

## 1. Create Table Queries

### Question 1:

Create a table `Customers` with the following columns:

- `CustomerID` (Primary Key, Auto Increment)
- `FirstName` (VARCHAR(50))
- `LastName` (VARCHAR(50))
- `Email` (VARCHAR(100), Unique)
- `PhoneNumber` (VARCHAR(15))
- `Address` (VARCHAR(200))

### Question 2:

Create a table `Accounts` with the following columns:

- `AccountNumber` (Primary Key)
- `CustomerID` (Foreign Key references `Customers.CustomerID`)
- `AccountType` (VARCHAR(20)) – values could be 'Saving', 'Checking', etc.
- `Balance` (Decimal)
- `DateCreated` (DATE)

## 2. Inserting Data

### Question 3:

Insert the following data into the `Customers` table:

CustomerID	FirstName	LastName	Email	PhoneNumber	Address
1	John	Doe	<a href="mailto:john.doe@email.com">john.doe@email.com</a>	1234567890	123 Main St, Cityville
2	Jane	Smith	<a href="mailto:jane.smith@email.com">jane.smith@email.com</a>	0987654321	456 Elm St, Townville
3	Mike	Johnson	<a href="mailto:mike.johnson@email.com">mike.johnson@email.com</a>	1122334455	789 Oak St, Villageville

### Question 4:

Insert the following data into the `Accounts` table:

AccountNumber	CustomerID	AccountType	Balance	DateCreated
1001	1	Saving	5000.00	2023-01-15
1002	1	Checking	1500.00	2023-02-20
1003	2	Saving	2000.00	2023-03-01
1004	3	Checking	3000.00	2023-03-10

## Set - 2.

1) a) Create table queries.

Create a table Customers

```
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY AUTO_INCREMENT,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Email VARCHAR(100), UNIQUE,
    PhoneNumber VARCHAR(15),
    Address VARCHAR(200)
);
```

b) Create a table Accounts

```
CREATE TABLE Accounts (
    AccountNumber INT PRIMARY KEY,
    CustomerID FOREIGN KEY,
    AccountType VARCHAR(20),
    Balance DECIMAL(10, 2),
    DateCreated DATE
);
```

## 2. Inserting data

### a) To Customers table:

```
INSERT INTO Customers (CustomerID, FirstName, LastName,  
Email, Phone Number, Address)
```

VALUES

```
(1, 'John', 'Doe', 'john.doe@gmail.com', '1234567890', '123 Main St,  
Cityville),
```

```
(2, 'Jane', 'Smith', 'jane.smith@gmail.com', '0987654321', '456 Elm St,  
Townville),
```

```
(3, 'Mike', 'Johnson', 'mike.johnson@gmail.com', '1122334455', '789 Oak St,  
Villageville);
```

### b) To Accounts table

```
INSERT INTO Accounts (AccountNumber, CustomerID, AccountType,-  
Balance, DateCreated)
```

VALUES

```
(1001, 1, 'Saving', 5000.00, '2023-01-15'),
```

```
(1002, 1, 'Checking', 1500.00, '2023-02-20'),
```

```
(1003, 2, 'Saving', 2000.00, '2023-03-01'),
```

```
(1004, 3, 'Checking', 3000.00, '2023-03-10');
```

3.) Update Data.

Q.5

UPDATE Accounts

SET Balance = 5500.00

WHERE AccountNumber = 1001;

Q.6

UPDATE Customers

SET Email = 'jane.smith@newdomain.com'

WHERE CustomerID = 2;

Q.7

UPDATE Accounts

SET Balance += (Balance \* 0.05)

WHERE AccountType = 'Saving';

## Q.4 Select Queries

Q.8

```
SELECT Customers.CustomerID, Customers.FirstName, Customers.LastName  
      Accounts.Balance  
FROM Customers  
JOIN Accounts ON Customers.CustomerID = Accounts.CustomerID  
WHERE Accounts.AccountType = 'Saving';
```

Q.no.9

```
SELECT Customers.CustomerID, Customers.FirstName, Customers.LastName  
      Accounts.Balance  
FROM Customers  
JOIN Accounts ON Customers.CustomerID = Accounts.CustomerID  
WHERE Accounts.Balance > 3000 AND Accounts.AccountType = 'Checking'
```

Q.no.10

```
SELECT CustomerID, AccountNumber, AccountType, Balance  
FROM Accounts  
WHERE Balance < 2000;
```

## 5.) Deleting Data

Q.11

```
DELETE FROM Accounts  
WHERE AccountNumber = 1002;
```

Q.12

```
DELETE FROM Customers  
WHERE PhoneNumber LIKE '123%';
```

Q.13

```
DELETE FROM Accounts  
WHERE DateCreate < '2023-02-01';
```

## 6.) Join Queries

Q.no.14

```
SELECT Customers.FirstName, Customers.LastName, Accounts.AccountType  
FROM Customers  
JOIN Accounts ON Customers.CustomerID = Accounts.CustomerID  
WHERE Accounts.Balance > 2000;
```

Q.15

```
SELECT AccountType, SUM(Balance) AS TotalBalance  
FROM Accounts  
WHERE AccountType = 'Saving'  
GROUP BY AccountType;
```

Q.no. 16

```
SELECT Customers.FirstName, Customers.LastName, Accounts.AccountNumber,  
       Accounts.Balance  
FROM Customers  
LEFT JOIN Accounts ON Customers.CustomerID = Accounts.CustomerID;
```

7) Constraints and Validation

17)

ALTER TABLE Accounts

ADD CONSTRAINT check\_Balance CHECK(Balance  $\geq 0$ );

18)

ALTER TABLE Customers

MODIFY COLUMN Email VARCHAR(100) NOT NULL UNIQUE;

19.

ALTER TABLE Accounts

ADD CONSTRAINT fk\_customer FOREIGN KEY(customer ID) REFERENCES  
Customers(customer ID);

20.

ALTER TABLE Accounts

ADD CONSTRAINT check\_account\_type CHECK(AccountType IN ('saving', 'check'))

## 8. Complex Queries.

Qno.21

```
SELECT Customers.FirstName, Customers.LastName, Accounts.AccountType,  
       MAX(Accounts.Balance) AS HighestBalance  
FROM Customers  
JOIN Accounts ON Customers.CustomerID = Accounts.CustomerID  
GROUP BY Customers.ID  
ORDER BY HighestBalance DESC;  
LIMIT 1;
```

Qno.22

```
BEGIN TRANSACTION;
```

```
    UPDATE Accounts  
    SET Balance -= 1000  
    WHERE AccountNumber = 1003;
```

```
    UPDATE Accounts  
    SET Balance += 1000  
    WHERE AccountNumber = 1001;
```

```
COMMIT;
```

23)

```
SELECT Customers.FirstName, Customers.LastName, SUM(Accounts.Balance)
FROM Customers
JOIN Accounts ON Customers.CustomerID = Accounts.CustomerID
GROUP BY Customers.CustomerID;
```

24) Aggregation

```
SELECT AVG(Balance) AS AverageBalance
FROM Accounts;
```

25.)

```
SELECT COUNT(*) AS TotalSavingAccount
FROM Accounts
WHERE AccountType = Saving;
```