National University of Computer and Emerging Sciences

Lab Manual

“Nested Queries”

Database Systems Lab

Department of Computer Science
FAST-NU, Lahore, Pakistan

1. Contents

2. Objective

- The purpose of this manual is to get started with nested queries. This lab will cover all the topics we have covered before. Starting from simple Select-From-Where, Joins, Order by, Aggregate functions & Group by, all of these will be used in combination with the nested queries.

3. Pre-requisites

- Lab manual 3,4,5 which includes:
  - Select-From-Where clause
  - Joins and all its types
  - Order By clause
  - Aggregate functions, Group By, and Having

Task Distribution

Total Time	170 Minutes
Nested Queries	30 Minutes
Exercise	120 Minutes
Evaluation	Last 20 Minutes

4. Nested Queries

For this in-lab manual, use the **InLab6TryThisSchema.sql** script to create database and practice the queries given below.

4.1.1. A subquery (inner query) is a SQL select query nested inside a another select query (outer query)

A subquery may occur in:

- SELECT clause of an outer query
- FROM clause of an outer query
- WHERE clause of an outer query (most commonly used)

4.1.2. A subquery can be nested inside:

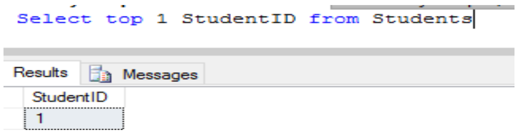
- SELECT statement
- INSERT statement
- UPDATE statement
- DELETE statement
- Another subquery.

4.1.3. There are two types of subqueries.

- Correlated subqueries: where we use some attribute of outer query in inner query, result of the inner query will then change according to the attribute of the outer query.
- Non-correlated subqueries: where no attribute of the outer query is used in an inner query, in this case, inner query always returns the same value.

4.1.4. Scalar Vs. Non-scalar

A select query can return a scalar value or a table. Scalar value means one column and one row
Example: the result of the following query is scalar



```
Select top 1 StudentID from Students
```

StudentID
1

A select query can also return non-scalar value, with more than one column and/or more than one row
Example:

```
Select StudentID from Students
```

Will give non-scalar result.

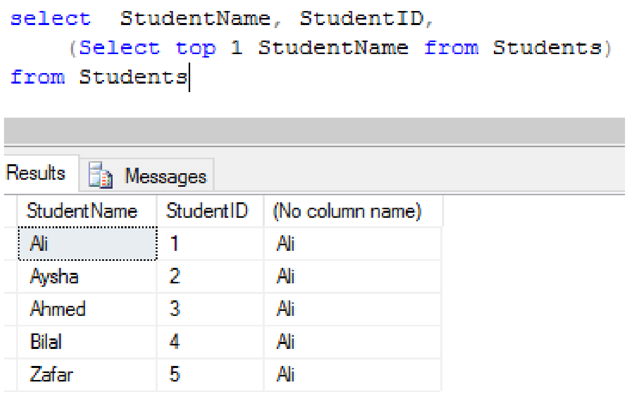
If you are writing a sub query in Select Clause, the inner query should be Scalar
If you are writiing a subquey in From Clause, inner query can be scalar or non-Scalar
If you are writing a subquery in Where Clause, inner query can be scalar or non-Scalar depending on conditon.

Non-Correlated Query:

4.1.5. Non-Correlated Subqueries in SELECT clause

```
SELECT <List of columns of T>
       (select ColumnName from <TableName>)
FROM <tablename> AS T
WHERE <condition>
**inner query should be scalar
```

TRY IT: Non-correlated nested query in Select is not very useful



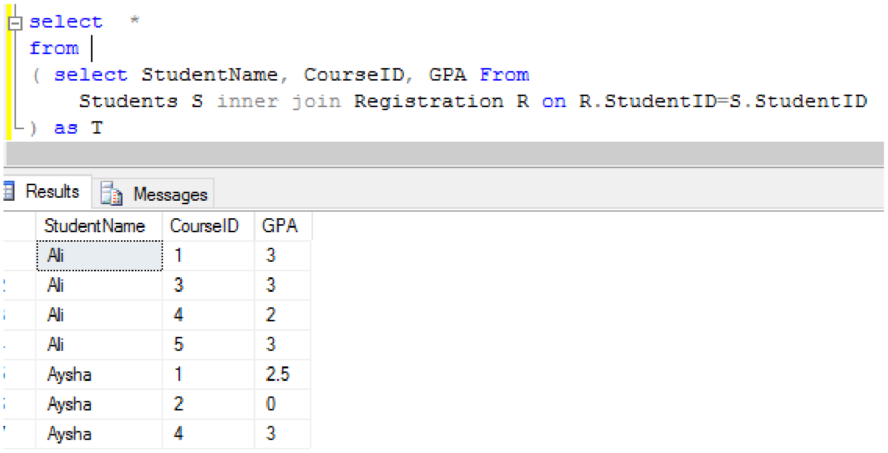
```
select  StudentName, StudentID,
    (Select top 1 StudentName from Students)
from Students
```

StudentName	StudentID	(No column name)
Ali	1	Ali
Aysha	2	Ali
Ahmed	3	Ali
Bilal	4	Ali
Zafar	5	Ali

4.1.6. Non-Correlated Subqueries in From Clause

```
SELECT <List of columns of T ( result of inner query)>
FROM (select ColumnName from <TableName>) as T WHERE <condition>
**inner query can be scalar of non-scalar
***always give alias to inner query in from clause
```

TRY THIS



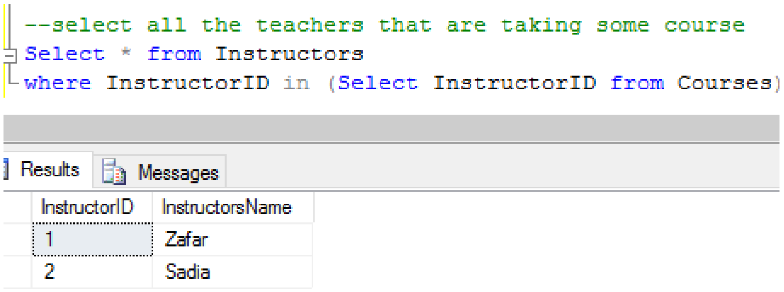
```
select  *
from
( select StudentName, CourseID, GPA From
    Students S inner join Registration R on R.StudentID=S.StudentID
) as T
```

StudentName	CourseID	GPA
Ali	1	3
Ali	3	3
Ali	4	2
Ali	5	3
Aysha	1	2.5
Aysha	2	0
Aysha	4	3

4.1.7. Non-Correlated Subqueries in Where Clause

```
SELECT <List of columns of T >
FROM TableName as T
WHERE <condition> (select ColumnName from <TableName>)
```

TRY THIS



```
--select all the teachers that are taking some course
Select * from Instructors
where InstructorID in (Select InstructorID from Courses)
```

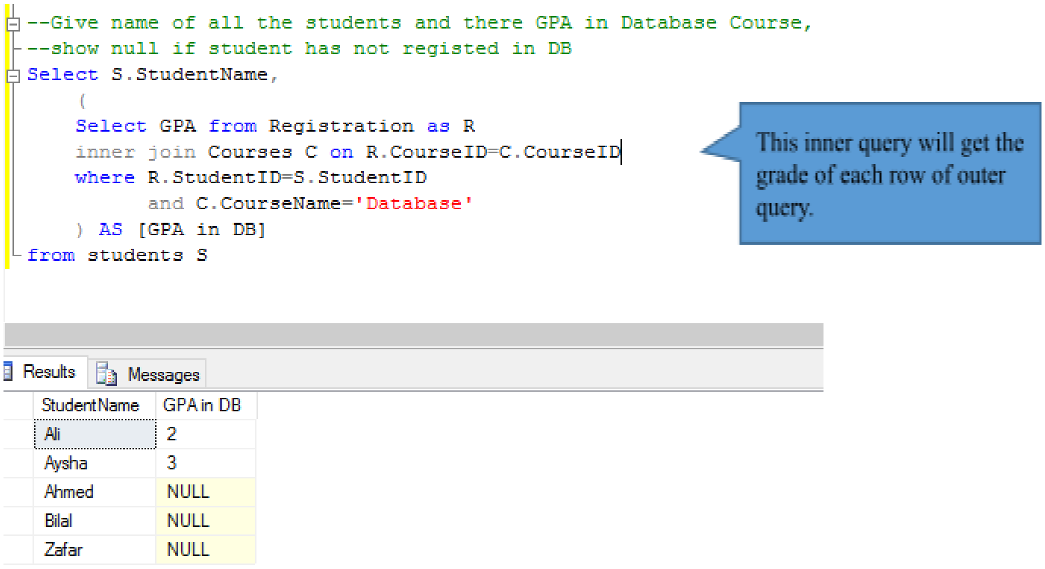
InstructorID	InstructorsName
1	Zafar
2	Sadia

Correlated queries

When inner query is correlated with outer query, then the inner query is executed for each row of outer query.

4.1.8. Correlated Subquery in Select Clause

TRY THIS



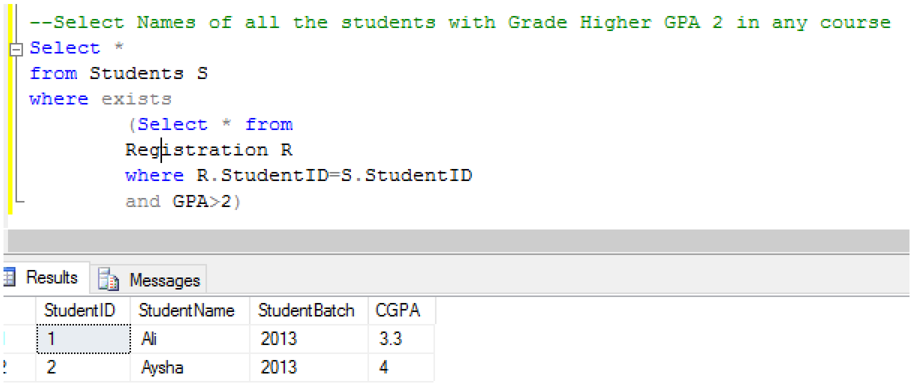
```
--Give name of all the students there GPA in Database Course,
--show null if student has not registed in DB
Select S.StudentName,
    (
    Select GPA from Registration as R
    inner join Courses C on R.CourseID=C.CourseID
    where R.StudentID=S.StudentID
          and C.CourseName='Database'
    ) AS [GPA in DB]
from students S
```

This inner query will get the grade of each row of outer query.

StudentName	GPA in DB
Ali	2
Aysha	3
Ahmed	NULL
Bilal	NULL
Zafar	NULL

4.1.9. Correlated Subquery in Where Clause

TRY THIS



```
--Select Names of all the students with Grade Higher GPA 2 in any course
Select *
from Students S
where exists
         (Select * from
         Registration R
         where R.StudentID=S.StudentID
         and GPA>2)
```

Results | Messages

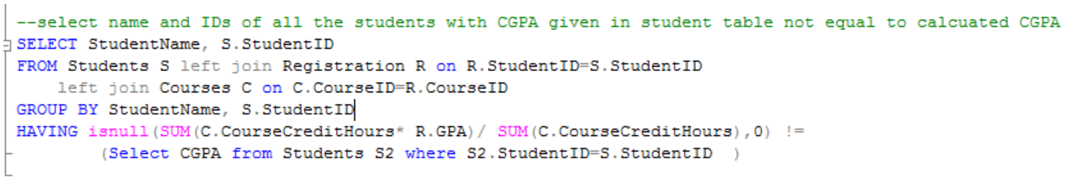
StudentID	StudentName	StudentBatch	CGPA
1	Ali	2013	3.3
2	Aysha	2013	4

** WHAT DOES THE EXIST CLAUSE DO?

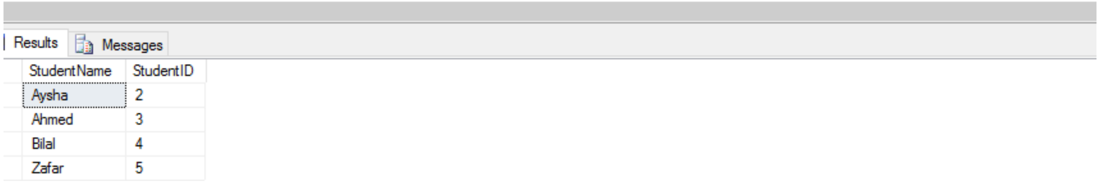
4.1.10. Correlated Subquery in Having Clause

You can also use subquery in having clause (correlated on non-correlated)

TRY THIS



```
--select name and IDs of all the students with CGPA given in student table not equal to calcuated CGPA
SELECT StudentName, S.StudentID
FROM Students S left join Registration R on R.StudentID=S.StudentID
    left join Courses C on C.CourseID=R.CourseID
GROUP BY StudentName, S.StudentID
HAVING isnull(SUM(C.CourseCreditHours* R.GPA)/ SUM(C.CourseCreditHours),0) !=
        (Select CGPA from Students S2 where S2.StudentID=S.StudentID  )
```



Results | Messages

StudentName	StudentID
Aysha	2
Ahmed	3
Bilal	4
Zafar	5

Modify the query given above to, Show name, IDs, Calcuated CGPA and CGPA given in Student table of all the students with CGPA given in student table lesser to calcuated CGPA