

Practise set 1

Question 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 2:

Insert Queries:

~~DELETE~~ INSERT INTO Students (StudentID, FirstName, LastName, Age, EnrollmentDate, 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 3: Update Data.

Update Queries:

- 1) UPDATE Students
SET Major = 'Data Science'
WHERE StudentID = 1;

2. ~~UPDATE~~ 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 = 'Undeclared'

WHERE Age < 20;

5. UPDATE Students

SET EnrollmentDate = '2024-01-01'

WHERE StudentID = 5;

6. UPDATE Students

SET Major = 'Physics'

WHERE Major = 'Chemistry' OR Major = 'Biology';

7. UPDATE Students

SET Age = 23

WHERE FirstName = 'Charlie';

8. ~~UPDATE~~ UPDATE Students

SET LastName = 'Williams'

WHERE Major = 'Mathematics';

9. UPDATE Students

SET FirstName = 'Alex'

WHERE Age = (SELECT MIN(Age) FROM Students);

10. UPDATE Students

SET Age = NULL

WHERE Major = 'Undeclared';

Question 4: Delete Records

Delete Queries.

1. DELETE FROM Students

WHERE StudentID = 3;

2. DELETE FROM Students

WHERE Major = 'Undeclared';

3. DELETE FROM Students

WHERE Enrollment > '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 EnrollmentDate < '2022-01-01';
9. DELETE FROM students,
WHERE Major = 'Physics';
10. DELETE FROM Students
11. UPDATE Students
SET Major = 'Computer science', Age = Age + 1
WHERE Age = (SELECT MIN(Age) FROM Students);

Practise Set 2:

1. ~~Create~~ Create table Queries:

Question 1: Create customers Table

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

Question 2: Create Accounts Table.

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

2. Inserting Data

Question 3: Insert Data into Customers Table.

Insert Data

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

VALUES

```
( 'John', 'Doe', 'john.doe@email.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' );
```

Question 4: Insert into Accounts.

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.

~~Question 5:~~

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 + 100

WHERE AccountType = 'Saving';

8. Select Queries.

SELECT customers, CustomerID,
FROM Customers

JOIN Accounts ON customers.CustomerID = Accounts.
CustomerID

~~JOIN~~ JOIN Accounts ON . WHERE Accounts.Account
Type = 'Saving';

9. SELECT customers.CustomerID, customers.FirstName, customers.

② LastName, Accounts.Balance
FROM Customers

JOIN Accounts ON ~~customer~~.CustomerID = Accounts.
CustomerID

WHERE Accounts.Balance > 3000 AND Accounts.AccountType =
'Checking';

10. SELECT CustomerID, AccountNumber, AccountType, Balance
FROM Accounts

WHERE Balance < 2000;

5. Deleting Data

11. DELETE FROM Accounts

WHERE AccountNumber = 1002;

12. DELETE FROM Customers

WHERE PhoneNumber LIKE '923456789';

13. DELETE FROM Accounts

WHERE DateCreated < '2023-02-01';

6. Join Queries.

14. SELECT Customers.FirstName, Customers.LastName, Accounts.AccountType
FROM Customers

JOIN Accounts ON Customers.CustomerID = Accounts.CustomerID
WHERE Account.Balance > 2000;

15. SELECT AccountType, SUM(Balance) AS TotalBalance
FROM Accounts

WHERE AccountType = 'Saving'
GROUP BY AccountType;

6. SELECT Customers.FirstName, Customers.LastName, Account.AccountNumber, Accounts.Balance

FROM Customers

LEFT JOIN Accounts ON Customers.CustomerID = Accounts.CustomerID;

7. Constraints & Validation.

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 customer (CustomerID);

20. ALTER TABLE Accounts

ADD CONSTRAINT check-account-type CHECK (Account
Type IN ('Saving', 'Checking'));

8. Complex Queries.

21. SELECT customers.FirstName, customers.LastName, Accounts.Account
Number, MAX(Accounts.Balance) AS HighestBalance
FROM customers

JOIN Accounts ON customers.CustomerID = Account.CustomerID

GROUP BY customer.CustomerID

ORDER BY Highest_Balance DESC

LIMIT 1;

22. BEGIN TRANSACTION;

UPDATE Accounts.

SET Balance = Balance - 1000

WHERE Account Number = 1003;

UPDATE Accounts

SET Balance = Balance + 1000

WHERE Account Number = ~~1003~~, 1001;

COMMIT;

23. SELECT Customer.FirstName, Customer.LastName, SUM (Accounts.Balance) AS TotalBalance

FROM Customers

JOIN Accounts ON Customers.CustomerID = Accounts.CustomerID

GROUP BY Customers.CustomerID;

g.. Aggregation

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

24. SELECT COUNT(*) AS TotalSavingAccounts
FROM ~~Accounts~~ Accounts
WHERE AccountType = 'Saving';

10. Transactions.

26. BEGIN TRANSACTION;

UPDATE Accounts

SET Balance = Balance - 500

WHERE Account Number = 1002;

UPDATE Accounts

SET Balance = Balance + 500

WHERE Account Number = 1003;

COMMIT;

library system set 3.

1. Create Table Query

```
CREATE TABLE Library (
    BookID INT AUTO_INCREMENT,
    Title VARCHAR(255),
    Author VARCHAR(255),
    Publisher VARCHAR(255),
    Genre VARCHAR(100),
    Published Year INT,
    ISBN VARCHAR(20) UNIQUE,
    Pages INT,
    Copies Available INT,
    Price DECIMAL(10,2)
);
```

~~ENDS~~ Insert Data:

```
INSERT INTO Library (Title, Author, Publisher, Genre, Published Year, ISBN, Pages, Copies Available, Price)
```

VALUES

('To Kill a Mockingbird', 'Harper Lee', 'J.B Lippincott', 'Fiction', 1960, '978-006120084', 324, 5, 15.99)

('1984', 'George Orwell', 'Harvill Secker', 'Dystopian', 1949, '978-0451524935', 328, 2, 9.99)

('The Great Gatsby', 'F. Scott Fitzgerald', 'Scribner', 'Fiction', 1925, '978-0743273565', 180, 3, 10.99),

('The Catcher in the Rye', 'J.D Salinger', 'Little; Brown',
 'Fiction', 1951, '978-0316709488', 297, 4, 12.99),

('The Hobbit', 'J.R.R. Tolkien', 'HarperCollins', 'Fantasy',
 1937, '978-0618968633', 310, 6, 13.99)

3 Update Queries

1. UPDATE Library

SET Price = ~~11.99~~ 11.99

WHERE ISBN = '978-0451524935'

AND Genre = 'Dystopian'

AND Published Year < 1950;

2. UPDATE Library

SET CopiesAvailable = 10

WHERE Genre = 'Fiction'

AND Published Year > 1950;

3. UPDATE Library

SET Price = Price * 0.95

WHERE Genre = 'Fiction'

AND Pages > 300;

4. UPDATE Library

SET Pages = 350

WHERE Copies Available > 4

AND Price < 14;

3. UPDATE Library

SET Price = Price * 1.10;

WHERE Genre = 'Fantasy'

AND PublishedYear < 1950

AND Pages < 300;

4. UPDATE Library

SET Copies Available = 0

WHERE Price > 12

AND genre IN ('Fiction', 'Dystopian');

5. UPDATE Library

SET PublishedYear = 2020

WHERE Author LIKE '%J.Rolkien.s.'

AND Price BETWEEN 10 AND 15;

6. UPDATE Library

SET Price = 8.99

WHERE Author LIKE 'George Orwell'

AND Pages > 300;

7. UPDATE Library

SET Price = Price * 0.85

WHERE genre = 'Fiction'

AND PublishedYear < 1950

AND Copies Available < 5;

10. UPDATE Library
SET Price = 17.99

WHERE Title = 'To Kill a Mockingbird'

AND Published Year = 1960

AND Copies Available > 4;

4 Delete Queries:

1. DELETE FROM Library

WHERE ISBN IN '978-0451524935', '978-0618968633'

AND Genre = 'Dystopian'

AND Copies Available > 2;

2. DELETE FROM Library

WHERE ISBN IN '978-0451524935', '978-0618968633'

AND Genre =

2. DELETE FROM Library

WHERE Published Year < 1950

AND Price = 10;

3. DELETE FROM Library

WHERE Genre = 'Fiction'

AND Copies Available < 3

AND Published Year < 1960;

5. DELETE FROM Library
WHERE Author = 'Harper Lee'
AND Price < 12;
6. DELETE FROM Library
WHERE CopiesAvailable = 0
AND Price > 15;
7. DELETE FROM Library
WHERE ISBN = '978-0343273565'
AND Genre = 'Fiction'
AND PublishedYear < 1950;
8. DELETE FROM Library
WHERE PublishedYear >= 2000
AND (Price Between 10 AND 1 OR Price > 15);
9. DELETE FROM Library
WHERE Genre = 'Fantasy'
AND CopiesAvailable > 3
AND CopiesAvailable < 10;
10. DELETE FROM Library
WHERE Author = 'J.D. Salinger'
AND Pages < 300
AND Price > 12;

5. Select Queries

1. SELECT *
FROM ~~LIB~~library
WHERE genre = 'Fiction'
AND Published Year > 1950
AND Pages > 200
AND Price BETWEEN 10 AND 15;

2. SELECT *
FROM Library
WHERE Pages > 300
AND Price > 12
AND genre 1 = 'Fantasy';

3. SELECT *
FROM Library
WHERE Published Year BETWEEN 1925 AND 1950
AND Price < 13
AND Copies Available > 3;

4. SELECT *
FROM Library
WHERE Author = 'Harper Lee'
AND Published Year > 1950
AND Price > 10;

5. SELECT *

FROM Library

WHERE Author = 'J.R.R. Tolkien'

AND Copies Available > 5

AND Price BETWEEN 12 AND 15;

6. SELECT *

FROM Library

WHERE (Genre = 'Dystopian' OR Genre = 'Fantasy')

AND Pages > 300

ORDER BY Price ~~DESC~~

LIMIT 1;

7. SELECT *

FROM Library

WHERE Author LIKE '%Tolkien%'

AND PublishedYear < 1990

AND Copies Available BETWEEN 4 AND 6;

8. SELECT *

FROM Library

WHERE Genre = 'Fiction'

AND Pages > 200

AND Price BETWEEN 10 AND 20

AND Copies Available BETWEEN 2 AND 5;

9. SELECT *
FROM Library
WHERE Published Year < 1950
AND Price < 15
AND Copies Available < 3;

10. SELECT Title , Author
FROM Library
WHERE Price > 12
AND Copies Available > 4
AND Published Year > 1930;