

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**LAB: CYCLE SHEET –2(Joins and Sub Query) - FALL SEMESTER 2023-2024****Programme Name & Branch: B.Tech****Course Name: Database Systems LAB****Course Code: BCSE302L**

Assume the following relations:

BOOKS(DocId, Title, Publisher, Year)

STUDENTS(StId, StName, Major, Age)

AUTHORS(AName, Address)

Borrows(DocId, StId, Date)

Has-written(DocId, AName)

Describes(DocId, Keyword)

TABLE CREATION:

```
SQL> CREATE TABLE BOOKS (  
2 DocId INT,  
3 Title VARCHAR(100),  
4 Publisher VARCHAR(100),  
5 Year INT,  
6 PRIMARY KEY (DocId)  
7 );
```

Table created.

```
SQL> CREATE TABLE STUDENTS (  
2 StId INT,  
3 StName VARCHAR(100),  
4 Major VARCHAR(50),  
5 Age INT,  
6 PRIMARY KEY (StId)  
7 );
```

Table created.

```
SQL> CREATE TABLE AUTHORS (  
2 AName VARCHAR(100),  
3 Address VARCHAR(200),  
4 PRIMARY KEY (AName)  
5 );
```

Table created.

```
SQL> CREATE TABLE Borrows (  
2 DocId INT,  
3 StId INT,  
4 BorrowDate DATE,  
5 FOREIGN KEY (DocId) REFERENCES BOOKS(DocId),  
6 FOREIGN KEY (StId) REFERENCES STUDENTS(StId)  
7 );
```

Table created.

```
SQL> CREATE TABLE Has_written (  
2 DocId INT,  
3 AName VARCHAR(100),  
4 FOREIGN KEY (DocId) REFERENCES BOOKS(DocId),  
5 FOREIGN KEY (AName) REFERENCES AUTHORS(AName)  
6 );
```

Table created.

```
SQL> CREATE TABLE Describes (  
2     DocId INT,  
3     Keyword VARCHAR(50),  
4     FOREIGN KEY (DocId) REFERENCES BOOKS(DocId)  
5 );
```

Table created.

INSERTING VALUES :

```
SQL> INSERT INTO AUTHORS (AName, Address)  
2 VALUES ('Author 1', 'Address 1');
```

1 row created.

```
SQL>  
SQL> INSERT INTO AUTHORS (AName, Address)  
2 VALUES ('Author 2', 'Address 2');
```

1 row created.

```
SQL>  
SQL> INSERT INTO AUTHORS (AName, Address)  
2 VALUES ('Author 3', 'Address 3');
```

1 row created.

```
SQL>  
SQL> INSERT INTO AUTHORS (AName, Address)  
2 VALUES ('Author 4', 'Address 4');
```

1 row created.

```
SQL>  
SQL> INSERT INTO AUTHORS (AName, Address)  
2 VALUES ('Author 5', 'Address 5');
```

1 row created.

```
SQL> INSERT INTO STUDENTS (StId, StName, Major, Age)  
2 VALUES (1, 'John Doe', 'Computer Science', 21);
```

1 row created.

```
SQL>  
SQL> INSERT INTO STUDENTS (StId, StName, Major, Age)  
2 VALUES (2, 'Jane Smith', 'English Literature', 20);
```

1 row created.

```
SQL>  
SQL> INSERT INTO STUDENTS (StId, StName, Major, Age)  
2 VALUES (3, 'Michael Johnson', 'History', 22);
```

1 row created.

```
SQL>  
SQL> INSERT INTO STUDENTS (StId, StName, Major, Age)  
2 VALUES (4, 'Emily Brown', 'Mathematics', 23);
```

1 row created.

```
SQL>  
SQL> INSERT INTO STUDENTS (StId, StName, Major, Age)  
2 VALUES (5, 'Daniel Wilson', 'Physics', 21);
```

1 row created.

```
SQL> INSERT INTO Borrows (DocId, StId, BorrowDate)  
2 VALUES (1, 1, TO_DATE('2023-06-01', 'YYYY-MM-DD'));
```

1 row created.

```
SQL>  
SQL> INSERT INTO Borrows (DocId, StId, BorrowDate)  
2 VALUES (2, 2, TO_DATE('2023-06-05', 'YYYY-MM-DD'));
```

1 row created.

```
SQL>  
SQL> INSERT INTO Borrows (DocId, StId, BorrowDate)  
2 VALUES (3, 3, TO_DATE('2023-06-07', 'YYYY-MM-DD'));
```

1 row created.

```
SQL>  
SQL> INSERT INTO Borrows (DocId, StId, BorrowDate)  
2 VALUES (4, 4, TO_DATE('2023-06-10', 'YYYY-MM-DD'));
```

1 row created.

```
SQL>  
SQL> INSERT INTO Borrows (DocId, StId, BorrowDate)  
2 VALUES (5, 5, TO_DATE('2023-06-12', 'YYYY-MM-DD'));
```

1 row created.

```
SQL> INSERT INTO BOOKS (DocId, Title, Publisher, Year)  
2 VALUES (1, 'Book 1', 'Publisher A', 2019);
```

1 row created.

```
SQL>  
SQL> INSERT INTO BOOKS (DocId, Title, Publisher, Year)  
2 VALUES (2, 'Book 2', 'Publisher B', 2020);
```

1 row created.

```
SQL>  
SQL> INSERT INTO BOOKS (DocId, Title, Publisher, Year)  
2 VALUES (3, 'Book 3', 'Publisher C', 2021);
```

1 row created.

```
SQL>  
SQL> INSERT INTO BOOKS (DocId, Title, Publisher, Year)  
2 VALUES (4, 'Book 4', 'Publisher D', 2022);
```

1 row created.

```
SQL>  
SQL> INSERT INTO BOOKS (DocId, Title, Publisher, Year)  
2 VALUES (5, 'Book 5', 'Publisher E', 2023);
```

1 row created.

```
SQL> INSERT INTO Has_written (DocId, AName)
2 VALUES (1, 'Author 1');

1 row created.

SQL>
SQL> INSERT INTO Has_written (DocId, AName)
2 VALUES (2, 'Author 2');

1 row created.

SQL>
SQL> INSERT INTO Has_written (DocId, AName)
2 VALUES (3, 'Author 3');

1 row created.

SQL>
SQL> INSERT INTO Has_written (DocId, AName)
2 VALUES (4, 'Author 4');

1 row created.

SQL>
SQL> INSERT INTO Has_written (DocId, AName)
2 VALUES (5, 'Author 5');

1 row created.
```

```
SQL> INSERT INTO Describes (DocId, Keyword)
2 VALUES (1, 'Science');

1 row created.

SQL>
SQL> INSERT INTO Describes (DocId, Keyword)
2 VALUES (2, 'Fiction');

1 row created.

SQL>
SQL> INSERT INTO Describes (DocId, Keyword)
2 VALUES (3, 'History');

1 row created.

SQL>
SQL> INSERT INTO Describes (DocId, Keyword)
2 VALUES (4, 'Technology');

1 row created.

SQL>
SQL> INSERT INTO Describes (DocId, Keyword)
2 VALUES (5, 'Art');

1 row created.
```

DISPLAY TABLES:

```
SQL> SELECT * FROM BOOKS;
```

DOCID	TITLE	PUBLISHER	YEAR
1	Book 1	Publisher A	2019
2	Book 2	Publisher B	2020
3	Book 3	Publisher C	2021
4	Book 4	Publisher D	2022
5	Book 5	Publisher E	2023

```
SQL> SELECT * FROM STUDENTS;
```

STID	STNAME	MAJOR	AGE
1	John Doe	Computer Science	21
2	Jane Smith	English Literature	20
3	Michael Johnson	History	22
4	Emily Brown	Mathematics	23
5	Daniel Wilson	Physics	21

```
SQL> SELECT * FROM AUTHORS;
```

ANAME	ADDRESS
Author 1	Address 1
Author 2	Address 2
Author 3	Address 3
Author 4	Address 4
Author 5	Address 5

```
SQL> SELECT * FROM Borrows;
DOCID  STID  BORROWDATE
-----
1      1      01-JUN-23
2      2      05-JUN-23
3      3      07-JUN-23
4      4      10-JUN-23
5      5      12-JUN-23

SQL> SELECT * FROM Has_written;
DOCID  ANAME
-----
1      Author 1
2      Author 2
3      Author 3
4      Author 4
5      Author 5
```

```
SQL> SELECT * FROM Describes;
DOCID  KEYWORD
-----
1      Science
2      Fiction
3      History
4      Technology
5      Art
```

1. Simple Queries and Joins:

List the year and title of each book.

```
SQL> SELECT Year, Title FROM BOOKS;
YEAR  TITLE
-----
2019  Book 1
2020  Book 2
2021  Book 3
2022  Book 4
2023  Book 5
```

List all information about students whose major is CS.

```
SQL> SELECT * FROM STUDENTS WHERE Major = 'CS';
no rows selected
```

List all students with the books they can borrow.

```
SQL> SELECT S.StName, B.Title
2 FROM STUDENTS S
3 JOIN Borrows BR ON S.StId = BR.StId
4 JOIN BOOKS B ON BR.DocId = B.DocId;
STNAME          TITLE
-----
John Doe        Book 1
Jane Smith      Book 2
Michael Johnson Book 3
Emily Brown     Book 4
Daniel Wilson   Book 5
SQL>
```

List all books published by McGraw-Hill before 1990.

```
SQL> SELECT * FROM BOOKS WHERE Publisher = 'McGraw-Hill' AND Year < 1990;

no rows selected
```

List the name of those authors who are living in Davis.

```
SQL> SELECT AName FROM AUTHORS WHERE Address LIKE '%Davis%';

no rows selected
```

List the name of students who are older than 30 and who are not studying CS.

```
SQL> SELECT StName FROM STUDENTS WHERE Age > 30 AND Major <> 'CS';

no rows selected
```

Rename AName in the relation AUTHORS to Name

```
SQL> ALTER TABLE AUTHORS RENAME COLUMN AName TO Name;

Table altered.

SQL>
SQL> select * from AUTHORS;
```

NAME	ADDRESS
Author 1	Address 1
Author 2	Address 2
Author 3	Address 3
Author 4	Address 4
Author 5	Address 5

List the names of all students who have borrowed a book and who are CS majors.

```
SQL> SELECT S.StName
2 FROM STUDENTS S
3 INNER JOIN Borrows B ON S.StId = B.StId
4 WHERE S.Major = 'CS';

no rows selected
```

List the title of books written by the author 'Silberschatz'.

```
SQL> SELECT B.Title
2 FROM BOOKS B
3 INNER JOIN Has_written HW ON B.DocId = HW.DocId
4 INNER JOIN AUTHORS A ON HW.AName = A.Name
5 WHERE A.Name = 'Silberschatz';

no rows selected
```

List No books that have the keyword 'database'.


```
SQL> SELECT B.Title
2 FROM BOOKS B
3 LEFT JOIN Describes D ON B.DocId = D.DocId
4 WHERE D.Keyword <> 'database' OR D.Keyword IS NULL;

TITLE
-----
Book 1
Book 2
Book 3
Book 4
Book 5
```

Find the name of the youngest student.

```
SQL> SELECT StName
2 FROM STUDENTS
3 WHERE Age = (SELECT MIN(Age) FROM STUDENTS);

STNAME
-----
Jane Smith
```

Find the title of the oldest book.

```
SQL> SELECT Title
2 FROM BOOKS
3 WHERE Year = (SELECT MIN(Year) FROM BOOKS);

TITLE
-----
Book 1
```

List each book with its keywords.

```
SQL> SELECT B.Title, D.Keyword
2 FROM BOOKS B
3 LEFT JOIN Describes D ON B.DocId = D.DocId;

TITLE                                     KEYWORD
-----
Book 1                                     Science
Book 2                                     Fiction
Book 3                                     History
Book 4                                     Technology
Book 5                                     Art
```

List each student with the books s/he has borrowed.

```
SQL> SELECT S.StName, B.Title
2 FROM STUDENTS S
3 JOIN Borrows BR ON S.StId = BR.StId
4 JOIN BOOKS B ON BR.DocId = B.DocId;
```

STNAME	TITLE
John Doe	Book 1
Jane Smith	Book 2
Michael Johnson	Book 3
Emily Brown	Book 4
Daniel Wilson	Book 5

List the title of books written by the author 'Ullman'.

```
SQL> SELECT B.Title
2 FROM BOOKS B
3 JOIN Has_written HW ON B.DocId = HW.DocId
4 JOIN AUTHORS A ON HW.AName = A.Name
5 WHERE A.Name = 'Ullman';

no rows selected
```

List the authors of the books the student 'Smith' has borrowed.

```
SQL> SELECT A.Name
2 FROM AUTHORS A
3 JOIN Has_written HW ON A.Name = HW.AName
4 JOIN BOOKS B ON HW.DocId = B.DocId
5 JOIN Borrows BR ON B.DocId = BR.DocId
6 JOIN STUDENTS S ON BR.StId = S.StId
7 WHERE S.StName = 'Smith';

no rows selected
```

Which books have both keywords 'database' and 'Programming'?

```
SQL> SELECT B.Title
2 FROM BOOKS B
3 JOIN Describes D ON B.DocId = D.DocId
4 WHERE D.Keyword IN ('database', 'Programming')
5 GROUP BY B.Title
6 HAVING COUNT(DISTINCT D.Keyword) = 2;

no rows selected
```

2. Assume Yourself at least 5 Non Correlated and 5 Correlated Sub-Query for the above Relational table and give your results

Non-correlated Subqueries:

1. Find the names of students who have borrowed at least one book:

```
SQL> SELECT StName
2 FROM STUDENTS
3 WHERE StId IN (SELECT DISTINCT StId FROM Borrows);

STNAME
-----
John Doe
Jane Smith
Michael Johnson
Emily Brown
Daniel Wilson
```

2. Find the titles of books published after the year of the oldest book:

```
SQL> SELECT Title
2 FROM BOOKS
3 WHERE Year > (SELECT MIN(Year) FROM BOOKS);

TITLE
-----
Book 2
Book 3
Book 4
Book 5
```

3. Find the names of authors who have written books with the keyword 'Science':

```
SQL> SELECT Name
2 FROM AUTHORS
3 WHERE Name IN (SELECT AName FROM Has_written WHERE DocId IN (SELECT DocId FROM Describes WHERE Keyword = 'Science'));

NAME
-----
Author 1
```

4. Find the titles of books borrowed by students majoring in 'History':

```
SQL> SELECT Title
2 FROM BOOKS
3 WHERE DocId IN (SELECT DocId FROM Borrows WHERE StId IN (SELECT StId FROM STUDENTS WHERE Major = 'History'));

TITLE
-----
Book 3
```

5. Find the names of students who have not borrowed any books:


```
SQL> SELECT StName
  2  FROM STUDENTS
  3  WHERE StId NOT IN (SELECT StId FROM Borrows);

no rows selected
```

Correlated Subqueries:

1. Find the names of students who have borrowed books published by the same publisher as the book 'Database Systems':

```
SQL> SELECT StName
  2  FROM STUDENTS S
  3  WHERE EXISTS (
  4      SELECT 1
  5      FROM Borrows BR
  6      JOIN BOOKS B ON BR.DocId = B.DocId
  7      WHERE BR.StId = S.StId
  8          AND B.Publisher = (SELECT Publisher FROM BOOKS WHERE Title = 'Database Systems')
  9  );

no rows selected
```

2. Find the names of students majoring in the same field as the student with the highest age:

```
SQL> SELECT StName
  2  FROM STUDENTS S1
  3  WHERE S1.Major = (SELECT Major FROM STUDENTS WHERE Age = (SELECT MAX(Age) FROM STUDENTS));

STNAME
-----
Emily Brown
```

3. Find the titles of books written by authors who live in the same city as the student 'John Doe':

```
SQL> SELECT Title
  2  FROM BOOKS
  3  WHERE DocId IN (SELECT DocId FROM Has_written WHERE AName IN (SELECT Name FROM AUTHORS WHERE Address IN (SELECT Address FROM STUDENTS WHERE StName = 'John Doe')));

TITLE
-----
Book 1
Book 2
Book 3
Book 4
Book 5
```

4. Find the names of authors who have written books borrowed by students majoring in 'Computer Science':

```
SQL> SELECT Name
  2   FROM AUTHORS A
  3   WHERE EXISTS (
  4     SELECT 1
  5     FROM Has_written HW
  6     JOIN BOOKS B ON HW.DocId = B.DocId
  7     JOIN Borrows BR ON B.DocId = BR.DocId
  8     JOIN STUDENTS S ON BR.StId = S.StId
  9     WHERE A.Name = HW.AName
 10           AND S.Major = 'Computer Science'
 11  );
```

NAME

Author 1

5. Find the names of students who have borrowed books with the same keyword as their major:

```
SQL> SELECT StName
  2   FROM STUDENTS S
  3   WHERE S.Major IN (SELECT Keyword FROM Describes D WHERE D.DocId IN (SELECT DocId FROM Borrows WHERE StId = S.StId));
```

STNAME

Michael Johnson

PAJAMA PADHAI