# PROGRAMACIÓN CON VISUAL STUDIO .NET

JOSÉ PABLO HERNÁNDEZ

# PROGRAMACIÓN CON VISUAL STUDIO .NET

2.6.3. Subconsultas - 2.

JOSÉ PABLO HERNÁNDEZ

# LAS SUBCONSULTAS

Definiciones

Referencias externas a otros servidores/bases de datos

Anidar subconsultas

Subconsulta en la lista de selección. Uso del operador IN

Subconsulta en las cláusulas WHERE y HAVING

### **EXIST y NOT EXIST**

La condición EXISTS se utiliza en combinación con una subconsulta. Devuelve VERDADERO siempre que la subconsulta devuelve uno o más valores.

En su forma más simple, la sintaxis de la condición EXISTS es

WHERE EXISTS (sub query)

### **EXIST y NOT EXIST**

Supongamos que debemos devolver todos los pedidos de ventas escritos por personal de ventas con sales year to date superiores a tres millones de dólares. Para hacerlo, podemos usar la cláusula EXISTS como se muestra en este ejemplo:

SELECT SalesOrderID,

RevisionNumber,

OrderDate

FROM Sales.SalesOrderHeader

WHERE EXISTS (SELECT 1

FROM sales.SalesPerson

WHERE SalesYTD > 3000000

AND SalesOrderHeader.SalesPersonID = Sales.SalesPerson.BusinessEntityID)

### **EXIST y NOT EXIST**

NOT EXISTS devuelve VERDADERO si se devuelven cero filas. Entonces, si queremos encontrar todos los pedidos de venta que fueron escritos por vendedores que no tuvieron 3,000,000 en ventas year-to-date, podemos usar la siguiente consulta:

SELECT SalesOrderID,

RevisionNumber,

OrderDate

FROM Sales.SalesOrderHeader

WHERE NOT EXISTS (SELECT 1

FROM sales.SalesPerson

WHERE SalesYTD > 3000000

AND SalesOrderHeader.SalesPersonID = Sales.SalesPerson.BusinessEntityID)

### EXIST y NOT EXIST. ¿Qué pasa con NULL?

Si una subconsulta devuelve un valor NULL, la instrucción EXISTS se resuelve en TRUE:

SELECT SalesOrderID,

RevisionNumber,

OrderDate

FROM Sales.SalesOrderHeader

WHERE EXISTS (SELECT NULL)

### IN y NOT IN

Cuando se usa con subconsultas, la lista de valores se reemplaza con una subconsulta.

SELECT SalesOrderID,

OrderDate,

AccountNumber,

CustomerID,

SalesPersonID,

TotalDue

FROM Sales.SalesOrderHeader

WHERE SalesPersonID IN (279, 286, 289)

	SalesOrderID	OrderDate	Account Number	CustomerID	SalesPersonID	TotalDue
1	43659	2005-07-01 00:00:00.000	10-4020-000676	29825	279	23153.2339
2	43660	2005-07-01 00:00:00.000	10-4020-000117	29672	279	1457.3288
3	43681	2005-07-01 00:00:00.000	10-4020-000423	29661	279	15524.0686
4	43684	2005-07-01 00:00:00.000	10-4020-000549	29912	279	6301.6258
5	43685	2005-07-01 00:00:00.000	10-4020-000458	30084	279	3082.0191
6	43694	2005-07-01 00:00:00.000	10-4020-000315	29549	279	23242.1865
7	43695	2005-07-01 00:00:00 000	10-4020-000027	29958	279	44344.8265

### IN y NOT IN

Cuando se usa con subconsultas, la lista de valores se reemplaza con una subconsulta.

SELECT SalesOrderID,

OrderDate,

AccountNumber,

CustomerID,

SalesPersonID,

TotalDue

	SalesOrderID	OrderDate	Account Number	CustomerID	SalesPersonID	TotalDue
1	43659	2005-07-01 00:00:00.000	10-4020-000676	29825	279	23153.2339
2	43660	2005-07-01 00:00:00.000	10-4020-000117	29672	279	1457.3288
3	43681	2005-07-01 00:00:00.000	10-4020-000423	29661	279	15524.0686
4	43684	2005-07-01 00:00:00.000	10-4020-000549	29912	279	6301.6258
5	43685	2005-07-01 00:00:00.000	10-4020-000458	30084	279	3082.0191
6	43694	2005-07-01 00:00:00.000	10-4020-000315	29549	279	23242.1865
7	43695	2005-07-01 00:00:00:00 000	10-4020-000027	29958	279	44344.8265

FROM Sales.SalesOrderHeader

WHERE SalesPersonID IN (SELECT BusinessEntityID

FROM Sales.SalesPerson

WHERE Bonus > 5000)

### IN y NOT IN

Al igual que con otras consultas, puede crear una subconsulta correlacionada para usar con la cláusula IN.

Aquí está la misma consulta que usamos con la cláusula EXIST.

SELECT SalesOrderID,

RevisionNumber,

OrderDate

FROM Sales.SalesOrderHeader

WHERE SalesPersonID IN (SELECT SalesPerson.BusinessEntityID

FROM sales.SalesPerson

WHERE SalesYTD > 3000000

AND SalesOrderHeader.SalesPersonID

= Sales.SalesPerson.BusinessEntityID)

### IN y NOT IN

Como IN devuelve VERDADERO si el valor comprobado se encuentra en la lista de comparación, NOT IN devuelve VERDADERO si no se encuentra el valor comprobado. Tomando la misma consulta anterior, podemos encontrar todos los pedidos de venta que fueron escritos por personal de ventas que no anotaron 3,000,000 en ventas del año hasta la fecha, podemos escribir la siguiente consulta:

SELECT SalesOrderID,

RevisionNumber,

OrderDate

FROM Sales.SalesOrderHeader

WHERE SalesPersonID NOT IN (SELECT SalesPerson.BusinessEntityID

FROM sales.SalesPerson

WHERE SalesYTD > 3000000

AND SalesOrderHeader.SalesPersonID

= Sales.SalesPerson.BusinessEntityID)

### Modificadores de comparación

Se pueden utilizar los operadores de comparación, como mayor que, menor que, igual y no igual.

En lugar de usar >, que solo tiene sentido cuando se compara con un solo valor ( escalar ), puede usar > ANY o > ALL para comparar un valor de columna con los resultados de una lista devueltos por la sub consulta.

Modificadores de comparación. ANY.

El operador de comparación > ANY significa mayor que uno o más elementos en la lista.

Esto es lo mismo que decir que es mayor que el valor MIN de la lista.

Sales > ANY (1000, 2000, 2500)

Es equivalente a

Sales > MIN(1000, 2000, 2500)

### Modificadores de comparación. ANY.

Vamos a buscar todos los productos que pueden tener un alto nivel de stock de seguridad. Para hacerlo, buscaremos todos los productos que tengan un SafetyStockLevel que sea mayor que el promedio de SafetyStockLevel para varios DaysToManufacture.

SELECT ProductID,

Name,

SafetyStockLevel,

DaysToManufacture

FROM Production.Product

WHERE SafetyStockLevel > ANY (SELECT AVG(SafetyStockLevel)

FROM Production.Product

GROUP BY DaysToManufacture)

### Modificadores de comparación. ANY.

Vamos a buscar todos los productos que pueden tener un alto nivel de stock de seguridad. Para hacerlo, buscaremos todos los productos que tengan un SafetyStockLevel que sea mayor que el promedio de SafetyStockLevel para varios DaysToManufacture.

SELECT ProductID,

Name,

SafetyStockLevel,

DaysToManufacture

FROM Production.Product

WHERE SafetyStockLevel > ANY (SELECT AVG(SafetyStockLevel)

FROM Production.Product

GROUP BY DaysToManufacture)

Modificadores de comparación. ALL.

El modificador > **ALL** funciona de manera similar, excepto que devuelve la fila externa si su valor de comparación es mayor que cada valor devuelto por la consulta interna.

```
SELECT p.BusinessEntityID,
    p.FirstName,
    p.LastName,
    s.Bonus,
    s.SalesYTD
FROM Person.Person AS p
    INNER JOIN Sales. Sales Person AS s
    ON p.BusinessEntityID = s.BusinessEntityID
WHERE s.Bonus > ALL (SELECT Bonus
                        FROM Sales.SalesPerson
                        WHERE Sales. Sales Person. Sales YT < 1000000)
```

Modificadores de comparación. Resumen.

Example Comparison	Result	Description						
	ANY – Satisfy the condition with one or more members of the list							
X > ANY (1,2,3)	True if X > 1	This is equivalent to X > MIN(1,2,3)						
X >= ANY (1,2,3)	True if X >= 1							
X < ANY (1,2,3)	True if X < 3	This is equivalent to X < MAX(1,2,3)						
X <= ANY (1,2,3)	True if X <= 3							
X <> ANY (1,2,3)	True if X <> 1 OR X <> 2 OR X <> 3	This will always return a true if there is more than one value in the list.						
X = ANY(1,2,3)	True if X = 1 OR X = 2 OR X = 3	This is equivalent to X IN (1,2,3)						
ALL – Satisfy condition with every member of the list								
X > ALL (1,2,3)	True if X > 3	This is equivalent to X > MAX(1,2,3)						
X >= ALL (1,2,3)	True if X >= 3							
X < ALL (1,2,3)	True if X < 1	This is equivalent to X < MIN(1,2,3)						
X <= ALL (1,2,3)	True if X <= 1							
X <> ALL (1,2,3)	True if X <> 1 AND X <> 2 AND X<>	This is equivalent to X NOT IN (1,2,3)						
X = ALL (1,2,3)	True if X = 1 AND X = 2 AND X = 3	Returns FALSE, X can't be more than one value at the same time.						

# SUBCONSULTAS EN CLÁUSULAS WHERE

## **EJERCICIOS**