Database Management Subqueries

By: Meyer Tanuan (2022), Glenn Paulley (2015), John Mckay (2011)

Nested Queries

- A subquery is a SELECT statement within the WHERE (or HAVING) clause of another SELECT statement:

```
SELECT …
    WHERE column <operator> (SELECT …)
SELECT …
    WHERE column IN (SELECT …)
SELECT …
    WHERE column NOT IN (SELECT …)
```

- These are examples of *nested queries*

CONESTOGA
Connect Life and Learning

Example 1: Two separate queries

Q1: What are the student numbers of those students who have committed Type “A” offences?

```
SELECT studentNumber
FROM StudentOffence
WHERE penaltyCode = 'A'
```

Q2: What are the final marks for students?

```
SELECT studentNumber, finalMark
FROM CourseStudent
```

Combine these with a nested query.

Example 1: (a) Using a join

What are the course grades for students who have committed Type “A” academic offences?

```
SELECT cs.studentNumber, cs.finalMark
FROM CourseStudent cs, StudentOffence so
WHERE cs.studentNumber = so.studentNumber
  AND so.penaltyCode = 'A'
```

CONESTOGA
Connect Life and Learning

Example 1: (b) Using a subquery

```
SELECT studentNumber, finalMark
FROM CourseStudent
WHERE studentNumber =
   (SELECT studentNumber
    FROM StudentOffence
    WHERE penaltyCode = 'A')
```

Semantics:

- The search condition (WHERE clause) is evaluated for each row in the CourseStudent table
- If the studentNumber column matches a value (i.e., studentNumber) computed by the subquery, then the WHERE condition evaluates to TRUE

Subqueries with relational operators

- If the subquery returns *a single value* ***and*** *a single row*, a relational operator (=, <>, <, <=, >, >=) can be used:

```
SELECT … WHERE column = (SELECT …)

SELECT … WHERE column <> (SELECT …)

SELECT … WHERE column < (SELECT …)

Etc.
```

- If the subquery returns a ***set of values***, use a quantifying operator (IN, SOME, ANY, ALL)

Example 2: Subquery and <> operator

- What are the course grades for students who have not committed Type "A" academic offences?

```
SELECT studentNumber, finalMark
FROM CourseStudent
WHERE studentNumber <>
  (SELECT studentNumber
   FROM StudentOffence
   WHERE penaltyCode = 'A')
```

- Relational operator works if the subquery returns a single value

Example 3: Subquery and run-time error

- What are the course grades for students who have a negative balance?

```
SELECT studentNumber, finalMark
FROM CourseStudent
WHERE studentNumber =
  (SELECT number
   FROM Student
   WHERE balance < 0)
```

- Result is a run-time error: Subquery returned more than 1 value.

Subqueries with IN and NOT IN

- If the subquery could return multiple rows, a **quantified** subquery predicate must be used – for example, IN or NOT IN:

```
SELECT …
    WHERE column IN (SELECT …)
SELECT …
    WHERE column NOT IN (SELECT …)
```

- Using ‘IN’ is equivalent to the syntax ‘= ANY()’ or ‘= SOME()’
- IN subqueries are very commonly used in practice

CONESTOGA
Connect Life and Learning

Example 4: Subqueries with IN and NOT IN

4a: Correction to original Example 3 nested query (using IN)

```
SELECT studentNumber, finalMark
FROM CourseStudent
WHERE studentNumber IN
  (SELECT number FROM Student WHERE balance < 0)
```

4b: And the negated version (using NOT IN)

```
SELECT studentNumber, finalMark
FROM CourseStudent
WHERE studentNumber NOT IN
  (SELECT number FROM Student WHERE balance < 0)
```

CONESTOGA
Connect Life and Learning

Example 5: Subquery with IN

Q. What are the names of the employees who work in the School of Business?

```
SELECT lastName, firstName
FROM Person
WHERE number IN
    (SELECT number
     FROM Employee
     WHERE schoolCode = 'BUS')
```

CONESTOGA
Connect Life and Learning

Example 6: Subqueries with = ANY() and = SOME()

6a (Variant 1): = ANY()

```
SELECT lastName, firstName
FROM Person
WHERE number = ANY
    (SELECT number FROM Employee WHERE schoolCode = 'BUS')
```

6b (Variant 2): = SOME()

```
SELECT lastName, firstName
FROM Person
WHERE number = SOME
    (SELECT number FROM Employee WHERE schoolCode = 'BUS')
```

CONESTOGA
Connect Life and Learning

Example 7: Subqueries with EXISTS

Q. What are the names of the employees who work in the School of Business?

```
SELECT lastName, firstName
FROM Person p
WHERE EXISTS
   (SELECT *
    FROM Employee e
    WHERE p.number = e.number
          AND schoolCode = 'BUS')
```

Note: p.number is a correlation reference to the column in the outer block.

EXISTS

- An EXISTS predicate evaluates a subquery for the existence of any rows – SQL Server will halt the execution of the subquery once the first row is found and the predicate evaluates to TRUE.
    - The expression in the SELECT list of the subquery doesn't matter, it is typically an asterisk (*) but it could be, for example, a literal constant.

```
SELECT lastname, firstname
FROM Person p
WHERE EXISTS
   (SELECT *
    FROM Employee e
    WHERE location = '4A17' AND p.number = e.number);
```

NOT EXISTS

- Similarly, a NOT EXISTS predicate is evaluated for the existence of any rows – SQL Server will halt the execution of the subquery once the first row is found
  - If a row is found, the predicate evaluates to FALSE

```
SELECT lastname, firstname
FROM Person p
WHERE NOT EXISTS
   (SELECT *
    FROM Employee e
    WHERE location = '4A17' AND p.number = e.number);
```

CONESTOGA
Connect Life and Learning

Example 8: Columns with comparable data type

- Column names don’t have to match, as long as the data type is comparable:

```
SELECT studentNumber, finalMark
FROM CourseStudent
WHERE studentNumber =
    (SELECT TOP 1 number
     FROM Student
     ORDER BY balance DESC)
```

- Note the use of TOP 1 for subquery to return a single value

Example 9a: Subquery in a SELECT List (1:1)

- Subquery is (logically) recomputed for each row in the output
- As with nested queries, a scalar subquery in a query’s SELECT list can reference expressions from its parent block
  - Such outer references (e.g., e.number) are treated as a constant value for that subquery’s execution

```
SELECT e.number, e.schoolCode
    , (SELECT p.firstName+' '+p.lastName
      FROM Person p
      WHERE p.number=e.number) AS EmployeeName
FROM Employee e
WHERE e.location = '4A17'
```

CONESTOGA
Connect Life and Learning

How to code a subquery in the SELECT list

To code a subquery properly, examine the relationships between tables

- The goal is to ensure the subquery produces ***one*** result row for each row of the master table

Q. Get the names of all people who paid more than $1,000 at a time and the amount they paid. Use the Person and Payment tables.

A. The relationship between the Person and Payment tables is one-to-many (1:M)

- i.e., One person can make multiple payments.

Example 9b: Subquery in a SELECT List (1:M)

- A subquery in a SELECT list must return a single value (scalar)
  - So we must start with the Payment table.
- If we start with the Person table
  - The query will fail because the subquery (Payment table) may return multiple values per person

```
SELECT p.FirstName, p.LastName
      , (SELECT amount FROM Payment WHERE studentNumber = p.Number)
FROM Person p
WHERE p.number IN
  (SELECT studentNumber FROM Payment WHERE amount > 1000);
```

CONESTOGA
Connect Life and Learning

Example 9c: Subquery in a SELECT List (M:1)

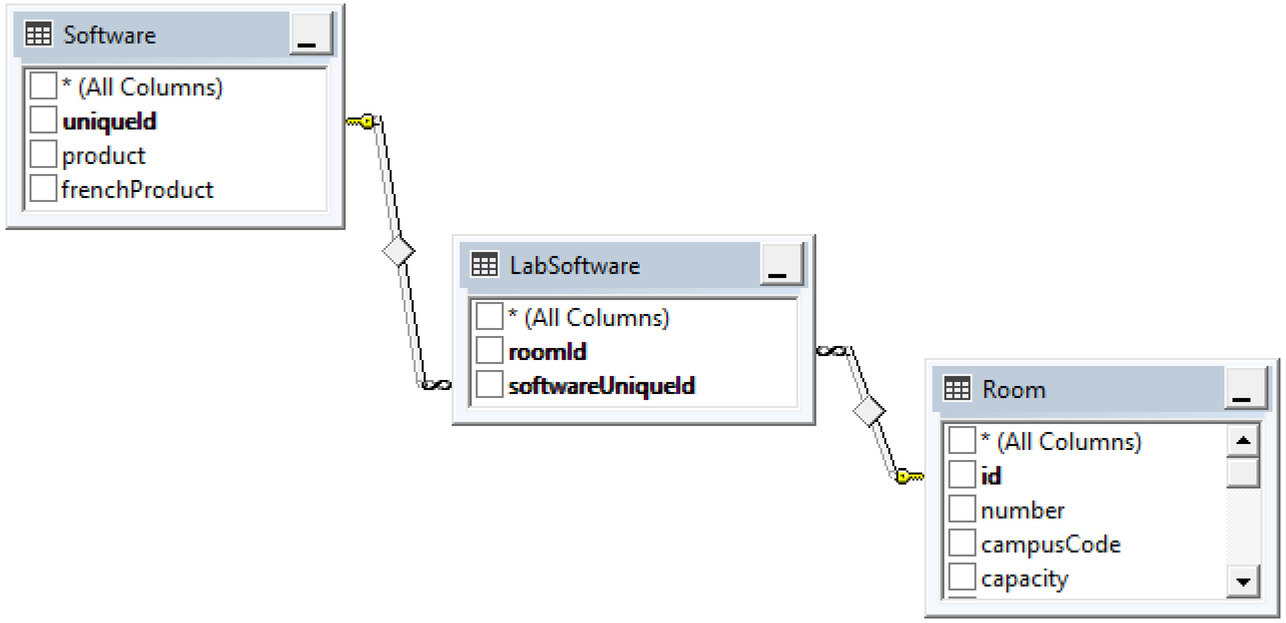
- If we start with the Payment table, the query works because the subquery returns a single value (i.e., one person per payment)

```
SELECT
     (SELECT firstName
    FROM Person
    WHERE number = pay.studentNumber)
  ,(SELECT lastName
      FROM Person
    WHERE number = pay.studentNumber)
  , pay.amount
FROM Payment pay
WHERE pay.amount > 1000;
```

Complex example for a subqueries in the SELECT list

Q. Find all software installed in all labs of the Doon campus using Software, LabSoftware, and Room.

A. Because LabSoftware is a join table and can have many Software rows connected to many Room rows, we need to start with that table.



Example 9d: Complex example (1:M,M:1)

```
SELECT
    (SELECT product
     FROM Software
     WHERE uniqueId = ls.softwareUniqueId)
   ,(SELECT number
     FROM Room
     WHERE id = ls.roomId)
FROM LabSoftware ls
WHERE ls.roomId IN
       (SELECT id
        FROM Room
     WHERE campusCode = 'D' and isLab = 1);
```

Conestoga
Connect Life and Learning

Practice: JOIN vs Subquery

Q. Find all courses which are offered in the CPA program. Show course names, campus, and semester for which the course is offered.

- Use these SIS tables: Program, Course, ProgramCourse, and Campus.

```
Campus     Course                                                      Semester
---------- ----------------------------------------------------------- -----------
Doon       Technology Infrastructure: Fundamentals                        1
Doon       Effective Technical Comunication I                             1
Doon       Mathematics for IT I                                           1
Doon       Programming: Fundamentals                                      1
. . .
```

Before you look at the sample solution in subquery.sql:

- Try using JOIN
- Try using Subquery
- Both should have the same result

CONESTOGA
Connect Life and Learning