



Universidad Tecnológica de Panamá
Facultad de Ingeniería de Sistemas Computacionales
BASE DE DATOS I
Laboratorio N°7



Facilitador: Víctor A. Fuentes T.

Estudiante: Johel Heraclio Batista Cárdenas

Cédula: 8-914-587

Grupo: 1IF-131

A. TÍTULO DE LA EXPERIENCIA:

Laboratorio No.7. Consultas multi tablas y Subconsultas anidadas.

B. TEMAS:

- a. Combinaciones Join
- b. Subconsultas con select anidados

C. OBJETIVO(S):

- Introducir al estudiante en el uso de técnicas avanzadas de consulta para acceder a los datos de una base de datos cuando se encuentran en tablas diferentes y cuando son productos de resultados previos.

D. METODOLOGÍA:

Para presentar el informe de los resultados obtenidos, haga captura de pantalla desde el SQL Server mostrando la instrucción y el resultado de la consulta generada por la misma.

Copie estas capturas de pantalla en la sección G (RESULTADOS) de esta guía, en el número mostrado en la sección E (PROCEDIMIENTO). Corte y sólo presente el área de trabajo donde aparece la instrucción y el resultado obtenido, no incluya el explorador de objetos, el menú de opciones ni la barra de herramientas estándar en su respuesta final.

E. PROCEDIMIENTO O ENUNCIADO DE LA EXPERIENCIA:

Todo lo indicado en color verde corresponde a acciones que usted deberá ejecutar. La primera sección es una serie de ejemplos explicativos que debe realizar para familiarizarse con el uso de las funciones y sintaxis que se tratan en este tema.

Para trabajar adecuadamente con algunas columnas, se hace necesario consultar información que residen en diferentes tablas. Hasta el momento solo se ha trabajado con una sola tabla, sin embargo, es posible emplear más de una tabla para satisfacer la consulta que se necesita. Para

estos casos, uno de los elementos en que se basa el funcionamiento de algunas formas de uniones en los sistemas de gestión de base de datos, es la teoría de conjuntos. A continuación, se introducen algunos de las formas de uniones más usadas en SQL Server.

Combinaciones Inner Join

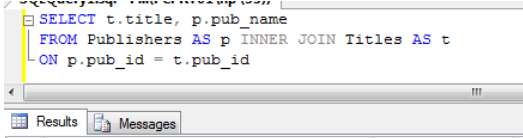
Una combinación **inner** es una combinación en la cual los valores de las columnas a ser combinadas son comparados a través de un operador de comparación.

Ejemplo 1

```
SELECT t.title, p.pub_name
FROM Publishers AS p INNER JOIN Titles AS t
ON p.pub_id = t.pub_id
```

ó

```
SELECT t.title, p.pub_name
FROM Publishers p INNER JOIN Titles t
ON p.pub_id = t.pub_id
```



	title	pub_name
1	The Busy Executive's Database Guide	Algodata Infosystems
2	Cooking with Computers: Sumptitious Balance Sheets	Algodata Infosystems
3	You Can Combat Computer Stress!	New Moon Books
4	Straight Talk About Computers	Algodata Infosystems
5	Silicon Valley Gastronomic Treats	Binnet & Hardley
6	The Gourmet Microwave	Binnet & Hardley
7	The Psychology of Computer Cooking	Binnet & Hardley
8	But Is It User Friendly?	Algodata Infosystems
9	Secrets of Silicon Valley	Algodata Infosystems
10	Net Etiquette	Algodata Infosystems
11	Computer Phobic AND Non-Phobic Individuals: Beha...	Binnet & Hardley
12	Is Anger the Enemy?	New Moon Books
13	Life Without Fear	New Moon Books
14	Prolonged Data Deprivation: Four Case Studies	New Moon Books
15	Emotional Security: A New Algorithm	New Moon Books
16	Onions, Leeks, and Garlic: Cooking Secrets of the M...	Binnet & Hardley
17	Fifty Years in Buckingham Palace Kitchens	Binnet & Hardley
18	Sushi, Anyone?	Binnet & Hardley

El comando SELECT recupera el nombre del libro (title) de la tabla titles y el nombre de la editora (pub_name) de la tabla publishers cuando coinciden los códigos pub_id en ambas tablas (usando PK y FK).

Hay que tener en cuenta que, si el nombre de una columna existe en más de una de las tablas indicadas en la cláusula FROM, hay que poner, obligatoriamente, el nombre o alias de la tabla de la que queremos obtener dicho valor. En caso contrario nos dará un error de ejecución, indicando que hay un nombre ambiguo.

Por ello otorgamos un alias a cada tabla (t y p respectivamente) para no tener que escribir su nombre completo cada vez que necesitamos usarlas.

Es una buena práctica acostumbrarse a colocar de qué tabla es cada campo con los alias.

Otra forma de **hacer lo mismo** es indicando las tablas en el from separadas por comas y colocar en el where los criterios de coincidencia que debemos validar (PK y FK)

```
SELECT t.title, p.pub_name

FROM Publishers p, Titles t

where p.pub_id = t.pub_id
```

En este caso podemos **usar el where** en lugar del **on** para establecer el vínculo entre PK y FK porque no teníamos ninguna condición adicional que validar.

```
SELECT t.title, p.pub_name
FROM Publishers p, Titles AS t
where p.pub_id = t.pub_id
```

	title	pub_name
1	The Busy Executive's Database Guide	Algodata Infosystems
2	Cooking with Computers: Suneptitious Balance Sheets	Algodata Infosystems
3	You Can Combat Computer Stress!	New Moon Books
4	Straight Talk About Computers	Algodata Infosystems
5	Silicon Valley Gastronomic Treats	Binnet & Hardley
6	The Gourmet Microwave	Binnet & Hardley
7	The Psychology of Computer Cooking	Binnet & Hardley
8	But Is It User Friendly?	Algodata Infosystems
9	Secrets of Silicon Valley	Algodata Infosystems
10	Net Etiquette	Algodata Infosystems
11	Computer Phobic AND Non-Phobic Individuals: Beha...	Binnet & Hardley
12	Is Anger the Enemy?	New Moon Books
13	Life Without Fear	New Moon Books
14	Prolonged Data Deprivation: Four Case Studies	New Moon Books
15	Emotional Security: A New Algorithm	New Moon Books
16	Onions, Leeks, and Garlic: Cooking Secrets of the M...	Binnet & Hardley
17	Fifty Years in Buckingham Palace Kitchens	Binnet & Hardley
18	Sushi, Anyone?	Binnet & Hardley

```
SELECT t.title, p.pub_name
FROM Publishers p join Titles t
on p.pub_id = t.pub_id
where pub_name = 'Algodata Infosystems'
```

	title	pub_name
1	The Busy Executive's Database Guide	Algodata Infosystems
2	Cooking with Computers: Suneptitious Balance Sh...	Algodata Infosystems
3	Straight Talk About Computers	Algodata Infosystems
4	But Is It User Friendly?	Algodata Infosystems
5	Secrets of Silicon Valley	Algodata Infosystems
6	Net Etiquette	Algodata Infosystems

Si particularmente sólo quisiéramos los títulos de libros de Algodata System, tendríamos que usar el **join** para definir qué tablas está usando, el **on** para unir las llaves y el **where** para la condición que desea.

Combinaciones Outer

SQL Server soporta tres tipos de combinaciones outer: izquierda (left), derecha (right), y completa (full).

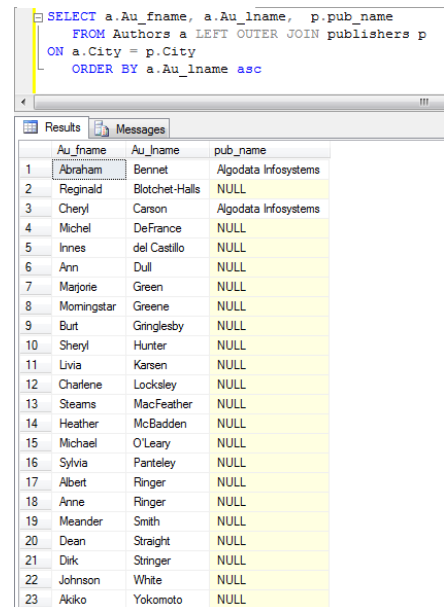
Todas las filas recuperadas desde la tabla izquierda son referenciadas con una combinación outer izquierda, y todas las filas de la tabla derecha son referenciadas en una combinación outer derecha. Todas las filas de ambas tablas son retornadas en una combinación outer completa.

Usar combinaciones outer izquierdas

En el siguiente comando SELECT, se usa una combinación outer izquierda para recuperar los apellidos y nombres de los autores y (cuando corresponda) los nombres de cualquier editor que se encuentre en la misma ciudad que el autor.

```
SELECT a.Au_fname, a.Au_lname, p.pub_name
FROM Authors a LEFT OUTER JOIN publishers p
ON a.City = p.City
ORDER BY a.au_lname asc
```

El conjunto de resultados para esta consulta listará el nombre de todos los autores de la tabla Autores. El conjunto de resultados incluirá sólo aquellos editores que se encuentren en las mismas ciudades de los autores. Si un editor no se encuentra en la ciudad del autor, un valor nulo es retornado para la columna



The screenshot shows a SQL query window with the following query:

```
SELECT a.Au_fname, a.Au_lname, p.pub_name
FROM Authors a LEFT OUTER JOIN publishers p
ON a.City = p.City
ORDER BY a.Au_lname asc
```

The Results pane displays the following data:

	Au_fname	Au_lname	pub_name
1	Abraham	Bennet	Algodata Infosystems
2	Reginald	Blotch-Halls	NULL
3	Cheryl	Carson	Algodata Infosystems
4	Michel	DeFrance	NULL
5	Innes	del Castillo	NULL
6	Ann	Dull	NULL
7	Marjorie	Green	NULL
8	Momingstar	Greene	NULL
9	Burt	Ginglesby	NULL
10	Sheryl	Hunter	NULL
11	Livia	Karsen	NULL
12	Charlene	Locksley	NULL
13	Stearns	MacFeather	NULL
14	Heather	McBadden	NULL
15	Michael	O'Leary	NULL
16	Sylvia	Panteley	NULL
17	Albert	Ringer	NULL
18	Anne	Ringer	NULL
19	Meander	Smith	NULL
20	Dean	Straight	NULL
21	Dik	Stringer	NULL
22	Johnson	White	NULL
23	Akiko	Yokomoto	NULL

Usar combinaciones outer derecha

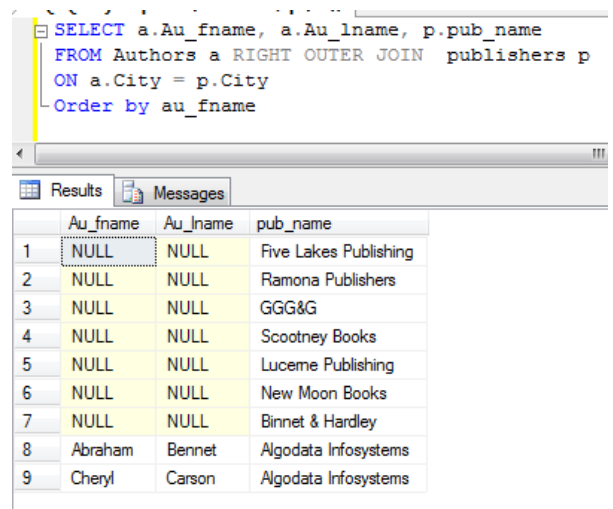
El resultado incluirá todas las filas de la tabla referenciada a la derecha del RIGHT OUTER JOIN. Las únicas columnas que serán recuperadas de la tabla izquierda serán aquellas que cumplan con la condición de la combinación.

En el siguiente comando SELECT, se usa una combinación outer derecha para obtener la lista de editores, los nombres y apellidos de los autores, si estos autores se encuentran en la misma ciudad de los editores.

```
SELECT a.Au_fname, a.Au_lname, p.pub_name
FROM Authors a RIGHT OUTER JOIN publishers p
ON a.City = p.City
Order by au_fname
```

El conjunto de resultados de esta consulta listará los nombres de todos los editores de la tabla Editores. El conjunto de resultados incluirá solo aquellos autores que se encuentren en la misma ciudad del editor.

Si un autor no se encuentra en la ciudad del editor, un valor nulo se retorna para las columnas Au_apellido y Au_nombre del conjunto de resultados.



The screenshot shows a SQL query window with the following query:

```
SELECT a.Au_fname, a.Au_lname, p.pub_name
FROM Authors a RIGHT OUTER JOIN publishers p
ON a.City = p.City
Order by au_fname
```

The Results pane displays the following data:

	Au_fname	Au_lname	pub_name
1	NULL	NULL	Five Lakes Publishing
2	NULL	NULL	Ramona Publishers
3	NULL	NULL	G&G
4	NULL	NULL	Scotney Books
5	NULL	NULL	Luceme Publishing
6	NULL	NULL	New Moon Books
7	NULL	NULL	Binnet & Hardley
8	Abraham	Bennet	Algodata Infosystems
9	Cheryl	Carson	Algodata Infosystems

Usar combinaciones outer completas

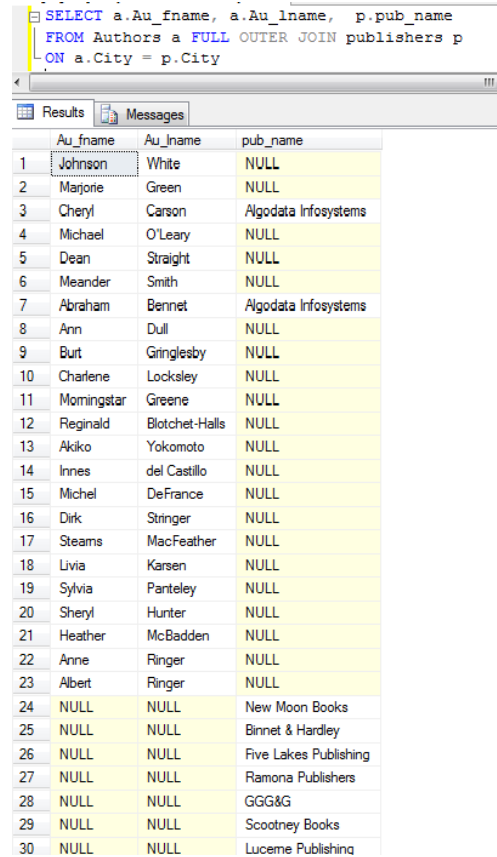
Los resultados incluirán todas las filas de ambas tablas, sin tener en cuenta si los valores cumplen con la condición de la combinación.

En el siguiente comando SELECT, se usa una combinación outer completa para obtener la lista de los editores y los nombres y apellidos de los autores

```
SELECT a.Au_fname, a.Au_lname, p.pub_name  
FROM Authors a FULL OUTER JOIN publishers p  
ON a.City = p.City
```

El conjunto de resultados de esta consulta listará el nombre de todos los editores de la tabla Editores y todos los autores de la tabla Autores. Si un autor no se encuentra ubicado en la misma ciudad del editor, un valor nulo se retorna en las columnas Au_apellido y Au_nombre del conjunto de resultados.

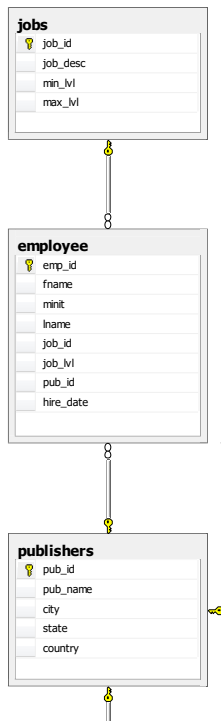
Si un editor no se encuentra en la misma ciudad del autor, se retorna un valor nulo en la columna del nombre. Cuando se cumple la condición todas las columnas del conjunto de resultados tendrán un valor. Ejecute esta instrucción en el analizador de consulta.



```
SELECT a.Au_fname, a.Au_lname, p.pub_name
FROM Authors a FULL OUTER JOIN publishers p
ON a.City = p.City
```

	Au_fname	Au_lname	pub_name
1	Johnson	White	NULL
2	Marjorie	Green	NULL
3	Cheryl	Carson	Algodata Infosystems
4	Michael	O'Leary	NULL
5	Dean	Straight	NULL
6	Meander	Smith	NULL
7	Abraham	Bennet	Algodata Infosystems
8	Ann	Dull	NULL
9	Burt	Gringlesby	NULL
10	Charlene	Locksley	NULL
11	Momingstar	Greene	NULL
12	Reginald	Blotch-Halls	NULL
13	Akiko	Yokomoto	NULL
14	Innes	del Castillo	NULL
15	Michel	DeFrance	NULL
16	Dirk	Stringer	NULL
17	Steams	MacFeather	NULL
18	Livia	Karsen	NULL
19	Sylvia	Panteley	NULL
20	Sheryl	Hunter	NULL
21	Heather	McBadden	NULL
22	Anne	Ringer	NULL
23	Albert	Ringer	NULL
24	NULL	NULL	New Moon Books
25	NULL	NULL	Binnet & Hardley
26	NULL	NULL	Five Lakes Publishing
27	NULL	NULL	Ramona Publishers
28	NULL	NULL	GGG&G
29	NULL	NULL	Scootney Books
30	NULL	NULL	Luceme Publishing

Ejemplo 2: Analicemos esta sección de la Base de datos PUB...



Si tenemos el nombre de la publicitaria y no su código y deseamos conocer que empleados laboran en esta publicitaria, debemos trabajar con las dos tablas. De employee sacaremos el nombre de los empleados, pero tenemos que condicionar la salida a los empleados que correspondan a la publicitaria que nos piden, (para este ejemplo 'New Moon Books') por lo que debo usar la tabla Publisher que es donde tengo el nombre de la publicitaria y establecer la unión de ambas tablas a través del atributo Pub_id (PK en Publisher y FK en Employee)

```
-- Trae el nombre de los empleados que trabajan
-- en la publicitaria llamada New Moon Books (pub_id=0736)

Select e.fname, e.lname
from employee e join publishers p
on e.pub_id = p.pub_id
where p.pub_name like 'New%'
```

Results	Messages	
1	Ralle	Ibsen
2	Karin	Josephs
3	Matti	Karttunen
4	Laurence	Lebihan
5	Roland	Mendel
6	Timothy	O'Rourke
7	Mary	Saveley
8	Howard	Snyder
9	Martin	Sommer
10	Gary	Thomas

Suponga que ahora deseamos el nombre del empleado y el trabajo que realizan en esta publicitaria. Esto involucra la unión de tres tablas.

```

select e.fname, e.lname, j.job_desc
from employee e join publishers p on e.pub_id = p.pub_id
      join jobs j on j.job_id = e.job_id
where pub_name like 'New%'

```

	fname	lname	job_desc
1	Palle	Ibsen	Marketing Manager
2	Karin	Josephs	Designer
3	Matti	Karttunen	Managing Editor
4	Laurence	Lebihan	Publisher
5	Roland	Mendel	Operations Manager
6	Timothy	O'Rourke	Sales Representative
7	Mary	Saveley	Public Relations Manager
8	Howard	Snyder	Editor
9	Martin	Sommer	Productions Manager
10	Gary	Thomas	Acquisitions Manager

Note que unimos primero dos tablas y colocamos en el ON, qué campos las relacionan. Luego otro join para unir la tercera tabla y con el on el campo a través de la cual las relacionó.

SUB CONSULTAS

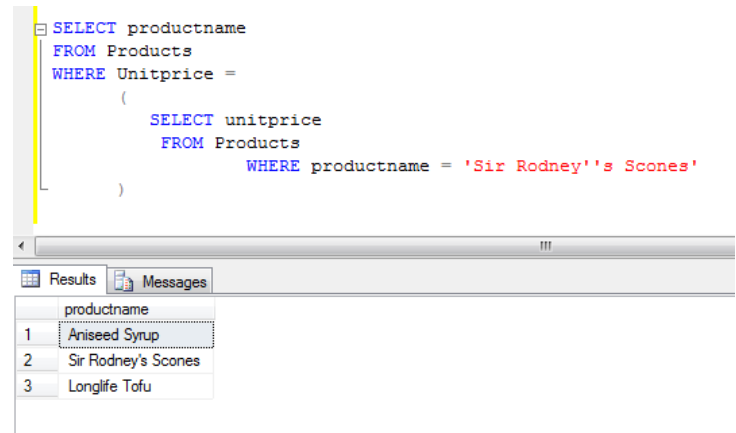
La mayoría de las combinaciones pueden ser reescritas como subconsultas (una consulta anidada dentro de otra consulta), y muchas subconsultas pueden ser reescritas como combinaciones.

Definir subconsultas dentro del comando SELECT

Una subconsulta es un comando SELECT que retorna un valor simple y se encuentra anidado dentro de un comando SELECT o dentro de otra subconsulta. Una subconsulta puede usarse en cualquier lugar en que una expresión lo permita.

En el ejemplo siguiente, se anida una subconsulta en la cláusula WHERE de un comando SELECT

```
USE Northwind
SELECT productname
FROM Products
WHERE Unitprice =
(
    SELECT unitprice FROM Products
    WHERE productname = 'Sir Rodney's Scones'
)
```



El comando SELECT interno; primero ubica el valor de PrecioUnit de los productos con nombre Sir Rodney's Scones, el cual es de \$10. Este valor (\$10) se usa luego en el comando SELECT exterior para obtener los nombres de los productos que tienen el mismo precio unitario. Si una tabla figura solo en la subconsulta y no en la consulta exterior, las columnas de esta tabla no pueden ser incluidas en el resultado de la consulta exterior.

Conceptualmente, el resultado de una subconsulta se sustituye dentro de la consulta exterior.

Tipos de subconsultas

Las subconsultas se pueden definir en muchos lugares dentro de un comando SELECT. Sin embargo, los comandos que incluyen subconsultas generalmente adoptan uno de los siguientes formatos:

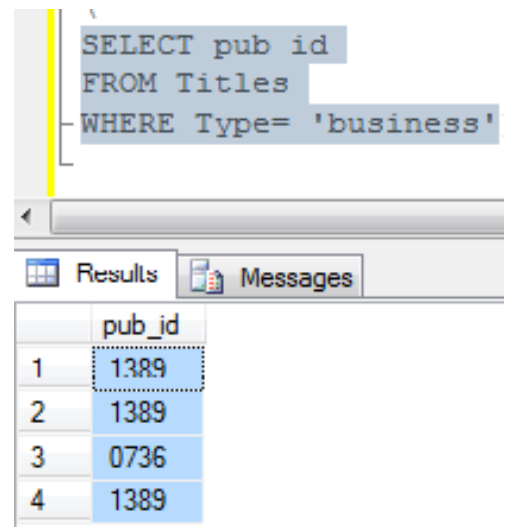
- WHERE <expresión> [NOT] IN (<subconsulta>)
- WHERE <expresión> <operador_de_comparación> [ANY | ALL] (<subconsulta>)
- WHERE [NOT] EXISTS (<subconsulta>)

Subconsultas que se usan con IN o con NOT IN

El resultado de una subconsulta introducida con IN (o con NOT IN) es una lista de cero o más valores. Después que la subconsulta devuelve el resultado, la consulta exterior lo utiliza. En el ejemplo siguiente. Una subconsulta se anida dentro de la cláusula WHERE, usando la palabra clave IN:

```
USE Pubs
SELECT pub_name
FROM publishers
WHERE pub_id IN
(
    SELECT pub_id
    FROM Titles
    WHERE Type= 'business'
)
```

Se puede evaluar este comando en dos pasos. Primero, la subconsulta retorna los números de ID de los editores que han publicado libros de negocios.



The screenshot shows a SQL query window with the following text:

```
SELECT pub id
FROM Titles
WHERE Type= 'business'
```

Below the query window, the 'Results' tab is active, displaying the output of the query:

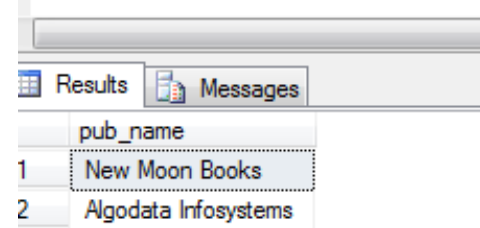
	pub_id
1	1389
2	1389
3	0736
4	1389

Luego estos valores son sustituidos en la consulta exterior, la cual encuentra los nombres que igualan estos números de identificación.

Primero, la subconsulta retorna los números de identificación de los editores que tienen libros de negocios publicados. Segundo, estos valores se sustituyen en la consulta exterior, la cual encuentra los nombres de los editores que tienen números de identificación dentro del conjunto de resultados de la subconsulta en la tabla Editores.

```
USE Pubs
SELECT pub_name
FROM publishers
WHERE pub_id IN
(
SELECT pub_id
FROM Titles
WHERE Type= 'business')

```



	pub_name
1	New Moon Books
2	Algodata Infosystems

Las subconsultas introducidas con la palabra clave **NOT IN** también retornan una lista de cero o más valores. La consulta trabaja exactamente igual que una con IN, excepto que con NOT IN se seleccionan a todas aquellas filas cuyos valores en la columna de comparación (pub_id en este caso) no se encuentren del conjunto de resultados de la subconsulta.

Subconsultas que son usadas con operadores de comparación

Los operadores de comparación que introduce una subconsulta se pueden modificar con las palabras claves **ALL** o **ANY**.

Las subconsultas introducidas con un operador de comparación modificado retornan una lista de cero o más valores y pueden incluir una cláusula GROUP BY o HAVING. Estas consultas pueden tener la cláusula **EXISTS**.

Las palabras clave ALL y ANY comparan un valor escalar con un conjunto de valores simples. **La palabra clave ALL** indica que se debe aplicar a todos los valores, mientras que ANY indica que al menos a un valor.

En el siguiente ejemplo, el operador de comparación mayor que (>) es usado con ANY

```

USE Pubs
SELECT Title
FROM Titles
WHERE Advance > ANY
(
SELECT advance
FROM publishers INNER JOIN Titles
ON Titles.pub_id = publishers.pub_id
AND pub_name = 'Algodata Infosystems'
)

```

Este comando encuentra los títulos que reciben un adelanto mayor que el mínimo adelanto pagado por Algodata Infosystems.

La cláusula WHERE en el comando SELECT exterior contiene una subconsulta que usa una combinación para recuperar monto de adelantos para Algodata Infosystems. El mínimo adelanto es utilizado para determinar que títulos obtener de la tabla Titulos.

```

SELECT advance
FROM publishers INNER JOIN Titles
ON Titles.pub_id = publishers.pub_id
AND pub_name = 'Algodata Infosystems'

```

	advance
1	5000.00
2	5000.00
3	5000.00
4	7000.00
5	8000.00
6	NULL

```

USE Pubs
SELECT Title
FROM Titles
WHERE Advance > ANY
(
SELECT advance
FROM publishers INNER JOIN Titles
ON Titles.pub_id = publishers.pub_id
AND pub_name = 'Algodata Infosystems'
)

```

	Title
1	You Can Combat Computer Stress!
2	The Gourmet Microwave
3	But Is It User Friendly?
4	Secrets of Silicon Valley
5	Computer Phobic AND Non-Phobic Individuals: Beha...
6	Life Without Fear
7	Onions, Leeks, and Garlic: Cooking Secrets of the M...
8	Sushi, Anyone?

Subconsultas que se usan con EXISTS y NOT EXISTS

Cuando se introduce una subconsulta con la palabra clave EXISTS. Esta funciona como un test de existencia. La cláusula WHERE de la consulta exterior comprueba por la existencia de las filas retornadas por la subconsulta. La subconsulta en realidad no produce ningún dato, solo retorna un valor de TRUE o FALSE.

En el siguiente ejemplo, la cláusula WHERE en el comando SELECT exterior contiene una subconsulta que usa EXISTS:

```

USE Pubs
SELECT pub_name
FROM publishers
WHERE EXISTS
(
SELECT * FROM Titles
WHERE Titles.pub_id = publishers.pub_id
AND Type = 'business'
)

```

The screenshot shows a SQL query window with the following text:

```

SELECT pub_name
FROM publishers
WHERE EXISTS
(
SELECT * FROM Titles
WHERE Titles.pub_id = publishers.pub_id
AND Type = 'business'
)

```

Below the query window, the 'Results' tab is active, displaying a table with two columns: 'pub_name' and 'Messages'. The table contains two rows of data:

	pub_name	Messages
1	New Moon Books	
2	Algodata Infosystems	

Para determinar el resultado de esta consulta, se toma cada fila de la tabla Editores y se verifica que exista dentro de las filas de la base título que sean de ese editor (Titles.Ed_id = Editores.Ed_id) y el tipo sea "negocios". Si existe algún título que cumpla, el nombre de ese editor figurará en el conjunto de resultados.

La palabra clave NOT EXISTS trabaja como EXISTS, excepto que la cláusula WHERE que tiene NOT EXIST será satisfecha solo si la subconsulta no devuelve ninguna fila.

F. RECURSOS:

Computador con acceso a internet, Software SQL Server, acceso a plataforma ecampus.utp.ac.pa/moodle, curso de Base de Datos 1.

G. RESULTADOS:

Usando la BD PUB

1. Liste el nombre y apellido de los empleados que trabajan en editoriales ubicadas en Francia (country France).

```

select n.[fname] + ' ' + n.[lname] as 'Nombre', p.[country]
from [dbo].[employee] as n inner join [dbo].[publishers] as p
on n.[pub_id] = p.[pub_id]
where [country] = 'France'

```

The screenshot shows a SQL query window with the following text:

```

select n.[fname] + ' ' + n.[lname] as 'Nombre', p.[country]
from [dbo].[employee] as n inner join [dbo].[publishers] as p
on n.[pub_id] = p.[pub_id]
where [country] = 'France'

```

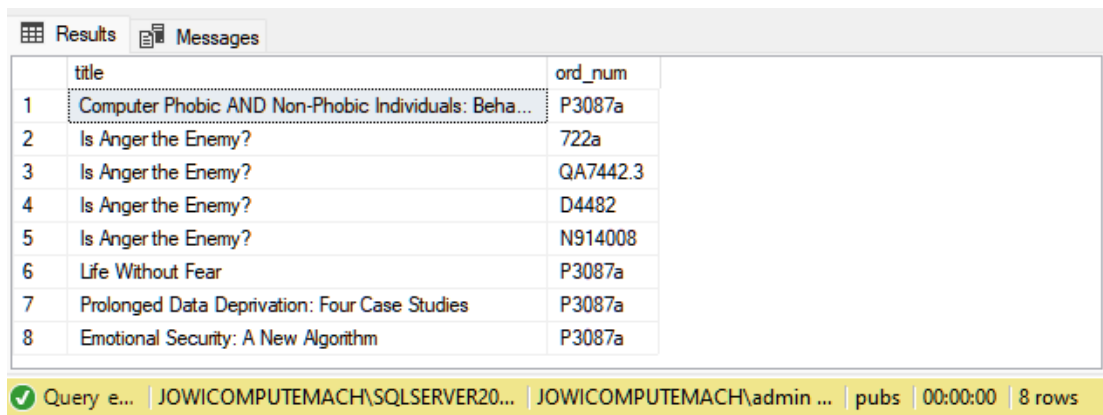
Below the query window, the 'Results' tab is active, displaying a table with two columns: 'Nombre' and 'country'. The table contains seven rows of data:

	Nombre	country
1	Carlos Hemadez	France
2	Karla Jablonski	France
3	Pirkko Koskitalo	France
4	Patricia McKenna	France
5	Helvetius Nagy	France
6	Manuel Pereira	France
7	Annette Roulet	France

At the bottom of the window, a status bar shows: Que... | JOWICOMPUTEMACH\SQLSERVER20... | JOWICOMPUTEMACH\admin ... | pubs | 00:00:00 | 7 rows

2. Muestre los nombres de los libros vendidos de psychology y el número de orden por el cual se compraron.

```
select t.[title], o.[ord_num]
  from [dbo].[titles] as t inner join [dbo].[sales] as o
 on t.[title_id] = o.[title_id]
 where t.[type] = 'psychology'
```

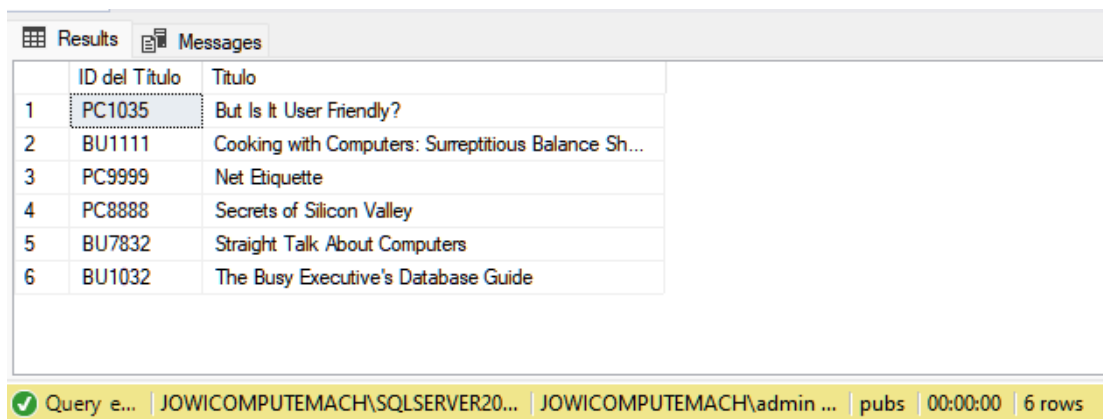


The screenshot shows a SQL Server query window with the 'Results' tab selected. The query results are displayed in a table with two columns: 'title' and 'ord_num'. There are 8 rows of data. The status bar at the bottom indicates 'Query e...' in the 'JOWICOMPUTEMACH\SQLSERVER20...' instance, 'JOWICOMPUTEMACH\admin ...' user, 'pubs' database, '00:00:00' duration, and '8 rows' returned.

	title	ord_num
1	Computer Phobic AND Non-Phobic Individuals: Beha...	P3087a
2	Is Anger the Enemy?	722a
3	Is Anger the Enemy?	QA7442.3
4	Is Anger the Enemy?	D4482
5	Is Anger the Enemy?	N914008
6	Life Without Fear	P3087a
7	Prolonged Data Deprivation: Four Case Studies	P3087a
8	Emotional Security: A New Algorithm	P3087a

3. Construya una consulta que incluya una subconsulta y la palabra clave IN para recuperar un conjunto de resultados de dos columnas (title_id y title) de la tabla Titles, de aquellos títulos que hayan sido publicados por editores de la ciudad de Berkeley, columna city (ciudad) de la tabla Publishers. Ordenar por title.

```
Select title_id 'ID del Título', title 'Titulo'
  from titles
 where pub_id in (select pub_id from publishers where city like 'Berkeley')
 order by title
```



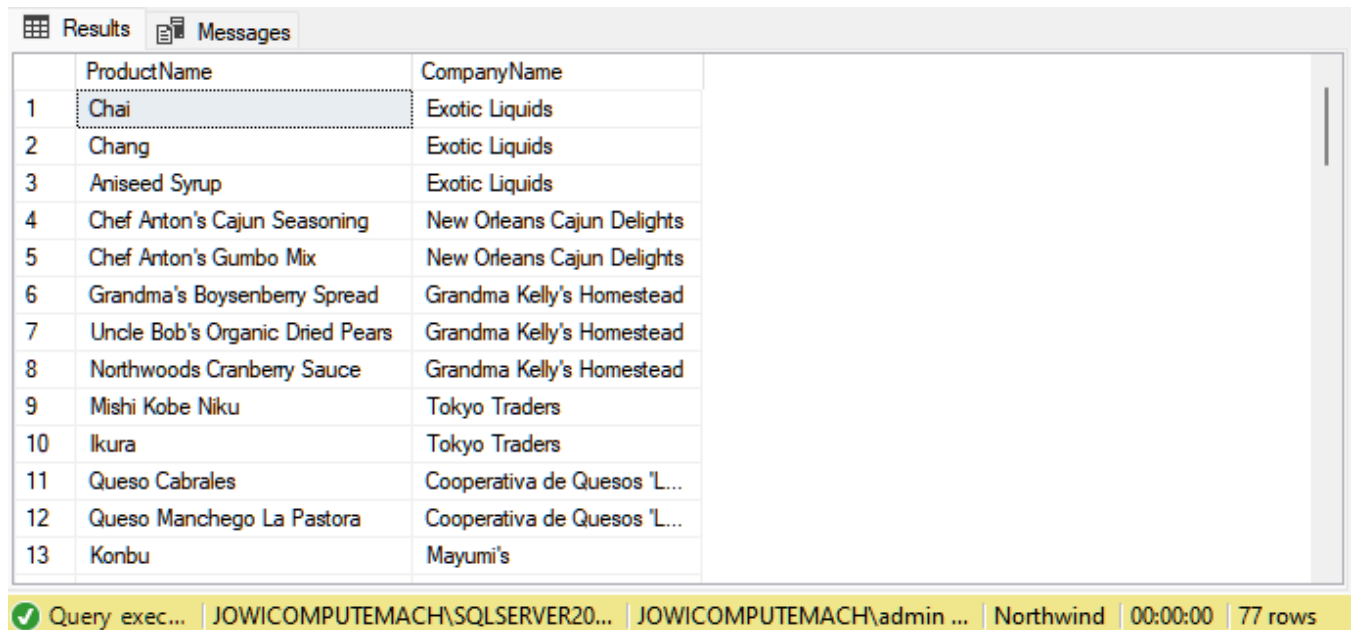
The screenshot shows a SQL Server query window with the 'Results' tab selected. The query results are displayed in a table with two columns: 'ID del Título' and 'Titulo'. There are 6 rows of data. The status bar at the bottom indicates 'Query e...' in the 'JOWICOMPUTEMACH\SQLSERVER20...' instance, 'JOWICOMPUTEMACH\admin ...' user, 'pubs' database, '00:00:00' duration, and '6 rows' returned.

	ID del Título	Titulo
1	PC1035	But Is It User Friendly?
2	BU1111	Cooking with Computers: Surreptitious Balance Sh...
3	PC9999	Net Etiquette
4	PC8888	Secrets of Silicon Valley
5	BU7832	Straight Talk About Computers
6	BU1032	The Busy Executive's Database Guide

Ahora usando la BD Northwind

4. Liste el nombre del producto (productname) y el nombre de la compañía que lo supe (companyname).

```
Select p.[ProductName], s.[CompanyName]
from [dbo].[Products] as p inner join [dbo].[Suppliers] as s
on p.[SupplierID] = s.[SupplierID]
```



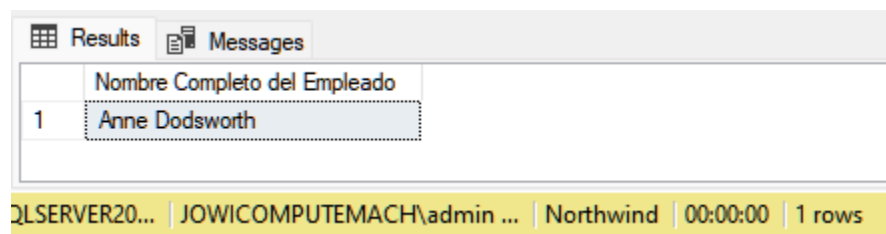
The screenshot shows a SQL query result in a table with two columns: ProductName and CompanyName. The results are listed in 13 rows. The status bar at the bottom indicates the query was executed successfully on the Northwind database, returning 77 rows.

	ProductName	CompanyName
1	Chai	Exotic Liquids
2	Chang	Exotic Liquids
3	Aniseed Syrup	Exotic Liquids
4	Chef Anton's Cajun Seasoning	New Orleans Cajun Delights
5	Chef Anton's Gumbo Mix	New Orleans Cajun Delights
6	Grandma's Boysenberry Spread	Grandma Kelly's Homestead
7	Uncle Bob's Organic Dried Pears	Grandma Kelly's Homestead
8	Northwoods Cranberry Sauce	Grandma Kelly's Homestead
9	Mishi Kobe Niku	Tokyo Traders
10	Ikura	Tokyo Traders
11	Queso Cabrales	Cooperativa de Quesos 'L...
12	Queso Manchego La Pastora	Cooperativa de Quesos 'L...
13	Konbu	Mayumi's

Query exec... | JOWICOMPUTEMACH\SQLSERVER20... | JOWICOMPUTEMACH\admin ... | Northwind | 00:00:00 | 77 rows

5. Deseamos saber el nombre del empleado que realizó la orden 10255

```
Select FirstName + ' ' + LastName 'Nombre Completo del Empleado'
from Employees
where EmployeeID in (Select EmployeeID from Orders where OrderID = '10255')
```



The screenshot shows a SQL query result in a table with one column: Nombre Completo del Empleado. The result is listed in one row: Anne Dodsworth. The status bar at the bottom indicates the query was executed successfully on the Northwind database, returning 1 row.

	Nombre Completo del Empleado
1	Anne Dodsworth

SQLSERVER20... | JOWICOMPUTEMACH\admin ... | Northwind | 00:00:00 | 1 rows

H. CONSIDERACIONES FINALES:

R/El uso del Álgebra Relacional es una de las armas más potentes que se tienen al momento del desarrollo de cualquier tipo de Modelo, ya sea Base de Datos o no, en el que existan relaciones entre sus conjuntos o sus subconjuntos; pero cuando se ve enfocado con una aplicación a ejemplos de la vida cotidiana como lo pueden ser directamente su uso en las Bases de Datos, especialmente con la simulación de un Diagrama de Venn entre dos

conjuntos (En este caso Entidades-Tablas), podemos seleccionar usando el comando “Join” y sus múltiples variaciones, cualquiera de los elementos que conformen esos dos conjuntos, siempre y cuando cumplan con ciertas reglas, como por ejemplo la dimensionalidad del Producto Cartesiano que se genera en ellas.

I. BIBLIOGRAFÍA:

- *A fondo SQL Server, Kalen Delaney, Serie de programación Microsoft, McGraw Hill profesional*

J. RÚBRICA:

- Los problemas tienen una ponderación de 20 puntos y se evaluará que aparezca el código como texto y la imagen respectiva de resultados.