Set - 1

```
Create Table Students (student_id int Primary Key
Auto_increment, FirstName varchar(50) not null, LastName
varchar(50) not null, Age int check(age>16), EnrollmentDate DATE, Major
varchar(100) default current_date);
```

```
INSERT INTO students (student_id, FirstName, lastName,
  Age, EnrollmentDate, Major)
VALUES (1, 'Alice', 'Johnson', 18, '2023-09-01', 'cow'),
       (2, 'Bob', 'Smith', 20, '2022-06-15', 'Maths'),
       (3, 'Charlie', 'Brown', 19, '2021-08-20', 'Physics'),
       (4, 'Daisy', 'Carter', 21, '2023-01-10', 'Bio'),
       (5, 'Ethan', 'Taylor', 22, '2023-03-25', 'Chemistry');
```

1)
```
UPDATE students
  SET Major = 'Data Science'
WHERE ~~stud~~ student_id = 1;
```

2)
```
UPDATE students
  SET Age = Age + 1
  WHERE EnrollmentDate < '2023-01-01';
```

3)
```
~~Upd~~ Update students
    SET lastName = 'Cooper'
    WHERE FirstName = 'Daisy';
```

4) UPDATE Students
SET Major = 'Undeclared'
WHERE Age < 20;

5) Update Students
SET EnrollmentDate = '2024-01-05'
WHERE student_id = 5;

6) ~~Up~~ UPDATE Students
SET Major = 'Physics'
WHERE Major = 'Biology';

7) UPDATE students
SET Age = 23
WHERE FirstName = 'Charlie';

8) UPDATE Students
SET LastName = 'Williams'
WHERE Major = 'Mathematics';

9) Update Students
SET FirstName = 'Alex'
WHERE Age = (SELECT MIN(Age) From Students)
LIMIT 1;

10) Update Students
SET Age = 'NULL'
WHERE Major = 'Undeclared';

DELETE

1) DELETE From Students
WHERE Student_id = 5;

2) DELETE From students
WHERE Major = 'Undeclared';

3) DELETE From students
WHERE EnrollmentDate > '2023-01-01';

4) DELETE From students
WHERE Age > 25;

5) DELETE FROM students
WHERE FirstName = 'Ethan' AND LastName = 'Taylor';

6) DELETE FROM students
WHERE Age = "NULL";

7) DELETE FROM students
WHERE LastName LIKE 'C%';

8) DELETE FROM students
WHERE EnrollmentDate < '2022-05-05';

9) DELETE FROM students
WHERE Major = 'Physics';

10) DELETE FROM students;

Set-2

Question-1

```
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));
```

Question-2

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

Set-2

Question-3

```
INSERT INTO Customers (FirstName, LastName, Email,
  PhoneNumber, Address)
VALUES
('John', 'Doe', 'john.doe@gmail.com', '1234567890',
  '123 Main St, Cityville'),
('Jane', 'Smith', 'jane.smith@email.com', '0987654321',
  '456 Elm St, Townville'),
('Mike', 'Johnson', 'mike.johnson@email.com', '1122334455',
. '789 Oak St, Villageville');
```

4)
```
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');
```

5)
```
UPDATE Accounts
SET Balance = 5500.00
WHERE AccountNumber = 1001;
```

6) UPDATE Customers
SET Email = 'jane.smith@newdomain.com'
WHERE CustomerID = 2;

7) UPDATE Accounts
SET Balance = Balance + 10/100 * Balance
WHERE AccountType = 'Saving';

8) SELECT Customers.CustomerID, Customers.FirstName
Customers.lastName, Accounts.Balance
FROM Customers
JOIN Accounts ON Customer.ID = Accounts.CustomerID
WHERE Accounts.AccountType = 'Saving';

9) SELECT Customers.CustomerID, Customers.FirstName
Customers.lastName, Accounts.Balance
FROM Customers
JOIN Accounts ON Customers.CustomerID = Accounts.Customer
WHERE Accounts.Balance > 3000 AND
Accounts.AccountType = 'Checking';

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

11) DELETE FROM Accounts
WHERE AccountNumber = 1002;

12) DELETE FROM Customers
WHERE PhoneNumber = '123%';

13) DELETE FROM Accounts
WHERE DateCreated < '2023-02-01';

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

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

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

17) ALTER TABLE Accounts
ADD CONSTRAINT check_balance
CHECK (Balance >= 0);

18) ALTER TABLE Customers
MODIFY COLUMN Email VARCHAR(100)
NOT NULL UNIQUE;

19) ALTER TABLE Accounts
ADD CONSTRAINT fk_customer FOREIGN KEY
(CustomerID) REFERENCES Customers(CustomerID);

20) ALTER TABLE Accounts
ADD CONSTRAINT check_account_type
CHECK (AccountType IN ('Saving', 'Checking'));

21) SELECT Customers.FirstName, Customers.LastName,
Accounts.AccountNumber, MAX(Accounts.Balance) AS
HighestBalance
FROM Customers
JOIN Accounts ON Customers.CustomerID = Accounts.Custom
GROUP BY ~~HighestBa~~ HighestBalance DESC
LIMIT 1;

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

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

25) SELECT COUNT(*) AS TotalSavingAccounts
FROM Accounts
WHERE AccountType = 'Saving';

Set-3

1)
```
CREATE TABLE Library (
  BookID INT PRIMARY KEY AUTO-INCREMENT,
  Title VARCHAR (100),
  Author VARCHAR (100),
  Publisher VARCHAR (100),
  Genre VARCHAR (100),
  PublishedYear INT,
  ISBN VARCHAR(100) UNIQUE,
  Pages INT,
  CopiesAvailable INT,
  Price DECIMAL (10, 2) );
```

2)
```
INSERT INTO Library(BookID, Title, Author,
  Publisher, Genre, PublishedYear, ISBN, Pages,
  CopiesAvailable, Price)
VALUES (
(1, 'To Kill a Mockingbird', 'Harper Lee', 'J.B. Lippincott',
  'Fiction', 1960, '978-0061120084', 324, 5, 15.99),
(2, '1984', 'George Orwell', 'Harvill Secker', 'Dystopian',
  1949, '978-0451524935', 328, 2, 9.99),
(3, 'The Great Gatsby', 'F. Scott Fitzgerald', 'Scribner',
  'Fiction', 1925, '978-0743273565', 180, 3, 10.99),
(4, 'The Catcher in the Rye', 'J.D. Salinger', 'Little, Brown',
  'Fiction', 1951, '978-0316769488', 277, 4, 12.99),
(5, 'The Hobbit', 'J.R.R. Tolkien', 'HarperCollins', 'Fantasy',
  1937, '978-0618968633', 310, 6, 13.99)
);
```

3) ~~UPDATE Library~~
~~SET ISBN = '978-0451524935'~~
~~WHERE~~

3) UPDATE Library
SET Price = 11.99
WHERE ISBN = '978-0451524935' AND
Genre = 'Dystopian' AND
PublishedYear < 1950 ;

4) UPDATE Library
CopiesAvailable = 10
WHERE Genre = 'Fiction' AND
PublisheYear > 1950 ;

5) UPDATE Library
Price = Price - 5/100 * Price
WHERE Genre = 'Fiction' AND
~~PublishedYear > 1950~~ Pages > 300 ;

6) UPDATE Library
Pages = 350
WHERE CopiesAvailable > 4 AND
Price < 14 ;

7) UPDATE Library
SET Price = Price + 10/100 * Price
WHERE Genre = 'Fantasy' AND
Published Year < 1950 AND
Pages < 300;

8) UPDATE Library
SET CopiesAvailable = 0
WHERE Price > 12 AND
(Genre = 'Fiction' OR Genre = 'Dystopian');

9) UPDATE Library
SET PublishedYear = 2020
WHERE Author = '%Tolkien%' AND
(Price > 10 AND Price < 15);

10) UPDATE Library
SET Price = 8.99
WHERE Author = 'George Orwell' AND
Pages > 300;

11) UPDATE Library
SET Price = Price - 15/100 * Price
WHERE Genre = 'Fiction' AND
PublishedYear < 1950 AND
CopiesAvailable < 5;

0.15
100

12) UPDATE Library
SET Price = 17.99
WHERE Title = 'To Kill a Mockingbird' AND
PublishedYear = 1960 AND
CopiesAvailable > 4;

13) DELETE FROM Library
WHERE (ISBN = '978-0451524935' ~~AND~~ AND
ISBN = '978-0618968633') AND
Genre = 'Dystopian' AND
CopiesAvailable > 2;

14) DELETE FROM Library
WHERE PublishedYear < 1950 AND
Price < 10;

15) DELETE FROM Library
WHERE Genre = 'Fiction' AND
CopiesAvailable < 3 AND
PublishedYear < 1960;

16) DELETE FROM Library
WHERE Pages < 200 AND
PublishedYear > 1920 AND
PublishedYear < 1960;

17) DELETE FROM Library
WHERE Author = 'Harper Lee' AND
Price < 12;

18) DELETE FROM Library
WHERE CopiesAvailable = 0 AND
Price > 15;

19) DELETE FROM Library
WHERE ISBN = '978-074' AND
Genre = 'Fiction' AND
PublishedYear < 1950;

20) DELETE FROM library
WHERE Published Year > 2000 AND
~~Price > 10 AND Price < 15~~
(Price < 10 OR Price > 15)

21) DELETE FROM library
WHERE Genre = 'Fantasy' AND
CopiesAvailable > 3 AND
CopiesAvailable < 10;

22) DELETE FROM library
WHERE Author = 'J. D. Salinger' AND
Pages < 300 AND
Price > 12;

23) SELECT * FROM Library
WHERE Genre = 'Fiction' AND
Published Year > 1950 AND
Pages > 200 AND
Price BETWEEN 10 AND 15;

24) SELECT * FROM library
WHERE Pages > 300 AND
Price > 12 AND
Genre != 'Fantasy';

25) SELECT * FROM Library
WHERE PublishedYear BETWEEN 1925 AND 1950
AND Price < 13
AND CopiesAvailable > 3;

26) SELECT * FROM Library
WHERE (Author = 'Harper Lee' OR Author = 'George Orwell') AND
PublishedYear > 1950 AND
Price > 10;

27) SELECT *FROM Library
WHERE Author = 'J.R.R. Tolkien' AND
CopiesAvailable > 5 AND
Price BETWEEN 12 AND 15;

28) SELECT * FROM Library
WHERE (Genre = 'Dystopian' OR Genre = 'Fantasy')
AND Pages > 300
ORDER BY Price DESC
LIMIT 1;

29) SELECT * FROM Library
WHERE Author LIKE '%Tolkien%' AND
PublishedYear < 1940 AND
CopiesAvailable BETWEEN 4 AND 6;

30) SELECT * FROM Library
WHERE Genre = 'Fiction' AND
Pages > 200 AND
Price BETWEEN 10 AND 20 AND
CopiesAvailable BETWEEN 2 AND 5;

31) SELECT * FROM Library
WHERE PublishedYear < 1950 AND
Price < 15 AND
CopiesAvailable < 3;

32) SELECT Title, Author FROM Library
WHERE Price > 12 AND
CopiesAvailable > 4 AND
PublishedYear > 1930;