1. **Create a Sales Table and Use Aggregate Functions a) Create a Sales table with columns: SaleID, ProductID, Quantity, SaleAmount, and SaleDate. b) Insert at least 10 sales records with different products and quantities. c) Write a query to calculate the total revenue generated using the SUM function. d) Find the product with the highest sale amount using the MAX function. e) Retrieve the average sale amount per transaction using the AVG function. Which recent tool or technology have you studied for database management, and can you briefly explain its key features and why it is used in the industry?**

➔

```
CREATE TABLE Sales (SaleID INT PRIMARY KEY,ProductID INT,Quantity INT,SaleAmount
DECIMAL(10,2),SaleDate DATE);


INSERT INTO Sales VALUES (1, 101, 5, 500.00, TO_DATE('2025-04-01', 'YYYY-MM-DD'));

INSERT INTO Sales VALUES (2, 102, 3, 300.00, TO_DATE('2025-04-02', 'YYYY-MM-DD'));

INSERT INTO Sales VALUES (3, 103, 2, 450.00, TO_DATE('2025-04-03', 'YYYY-MM-DD'));

INSERT INTO Sales VALUES (4, 104, 7, 700.00, TO_DATE('2025-04-04', 'YYYY-MM-DD'));

INSERT INTO Sales VALUES (5, 105, 4, 600.00, TO_DATE('2025-04-05', 'YYYY-MM-DD'));

INSERT INTO Sales VALUES (6, 106, 1, 150.00, TO_DATE('2025-04-06', 'YYYY-MM-DD'));

INSERT INTO Sales VALUES (7, 107, 6, 800.00, TO_DATE('2025-04-07', 'YYYY-MM-DD'));

INSERT INTO Sales VALUES (8, 108, 2, 200.00, TO_DATE('2025-04-08', 'YYYY-MM-DD'));

INSERT INTO Sales VALUES (9, 109, 3, 350.00, TO_DATE('2025-04-09', 'YYYY-MM-DD'));

INSERT INTO Sales VALUES (10, 110, 5, 750.00, TO_DATE('2025-04-10', 'YYYY-MM-DD'));


select * from Sales


SELECT SUM(SaleAmount) AS TotalRevenue FROM Sales;


SELECT ProductID, SaleAmount FROM Sales WHERE SaleAmount = (SELECT MAX(SaleAmount)
FROM Sales);


SELECT AVG(SaleAmount) AS AverageSaleAmount FROM Sales;
```

Recent Database Tool I Studied

Tool: MySQL 8.0

☑ Key Features:

- Improved Performance (faster queries with indexes)
- Data Security (user authentication, SSL)
- JSON Support (store and query JSON data)
- Window Functions (like ROW_NUMBER(), RANK())
- Backup and Recovery Tools
- Replication (creating copies for load balancing)

☑ Why it is used:

- Open-source and free.
- Very reliable for small to enterprise-level applications.
- Widely supported in cloud platforms like AWS, Azure.

2. **Use DDL and DML Commands a) Create a Products table with columns for ProductID, ProductName, Price, and StockQuantity using DDL commands. b) Insert five product records and display all products using a SELECT query. c) Update the price of a product with ProductID = 3 and check the changes using a SELECT statement. d) Delete a product from the table and verify whether the changes are reflected. e) Alter the table to add a new column Discount and set a default value of 5%.**
Which recent tool or technology have you studied for database management, and can you briefly explain its key features and why it is used in the industry?

➔

CREATE TABLE Products (ProductID INT PRIMARY KEY,ProductName VARCHAR2(50),Price DECIMAL(10,2),StockQuantity INT);

INSERT INTO Products VALUES (1, 'Laptop', 55000.00, 20);

INSERT INTO Products VALUES (2, 'Smartphone', 30000.00, 50);

INSERT INTO Products VALUES (3, 'Headphones', 1500.00, 100);

INSERT INTO Products VALUES (4, 'Keyboard', 800.00, 75);

INSERT INTO Products VALUES (5, 'Mouse', 600.00, 90);

SELECT * FROM Products;

UPDATE Products SET Price = 1800.00 WHERE ProductID = 3;

DELETE FROM Products WHERE ProductID = 5;

```
ALTER TABLE Products ADD Discount NUMBER(5,2) DEFAULT 5.00;
```

```
SELECT * FROM Products;
```

Recent Database Management Tool I Studied

Tool: PostgreSQL 15

☑ Key Features:

- Open-source and free.
- Highly reliable and robust for complex applications.
- Advanced indexing (GIN, GiST, BRIN).
- Supports JSON, XML, and full-text search.
- Partitioning for very large tables.
- Extensible: You can define your own functions, data types, operators!

☑ Why it is used:

- PostgreSQL is preferred for large-scale, complex systems where performance, scalability, and customizations are important.
- It's commonly used in startups, big tech companies, and cloud-based applications like AWS, Azure.

3. **Create a Customer Table with Integrity Constraints a) Create a Customers table with constraints: CustomerID (PRIMARY KEY), Email (UNIQUE), Age (CHECK Age > 18). b) Insert a valid customer record and verify that the default country is assigned if not explicitly provided. c) Attempt to insert a customer with an age of 16 and observe the CHECK constraint violation. d) Try inserting two customers with the same email ID and observe the UNIQUE constraint violation. e) Retrieve all customers who are older than 25 and belong to a country other than 'India'.**
   Which recent tool or technology have you studied for database management, and can you briefly explain its key features and why it is used in the industry?

➔

```
CREATE TABLE Customers (CustomerID INT PRIMARY KEY,CustomerName VARCHAR2(50),Email
VARCHAR2(100) UNIQUE,Age INT CHECK (Age > 18),Country VARCHAR2(50) DEFAULT 'India');
```

```
INSERT INTO Customers (CustomerID, CustomerName, Email, Age) VALUES (1, 'John Doe',
'john.doe@example.com', 28);
```

```
select * from Customers


// below command Shows error which we have to show the error

INSERT INTO Customers (CustomerID, CustomerName, Email, Age) VALUES (2, 'Jane Smith',
'jane.smith@example.com', 16);


// Try to insert same email ID Customer

INSERT INTO Customers (CustomerID, CustomerName, Email, Age) VALUES (2, 'Alice Brown',
'alice.brown@example.com', 30);

INSERT INTO Customers (CustomerID, CustomerName, Email, Age) VALUES (3, 'Bob Green',
'alice.brown@example.com', 35);


INSERT INTO Customers (CustomerID, CustomerName, Email, Age, Country) VALUES (4, 'Chris Red',
'chris.red@example.com', 32, 'USA');


SELECT * FROM Customers WHERE Age > 25 AND Country != 'India';
```

Recent Database Management Tool I Studied

Tool: MongoDB Atlas

☑ Key Features:

- NoSQL, document-oriented database (not table-based like SQL).
- Cloud managed: no installation needed.
- Auto-scaling and high availability features.
- Supports flexible schemas — no need for strict table definitions.
- Great for modern web and mobile apps needing fast development.

☑ Why it is used:

- Companies use MongoDB for high-speed, flexible applications like e-commerce, analytics, IoT systems, and real-time apps.
- Cloud services (like Atlas) allow teams to easily deploy and manage large databases globally.

4. **Create a Table with Constraints a) Create an EmployeeDetails table with EmployeeID as the PRIMARY KEY and DepartmentID as a FOREIGN KEY referencing a Department table. b) Insert a valid employee record with an existing DepartmentID, then attempt to insert an employee with a non-existent DepartmentID and observe the constraint violation. c) Insert an employee with a duplicate EmployeeID and check how the primary key constraint prevents duplicate entries. d) Modify the Salary column to have a UNIQUE constraint and attempt to insert two employees with the same salary to test the constraint. e) Write a query to delete an employee from EmployeeDetails and ensure that the deletion does not violate any referential integrity constraints.**
Which recent tool or technology have you studied for database management, and can you briefly explain its key features and why it is used in the industry?

➔

```
CREATE TABLE Department (DepartmentID INT PRIMARY KEY,DepartmentName
VARCHAR2(50));
```

```
INSERT INTO Department VALUES (1, 'HR');
```

```
INSERT INTO Department VALUES (2, 'Finance');
```

```
INSERT INTO Department VALUES (3, 'IT');
```

```
CREATE TABLE EmployeeDetails (EmployeeID INT PRIMARY KEY,EmployeeName
VARCHAR2(50),Salary DECIMAL(10,2),DepartmentID INT,CONSTRAINT
fk_department FOREIGN KEY (DepartmentID) REFERENCES
Department(DepartmentID));
```

```
INSERT INTO EmployeeDetails VALUES (101, 'Rahul Sharma', 50000.00, 2);
```

//Invalid Command

```
INSERT INTO EmployeeDetails VALUES (102, 'Neha Patil', 45000.00, 5);
```

```
INSERT INTO EmployeeDetails VALUES (101, 'Vikas Mehra', 60000.00, 1);
```

//Valid Commands

```
ALTER TABLE EmployeeDetails ADD CONSTRAINT uq_salary UNIQUE (Salary);
```

INSERT INTO EmployeeDetails VALUES (103, 'Kiran Joshi', 70000.00, 3);

//Invalid commands

INSERT INTO EmployeeDetails VALUES (104, 'Priya Nair', 70000.00, 1);

//valid commands

DELETE FROM EmployeeDetails WHERE EmployeeID = 101;

Recent Database Management Tool I Studied

Tool: Amazon RDS (Relational Database Service)

☑ Key Features:

- Fully managed cloud database service for SQL databases like MySQL, PostgreSQL, Oracle, SQL Server.
- Automatic backups, multi-zone replication, automatic patching.
- Highly available and scalable without manual maintenance.
- Monitoring and performance tuning included.

☑ Why it is used:

- Used by companies to save time, reduce cost, and ensure reliability without manually handling databases.
- Ideal for web apps, e-commerce sites, banking systems, large data applications.

5. **Create an Employee Table with Various Columns a) Create a table Employee with attributes: EmployeeID (INT, PRIMARY KEY), Name (VARCHAR), Salary (DECIMAL), JoiningDate (DATE), and ActiveStatus (BOOLEAN). b) Insert five sample employee records and ensure each employee has a unique EmployeeID. c) Write a query to find all employees who joined before January 1, 2023. d) Update the salary of an employee named 'Amit Sharma' by 10% and display the updated record. e) Retrieve all employees who are currently active (ActiveStatus = TRUE).**
   Which recent tool or technology have you studied for database management, and can you briefly explain its key features and why it is used in the industry?
   ➜

```sql
CREATE TABLE Employee (EmployeeID INT PRIMARY KEY,Name
VARCHAR2(50),Salary DECIMAL(10,2),JoiningDate DATE,ActiveStatus CHAR(1));

INSERT INTO Employee VALUES (1, 'Amit Sharma', 50000.00, TO_DATE('2022-05-10',
'YYYY-MM-DD'), 'Y');

INSERT INTO Employee VALUES (2, 'Neha Sinha', 55000.00, TO_DATE('2023-03-20',
'YYYY-MM-DD'), 'Y');

INSERT INTO Employee VALUES (3, 'Raj Patel', 48000.00, TO_DATE('2021-11-15',
'YYYY-MM-DD'), 'N');

INSERT INTO Employee VALUES (4, 'Priya Verma', 60000.00, TO_DATE('2022-09-01',
'YYYY-MM-DD'), 'Y');

INSERT INTO Employee VALUES (5, 'Vikas Gupta', 45000.00, TO_DATE('2020-08-25',
'YYYY-MM-DD'), 'N');

SELECT * FROM Employee WHERE JoiningDate < TO_DATE('2023-01-01', 'YYYY-
MM-DD');

UPDATE Employee SET Salary = Salary * 1.10 WHERE Name = 'Amit Sharma';

SELECT * FROM Employee WHERE Name = 'Amit Sharma';

SELECT * FROM Employee WHERE ActiveStatus = 'Y';
```

Recent Database Management Tool I Studied:

Tool: MongoDB Atlas

☑ Key Features:

- Fully managed cloud database for MongoDB (NoSQL).
- Supports document-based storage (flexible and scalable).
- Auto-sharding, global clusters, automatic backups, and real-time analytics.
- Built-in security, monitoring, and data migration tools.

☑ Why It Is Used in Industry:

- Companies use MongoDB Atlas for high-speed, flexible app development (especially web apps, IoT, AI/ML projects).
- It removes server management headaches — developers can focus on building applications.
- Supports millions of concurrent users and huge datasets easily.

6. **Aggregate Functions (on a single table: Create a Sales table with columns: SaleID, ProductID, ProductName, Quantity, Discount, SaleAmount, and SaleDate.) a) From the Sales table, calculate the total sales amount (SUM) generated in the month of February 2025. b) Find the average (AVG) billing amount from the Sales table to assess customer spending behavior. c) Identify the minimum (MIN) quantity of products sold in any transaction using the Sales table. d) Determine the highest (MAX) discount applied on any sale using the Sales table. e) Use the COUNT function to find how many transactions were recorded in the Sales table for the product "Laptop".**
   Which recent tool or technology have you studied for database management, and can you briefly explain its key features and why it is used in the industry?

➔

```
CREATE TABLE Sales (SaleID INT PRIMARY KEY,ProductID INT,ProductName VARCHAR2(50),Quantity INT,Discount DECIMAL(5,2),SaleAmount DECIMAL(10,2),SaleDate DATE);
```

```
INSERT INTO Sales VALUES (1, 101, 'Laptop', 2, 5.00, 90000.00, TO_DATE('2025-02-10', 'YYYY-MM-DD'));
```

```
INSERT INTO Sales VALUES (2, 102, 'Mobile', 5, 3.00, 50000.00, TO_DATE('2025-02-12', 'YYYY-MM-DD'));
```

```
INSERT INTO Sales VALUES (3, 103, 'Tablet', 3, 2.50, 30000.00, TO_DATE('2025-01-20', 'YYYY-MM-DD'));
```

```
INSERT INTO Sales VALUES (4, 104, 'Laptop', 1, 7.00, 45000.00, TO_DATE('2025-02-15', 'YYYY-MM-DD'));
```

```
INSERT INTO Sales VALUES (5, 105, 'Headphones', 10, 10.00, 15000.00, TO_DATE('2025-02-05', 'YYYY-MM-DD'));
```

```
INSERT INTO Sales VALUES (6, 106, 'Monitor', 4, 4.00, 40000.00, TO_DATE('2025-03-10', 'YYYY-MM-DD'));
```

```
INSERT INTO Sales VALUES (7, 107, 'Keyboard', 6, 3.50, 6000.00, TO_DATE('2025-02-20', 'YYYY-MM-DD'));
```

```
INSERT INTO Sales VALUES (8, 108, 'Mouse', 8, 2.00, 4000.00, TO_DATE('2025-02-22', 'YYYY-MM-DD'));
```

```
INSERT INTO Sales VALUES (9, 109, 'Charger', 7, 1.50, 7000.00, TO_DATE('2025-01-25', 'YYYY-MM-DD'));
```

```
INSERT INTO Sales VALUES (10, 110, 'Laptop', 1, 5.50, 48000.00, TO_DATE('2025-02-25', 'YYYY-MM-DD'));
```

```
Select * from Sales
```

```
SELECT SUM(SaleAmount) AS TotalSalesFebruary FROM Sales WHERE TO_CHAR(SaleDate, 'MM-YYYY') = '02-2025';
```

SELECT AVG(SaleAmount) AS AverageBillingAmount FROM Sales;

SELECT MIN(Quantity) AS MinimumQuantitySold FROM Sales;

SELECT MAX(Discount) AS HighestDiscount FROM Sales;

SELECT COUNT(*) AS LaptopTransactions FROM Sales WHERE ProductName = 'Laptop';

Recent Database Management Tool I Studied:

Tool: PostgreSQL

☑ Key Features:

- Open-source, enterprise-level relational database.

- Supports advanced SQL, full ACID compliance, procedures, functions, and triggers.

- Has powerful features like JSON support, full-text search, geospatial data with PostGIS.

- Very scalable — can handle very large applications with millions of records easily.

☑ Why It Is Used in Industry:

- Companies like Apple, Instagram, Spotify use PostgreSQL because it is free, reliable, and scales very well.

- It is used for banking systems, web apps, data warehouses, and even AI/ML projects needing strong data consistency.

7. **Constraints (on a single table: Employees) a) Create the Employees table with EmployeeID as PRIMARY KEY, Email as UNIQUE, and Salary with a CHECK (Salary > 10000) constraint. b) Add a NOT NULL constraint on the Name column in the Employees table and try inserting a record without the name. c) Add a DEFAULT value 'Active' to the Status column in Employees, and insert a record without specifying the status to verify the default. d) Insert a record into Employees where Salary is less than 10000 to test the CHECK constraint. e) Try inserting two employees with the same Email ID to verify the enforcement of the UNIQUE constraint.**
   Which recent tool or technology have you studied for database management, and can you briefly explain its key features and why it is used in the industry?

➔

```
CREATE TABLE Employees (EmployeeID INT PRIMARY KEY,Name VARCHAR2(100),Email
VARCHAR2(100) UNIQUE,Salary DECIMAL(10,2) CHECK (Salary > 10000),Status VARCHAR2(20)
DEFAULT 'Active');
```

```
ALTER TABLE Employees MODIFY Name VARCHAR2(100) NOT NULL;
```

//Invalid Command

```
INSERT INTO Employees (EmployeeID, Email, Salary) VALUES (1, 'test1@email.com', 15000);
```

//Valid Command

```
INSERT INTO Employees (EmployeeID, Name, Email, Salary) VALUES (2, 'Amit Sharma',
'amit@email.com', 25000);
```

```
SELECT * FROM Employees;
```

//Invalid Command

```
INSERT INTO Employees (EmployeeID, Name, Email, Salary) VALUES (3, 'Rahul Gupta',
'rahul@email.com', 8000);
```

//Invalid Command

```
INSERT INTO Employees (EmployeeID, Name, Email, Salary) VALUES (4, 'Sneha Patil',
'amit@email.com', 30000);
```

Recent Database Management Tool I Studied:

Tool: Oracle Database 21c

☑ Key Features:

- Supports multi-model databases (Relational + JSON + Graph + Spatial).
- Provides automatic indexing, blockchain tables, and in-memory processing.
- Strong security features like encryption, fine-grained access control.
- Very good for large enterprises where data reliability and performance are critical.

☑ Why It Is Used in Industry:

- Oracle is widely used in banking, telecom, healthcare, and government sectors because of its high scalability, reliability, and security.

- It is ideal for critical transaction processing systems.

8. **DDL and DML Commands a) Use DDL commands to create a Library database and define a Books table with fields: BookID, Title, Author, Genre, and Price. b) Insert at least five sample records into the Books table using INSERT (DML) and verify them using a SELECT query. c) A new column PublicationYear needs to be added. Use ALTER TABLE to modify the existing table structure. d) Update the price of all books published before 2020 by increasing 10% using the UPDATE statement. e) Use DELETE to remove all books where the genre is 'Outdated Technology' and validate the change with a SELECT query.**
Which recent tool or technology have you studied for database management, and can you briefly explain its key features and why it is used in the industry?
➔

```
CREATE TABLE Books (BookID INT PRIMARY KEY,Title VARCHAR2(200),Author VARCHAR2(100),Genre VARCHAR2(50),Price DECIMAL(8,2));
```

```
INSERT INTO Books (BookID, Title, Author, Genre, Price) VALUES (1, 'Learning SQL', 'Alan Beaulieu', 'Education', 500.00);
```

```
INSERT INTO Books (BookID, Title, Author, Genre, Price) VALUES (2, 'Modern Web Development', 'Mark Myers', 'Technology', 750.00);
```

```
INSERT INTO Books (BookID, Title, Author, Genre, Price) VALUES (3, 'Data Science Essentials', 'Cathy O\Neil', 'Education', 850.00);
```

```
INSERT INTO Books (BookID, Title, Author, Genre, Price) VALUES (4, 'Ancient Computers', 'John Doe', 'Outdated Technology', 300.00);
```

```
INSERT INTO Books (BookID, Title, Author, Genre, Price) VALUES (5, 'Python Programming', 'Eric Matthes', 'Technology', 650.00);
```

```
SELECT * FROM Books;
```

```
ALTER TABLE Books ADD PublicationYear INT;
```

```
UPDATE Books SET PublicationYear = 2015 WHERE BookID = 1;
```

```
UPDATE Books SET PublicationYear = 2019 WHERE BookID = 2;
```

```
UPDATE Books SET PublicationYear = 2021 WHERE BookID = 3;
```

```
UPDATE Books SET PublicationYear = 2008 WHERE BookID = 4;
```

```
UPDATE Books SET PublicationYear = 2022 WHERE BookID = 5;
```

select * from Books

UPDATE Books SET Price = Price * 1.10 WHERE PublicationYear < 2020;

DELETE FROM Books WHERE Genre = 'Outdated Technology';

SELECT * FROM Books;

Recent Database Management Tool I Studied:

Tool: PostgreSQL 15

☑ Key Features:

- Open-source and enterprise-grade relational database system.

- Support for JSON, XML, Key-Value storage along with traditional relational data.

- Advanced indexing techniques (GIN, GiST, BRIN) for faster search.

- Supports stored procedures, triggers, and custom functions.

- Highly extensible - can add new data types, operators, and index methods.

- Focus on data integrity and ACID compliance (Atomicity, Consistency, Isolation, Durability).

☑ Why It Is Used:

- Popular in startups, tech companies, and financial sectors for scalable, reliable, and cost-effective solutions.

- Works extremely well for complex queries, data analytics, and OLTP (Online Transaction Processing) applications.

9. **DDL and DML Commands (on a single table: Books) a) Create a table Books using DDL with fields: BookID, Title, Author, Price, and StockAvailable. b) Insert 5 book records into the Books table using the INSERT command. c) Modify the structure of Books table by adding a new column Genre using the ALTER TABLE command. d) Use the UPDATE command to increase the price of all books by RS 50 in the Books table. e) Delete all records from the Books table where StockAvailable is 0 using the DELETE command.**
Which recent tool or technology have you studied for database management, and can you briefly explain its key features and why it is used in the industry?

➔

CREATE TABLE Books (BookID INT PRIMARY KEY,Title VARCHAR2(200),Author VARCHAR2(100),Price DECIMAL(8,2),StockAvailable INT);

```sql
INSERT INTO Books (BookID, Title, Author, Price, StockAvailable) VALUES (1, 'Database Systems', 'C.J. Date', 600.00, 10);

INSERT INTO Books (BookID, Title, Author, Price, StockAvailable) VALUES (2, 'Learning SQL', 'Alan Beaulieu', 550.00, 5);

INSERT INTO Books (BookID, Title, Author, Price, StockAvailable) VALUES (3, 'Python Crash Course', 'Eric Matthes', 750.00, 8);

INSERT INTO Books (BookID, Title, Author, Price, StockAvailable) VALUES (4, 'Web Development Basics', 'John Smith', 400.00, 0);

INSERT INTO Books (BookID, Title, Author, Price, StockAvailable) VALUES (5, 'Artificial Intelligence', 'Stuart Russell', 850.00, 2);


ALTER TABLE Books ADD Genre VARCHAR2(50);


UPDATE Books SET Price = Price + 50;


DELETE FROM Books WHERE StockAvailable = 0;
```

Recent Tool or Technology Studied:

Tool: MongoDB (NoSQL Database)

☑ Key Features:

- Document-Oriented: Stores data in flexible, JSON-like documents.

- Schema-less: No need for a strict schema; fields can vary across documents.

- Horizontal Scalability: Easily handles large data volumes with sharding.

- High Performance: Optimized for read/write operations.

- Powerful Query Language: Supports rich queries, aggregations, and indexing.

- Built-in Replication: Provides high availability through replica sets.

☑ Why Used in the Industry:

- Perfect for big data applications, real-time analytics, mobile apps, and IoT solutions.

- Used by companies like Uber, eBay, and Forbes for handling large, complex, and changing datasets efficiently.

10. Analyze Sales Performance Using Aggregate Functions Create a Sales table with columns: SaleID, ProductID, ProductName, Quantity, Discount, SaleAmount, and SalesPerson.) a) Calculate the total quantity of products sold across all transactions in the Sales table. b) Find the average sale amount for transactions made in March 2025. c) Identify the product with the minimum sale quantity from the Sales table. d) Determine the maximum discount offered in February 2025. e) Count how many sales were made by each salesperson using GROUP BY SalesPerson.

Which recent tool or technology have you studied for database management, and can you briefly explain its key features and why it is used in the industry?

➔

```
CREATE TABLE Sales (SaleID INT PRIMARY KEY,ProductID INT,ProductName VARCHAR2(100),Quantity
INT,Discount DECIMAL(5,2),SaleAmount DECIMAL(10,2),SaleDate DATE,SalesPerson
VARCHAR2(100));
```

```
INSERT INTO Sales (SaleID, ProductID, ProductName, Quantity, Discount, SaleAmount, SaleDate,
SalesPerson)

VALUES (1, 101, 'Laptop', 5, 10.00, 50000, TO_DATE('2025-03-05', 'YYYY-MM-DD'), 'John');
```

```
INSERT INTO Sales (SaleID, ProductID, ProductName, Quantity, Discount, SaleAmount, SaleDate,
SalesPerson)

VALUES (2, 102, 'Smartphone', 3, 5.00, 30000, TO_DATE('2025-02-15', 'YYYY-MM-DD'), 'Jane');
```

```
INSERT INTO Sales (SaleID, ProductID, ProductName, Quantity, Discount, SaleAmount, SaleDate,
SalesPerson)

VALUES (3, 103, 'Tablet', 7, 12.00, 35000, TO_DATE('2025-03-18', 'YYYY-MM-DD'), 'John');
```

```
INSERT INTO Sales (SaleID, ProductID, ProductName, Quantity, Discount, SaleAmount, SaleDate,
SalesPerson)

VALUES (4, 104, 'Headphones', 4, 8.00, 16000, TO_DATE('2025-02-25', 'YYYY-MM-DD'), 'Alice');
```

```
INSERT INTO Sales (SaleID, ProductID, ProductName, Quantity, Discount, SaleAmount, SaleDate,
SalesPerson)

VALUES (5, 105, 'Monitor', 6, 15.00, 18000, TO_DATE('2025-03-01', 'YYYY-MM-DD'), 'Alice');
```

```
SELECT SUM(Quantity) AS TotalQuantitySold FROM Sales;
```

SELECT AVG(SaleAmount) AS AverageSaleAmount FROM Sales WHERE SaleDate BETWEEN TO_DATE('2025-03-01', 'YYYY-MM-DD') AND TO_DATE('2025-03-31', 'YYYY-MM-DD');

SELECT ProductName, MIN(Quantity) AS MinQuantitySold FROM Sales GROUP BY ProductName;

SELECT MAX(Discount) AS MaxDiscount FROM Sales WHERE SaleDate BETWEEN TO_DATE('2025-02-01', 'YYYY-MM-DD') AND TO_DATE('2025-02-28', 'YYYY-MM-DD');

SELECT SalesPerson, COUNT(SaleID) AS TotalSales FROM Sales GROUP BY SalesPerson;

Recent Tool or Technology Studied for Database Management

Tool: Microsoft SQL Server

Key Features:

- **Transaction Control**: Supports ACID properties to ensure data integrity.
- **Advanced Query Optimization**: Includes the Query Optimizer for faster execution of SQL queries.
- **Scalability**: Suitable for both small-scale and enterprise-level applications.
- **Security**: Offers robust security features like encryption, authentication, and row-level security.
- **In-memory OLTP**: Provides high performance for transactional workloads by storing data in memory.

Why it is used in the industry:

- It's widely used in enterprise environments because of its **reliability**, **robust security features**, and **scalability**.
- SQL Server is a popular choice for **data warehousing**, **business intelligence (BI)**, and **transactional systems** in organizations of various sizes.