جامعة طرابلس
كلية تقنية المعلومات

Advanced Databases قواعد البيانات المتقدمة

IT IS–325

د. عبدالسلام منصور الشريف

a.abdoessalam@uot.edu.ly

المحاضرة الخامسة – استرجاع البيانات III

Retrieving Data III

Contents

- Retrieving Data
  - SELECT
  - SUBQUERIES

Select Statement - General Structure

SELECT ..... FROM ..... WHERE .....

SELECT [ALL / DISTINCT] *expr1* [AS *col1*], *expr2* [AS *col2*] ;

FROM *tablename* WHERE *condition*

Sub-Queries

- A subquery is a query that is nested inside a SELECT, INSERT, UPDATE, or DELETE statement, or inside another subquery.
- A subquery is also called an ***inner query*** or inner select, while the statement containing a subquery is also called an ***outer query*** or outer select.
- A subquery nested in the outer SELECT statement has the following components:
  - A regular SELECT query including the regular select list components.
  - A regular FROM clause including one or more table or view names.
  - An optional WHERE clause.
  - An optional GROUP BY clause.
  - An optional HAVING clause.

Sub-Queries

- A subquery can be used as an expression in the SELECT.
- A subquery can be nested inside the WHERE or HAVING clause of an outer SELECT, INSERT, UPDATE, or DELETE statement, or inside another subquery.
- Up to 32 levels of nesting is possible, although the limit varies based on available memory and the complexity of other expressions in the query.
- If a table appears only in a subquery and not in the outer query, then columns from that table cannot be included in the output (the select list of the outer query).

Sub-Queries

- Statements that include a subquery usually take one of these formats:
  - WHERE expression \[NOT] IN (subquery)
  - WHERE expression comparison_operator \[ANY | ALL] (subquery)
  - WHERE \[NOT] EXISTS (subquery)
- There are three basic types of subqueries. Those that:
  - Operate on lists introduced with IN, or those that a comparison operator modified by ANY or ALL.
  - Are introduced with an unmodified comparison operator and must return a single value.
  - Are existence tests introduced with EXISTS.

Sub-Queries

- A subquery is subject to the following restrictions:
  - The select list of a subquery introduced with a comparison operator can include only one expression or column name (except that EXISTS and IN operate on SELECT * or a list, respectively).
  - If the WHERE clause of an outer query includes a column name, it must be join-compatible with the column in the subquery select list.
  - Because they must return a single value, subqueries introduced by an unmodified comparison operator (one not followed by the keyword ANY or ALL) cannot include GROUP BY and HAVING clauses.
  - The DISTINCT keyword cannot be used with subqueries that include `GROUP BY.

Sub-Queries – in SELECT

- List the student with their highest marks

```
SELECT Id,Name,
       (SELECT MAX(Mark)  FROM dbo.Semesters sem
           WHERE stu.id = sem.StudentId
       ) Maxmark
FROM dbo.Students stu
```

Id	Name	Maxmark
9223082	Salem Ahmed Najem	NULL
9223312	Sabria Altaher Omar	NULL
9223333	Ahmed Othman kalaf	99
9223334	Samir Alhadi Qabaj	NULL
9223338	Fatim Hussni Mahmood	99.5
9223363	Ali Ahmed Salem	NULL
9223373	Alfirjani Adel Muftah	NULL
9223381	Zahra Fouad Aljosh	99.5
9223382	Asma Altaher Omar	NULL
9225582	Sumia Adel Rajab	NULL
9243282	UmKulthom Ahmed Kamel	NULL

Sub-Queries – in FROM

- Find The max mark by semester type ordered by maxmark

```
SELECT SemesterType, Maxmark FROM
    (SELECT SemesterType, MAX(Mark) Maxmark
         FROM dbo.Semesters GROUP BY SemesterType
    ) sem
ORDER BY Maxmark DESC
```

SemesterType	Maxmark
Spring	99.5
Fall	95

Sub-Queries – in FROM

- The max of average mark

```
SELECT MAX(Avgmark) Maxavg
 FROM (SELECT StudentId, AVG(Mark) Avgmark
        FROM dbo.Semesters GROUP BY StudentId) sem
```

StudentId	Avgmark
9223333	66.545454
9223338	71.681818
9223381	90.590909

Maxavg
90.59091

Sub-Queries – in WHERE

- Students who have marks above the average

```
SELECT DISTINCT StudentId
  FROM dbo.Semesters
  WHERE Mark >
     (SELECT AVG(Mark) FROM dbo.Semesters)
```

StudentId
9223333
9223338
9223381

Sub-Queries – in WHERE. Cont.

- Find information about student with marks above the average

```
SELECT stu.Id,stu.Name
  FROM dbo.Students stu
  WHERE Id in
    (SELECT DISTINCT StudentId FROM dbo.Semesters
       WHERE Mark > (SELECT AVG(Mark) FROM dbo.Semesters))
```

Id	Name
9223333	Ahmed Othman kalaf
9223338	Fatim Hussni Mahmood
9223381	Zahra Fouad Aljosh

Sub-Queries –ANY, SOME

- Compares a scalar value with a single-column set of values. SOME and ANY are equivalent.
- SOME requires the *scalar_expression* to compare positively to at least one value returned by the subquery.

scalar_expression { = | < > | ! = | > | > = | ! > | < | < = | ! < } { SOME | ANY } (subquery)

- SOME or ANY returns TRUE when the comparison specified is TRUE for any pair (scalar_expression,x) where x is a value in the single-column set; otherwise, returns FALSE.
- The =ANY operator is equivalent to IN.
- The <>ANY operator, not equivalent to NOT IN:
  - <>ANY means not = a, or not = b, or not = c
  - NOT IN means not = a, and not = b, and not = c
  - <>ALL means the same as NOT IN

Sub-Queries – ANY, SOME

- List courses whose marks are greater than or equal to the maximum mark of any classfication

```
SELECT DISTINCT CourseId FROM [dbo].[Semesters] sem
WHERE sem.Mark >= ANY (
        SELECT MAX(sem.Mark) FROM [dbo].[Semesters] sem
        INNER JOIN [dbo].[Courses] cou
        ON sem.CourseId = cou.Id
        GROUP BY cou.Classification)
```

CourseId
ITAR111
ITEL111
ITGS113
ITGS122
ITGS124
ITGS213
ITGS219
ITGS223
ITGS302
ITNT311
ITNT411

Sub-Queries –ALL

- Compares a scalar value with all column set of values.

scalar_expression { = | < > | ! = | > | > = | ! > | < | < = | ! < } ALL (subquery)

- ALL requires the scalar_expression to compare positively to all of the values returned by the subquery.

Sub-Queries – EXISTS

- When a subquery is introduced with the keyword EXISTS, the subquery functions as an existence test.
- The WHERE clause of the outer query tests whether the rows that are returned by the subquery exist.
- The subquery does not actually produce any data; it returns a value of TRUE or FALSE.

- `WHERE [NOT] EXISTS` (subquery)

- Notice that subqueries that are introduced with EXISTS are different from other subqueries in the following ways:
  - The keyword EXISTS is not preceded by a column name, constant, or other expression.
  - The select list of a subquery introduced by EXISTS almost always consists of an asterisk (*). There is no reason to list column names because you are just testing whether rows that meet the conditions specified in the subquery exist.