

Set 1

• Create Table

```
CREATE TABLE Students (
    StudentID INT PRIMARY KEY AUTO_INCREMENT,
    FirstName VARCHAR (50) NOT NULL,
    LastName VARCHAR (50) NOT NULL,
    Age INT CHECK (Age >= 10),
    EnrollmentDate DATE DEFAULT CURRENT_DATE,
    Major VARCHAR (100)
```

;

• Insert Queries

```
INSERT INTO Students (StudentID, FirstName, LastName,
    Age, EnrollmentDate, Major) VALUES
(1, 'Alice', 'Johnson', 18, '2023-09-01', 'Comp'),
(2, 'Bob', 'Smith', 20, '2022-06-15', 'Math'),
(3, 'Charlie', 'Brown', 19, '2021-08-20', 'Phys'),
(4, 'Daisy', 'Carter', 21, '2023-01-10', 'Biology'),
(5, 'Ethan', 'Taylor', 22, '2023-03-25', 'Chemistry');
```

• Update 'Queries'.

```
1. UPDATE Students
   SET Major = 'Data Science'
 WHERE StudentID = 1;
```

2 UPDATE Students

SET Age = Age + 1

WHERE Enrollment Date < '2023-01-01'

3 UPDATE Students

SET LastName = 'Cooper'

WHERE FirstName = 'Darsy';

4 UPDATE Students

SET Major = 'Undeclared'

WHERE Age < 20;

5 UPDATE Students

SET Enrollment Date > '2024-01-01'

WHERE Student ID = S;

6 UPDATE Students

SET Major = 'Physics'

WHERE Major = 'Biology';

7 UPDATE Students

SET LastName = 'Williams'

WHERE Major = 'Mathematics';

8 UPDATE Students

SET Age = NULL

WHERE Major = 'Undeclared';

10. UPDATE Students

SET FirstName = 'Alex'

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

, DELETE Queries

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 IS NULL;

7. DELETE FROM Students

WHERE LastName LIKE 'C.%';

8. DELETE FROM Students
WHERE Enrollment Date < '2022-01-01' - 2
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)

~~Practice Set 2~~

g. CREATE TABLE Customers
(Customer ID INT AUTO_INCREMENT PRIMARY KEY,
FirstName VARCHAR (50),
LastName VARCHAR (50),
Email VARCHAR (100) UNIQUE,
PhoneNumber VARCHAR (15),
Address VARCHAR (200))

2. CREATE TABLE Accounts

AccountNumber INT PRIMARY KEY ,

CustomerID INT ,

AccountType VARCHAR (30) ,

Balance DECIMAL (10,2) ,

DateCreated DATE ,

FOREIGN KEY (CustomerID) REFERENCES

(Customers (CustomerID)) ;

3. INSERT DATA

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

VALUES

('John' , 'Doe' , 'John.Doe@gmail.com' , '123USA28' ,
'1123 Main St , Cityville') ,

('Jane' , 'Smith' , '123@jane.smith@gmail.com' , '09876543'
'456 Elm St , Townville') ,

('Mike' , 'Johnson' , 'mike.johnson@gmail.com'
'111223345' , '789 Oak St , Villageville') ;

4. INSERT INTO Accounts (AccountNumber , CustomerID , AccountType , Balance , DateCreated) VALUES

(1001 , 1 , 'Savings' , 500.00 , '2023-01-15') ,

(1002 , 2 , 'Checking' , 1500.00 , '2023-02-20') ,

(1003 , 3 , 'Savings' , 2000.00 , '2023-03-01') ,

(1004 , 4 , 'Checking' , 3000.00 , '2023-03-10') ,

Update Data

5. UPDATE Account

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 * 1.10

WHERE AccountType = 'Saving';

SELECT Queries

8. SELECT Customers.CustomerID, Customers.FirstName,
Customers.LastName, Accounts.Balance
FROM Customers

JOIN Accounts ON Customers.CustomerID = Accounts.
CustomerID

WHERE Accounts.AccountType = 'Saving';

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';

10. SELECT Customer ID, Account Number, Account Type,
Balance FROM Accounts
WHERE Balance < 2000;

DELETING DATA

11. DELETE FROM Accounts
WHERE Account Number = 100;

12. DELETE FROM Account Customers
WHERE Phone Number LIKE '123-1111';

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

JOIN Queries

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 = 'Savings'
GROUP BY AccountType;

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 ('Savings', 'Checking'));

COMPLEX Queries

21. SELECT Customers.FirstName, Customers.LastName,
Accounts.AccountNumber, MAX(Accounts.Balance)
AS HighestBalance FROM Customers
JOIN Accounts ON Customer.CustomerID = Account-
Customer.ID GROUP BY Customer.CustomerID
ORDER BY HighestBalance DESC
LIMIT 5;

2) BEING TRANSACTION;

UPDATE Account
SET Balance = Balance - 1000
WHERE Account Number = 1003;

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

COMMIT;

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

9. Aggregation

10. SELECT AVG(Balance) AS Average Balance
FROM Accounts;

21. SELECT COUNT(*) AS Total Saving Accounts
FROM Accounts
WHERE Account Type = 'Saving';

2F. BEGIN TRANSACTION;

UPDATE Account

SET Balance = Balance - 500

WHERE Account Number = 1002;

UPDATE Account

SET Balance = Balance + 500

WHERE Account Number = 1003;

COMMIT;

~~Practiceset 3~~

1. CREATE TABLE library C

BookID INT AUTO_INCREMENT PRIMARY KEY,

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)

) ;

2. INSERT INTO library (Title, Author, Publisher, Genre, Published Year, ISBN, Page s, Copies Available, Price) VALUES

(('To kill a mockingbird', 'Harper Lee', 'J.B. Lippincott', 'Fiction', 1960, '978-0061120084', 320, 5, 15.99),
(('1984', 'George Orwell', 'Harcourt Secker', 'Dystopian', 1984, '978-0451524935', 328, 2, 9.99),
('The Great Gatsby', 'F. Scott Fitzgerald', 'Scribner', 'Fiction', 1925, '978-0743273515', 160, 3, 10.99),
('The Catcher in the Rye', 'J.D. Salinger', 'Little, Brown', 'Fiction', 1951, '978-0316769488', 272, 4, 12.99),
('The Hobbit', 'J.R.R. Tolkien', 'Harper Collins', 'Fantasy', 1937, '978-0007891863', 310, 6, 13.99),

3. UPDATE library

SET Price = 11.99

WHERE ISBN = '978-0451524935'

AND Genre = 'Dystopian'

AND Published Year > 1950;

4. UPDATE library

SET Copies Available = 10

WHERE Genre = 'Fiction'

AND Published Year > 1950;

5. UPDATE library
SET Pages = 350
WHERE Copies Available > 4
AND Price < 14;
6. UPDATE library
SET Copies Available = 0
WHERE Price > 12
AND Genre IN ('Fiction', 'Dystopian');
7. UPDATE library
SET Published Year = 2020
WHERE Author like 'J.R.R.Tolkien'
AND Price BETWEEN 10 AND 15;
8. UPDATE library
SET Price = Price * 0.99
WHERE Author like 'George Orwell'
AND Pages > 300;
9. UPDATE library
SET Price = Price * 0.95
WHERE Genre = 'Fiction'
AND Published Year < 1950
AND (Copies Available < 5,

AND (Genre = 'OR' Genre = '')

AND (Genre IN (' ',))

Date _____
Page _____

10. DELETE FROM library
WHERE ISBN IN ('978-051520933')
AND Genre = 'Dystopian'
AND (Copies Available > 2);

11. DELETE FROM library
WHERE Published Year < 1950
AND Price < 10;

12. DELETE FROM library
WHERE Genre = 'Fiction'
AND (Copies Available < 3
AND Published Year < 1960);

13. DELETE FROM library
WHERE Pages < 200
AND Published Year > 1920
AND Published Year < 1960;

14. DELETE FROM library
WHERE Author = 'Floriper Lee'
AND Price < 12;

15. DELETE FROM library
WHERE (Copies Available = 0
AND Price > 15);

16. DELETE FROM library
WHERE Published Year > 2000
AND (Price Between 10 AND 15)
AND Price > 15;

17. DELETE FROM library
WHERE Genre = 'Fantasy'
AND Copies Available > 3
AND Copies Available < 10;

18. DELETE FROM library
WHERE Author = 'J. R. Salinger'
AND Pages < 300
AND Price > 12;

SELECT query

19.
SELECT * FROM library
WHERE Genre = 'Fiction'
AND Published Year > 1950
AND Pages > 200
AND Price Between 10 AND 15;

Date _____
Page _____

2. SELECT X FROM library
WHERE Pages > 300
AND Price > 12
AND Genre != "Fantasy";
3. SELECT X FROM library
WHERE PublishedYear BETWEEN 1925 AND 1950
AND Price < 13
AND CopiesAvailable > 3;
4. SELECT X FROM library
WHERE (Author = "Harper Lee" OR Author = "George Orwell") AND PublishedYear > 1950
AND Price > 10;
5. SELECT X FROM library
WHERE (Genre = "Dystopian" OR Genre = "Fantasy")
AND Pages > 300
ORDER BY Price DESC
LIMIT 2;
6. ~~SELECT~~ SELECT X FROM library
WHERE Author LIKE "%J. R. Tolkien%"
AND PublishedYear < 1940
AND CopiesAvailable BETWEEN 4 AND 6;

Date _____
Page _____

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;