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 NUMBER PRIMARY KEY, ProductID NUMBER, Quantity NUMBER, SaleAmount NUMBER(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, 2, 300.00, TO\_DATE('2025-04-02', 'YYYY-MM-DD'));

INSERT INTO Sales VALUES (3, 103, 1, 150.00, TO\_DATE('2025-04-03', 'YYYY-MM-DD'));

INSERT INTO Sales VALUES (4, 101, 3, 300.00, TO\_DATE('2025-04-04', 'YYYY-MM-DD'));

INSERT INTO Sales VALUES (5, 104, 4, 800.00, TO\_DATE('2025-04-05', 'YYYY-MM-DD'));

INSERT INTO Sales VALUES (6, 105, 2, 400.00, TO\_DATE('2025-04-06', 'YYYY-MM-DD'));

INSERT INTO Sales VALUES (7, 103, 6, 900.00, TO\_DATE('2025-04-07', 'YYYY-MM-DD'));

INSERT INTO Sales VALUES (8, 102, 5, 750.00, TO\_DATE('2025-04-08', 'YYYY-MM-DD'));

INSERT INTO Sales VALUES (9, 106, 7, 1050.00, TO\_DATE('2025-04-09', 'YYYY-MM-DD'));

INSERT INTO Sales VALUES (10, 107, 3, 600.00, TO\_DATE('2025-04-10', 'YYYY-MM-DD'));

SELECT \* from Sales

SELECT SUM(SaleAmount) AS Total\_Revenue FROM Sales;

SELECT MAX(SaleAmount) AS Highest\_Sale\_Amount FROM Sales;

SELECT AVG(SaleAmount) AS Average\_Sale\_Amount FROM Sales;

SELECT \* 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%.

CREATE TABLE Products (ProductID NUMBER PRIMARY KEY, ProductName VARCHAR2(100), Price NUMBER(10, 2), StockQuantity NUMBER);

INSERT INTO Products (ProductID, ProductName, Price, StockQuantity)

VALUES (1, 'Laptop', 50000.00, 20);

INSERT INTO Products (ProductID, ProductName, Price, StockQuantity)

VALUES (2, 'Smartphone', 20000.00, 50);

INSERT INTO Products (ProductID, ProductName, Price, StockQuantity)

VALUES (3, 'Headphones', 1500.00, 100);

INSERT INTO Products (ProductID, ProductName, Price, StockQuantity)

VALUES (4, 'Keyboard', 800.00, 30);

INSERT INTO Products (ProductID, ProductName, Price, StockQuantity)

VALUES (5, 'Monitor', 10000.00, 15);

SELECT \* FROM Products;

UPDATE Products SET Price = 1800.00 WHERE ProductID = 3;

SELECT \* FROM Products WHERE ProductID = 3;

DELETE FROM Products WHERE ProductID = 5;

SELECT \* FROM Products;

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

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'.

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

INSERT INTO Customers (CustomerID, CustomerName, Email, Age)

VALUES (1, 'John Doe', 'john.doe@example.com', 30);

SELECT \* FROM Customers WHERE CustomerID = 1;

INSERT INTO Customers (CustomerID, CustomerName, Email, Age)

VALUES (2, 'Alice Smith', 'alice.smith@example.com', 16);

INSERT INTO Customers (CustomerID, CustomerName, Email, Age)

VALUES (3, 'Bob Johnson', 'bob.johnson@example.com', 25);

INSERT INTO Customers (CustomerID, CustomerName, Email, Age)

VALUES (4, 'Charlie Brown', 'bob.johnson@example.com', 35);

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

SELECT \* FROM Customers

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.

CREATE TABLE Department (DepartmentID NUMBER PRIMARY KEY, DepartmentName VARCHAR2(100));

CREATE TABLE EmployeeDetails (EmployeeID NUMBER PRIMARY KEY, EmployeeName VARCHAR2(100), DepartmentID NUMBER, Salary NUMBER(10, 2), FOREIGN KEY (DepartmentID) REFERENCES Department (DepartmentID));

INSERT INTO Department (DepartmentID, DepartmentName)

VALUES (1, 'HR');

INSERT INTO EmployeeDetails (EmployeeID, EmployeeName, DepartmentID, Salary)

VALUES (101, 'John Doe', 1, 50000);

INSERT INTO EmployeeDetails (EmployeeID, EmployeeName, DepartmentID, Salary)

VALUES (102, 'Jane Smith', 999, 60000);

INSERT INTO EmployeeDetails (EmployeeID, EmployeeName, DepartmentID, Salary)

VALUES (101, 'Alice Brown', 1, 55000);

ALTER TABLE EmployeeDetails ADD CONSTRAINT unique\_salary UNIQUE (Salary);

INSERT INTO EmployeeDetails (EmployeeID, EmployeeName, DepartmentID, Salary)

VALUES (103, 'Bob Johnson', 1, 70000);

INSERT INTO EmployeeDetails (EmployeeID, EmployeeName, DepartmentID, Salary)

VALUES (104, 'Charlie Lee', 2, 70000);

DELETE FROM EmployeeDetails WHERE EmployeeID = 101;

SELECT \* FROM EmployeeDetails;

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).

CREATE TABLE Employee ( EmployeeID NUMBER PRIMARY KEY, Name VARCHAR2(100), Salary NUMBER(10, 2), JoiningDate DATE, ActiveStatus CHAR(1) CHECK (ActiveStatus IN ('Y', 'N')));

INSERT INTO Employee (EmployeeID, Name, Salary, JoiningDate, ActiveStatus)

VALUES (1, 'Amit Sharma', 60000.00, TO\_DATE('2020-05-15', 'YYYY-MM-DD'), 'Y');

select \* from Employee;

INSERT INTO Employee (EmployeeID, Name, Salary, JoiningDate, ActiveStatus)

VALUES (2, 'Priya Verma', 55000.00, TO\_DATE('2021-03-20', 'YYYY-MM-DD'), 'Y');

INSERT INTO Employee (EmployeeID, Name, Salary, JoiningDate, ActiveStatus)

VALUES (3, 'Ravi Kumar', 48000.00, TO\_DATE('2022-01-10', 'YYYY-MM-DD'), 'N');

INSERT INTO Employee (EmployeeID, Name, Salary, JoiningDate, ActiveStatus)

VALUES (4, 'Sneha Gupta', 70000.00, TO\_DATE('2019-08-05', 'YYYY-MM-DD'), 'Y');

INSERT INTO Employee (EmployeeID, Name, Salary, JoiningDate, ActiveStatus)

VALUES (5, 'Vijay Singh', 62000.00, TO\_DATE('2020-11-12', 'YYYY-MM-DD'), 'Y');

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”.

CREATE TABLE Sale ( SaleID NUMBER PRIMARY KEY, ProductID NUMBER, ProductName VARCHAR2(100), Quantity NUMBER, Discount NUMBER(5,2), SaleAmount NUMBER(10, 2), SaleDate DATE);

INSERT INTO Sale VALUES (1, 101, 'Laptop', 2, 10.50, 1500.00, TO\_DATE('2025-02-15', 'YYYY-MM-DD'));

INSERT INTO Sale VALUES (2, 102, 'Mouse', 5, 5.00, 250.00, TO\_DATE('2025-02-20', 'YYYY-MM-DD'));

INSERT INTO Sale VALUES (3, 101, 'Laptop', 1, 15.00, 1200.00, TO\_DATE('2025-03-01', 'YYYY-MM-DD'));

INSERT INTO Sale VALUES (4, 103, 'Keyboard', 3, 8.00, 450.00, TO\_DATE('2025-01-28', 'YYYY-MM-DD'));

SELECT SUM(SaleAmount) AS TotalSale FROM Sale WHERE SaleDate BETWEEN TO\_DATE('2025-02-01', 'YYYY-MM-DD')

AND TO\_DATE('2025-02-28', 'YYYY-MM-DD');

SELECT AVG(SaleAmount) AS AverageSaleAmount FROM Sale;

SELECT MIN(Quantity) AS MinQuantitySold FROM Sale;

SELECT MAX(Discount) AS MaxDiscount FROM Sale;

SELECT COUNT(\*) AS LaptopTransactions FROM Sale WHERE UPPER(ProductName) = 'LAPTOP';

select \* from Sale;

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.

CREATE TABLE Employees ( EmployeeID NUMBER PRIMARY KEY, Name VARCHAR2(100) NOT NULL, Email VARCHAR2(100) UNIQUE, Salary NUMBER(10, 2) CHECK (Salary > 10000), Status VARCHAR2(10) DEFAULT 'Active');

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

INSERT INTO Employees (EmployeeID, Email, Salary) VALUES (1, 'john.doe@example.com', 15000.00);

ALTER TABLE Employees MODIFY Status VARCHAR2(10) DEFAULT 'Active';

INSERT INTO Employees (EmployeeID, Name, Email, Salary) VALUES (2, 'Jane Smith', 'jane.smith@example.com', 25000.00);

SELECT \* FROM Employees WHERE EmployeeID = 2;

INSERT INTO Employees (EmployeeID, Name, Email, Salary, Status) VALUES (3, 'Tom Brown', 'tom.brown@example.com', 8000.00, 'Active');

INSERT INTO Employees (EmployeeID, Name, Email, Salary, Status) VALUES (4, 'Emily White', 'emily.white@example.com', 12000.00, 'Active');

INSERT INTO Employees (EmployeeID, Name, Email, Salary, Status) VALUES (5, 'John Green', 'emily.white@example.com', 13000.00, 'Active');

SELECT \* FROM Employees;

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.

CREATE TABLE Books ( BookID NUMBER PRIMARY KEY, Title VARCHAR2(100), Author VARCHAR2(100), Genre VARCHAR2(50), Price NUMBER(10, 2));

INSERT INTO Books (BookID, Title, Author, Genre, Price)

VALUES (1, 'The Great Gatsby', 'F. Scott Fitzgerald', 'Classic', 250.00);

INSERT INTO Books (BookID, Title, Author, Genre, Price)

VALUES (2, '1984', 'George Orwell', 'Dystopian', 300.00);

INSERT INTO Books (BookID, Title, Author, Genre, Price)

VALUES (3, 'To Kill a Mockingbird', 'Harper Lee', 'Fiction', 350.00);

INSERT INTO Books (BookID, Title, Author, Genre, Price)

VALUES (4, 'The Catcher in the Rye', 'J.D. Salinger', 'Classic', 200.00);

INSERT INTO Books (BookID, Title, Author, Genre, Price)

VALUES (5, 'Sapiens', 'Yuval Noah Harari', 'Non-Fiction', 400.00);

SELECT \* FROM Books;

ALTER TABLE Books ADD PublicationYear NUMBER(4);

DESCRIBE Books;

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

SELECT \* FROM Books;

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.

CREATE TABLE Book ( BookID NUMBER PRIMARY KEY, Title VARCHAR2(100), Author VARCHAR2(100), Price NUMBER(10, 2), StockAvailable NUMBER);

INSERT INTO Book (BookID, Title, Author, Price, StockAvailable)

VALUES (1, 'The Great Gatsby', 'F. Scott Fitzgerald', 250.00, 10);

INSERT INTO Book (BookID, Title, Author, Price, StockAvailable)

VALUES (2, '1984', 'George Orwell', 300.00, 5);

INSERT INTO Book (BookID, Title, Author, Price, StockAvailable)

VALUES (3, 'To Kill a Mockingbird', 'Harper Lee', 350.00, 8);

INSERT INTO Book (BookID, Title, Author, Price, StockAvailable)

VALUES (4, 'The Catcher in the Rye', 'J.D. Salinger', 200.00, 0);

INSERT INTO Book (BookID, Title, Author, Price, StockAvailable)

VALUES (5, 'Sapiens', 'Yuval Noah Harari', 400.00, 3);

ALTER TABLE Book ADD Genre VARCHAR2(50);

DESCRIBE Book;

UPDATE Book SET Price = Price + 50;

SELECT \* FROM Book;

DELETE FROM Book WHERE StockAvailable = 0;

SELECT \* FROM Book;

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.

CREATE TABLE Salesss (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 Salesss (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 Salesss (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 Salesss (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 Salesss (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 Salesss (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 Salesss;

SELECT AVG(SaleAmount) AS AverageSaleAmount FROM Salesss 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 Salesss GROUP BY ProductName;

SELECT MAX(Discount) AS MaxDiscount FROM Salesss 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 Salesss GROUP BY SalesPerson;

SELECT \* FROM Salesss;

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.