**DATABASE MANGEMENT SYSTEMS LAB**

**ASSIGNMENT -2**

**1.** **Create a table for product information: Write a query to create a table**

**named Products with the following fields:**

** product\_id (Integer, Primary Key)**

** product\_name (Varchar, 50)**

** category (Varchar, 30)**

** price (Decimal, 10, 2)**

** stock\_quanƟty (Integer)**

ANS: DROP TABLE IF EXISTS Products;

DROP TABLE IF EXISTS Customer\_Details;

DROP TABLE IF EXISTS Orders;

CREATE TABLE Products (

product\_id INT PRIMARY KEY,

product\_name VARCHAR(50),

category VARCHAR(30),

price DECIMAL(10, 2),

stock\_quantity INT

);

**Add a column to an existing table: You have an existing table Customer\_Details. Write a query to add a new column email to the table with a maximum length of 50 characters.**

CREATE TABLE Customer\_Details (

customer\_id INT PRIMARY KEY,

customer\_name VARCHAR(50),

phone\_number VARCHAR(15) -- Initial length of phone\_number);

ALTER TABLE Customer\_Details

ADD email VARCHAR(50);

**Modify an existing column in a table: You need to modify the phone\_number column in the Customer\_Details table to increase its length from 15 to 20 characters. Write the query to do this.**

ALTER TABLE Customer\_Details

MODIFY phone\_number VARCHAR(20);

**Delete records from a table: Write a query to delete all records from the Products table where the stock\_quanƟty is less than 5.**

INSERT INTO Products (product\_id, product\_name, category, price, stock\_quantity) VALUES

(1, 'Product A', 'Category 1', 10.00, 5),

(2, 'Product B', 'Category 1', 15.00, 3),

(3, 'Product C', 'Category 2', 20.00, 10);

INSERT INTO Customer\_Details (customer\_id, customer\_name, phone\_number, email) VALUES

(1, 'John Doe', '1234567890', 'john@example.com'),

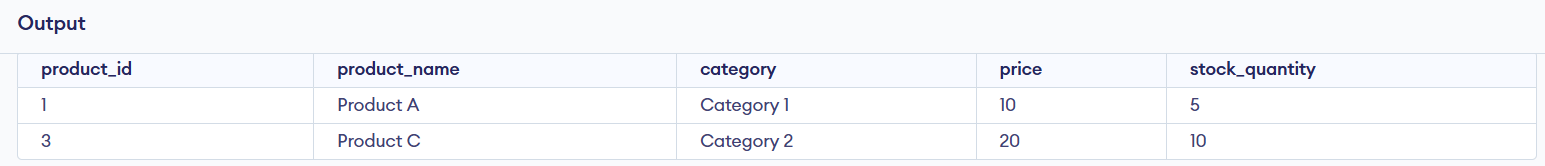
(2, 'Jane Smith', '0987654321', 'jane@example.com');

DELETE FROM Products

WHERE stock\_quantity < 5;

**Drop a table from the database: Write a query to delete (drop) the entire Orders table from the database, assuming this table contains order details.**

DROP TABLE Customer\_Details;



**2. Create a table for employee information: Write a query to create a table**

**named Employees with the following fields:**

** employee\_id (Integer, Primary Key)**

** first\_name (Varchar, 30)**

** last\_name (Varchar, 30)**

** department (Varchar, 30)**

** salary (Decimal, 10, 2)**

** date\_of\_joining (Date)**

**Ans:**

DROP TABLE IF EXISTS Employees;

DROP TABLE IF EXISTS Orders;

DROP TABLE IF EXISTS Products;

CREATE TABLE Employees (employee\_id INT PRIMARY KEY,first\_name VARCHAR(30),last\_name VARCHAR(30),department VARCHAR(30),salary DECIMAL(10, 2),date\_of\_joining DATE

);

**Alter a table to add a new column: The Orders table currently does not track shipment status. Write a query to add a new column shipment\_status (Varchar, 20) to track whether the order is "Shipped", "Pending", or "Delivered".**

CREATE TABLE Orders ( order\_id INT PRIMARY KEY, order\_date DATE, customer\_id INT );

ALTER TABLE Orders

ADD shipment\_status VARCHAR(20);

**Delete specific records from a table: You have a table Employees where some employees have not been acƟve for a long Ɵme. Write a query to delete all employees whose date\_of\_joining is before '2015-01-01' and whose salary is less than 3000.**

INSERT INTO Employees (employee\_id, first\_name, last\_name, department, salary, date\_of\_joining)

VALUES

(1, 'Alice', 'Johnson', 'HR', 3500.00, '2014-06-15'),

(2, 'Bob', 'Smith', 'IT', 2800.00, '2013-04-20'),

(3, 'Charlie', 'Brown', 'Finance', 4000.00, '2016-02-01'),

(4, 'Diana', 'Prince', 'IT', 4500.00, '2018-08-10');

DELETE FROM Employees

WHERE date\_of\_joining < '2015-01-01' AND salary < 3000;

**Modify an existing column in a table: You have a Products table where the price column needs to have a higher precision to account for discount rates. Write a query to modify the price column from DECIMAL(10,2) to DECIMAL(12,4).**

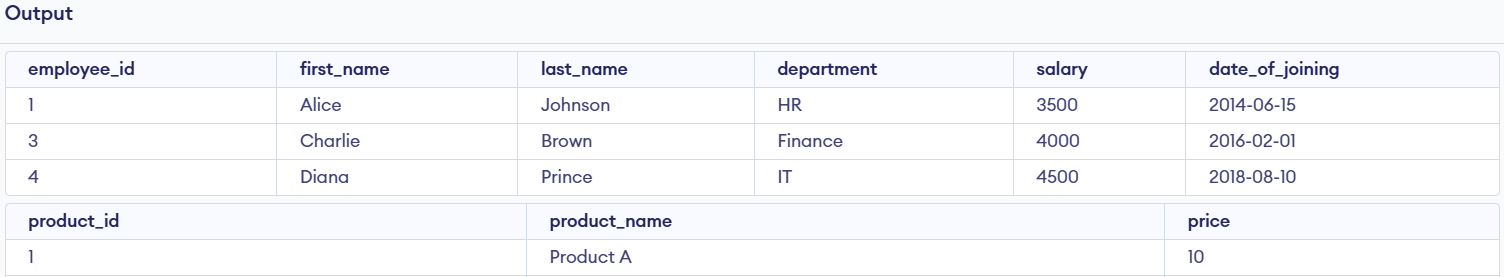
CREATE TABLE Products ( product\_id INT PRIMARY KEY, product\_name VARCHAR(50), price DECIMAL(10, 2) );

ALTER TABLE Products

MODIFY price DECIMAL(12, 4);

INSERT INTO Products (product\_id, product\_name, price) VALUES

(1, 'Product A', 10.00),



**3. Create a table for order details: Write a query to create a table named**

**Order\_Details with the following fields:**

** order\_id (Integer, Primary Key)**

** customer\_id (Integer, Foreign Key referencing Customer\_Details)**

** product\_id (Integer, Foreign Key referencing Products)**

** order\_date (Date)**

** quantity (Integer)**

DROP TABLE IF EXISTS Order\_Details;

DROP TABLE IF EXISTS Orders;

DROP TABLE IF EXISTS Customer\_Details;

DROP TABLE IF EXISTS Products;

DROP TABLE IF EXISTS Employees;

CREATE TABLE Products ( product\_id INT PRIMARY KEY, product\_name VARCHAR(50), price DECIMAL(10, 2) );

CREATE TABLE Customer\_Details ( customer\_id INT PRIMARY KEY, first\_name VARCHAR(30), middle\_name VARCHAR(30),last\_name VARCHAR(30) );

CREATE TABLE Orders ( order\_id INT PRIMARY KEY, customer\_id INT, order\_date DATE, FOREIGN KEY (customer\_id) REFERENCES Customer\_Details(customer\_id) );

CREATE TABLE Order\_Details ( order\_id INT PRIMARY KEY, customer\_id INT, product\_id INT, order\_date DATE, quantity INT, FOREIGN KEY (customer\_id) REFERENCES Customer\_Details(customer\_id), FOREIGN KEY (product\_id) REFERENCES Products(product\_id) );

**Alter a table to modify an existing column: The Employee table has a column department which is currently of type Varchar(30). You want to increase the length to 50 characters. Write a query to alter the department column to Varchar(50).**

CREATE TABLE Employees ( employee\_id INT PRIMARY KEY, first\_name VARCHAR(30), last\_name VARCHAR(30), department VARCHAR(50), -- Increased length to 50 salary DECIMAL(10, 2), date\_of\_joining DATE );

ALTER TABLE Employee

MODIFY department VARCHAR(50);

**Delete specific records based on a condition: You have a table Orders that records customer orders. Write a query to delete all orders where the order\_date is before '2020-01-01'.**

INSERT INTO Products (product\_id, product\_name, price) VALUES (1, 'Product A', 10.00), (2, 'Product B', 15.00), (3, 'Product C', 20.00);

INSERT INTO Customer\_Details (customer\_id, first\_name, last\_name) VALUES (1, 'John', Stephen”,'Doe'), (2, 'Jane',’marry’, 'Doe');

INSERT INTO Orders (order\_id, customer\_id, order\_date) VALUES (1, 1, '2021-05-01'), (2, 2, '2019-12-15'), (3, 1, '2022-03-10');

INSERT INTO Order\_Details (order\_id, customer\_id, product\_id, order\_date, quantity) VALUES (1, 1, 1, '2021-05-01', 2), (2, 2, 2, '2019-12-15', 1), (3, 1, 3, '2022-03-10', 5);

DELETE FROM Orders

WHERE order\_date < '2020-01-01';

**Drop a column from an existing table: The Customer\_Details table contains an unnecessary column middle\_name. Write a query to drop the middle\_name column from the Customer\_Details table.**

ALTER TABLE Customer\_Details

DROP COLUMN middle\_name;

**Create a table for aƩendance records: Write a query to create a table**

**named AƩendance with the following fields:**

** aƩendance\_id (Integer, Primary Key)**

** employee\_id (Integer, Foreign Key referencing Employees)**

** aƩendance\_date (Date)**

** status (Varchar(10), 'Present' or 'Absent')**

** remarks (Varchar(100))**

**Ans:**

DROP TABLE IF EXISTS Order\_Details;

DROP TABLE IF EXISTS Products;

DROP TABLE IF EXISTS Customer\_Details;

DROP TABLE IF EXISTS Attendance;

DROP TABLE IF EXISTS Employees;

DROP TABLE IF EXISTS Suppliers;

DROP TABLE IF EXISTS Orders;

CREATE TABLE Employees ( employee\_id INTEGER PRIMARY KEY, employee\_name VARCHAR(100) );

CREATE TABLE Attendance ( attendance\_id INTEGER PRIMARY KEY, employee\_id INTEGER, attendance\_date DATE, status VARCHAR(10) CHECK (status IN ('Present', 'Absent')), remarks VARCHAR(100), FOREIGN KEY (employee\_id) REFERENCES Employees(employee\_id));

**Alter a table to rename a column: The Order\_Details table has a column quantity which should be renamed to order\_quantity. Write a query to rename the quantity column to order\_quantity**

CREATE TABLE Orders ( order\_id INTEGER PRIMARY KEY, order\_date DATE );

CREATE TABLE Suppliers ( supplier\_id INTEGER PRIMARY KEY, supplier\_name VARCHAR(100) );

CREATE TABLE Products ( product\_id INTEGER PRIMARY KEY, product\_name VARCHAR(100), price DECIMAL(10, 2), stock\_quantity INTEGER, supplier\_id INTEGER, FOREIGN KEY (supplier\_id) REFERENCES Suppliers(supplier\_id) );

CREATE TABLE Order\_Details ( order\_detail\_id INTEGER PRIMARY KEY, order\_id INTEGER, product\_id INTEGER, quantity INTEGER, price DECIMAL(10, 2), FOREIGN KEY (order\_id) REFERENCES Orders(order\_id), FOREIGN KEY (product\_id) REFERENCES Products(product\_id) );

ALTER TABLE Order\_Details

CHANGE COLUMN quantity order\_quantity INT;

**Delete duplicate records from a table: You have a Customer\_Details table that contains duplicate customer names. Write a query to delete duplicate rows based on the customer\_name column, keeping only one record for each customer**

CREATE TABLE Customer\_Details ( customer\_id INTEGER PRIMARY KEY, customer\_name VARCHAR(100), contact\_info VARCHAR(100) );

INSERT INTO Customer\_Details (customer\_id, customer\_name, contact\_info) VALUES (1, 'Customer A', 'contactA@example.com');

INSERT INTO Customer\_Details (customer\_id, customer\_name, contact\_info) VALUES (2, 'Customer B', 'contactB@example.com');

INSERT INTO Customer\_Details (customer\_id, customer\_name, contact\_info) VALUES (2, 'Customer B', 'contactB@example.com');

DELETE FROM Customer\_Details

WHERE customer\_id NOT IN (

SELECT MIN(customer\_id)

FROM Customer\_Details

GROUP BY customer\_name

);

**Add a Foreign Key constraint to an existing table: The Products table currently has no relation to the Supplier table. Write a query to add a foreign key constraint supplier\_id (Integer) to the Products table, referencing the supplier\_id from the Suppliers table.**

ALTER TABLE Products

ADD supplier\_id INT,

ADD CONSTRAINT fk\_supplier

FOREIGN KEY (supplier\_id) REFERENCES Suppliers(supplier\_id);

**5. Create a table for project assignments: Write a query to create a table**

**named Project\_Assignments with the following fields:**

** assignment\_id (Integer, Primary Key)**

** project\_id (Integer, Foreign Key referencing Projects)**

** employee\_id (Integer, Foreign Key referencing Employees)**

** role (Varchar(50))**

** start\_date (Date)**

** end\_date (Date)**

CREATE TABLE IF NOT EXISTS Projects (project\_id INTEGER PRIMARY KEY,project\_name TEXT);

CREATE TABLE IF NOT EXISTS Employees (employee\_id INTEGER PRIMARY KEY,employee\_name TEXT);

CREATE TABLE IF NOT EXISTS Project\_Assignments (assignment\_id INTEGER PRIMARY KEY,project\_id INTEGER,employee\_id INTEGER,role TEXT,start\_date DATE,end\_date DATE,

FOREIGN KEY (project\_id) REFERENCES Projects(project\_id),

FOREIGN KEY (employee\_id) REFERENCES Employees(employee\_id));

**Alter a table to set a column as NOT NULL: The Customer\_Details table currently allows null values for the phone\_number column. Write a query to alter the phone\_number column so that it cannot accept null values.**

CREATE TABLE IF NOT EXISTS Customer\_Details (customer\_id INTEGER PRIMARY KEY,customer\_name TEXT,phone\_number TEXT -- Initial table without NOT NULL for demo purposes);

INSERT INTO Customer\_Details (customer\_id, customer\_name, phone\_number)

VALUES (1, 'Alice', '1234567890');

CREATE TABLE Customer\_Details\_New (

customer\_id INTEGER PRIMARY KEY,

customer\_name TEXT,

phone\_number TEXT NOT NULL

);

INSERT INTO Customer\_Details\_New (customer\_id, customer\_name, phone\_number)

SELECT customer\_id, customer\_name, phone\_number

FROM Customer\_Details;

DROP TABLE Customer\_Details;

ALTER TABLE Customer\_Details\_New RENAME TO Customer\_Details;

CREATE TABLE IF NOT EXISTS Suppliers (

supplier\_id INTEGER PRIMARY KEY,

name TEXT,

location TEXT

);

CREATE TABLE IF NOT EXISTS Products (

product\_id INTEGER PRIMARY KEY,

name TEXT,

supplier\_id INTEGER,

FOREIGN KEY (supplier\_id) REFERENCES Suppliers(supplier\_id)

);

**Delete records based on a JOIN condiƟon: You have a Products table and a Suppliers table. Write a query to delete all products from the Products table that are supplied by a supplier located in 'New York', using a join with the Suppliers table.**

INSERT INTO Suppliers (supplier\_id, name, location)

VALUES (1, 'Supplier A', 'New York'),(2, 'Supplier B', 'Los Angeles');

INSERT INTO Products (product\_id, name, supplier\_id)

VALUES (1, 'Product X', 1),(2, 'Product Y', 2);

DELETE FROM Products

WHERE supplier\_id IN (

SELECT supplier\_id

FROM Suppliers

WHERE location = 'New York'

);

**Drop a table from the database: The Attendance table is no longer needed. Write a query to drop the Attendance table from the database**

DROP TABLE IF EXISTS Attendance;

SELECT \* FROM Products;

SELECT \* FROM Suppliers

THANK YOU