PRACTICE 1.

Q.N.1.

```
→   CREATE TABLE Students(
    StudentID INT PRIMARY KEY AUTO_INCREMENT,
    FirstNAME VARCHAR(50) NOT NULL,
    LastNAME VARCHAR(50) NOT NULL,
    Age INT CHECK (Age >=16),
    EnrollmentDate DATE DEFAULT CURRENT_DATE,
    Major VARCHAR(100)
    );
```

Question No. 2

```
→ INSERT INTO Students(studentID, FirstNAME, LastNAME
  Age, Enrollment Date, Major)
  VALUES
 (1,'Alice','Johnson', 18,'2023-09-01','computer science'
 (2,'Bob', 'Smith',20,'2022-06-15','Mathematics'),
 (3,'Charlie','Brown', 19,'2021-08-20','Physics'),
 (4,'Daisy','Carter', 21,'2023-01-10','Biology'),
 (5,'Ethan','Taylor', 22, 2023-03-25','Chemistry');
```

Question No. 3

```
1. UPDATE students
   SET Major = ' Data Science'
   WHERE Student ID = 1;
2. UPDATE students
   SET Age = Age +1
   WHERE EnrollmentDate <'2023-01-01' ;
3. UPDATE students
   SET LastNAME = 'cooper'
   WHERE FirstName = 'Daisy' ;
4. UPDATE Students
   SET Major='undecleared
   WHERE Age<20;
```

Question No. 4

Contd...

```
6. UPDATE Students
   SET Major = ' Physics'
   WHERE Major = 'Biology';

5. UPDATE Students
   SET EnrollmentDate = '2024-0101'
   WHERE StudentID = 5 ;

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 = ~~NULL~~ (SELECT MIN(Age) FROM students;
   ~~WHERE Major = 'Undeclared';~~

10. UPDATE Students
    SET Age = NULL
    WHERE Major= 'Undeclared';

11. UPDATE Students
    SET Major= 'DBMS'
    WHERE Age =(SELECT MIN(Age) FROM students)
```

[illegible] NO. 9

PRACTICE -2

Question NO.1

```
CREATE TABLE Customers(
 CustomerID INT ~~Auto~~ AUTO_INCREAMENT PRIMARYKEY,
 firstName VARCHAR(50),
 LastName VARCHAR(50),
 Email ~~Varcha~~ VARCHAR (100) UNIQUE,
 PhoneNumber VARCHAR (15),
 Address VARCHAR(200)
 );
```

Question NO. 2

```
CREATE TABLE Accounts(
 Account Number INT PRIMARY KEY,
 Customer ID INT,
 AccountType VARCHAR (20),
 Balance DECIMAL(10,2)
 Date created DATE,
 FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
 );
```

Question NO. 3

```
INSERT INTO Customers(firstName, LastName, Email,
PhoneNumber, Address)
VALUES
('John', 'Doe' 'John.doe@email.com', '1234567890', '123Mainst',
 'cityville'),
('Jane', 'smith', 'Jane.smith@email.com', '0987654321', '456Elmst', 'Townville'),
('Mike', 'Johnson', 'mike.Johnson@email.com', '1122334455', '789 oak st', 'villageville');
```

Question No. 4

1. DELETE FROM Students
   WHERE StudentID = 3;

2. DELETE FROM Students
   WHERE Major = 'undeclared';

3. DELETE FROM Students
   WHERE EnrollmentDate > '2023-01-01';

4. DELETE FROM Students
   WHERE Age > 21;

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

6. DELETE FROM Students
   WHERE Age IS NULL;

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

8. DELETE FROM students
   WHERE Enrollment Date < '2022-01-01';

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

10. DELETE FROM students;
    ~~WHERE~~

Question No. 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');
```

Question No. 5

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

Question No. 6

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

Question No. 7

```
UPDATE Accounts
SET Balance = Balance * 1.10
WHERE AccountType = 'Saving';
```

Question No. 8

```
SELECT Customers.CustomerID, Customer.FirstName,
Customers.LastName, Accounts.Balance
FROM customers
Join Accounts.AccountType = 'Saving';
```

.. Total balance

Question No. 9

```
SELECT Customers.Customer ID, Customers.
FirstName, Customers.LastName, Accounts.Balance
FROM customers
JOIN Accounts ON customers.Customer ID=Accounts.CustomerID
WHERE Accounts.Balance> 3000 AND Accounts.Account Type='checking'
```

Question No. 10

```
SELECT CustomerID, AccountNumber, Account Type, Balanc
FROM Accounts
WHERE Balance <2000;
```

Question No. 11

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

Q.NO. 12

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

Q.No. 13

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

Q.N. 14

```
SELECT Customers.firstName, Customer.LastName,
Accounts. Account Type
FROM Customers
JOIN Accounts ON Customers.CustomerID =Accounts.CustomerID
 WHERE Account.Balance >2000;
```

Q.No. 15

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

Q.N. 16

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

Q.N. 17

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

Q.N. 18

```
ALTER TABLE Cotomers
MODIFY COLUMN Email VARCHAR(100) NOT NULL UNIQUE;
```

Q.N.19

```
ALTER TABLE Accounts
ADD ~~Cons~~ CONSTRAINT fk_customer FOREIGN KEY (customerID)
REFERANCES Customers (CustomerID);
```

Q.N. 20

```
ALTER TABLE Accounts
ADD CONSTRAINT check_account_type CHECK
(Account Type IN ('Saving', 'checking'));
```

Q.NO. 21

```
SELECT Customers. firstName, Customers. LastName,
Accounts. AccountNumber, MAX( Account. Balance) AS
HighestBalance
FROM customers
Join Accounts ON customers. Customer ID = Accounts. Customer ID
 GROUP BY customers. CustomerID
 ORDER BY Highest Balance DESC
LIMIT 1;
```

Q.N. 22

```
 BEGIN TRANSACTION;
UPDATE Accounts
SET Balance = Balance -1000
WHERE Account Number = 1003;

UPDATE Accounts
 SET Balance = Balance + 1000
WHERE Account Number = 1001;

 COMMIT;
```

Q.N. 23

```
SELECT Customers. firstName, Customers. LastName, SUM(
 Accounts. Balance) As Total Balance
FROM Customers
JOIN Accounts ON Customers. Customer ID
                = Accounts. CustomerID
 GROUP BY Customers. CustomerID;
```

Q.N.24

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

Q.N.25

```
SELECT COUNT (*) AS Total SavingAccounts
FROM Accounts
WHERE AccountType= 'saving';
```

Q.N.26

```
BEGIN TRANSACTION;
UPDATE Accounts
SET BALANCE = Balance -500
WHERE AccountNumber= 1002;
UPDATE Accounts
SET Balance = Balance +500
WHERE AccountNumber= 1003;
COMMIT;
```