------------|
| Component I  | Widget A        |
| Equipment G  | Widget A        |
| Device E     | Widget A        |
| Tool C       | Widget A        |
| Tool J       | Gadget B        |
| Gizmo F      | Gadget B        |
| Component I  | Tool C          |
| Equipment G  | Tool C          |
| Device E     | Tool C          |
| Widget A     | Tool C          |
| Component I  | Device E        |
| Equipment G  | Device E        |
| Tool C       | Device E        |
| Widget A     | Device E        |
| Component I  | Device E        |
| Equipment G  | Device E        |
| Tool C       | Device E        |
| Widget A     | Device E        |
| Tool J       | Gizmo F         |
| Gadget B     | Gizmo F         |



|             |             |
|-------------|-------------|
| Component I | Equipment G |
| Device E    | Equipment G |
| Tool C      | Equipment G |
| Widget A    | Equipment G |
| Equipment G | Component I |
| Device E    | Component I |
| Tool C      | Component I |
| Widget A    | Component I |
| Gizmo F     | Tool J      |
| Gadget B    | Tool J      |

*/\*25. Cross Join to Generate Combinations:\*/*

```
SELECT p.product_name, s.supplier_name
FROM product p
CROSS JOIN supplier s;
```

| product_name | supplier_name   |
|--------------|-----------------|
| Tool C       | UVW Corporation |
| Gadget B     | UVW Corporation |
| Widget A     | UVW Corporation |
| Tool J       | GHI Enterprises |
| Component I  | GHI Enterprises |
| Machine H    | GHI Enterprises |
| Equipment G  | GHI Enterprises |
| Gizmo F      | GHI Enterprises |
| Device E     | GHI Enterprises |
| Appliance D  | GHI Enterprises |
| Tool C       | GHI Enterprises |
| Gadget B     | GHI Enterprises |
| Widget A     | GHI Enterprises |
| Tool J       | MNO Solutions   |
| Component I  | MNO Solutions   |
| Machine H    | MNO Solutions   |
| Equipment G  | MNO Solutions   |
| Gizmo F      | MNO Solutions   |
| Device E     | MNO Solutions   |
| Appliance D  | MNO Solutions   |
| Tool C       | MNO Solutions   |
| Gadget B     | MNO Solutions   |

|             |                  |
|-------------|------------------|
| Widget A    | MNO Solutions    |
| Tool J      | DEF Supplies     |
| Component I | DEF Supplies     |
| Machine H   | DEF Supplies     |
| Equipment G | DEF Supplies     |
| Gizmo F     | DEF Supplies     |
| Device E    | DEF Supplies     |
| Appliance D | DEF Supplies     |
| Tool C      | DEF Supplies     |
| Gadget B    | DEF Supplies     |
| Widget A    | DEF Supplies     |
| Tool J      | STU Technolog... |
| Component I | STU Technolog... |
| Machine H   | STU Technolog... |
| Equipment G | STU Technolog... |
| Gizmo F     | STU Technolog... |
| Device E    | STU Technolog... |
| Appliance D | STU Technolog... |
| Tool C      | STU Technolog... |
| Gadget B    | STU Technolog... |
| Widget A    | STU Technolog... |

```

/*26. **Joining with Multiple Conditions:*/
SELECT p.product_name, i.quantity_on_hand
FROM product p
JOIN inventory i ON p.product_ID = i.product_ID
AND i.quantity_on_hand > 100;

```

| product_name | quantity_on_hand |
|--------------|------------------|
| Gadget B     | 200              |
| Tool C       | 150              |
| Device E     | 120              |
| Equipment G  | 130              |
| Machine H    | 110              |
| Tool J       | 140              |

/\*27. Joining with Subquery:\*/

```
SELECT p.product_name, t.transaction_type, t.quantity
```

```
FROM product p
```

```
JOIN (
```

```
    SELECT * FROM transaction WHERE transaction_date = '2024-03-01'
```

```
) t ON p.product_ID = t.product_ID;
```

| product_name | transaction_type | quantity |
|--------------|------------------|----------|
| Widget A     | Sale             | 2        |

## **Self -Learning beyond classroom**

During the course of the research, We explored several topics that went beyond the confines of a regular classroom. Here are a few of the fresh domains I explored into:

**Real life Application:** Although classroom education imparts fundamental knowledge, working on the project enabled me to put that knowledge to use in a real-world situation. I gained experience navigating issues with data integration, system compatibility, and security.

**Problem solving skills :** Problems are frequently presented in a systematic manner in a classroom. On the other hand, unexpected problems that arose during the project called for inventive problem-solving. I learned how to think critically, adjust to changing conditions, and come up with creative answers from this.

**Team Collaboration:** Working together on a challenging project with colleagues taught me valuable lessons about efficient teamwork. I gained knowledge on how to assign work effectively, express ideas properly, and play to the talents of each team member in order to accomplish shared objectives.

**Project Management:** Three key skills I learned were managing the project timeline, assigning resources, and dividing up tasks among team members. My comprehension of project planning, organisation, and execution in a real-world context has increased.

**Technical Proficiency:** While working on the project improved my technical skills, classroom instruction introduced me to database administration concepts. Through practical experience, I was able to create effective queries, optimise system performance, and build database schemas

## **Learning from the Project**

In simpler terms, we've learned how to organize and keep track of important information about different parts of a business, like its departments and various aspects of its operations. This helps everything run more smoothly because we have all the necessary details at our fingertips. For example, we can easily find out who works where, what products we have in stock, and which customers have placed orders.

Not only that, but we've also improved our coding skills. We've figured out how to create structured systems for storing and managing this data. Imagine it like building a big, organized filing cabinet for all the business's paperwork. Each piece of information has its own place, making it easy to find whenever we need it.

By doing this, we're making things simpler and more efficient. Instead of digging through piles of papers or scrolling endlessly through computer files, we can quickly access the exact information we need. Plus, having everything neatly organized helps prevent mistakes and ensures that nothing important gets overlooked.

Overall, we've taken a big step forward in our ability to handle data effectively. We're better equipped to support the business and keep things running smoothly, thanks to our improved coding skills and structured approach to managing information.

## **Challenges Faced**

One big challenge is making sure that the data in the system is accurate and stays that way, especially when you're dealing with lots of data or different sources of data. It's also tough to get different parts of the system to work together smoothly, especially if they're made with different technologies. Keeping the data safe from hackers and other threats is a big concern too, as well as making sure the system can handle a lot of data without slowing down. And don't forget about managing the project itself—keeping everything on track and making sure everyone involved knows what they're supposed to do can be a challenge too.

## **Conclusion**

In wrapping up, creating this database management system project has been quite the adventure, full of ups and downs. We faced many challenges, but each one taught us something new. From dealing with tricky technical stuff to making sure all the different parts of the project work well together, we learned a ton about software development, managing databases, and keeping everything organized.

Despite the bumps in the road, we managed to build a system that's strong and reliable. We tackled problems like making sure different pieces of software could talk to each other and keeping our data safe from hackers. We also made sure our system could handle lots of data without slowing down, kind of like making sure our car runs smoothly even when it's carrying a heavy load.

Looking ahead, the lessons we've learned will be super helpful for future projects. They'll be like tools in our toolbox, helping us build even better systems in the future. With these experiences under our belts, we're ready to keep growing and coming up with new ideas in the world of database management and software engineering.