

CRUD practice questions:-

Set 1

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 INTO Students (StudentID,
FirstName, LastName, Age, EnrollmentDate,
Major)

VALUES

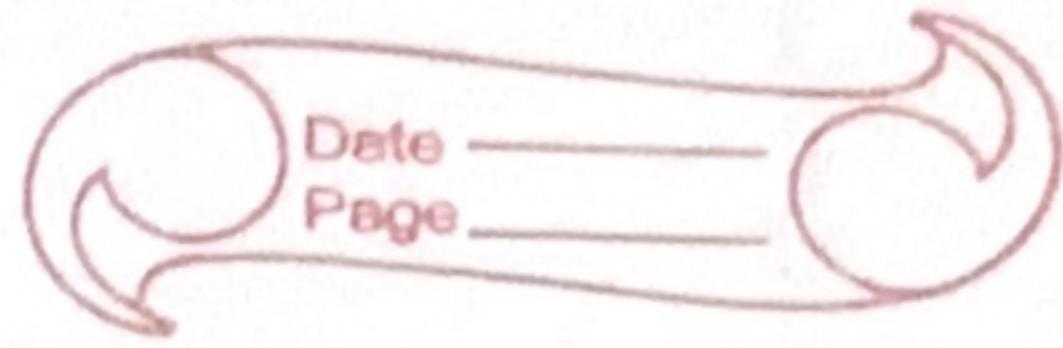
(1, 'Alice', 'Johnson', 18, '2023-09-01',
'Computer Science')

(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:

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 Age
Last Name = "Cooper"
WHERE First Name = "Daisy";
4. UPDATE Students SET Major = "Undeclared"
WHERE Age < 20;
5. UPDATE Students SET
EnrollmentDate = "2024-01-01"
WHERE StudentID = 5;
6. UPDATE Student SET Major = "Physics"
WHERE Major = "Biology";
7. UPDATE Students SET Age = 23
WHERE First Name = "Charlie";

9. UPDATE Students SET FirstName = 'Alex'
WHERE Age = (SELECT MIN(Age)
FROM Students);

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

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

Question 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";

6 DELETE FROM Students
WHERE Age != NULL;
IS

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

Practice Set 2 :-

1. Create table queries

```
1. CREATE TABLE customers (
    CustomerID INT PRIMARY KEY,
    AUTO_INCREMENT,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Email      VARCHAR(100) UNIQUE,
    PhoneNumber VARCHAR(200),
);
```

```
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),
);
```

2.) Inserting data.

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@gmail.com’,`
`‘0987654321’, ‘456 Elm St, Townville’),`

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

Q4. `INSERT INTO Accounts (AccountNumber, CustomerID, AccountType, Balance, DateCreated)`

`VALUES`

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

`(1002, 1, ‘Checking’, 1500.00, ‘2023-03-01’),`

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

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

3. Update Data.

Q5. UPDATE Accounts .

SET Balance = 5500.00

WHERE AccountNumber = 1001;

Q6 UPDATE Customers

SET Email = "jane.smith@hotmail.com"

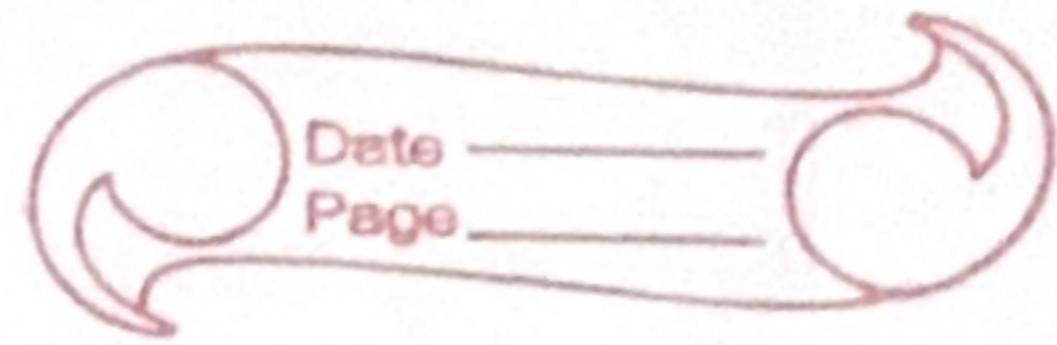
WHERE CustomerID = 2;

Q7. UPDATE Accounts

SET Balance = Balance * 1.10

WHERE AccountType = "Saving";

4. Select Queries



Q8.

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

Q9.

```
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";
```

Q10.

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

5.

Deleting Data

Q11

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

Q12

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

Q13

```
DELETE FROM Account  
WHERE DateCreated < '2023-02-01'
```

6.

Join Queries

Q14.

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

```
JOIN Accounts ON Customers.CustomerID  
= Accounts.AccountID CustomerID
```

```
WHERE Account.Balance > 2000;
```

Q15. SELECT AccountType , SUM(Balance)
AS TotalBalance
FROM Accounts
WHERE AccountType = 'Saving'
GROUP BY AccountType ;

Q16. SELECT Customers.FirstName , Customers.LastName
Accounts.AccountNumber , Accounts.Balance
FROM Customers
LEFT JOIN Accounts ON Customers.
CustomerID = Accounts.CustomerID ;

7 Constraints and Validation

Q17. ALTER TABLE Accounts
ADD CONSTRAINT check_balance
CHECK (Balance >= 0);

Q18. ALTER TABLE Customers
MODIFY COLUMN Email VARCHAR(100)
NOT NULL UNIQUE ;

Q19 ALTER TABLE Accounts

ADD CONSTRAINT fk_customer

FOREIGN KEY (CustomerID)

REFERENCES customers (CustomerID);

Q20 ALTER TABLE Accounts

ADD CONSTRAINT check_account_type

CHECK (AccountType IN ('Saving', 'Checking'));

~~Q21~~

Complex Queries

Q21

SELECT Customers.FirstName,

Customers.LastName, Accounts.AccountNumber,

MAX(Accounts.Balance) AS HighestBalance

FROM customers

JOIN Accounts ON Customers.CustomerID

= Accounts.CustomerID

GROUP BY HighestBalance DESC

LIMIT 1;



BEGIN TRANSACTION;

UPDATE Accounts

SET Balance = Balance - 1000

WHERE Account Number = 1003;

UPDATE Accounts

SET Balance = Balance + 1000

WHERE Account Number = 1001;

COMMIT;

Q23.

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

9.

Aggregation

Q24

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

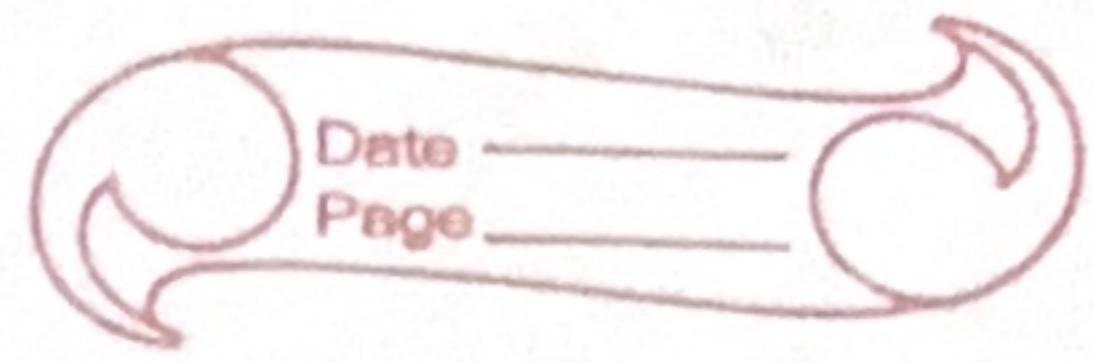
Q25

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

10.

Transactions

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



Practice set 3

1. Create table query

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

2. INSERT DATA

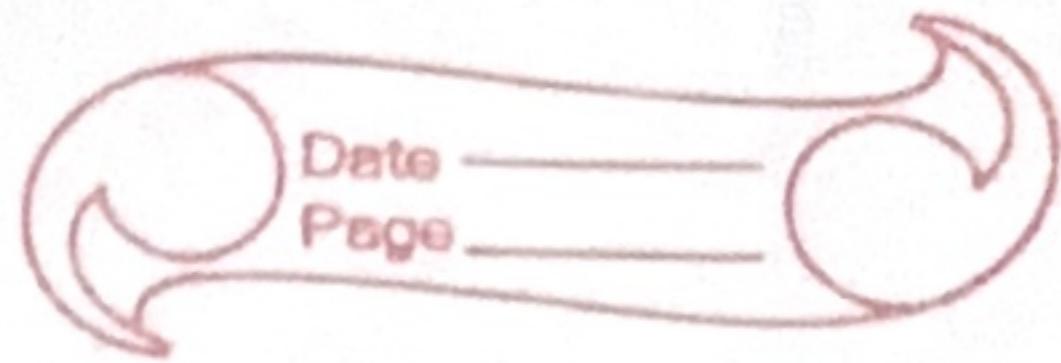
INSERT INTO Library (Title,
Author, Publisher, genre, Published Year
ISBN, Pages, copies Available, Price)

VALUES

(‘To Kill a Mockingbird’, ‘Harper Lee’,
‘J. B. Lippincott’, ‘Fiction’, 1960,
‘9780061120084’, 324, 5, 15.99),
(‘1984’, ‘George Orwell’, ‘Harcourt Secker’,
‘Dystopian’, 1949, ‘9780451524935’,
328, 2, 9.99),

(‘The Great Gatsby’, ‘F. Scott
Fitzgerald’, ‘Scribner’, ‘Fiction’,
1925, ‘9780743273565’, 180,
3, 10.99),

(‘The Catcher in the Eye’, ‘J. D. Salinger’,
‘Little Brown’, ‘Fiction’, 1951,
‘9780316769488’, 277, 4, 12.99),



3. Update queries with multiple conditions

1. UPDATE Library

SET Price = 11.99

WHERE ISBN = "9780451524935"

AND Genre = "Dystopian"

AND PublishedYear < 1950;

2. UPDATE Library

SET CopiesAvailable = 10

WHERE Genre = "Fiction"

AND PublishedYear > 1950;

3. UPDATE Library

SET Price = Price * 0.95

WHERE Genre = "Fiction"

AND Pages > 300;

4. UPDATE Library

SET Pages = 350

WHERE CopiesAvailable > 4

AND Price < 14

5.

UPDATE Library

SET Price = Price * 1.10;

WHERE genre = "Fantasy"

AND PublishedYear < 1950

AND Pages < 300;

6.

UPDATE Library

SET CopiesAvailable = 0

WHERE Price > 12

AND genre IN ("Fiction", "Dystopian")

7.

UPDATE Library

SET PublishedYear = 2020 Price = 8.99

WHERE Author LIKE "%George Orwell%"

AND Pages > 300;

8.

UPDATE Library

SET PublishedYear = 2020

WHERE Author LIKE "%J.R.R. Tolkien%"

AND Price BETWEEN 10 AND 15-

9. UPDATE Library

SET Price = Price * 0.85

WHERE Genre = 'Fiction'

AND PublishedYear < 1950

AND CopiesAvailable < 5;

10. UPDATE Library

SET Price = 17.99

WHERE Title = 'To kill a mockingbird'

AND PublishedYear = 1960

AND CopiesAvailable > 4;

4 → Delete queries with multiple conditions

1. DELETE FROM Library

WHERE ISBN IN ('9780451524939',
'9780618968633')

AND Genre = 'Dystopian'

AND CopiesAvailable > 2;

2

DELETE FROM Library
WHERE PublishedYear < 1950
AND Price < 10°;

3

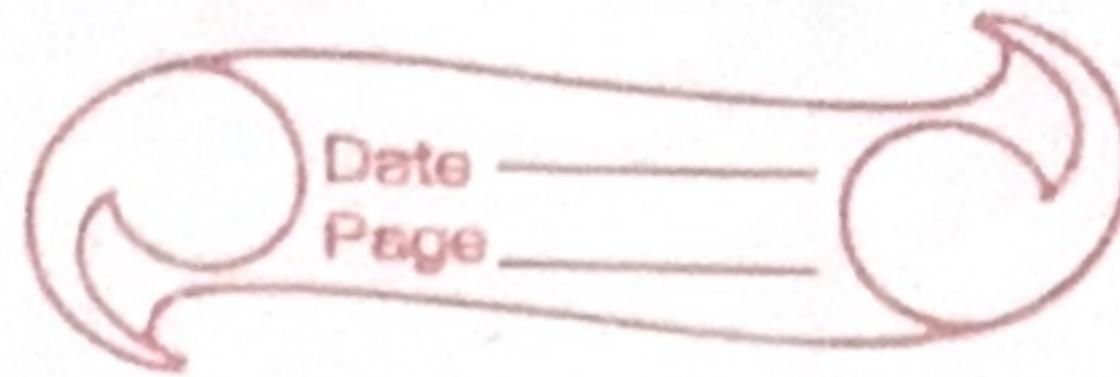
DELETE FROM Library
WHERE Genre = 'Fiction'
AND CopiesAvailable < 3
AND PublishedYear < 1960°;

4

DELETE FROM Library
WHERE Pages < 200
AND PublishedYear > 1920
AND PublishedYear < 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 = '9780743273565'
AND Genre = 'Fiction'
AND PublishedYear < 1950;

8. DELETE FROM Library

WHERE PublishedYear > 2000
AND (Price BETWEEN 10 AND 15
OR Price > 15);

9. DELETE FROM Library

WHERE Genre = 'Fantasy'
AND CopiesAvailable > 3
AND CopiesAvailable < 10;

D.

DELETE FROM Library

WHERE Author = "J.D.Salinger"

AND Pages < 300

AND Price > 12;

5 →

Select Queries with multiple
conditions

1.

SELECT * FROM LIBRARY

WHERE genre = "Fiction"

AND PublishedYear > 1950

AND Pages > 200

AND Price BETWEEN 10 AND 15

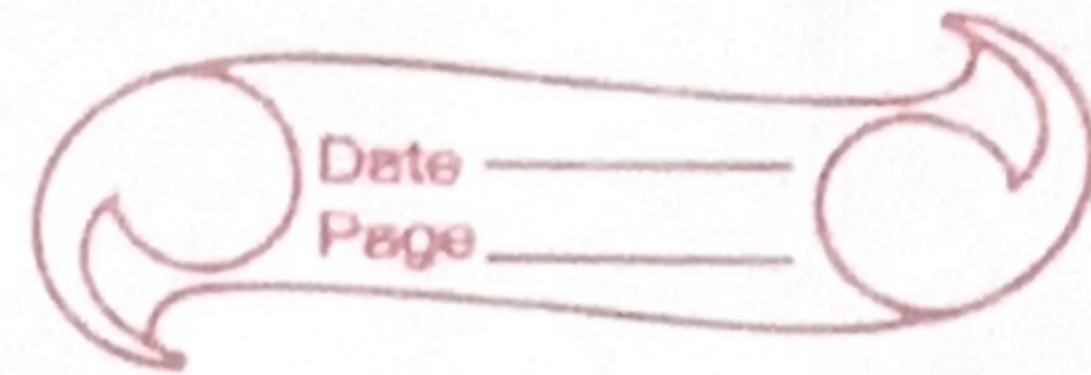
2

SELECT * FROM LIBRARY

WHERE Pages > 300

AND Price > 12

AND genre != "Fantasy";



3

```
SELECT * FROM Library
```

```
WHERE PublishedYear <= Between
```

```
BETWEEN 1925 AND 1950;
```

```
AND Price < 13;
```

```
AND CopiesAvailable > 3;
```

4

```
SELECT * FROM Library
```

```
WHERE Author = 'Harper Lee' OR
```

```
Author = 'George Orwell')
```

```
AND PublishedYear > 1950
```

```
AND Price > 10;
```

5

```
SELECT * FROM Library
```

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

```
AND CopiesAvailable > 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 < 1940
AND CopiesAvailable BETWEEN
4 AND 6 ;
8. SELECT * FROM Library
WHERE Genre = 'Fiction'
AND Pages > 200
AND Price BETWEEN 10 AND 20
AND CopiesAvailable BETWEEN
2 AND 5 ;
9. SELECT * FROM Library
WHERE PublishedYear < 1950
AND Price < 15
AND CopiesAvailable < 3 ;
10. SELECT Title, Author
FROM Library
WHERE Price > 12
AND CopiesAvailable > 4
AND PublishedYear > 1930 ;