Task 2

**1)Create a user-defined database named EmployeeRecords with specific file paths for the .mdf and .ldf files.**

CREATE DATABASE EmployeeRecords;

**2) Rename the EmployeeRecords database to HR\_Database using T-SQL.**

USE master;

ALTER DATABASE EmployeeRecords MODIFY NAME = HR\_Database;

**3)Drop the HR\_Database safely, ensuring there are no active connections before deletion.**

ALTER DATABASE HR\_Database SET SINGLE\_USER WITH ROLLBACK IMMEDIATE;

DROP DATABASE HR\_Database;

**4)Identify at least five commonly used data types in SQL Server and explain their use cases.**

CREATE TABLE Customers (

CustomerID INT PRIMARY KEY,

FullName VARCHAR(100),

Email VARCHAR(100),

Phone VARCHAR(15),

DateJoined DATE

);

**5)Create a Customers table with appropriate columns (CustomerID, FullName, Email, Phone, DateJoined).**

CREATE TABLE Customers (

CustomerID INT PRIMARY KEY,

FullName VARCHAR(100),

Email VARCHAR(100),

Phone VARCHAR(15),

DateJoined DATE,

Address VARCHAR(200)

);

**6)Modify the Customers table to add a new column Address.**

ALTER TABLE Customers

ADD Address VARCHAR(200);

**7)Rename the Customers table to ClientDetails.**

EXEC sp\_rename 'Customers', 'ClientDetails';

**8)Drop the ClientDetails table safely.**

DROP TABLE ClientDetails;

**9)Insert five sample records into the Customers table.**

**10)Update the email of a customer whose CustomerID = 3.**

UPDATE Customers

SET Email = 'james.smith@university.edu'

WHERE CustomerID = 3;

**11)Delete a customer record where the CustomerID = 5.**

DELETE FROM Customers

WHERE CustomerID = 5;

**12)Demonstrate inserting multiple records in a single query for better efficiency.**

INSERT INTO Customers (CustomerID, FullName, Email, Phone, DateJoined, Address)

VALUES

(6, 'Julia ', 'julia@gmail.com', '3334445566', '2023-01-25', 'Miami'),

(7, 'John ', 'john@gmail.com', NULL, '2017-12-10', 'New York');

**13)Retrieve and display only the FullName and Email of all customers.**

SELECT FullName, Email FROM Customers;

**14)Retrieve all customers who joined after 2020-01-01.**

SELECT \* FROM Customers

WHERE DateJoined > '2020-01-01';

**15)Fetch all customers whose names start with 'J' using a LIKE query.**

SELECT \* FROM Customers

WHERE FullName LIKE 'J%';

**16)Retrieve customers where Phone is NULL (i.e., customers who haven't provided a phone number).**

SELECT \* FROM Customers

WHERE Phone IS NULL;

**17)Filter customers using IN—Retrieve records where CustomerID is either 1, 3, or 7.**

SELECT \* FROM Customers

WHERE CustomerID IN (1, 3, 7);

**18)Use DISTINCT to list unique domain names from customer emails (e.g., gmail.com, yahoo.com).**

SELECT DISTINCT

RIGHT(Email, LEN(Email) - CHARINDEX('@', Email)) AS EmailDomain

FROM Customers;

**19)Use AND and OR together—Retrieve customers who either live in 'New York' OR have joined before 2019-06-01.**

SELECT \* FROM Customers

WHERE Address = 'New York' OR DateJoined < '2019-06-01';

**20)Retrieve customers where DateJoined is BETWEEN 2018-01-01 AND 2023-12-31.**

SELECT \* FROM Customers

WHERE DateJoined BETWEEN '2018-01-01' AND '2023-12-31';

**21)Use column and table aliases to rename output fields while selecting.**

SELECT

C.FullName AS Name,

C.Email AS EmailAddress,

C.Address AS City

FROM Customers AS C;

**22)Demonstrate a query that filters using multiple conditions, such as WHERE Age > 25 AND City = 'Chicago'.**

SELECT \* FROM Customers

WHERE Age > 25 AND Address = 'Chicago';

**23)Execute and analyze filtering queries to optimize performance using EXPLAIN plans (if applicable).**