WEB TECH DOCUMENTATION

(QUESTIONS | FILE LOCATION & CODE OUTPUT)

(All SQL Assignment Files are located in DBMS Assignment Folder)

Cephas Cardozo | 12 CT | 2024-25

QUESTIONS

Create database Employee.
Table : Emp_tbl.

Fields : Employee_id, Name, Address, Salary, Age.

Construct MySQL queries with reference to the above structure.

a. Enter a data record in Emp_tbl.
b. List all the records from eEmp_tbl.
c. Add a coloum date of birth between address and salary.
d. Display Employee_id and Names of employees whose age is less than 30.

1) Above Image is Connected to SQL File A

The Salary table has fields EmployeeId, Name, BasicSalary.

Construct Mysql queries:

a. To add a record to the table.
b. To view all the records from the table.
c. To display the highest BasicSalary from table.
d. To remove column Name from table.

2) Above Image is Connected to SQL File B

The Invoice table has the fields InvoiceNumber, Description, Quantity, Amount.

Construct Mysql Queries :

a. To display the structure of the table.
b. To remove all the records whose amount is below 5000.
c. To insert invoice_date between InvoiceNumber and description.
d. To display the total amount of all the records from the table.

3) Above Image is Connected to SQL File C

The BANKS table has the fields CustomerID, Name, Age, Amount

Construct MYSQL queries for the following:

a. Add a record.
b. Delete records of customer whose age is less than 40.
c. Change the name of the Field amount to deposit.
d. Insert a field address between name and age.

4) Above Image is Connected to SQL File D

The PAY table has the fields EmployeeID, Name, date_of_birth, salary
Construct MYSQL queries for the following:
a. Create pay table.
b. Display total salary of employees.
c. Display EmployeeID, name and salary in descending order of name.
d. Change the name of table Pay to Pay_tbl.

5) Above Image is Connected to SQL File E

Table: Student
Fields: Roll no, Name, DOB.

Construct MYSQL queries for the following:

a. Display name, date of birth and day on which student is born.
b. Display name and age of each student.
c. Display names of all students who are born in December.
d. To change date of birth of a student as 01 Jan 2000 whose name is “ANMOL”.

6) Above Image is Connected to SQL File F

Table: Payroll
Fields: Emp_id, name, state, salary.

Construct MYSQL queries for the following:

a. Insert another field 'DESIGNATION' in the table with length 10 and of character data type after field name.
b. To arrange records is descending order based on their salary.
c. Display all the employees whose designation is programmer and salary is more than 50,000.
d. To display total number of employee from each state.

7) Above Image is Connected to SQL File G

With reference to the given structure of the table, construct the following queries:

The database 'Employee' contains table 'E_tbl' table with fields Emp_id, Emp_Name, Dob, State, Salary.

a. Display Emp_id, Emp_name and Salary in the descending order of the Emp_name.
b. Display state and the total number of employees from each state.
c. Insert another field ' Gender' in the table with length 6 and of char data type, after the field Emp_name.
d. Change the value of the field 'Dob' to 2012-10-12 whose salary is greater than 25000.

8) Above Image is Connected to SQL File H

Construct MySQL queries for the following: Table : Inventory.
Fields : itemid, itemname, price, qty.

a. To arrange the records as per price in descending order.
b. Display itemid, itemname of all the items where quantity is between 5 to 10.
c. To remove the records where itemname is “Transistor”.

d. Display name of all items whose price is greater than 10000.

9) Above Image is Connected to SQL File I

Create a Database: Customer.

Table 1: Cust_tbl.

Fields: Customerid, Name, Address.

Table 2: Bill_tbl.

Fields: Customerid, Productid, Quantity, Amount.

Construct MYSQL queries for the following:

a. To list Customerid, Productid, Amount from Bill_tbl whose Amount exceeds 1,000.
b. To enter a data record in Cust_tbl.
c. To display all the records from multiple tables by combining Cust_tbl and Bill_tbl.
d. To remove Address column from Cust_tbl.

10) Above Image is Connected to SQL File J

OUTPUT & CODE

---

1)

emp_id	emp_name	address	salary	age
1	Cephas Cardozo	Veroda,Cuncolim-Goa, 403703	900000	23
2	Jolny Fernandes	KTC Bus Stand,Margao-Goa	20000	18
NULL	NULL	NULL	NULL	NULL

emp_id	emp_name
1	Cephas Cardozo
2	Jolny Fernandes
3	Cephas Cardozo
5	Andy Fernandes
6	Afra D'Souza
9	Michael D'Souza
10	Emily Roy
11	Aiden Pereira
12	Nina Thomas
NULL	NULL

```
-- DB INITIALIZATION
CREATE DATABASE IF NOT EXISTS Employee;
USE Employee;

-- TABLE INITIALIZATION
CREATE TABLE Emp_tbl(
        emp_id SERIAL PRIMARY KEY NOT NULL,
   emp_name VARCHAR(60) NOT NULL,
   address VARCHAR(300) NOT NULL,
   salary BIGINT NOT NULL,
   age SMALLINT NOT NULL
);

-- a) Enter a data record in Emp_tbl
-- Inserting data into the table with specified columns
INSERT INTO Emp_tbl (emp_name, address, dob, salary, age)
VALUES
("Cephas Cardozo", "Veroda, Cuncolim-Goa, 403703", '2000-08-13', 900000, 23),
("Jolny Fernandes", "KTC Bus Stand, Margao-Goa", '2005-07-01', 20000, 35),
("Andy Fernandes", "Colva Beach, Colva-Goa", '1995-05-10', 550000, 29),
("Afra D'Souza", "Fatorda, Margao-Goa", '1998-11-25', 750000, 26),
("Ryan D'Silva", "Vasco da Gama, Goa", '2003-02-17', 350000, 78),
("Sophie Mendes", "Panjim, Goa", '1997-09-30', 450000, 46),
("Michael D'Souza", "Margao, Goa", '1999-04-12', 500000, 25),
("Emily Roy", "Mapusa, Goa", '1996-12-05', 300000, 27),
("Aiden Pereira", "Benaulim, Goa", '2002-01-21', 150000, 22),
("Nina Thomas", "Dona Paula, Goa", '2004-06-15', 250000, 20);


-- b. List all the records from Emp_tbl
SELECT * FROM Emp_tbl;

-- c. Add a column date of birth between address and salary
ALTER TABLE Emp_tbl
ADD COLUMN dob DATE NULL AFTER address;

-- d. Display Employee_id and Names of employees whose age is less than 30
SELECT emp_id, emp_name
FROM Emp_tbl
WHERE age < 30;
```

2)

EmployeeId	Name	BasicSalary
2	Jolny Fernandes	520000.00
3	Andy Fernandes	450000.00
4	Afra D'Souza	600000.00
5	Ryan D'Silva	310000.00
6	Sophie Mendes	470000.00
7	Michael D'Souza	490000.00
8	Emily Roy	380000.00
9	Aiden Pereira	410000.00
10	Nina Thomas	530000.00

HighestSalary
600000.00

EmployeeId	BasicSalary
2	520000.00
3	450000.00
4	600000.00
5	310000.00
6	470000.00
7	490000.00
8	380000.00
9	410000.00
10	530000.00

```
--        DB INITIALIZATION
CREATE DATABASE IF NOT EXISTS Employee;
USE Employee;

-- TABLE INITIALIZATION
-- a. To create a table & insert data
CREATE TABLE Salary (
   EmployeeId INT PRIMARY KEY,
   Name VARCHAR(100) NOT NULL,
   BasicSalary DECIMAL(10, 2) NOT NULL
);

INSERT INTO Salary (EmployeeId, Name, BasicSalary)
VALUES
(2, 'Jolny Fernandes', 520000.00),
(3, 'Andy Fernandes', 450000.00),
(4, 'Afra D\'Souza', 600000.00),
(5, 'Ryan D\'Silva', 310000.00),
(6, 'Sophie Mendes', 470000.00),
(7, 'Michael D\'Souza', 490000.00),
(8, 'Emily Roy', 380000.00),
(9, 'Aiden Pereira', 410000.00),
(10, 'Nina Thomas', 530000.00);


-- b. To view all the records from the table.
SELECT * FROM Salary;

-- c. To display highest salary
SELECT MAX(BasicSalary) AS HighestSalary
FROM Salary;

-- d. To remove column Name from table
ALTER TABLE Salary
DROP COLUMN Name;
```

3)

Field	Type	Null	Key	Default	Extra
InvoiceNumber	int	NO	PRI	NULL	
Description	varchar(255)	NO		NULL	
Quantity	int	NO		NULL	
Amount	decimal(10,2)	NO		NULL	

TotalAmount
368500.00

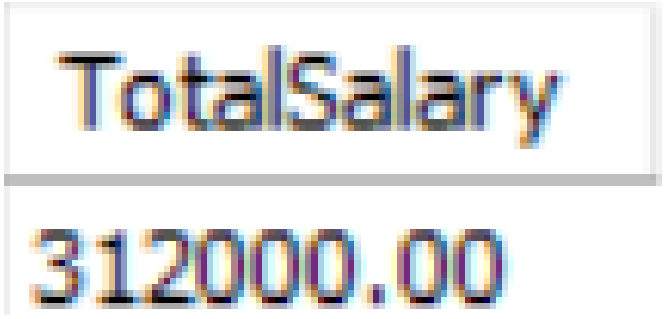
```
--        DB INITIALIZATION
CREATE DATABASE IF NOT EXISTS sales;
USE sales;

-- TABLE INITIALIZATION
-- a. Create the Invoice table
CREATE TABLE Invoice (
   InvoiceNumber INT PRIMARY KEY,
   Description VARCHAR(255) NOT NULL,
   Quantity INT NOT NULL,
   Amount DECIMAL(10, 2) NOT NULL
);

-- b. Display the structure of the Invoice table
-- The DESCRIBE Command is used to view table structure only!, & not the data within.
DESCRIBE Invoice;

-- Insert sample data into the Invoice table
INSERT INTO Invoice (InvoiceNumber, invoice_date, Description, Quantity, Amount) VALUES
(1, '2024-01-15', 'Laptop', 2, 150000.00),
(2, '2024-02-20', 'Smartphone', 5, 80000.00),
(3, '2024-03-05', 'Office Chair', 10, 30000.00),
(4, '2024-07-18', 'Mouse', 25, 12500.00),
(5, '2024-08-15', 'Tablet', 6, 60000.00),
(6, '2024-09-01', 'Headphones', 12, 36000.00);

-- c. Remove all records whose amount is below 5000
DELETE FROM Invoice
WHERE Amount < 5000;

-- d. Insert invoice_date between InvoiceNumber and Description
ALTER TABLE Invoice ADD COLUMN invoice_date DATE NULL AFTER InvoiceNumber;

-- e. Display the total amount of all the records from the table
SELECT SUM(Amount) AS TotalAmount
FROM Invoice;
```

4)

CustomerID	Name	Age	Deposit
1	John Doe	45	10000.00
2	Jane Smith	38	5000.00
3	Michael Brown	50	15000.00
4	Emily Davis	42	20000.00
5	Chris Wilson	39	8000.00
NULL	NULL	NULL	NULL

```
--        DB INITIALIZATION
CREATE DATABASE IF NOT EXISTS bank_info;
USE bank_info;

-- TABLE INITIALIZATION
-- Create the BANKS table
CREATE TABLE BANKS (
   CustomerID INT PRIMARY KEY,
   Name VARCHAR(100) NOT NULL,
   Age INT NOT NULL,
   Amount DECIMAL(10, 2) NOT NULL
);

-- a. Add sample records to the BANKS table
INSERT INTO BANKS (CustomerID, Name, Age, Amount) VALUES
(1, 'John Doe', 45, 10000.00),
(2, 'Jane Smith', 38, 5000.00),
(3, 'Michael Brown', 50, 15000.00),
(4, 'Emily Davis', 42, 20000.00),
(5, 'Chris Wilson', 39, 8000.00);

-- b. Delete records of customers whose age is less than 40
DELETE FROM BANKS
WHERE Age < 40;

-- c. Change the name of the field 'Amount' to 'Deposit'
ALTER TABLE BANKS
CHANGE COLUMN Amount Deposit DECIMAL(10, 2);

-- d. Insert a field 'Address' between 'Name' and 'Age'
ALTER TABLE BANKS ADD COLUMN Address VARCHAR(255) NULL AFTER Name;

-- Update the table with addresses for the existing records as Sample Data
UPDATE BANKS
SET Address = '123 Main St, Springfield' WHERE CustomerID = 1;
UPDATE BANKS
SET Address = '456 Oak Ave, Riverside' WHERE CustomerID = 3;

-- Viewing Data
SELECT * FROM BANKS;
```

5)



EmployeeID	Name	salary
5	Emily Davis	72000.00
4	Diana Ross	55000.00
3	Charlie Brown	65000.00
2	Bob Smith	50000.00
1	Alice Johnson	70000.00
NULL	NULL	NULL

EmployeeID	Name	date_of_birth	salary
1	Alice Johnson	1985-03-25	70000.00
2	Bob Smith	1990-07-18	50000.00
3	Charlie Brown	1982-11-12	65000.00
4	Diana Ross	1995-05-20	55000.00
5	Emily Davis	1988-09-10	72000.00
NULL	NULL	NULL	NULL

```
--        DB INITIALIZATION
CREATE DATABASE IF NOT EXISTS payments;
USE payments;

-- TABLE INITIALIZATION
-- a. Create the PAY table
CREATE TABLE PAY (
    EmployeeID INT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    date_of_birth DATE NOT NULL,
    salary DECIMAL(10, 2) NOT NULL
);

-- Insert sample data into the PAY table
INSERT INTO PAY (EmployeeID, Name, date_of_birth, salary) VALUES
(1, 'Alice Johnson', '1985-03-25', 70000.00),
(2, 'Bob Smith', '1990-07-18', 50000.00),
(3, 'Charlie Brown', '1982-11-12', 65000.00),
(4, 'Diana Ross', '1995-05-20', 55000.00),
(5, 'Emily Davis', '1988-09-10', 72000.00);

-- b. Display total salary of employees
SELECT SUM(salary) AS TotalSalary
FROM PAY;

-- c. Display EmployeeID, Name, and salary in descending order of Name
SELECT EmployeeID, Name, salary
FROM PAY
ORDER BY Name DESC;

-- d. Change the name of the table PAY to PAY_tbl
RENAME TABLE PAY TO PAY_tbl;

-- Viewing Data
SELECT * FROM PAY_tbl;
```

6)

Name	DayOfWeek
Anmol	2000-01-01
Bina	1999-12-15
Chirag	2001-05-23
Disha	2002-12-05
Esha	2003-11-30

Name	Age
Anmol	24
Bina	24
Chirag	23
Disha	21
Esha	20

Name
Bina
Disha

```
--          DB INITIALIZATION
```

```
CREATE DATABASE IF NOT EXISTS student_info;
USE student_info;

-- TABLE INITIALIZATION
-- Create the Student table
CREATE TABLE Student (
    Roll_no INT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    DOB DATE NOT NULL,
    Age INT NOT NULL
);

-- Insert sample data into the Student table
INSERT INTO Student (Roll_no, Name, DOB, Age) VALUES
(1, 'Anmol', '2000-01-01', 24),
(2, 'Bina', '1999-12-15', 24),
(3, 'Chirag', '2001-05-23', 23),
(4, 'Disha', '2002-12-05', 21),
(5, 'Esha', '2003-11-30', 20);

-- a. Display name, date of birth, and the day of the week on which the student was born
SELECT
    Name,
    DOB AS DayOfWeek
    FROM Student;

-- b. Display name and age of each student
SELECT
    Name,
    Age
FROM Student;

-- c. Display names of all students who are born in December
SELECT
    Name
FROM Student
WHERE MONTH(DOB) = 12;

-- d. Change the date of birth of a student named "Anmol" to 01 Jan 2000
UPDATE Student
SET DOB = '2000-01-01'
WHERE Name = 'Anmol';
```

7)

Emp_id	Name	DESIGNATION	State	Salary
5	Emily Davis	Programmer	Texas	75000.00
1	Alice Johnson	Manager	California	70000.00
4	Diana Ross	Manager	California	65000.00
2	Bob Smith	Programmer	Texas	55000.00
3	Charlie Brown	NULL	New York	45000.00
NULL	NULL	NULL	NULL	NULL

Emp_id	Name	DESIGNATION	State	Salary
2	Bob Smith	Programmer	Texas	55000.00
5	Emily Davis	Programmer	Texas	75000.00
NULL	NULL	NULL	NULL	NULL

State	TotalEmployees
California	2
Texas	2
New York	1

```
--        DB INITIALIZATION
CREATE DATABASE IF NOT EXISTS PayrollDB;
USE PayrollDB;

-- TABLE INITIALIZATION
-- Create the Payroll table
CREATE TABLE Payroll (
    Emp_id INT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    State VARCHAR(100) NOT NULL,
    Salary DECIMAL(10, 2) NOT NULL
);

-- Insert sample data into the Payroll table
INSERT INTO Payroll (Emp_id, Name, State, Salary) VALUES
(1, 'Alice Johnson', 'California', 70000.00),
(2, 'Bob Smith', 'Texas', 55000.00),
(3, 'Charlie Brown', 'New York', 45000.00),
(4, 'Diana Ross', 'California', 65000.00),
(5, 'Emily Davis', 'Texas', 75000.00);

-- a. Add the DESIGNATION column after the Name column
ALTER TABLE Payroll
ADD COLUMN DESIGNATION VARCHAR(10) AFTER Name;

-- b. Update the sample data with designations
-- Update the DESIGNATION for employee with Emp_id 1
UPDATE Payroll
SET DESIGNATION = 'Manager'
WHERE Emp_id = 1;

-- Update the DESIGNATION for employee with Emp_id 2
UPDATE Payroll
SET DESIGNATION = 'Programmer'
WHERE Emp_id = 2;

-- Update the DESIGNATION for employee with Emp_id 3
UPDATE Payroll
SET DESIGNATION = 'Analyst'
WHERE Emp_id = 3;
```

-- Update the DESIGNATION for employee with Emp_id 4
UPDATE Payroll
SET DESIGNATION = 'Manager'
WHERE Emp_id = 4;

-- Update the DESIGNATION for employee with Emp_id 5
UPDATE Payroll
SET DESIGNATION = 'Programmer'
WHERE Emp_id = 5;

-- c. Arrange records in descending order based on their salary
SELECT * FROM Payroll
ORDER BY Salary DESC;

-- d. Display all employees whose designation is Programmer and salary is more than 50,000
SELECT * FROM Payroll
WHERE DESIGNATION = 'Programmer' AND Salary > 50000;

-- e. Display the total number of employees from each state
SELECT State, COUNT(*) AS TotalEmployees
FROM Payroll
GROUP BY State;

8)

Emp_id	Emp_Name	Salary
5	Emily Davis	45000.00
4	Diana Ross	25000.00
3	Charlie Brown	30000.00
2	Bob Smith	55000.00
1	Alice Johnson	70000.00
NULL	NULL	NULL

State	TotalEmployees
California	2
Texas	2
New York	1

```
--          DB INITIALIZATION
CREATE DATABASE IF NOT EXISTS Employee;
USE Employee;

-- TABLE INITIALIZATION
-- Create the E_tbl table
CREATE TABLE E_tbl (
   Emp_id INT PRIMARY KEY,
   Emp_Name VARCHAR(100) NOT NULL,
   Dob DATE NOT NULL,
   State VARCHAR(100) NOT NULL,
   Salary DECIMAL(10, 2) NOT NULL
);

-- Insert sample data into the E_tbl table
INSERT INTO E_tbl (Emp_id, Emp_Name, Dob, State, Salary) VALUES
(1, 'Alice Johnson', '1985-07-15', 'California', 70000.00),
(2, 'Bob Smith', '1990-05-23', 'Texas', 55000.00),
(3, 'Charlie Brown', '1988-03-12', 'New York', 30000.00),
(4, 'Diana Ross', '1992-11-25', 'California', 25000.00),
(5, 'Emily Davis', '1994-02-10', 'Texas', 45000.00);

-- a. Display Emp_id, Emp_Name, and Salary in descending order of Emp_Name
SELECT
Emp_id,
Emp_Name,
Salary
FROM E_tbl
ORDER BY Emp_Name DESC;

-- b. Display state and the total number of employees from each state
SELECT State,
COUNT(*) AS TotalEmployees
FROM E_tbl
GROUP BY State;

-- c. Insert another field 'Gender' in the table with length 6 and of CHAR data type, after Emp_Name
ALTER TABLE E_tbl
ADD COLUMN Gender CHAR(6) AFTER Emp_Name;

-- d. Change the value of the field 'Dob' to 2012-10-12 for employees whose Salary is greater than 25000
UPDATE E_tbl
SET Dob = '2012-10-12'
WHERE Salary > 25000;
```

9)

itemid	itemname	price	qty
1	Laptop	120000.00	7
4	Monitor	25000.00	8
6	Printer	15000.00	4
7	External Hard Drive	8000.00	6
3	Keyboard	800.00	12
2	Mouse	500.00	15
5	Transistor	50.00	25
NULL	NULL	NULL	NULL

itemid	itemname
1	Laptop
4	Monitor
7	External Hard Drive
NULL	NULL

itemname
Laptop
Monitor
Printer

```
--        DB INITIALIZATION
CREATE DATABASE IF NOT EXISTS InventoryDB;
USE InventoryDB;

-- TABLE INITIALIZATION
-- Create the Inventory table
CREATE TABLE Inventory (
    itemid INT PRIMARY KEY,
    itemname VARCHAR(100) NOT NULL,
    price DECIMAL(10, 2) NOT NULL,
    qty INT NOT NULL
);

-- Insert sample data into the Inventory table
INSERT INTO Inventory (itemid, itemname, price, qty) VALUES
(1, 'Laptop', 120000.00, 7),
(2, 'Mouse', 500.00, 15),
(3, 'Keyboard', 800.00, 12),
(4, 'Monitor', 25000.00, 8),
(5, 'Transistor', 50.00, 25),
(6, 'Printer', 15000.00, 4),
(7, 'External Hard Drive', 8000.00, 6);

-- a. To arrange the records as per price in descending order
SELECT *
FROM Inventory
ORDER BY price DESC;

-- b. Display itemid, itemname of all the items where quantity is between 5 to 10
SELECT itemid, itemname
FROM Inventory
WHERE qty BETWEEN 5 AND 10;

-- c. To remove the records where itemname is “Transistor”
DELETE FROM Inventory WHERE itemname = 'Transistor';

-- d. Display name of all items whose price is greater than 10000
SELECT itemname
FROM Inventory
WHERE price > 10000;
```

10)

Customerid	Productid	Amount
1	101	1500.00
2	103	2000.00

Customerid	Name	Address	Productid	Quantity	Amount
1	John Doe	123 Elm Street, Springfield	101	2	1500.00
1	John Doe	123 Elm Street, Springfield	102	1	750.00
2	Jane Smith	456 Oak Avenue, Springfield	103	5	2000.00
3	Jim Brown	789 Pine Road, Springfield	104	3	300.00

```
--        DB INITIALIZATION
CREATE DATABASE IF NOT EXISTS Customer;
USE Customer;

-- TABLE INITIALIZATION
-- Create the Cust_tbl table
CREATE TABLE Cust_tbl (
   Customerid INT PRIMARY KEY,
   Name VARCHAR(100) NOT NULL,
   Address VARCHAR(255) NOT NULL
);

-- Create the Bill_tbl table
CREATE TABLE Bill_tbl (
   Customerid INT,
   Productid INT,
   Quantity INT,
   Amount DECIMAL(10, 2),
   FOREIGN KEY (Customerid) REFERENCES Cust_tbl(Customerid)
);

-- a. To list Customerid, Productid, Amount from Bill_tbl whose Amount exceeds 1,000
SELECT
   Customerid,
   Productid,
   Amount
FROM Bill_tbl
WHERE Amount > 1000;

-- b. To enter a data record in Cust_tbl
-- Insert sample data into Cust_tbl
INSERT INTO Cust_tbl (Customerid, Name, Address) VALUES
(1, 'John Doe', '123 Elm Street, Springfield'),
(2, 'Jane Smith', '456 Oak Avenue, Springfield'),
(3, 'Jim Brown', '789 Pine Road, Springfield');

-- Insert sample data into Bill_tbl
INSERT INTO Bill_tbl (Customerid, Productid, Quantity, Amount) VALUES
(1, 101, 2, 1500.00),
(1, 102, 1, 750.00),
(2, 103, 5, 2000.00),
(3, 104, 3, 300.00);

-- c. To display all the records from multiple tables by combining Cust_tbl and Bill_tbl
```

```
SELECT
   c.Customerid,
   c.Name,
   c.Address,
   b.Productid,
   b.Quantity,
   b.Amount
FROM Cust_tbl c
JOIN Bill_tbl b ON c.Customerid = b.Customerid;

-- d. To remove Address column from Cust_tbl
ALTER TABLE Cust_tbl DROP COLUMN Address;
```
