**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

1. **TÍTULO** **DE LA EXPERIENCIA:**

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

1. **TEMAS:**
   1. Combinaciones Join
   2. Subconsultas con select anidados
2. **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.

1. **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.

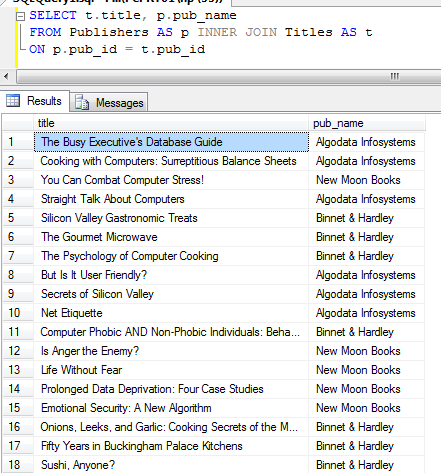
Copie estas capturas de pantalla en la sección G (RESULTADOS) de esta guía, en el número mostrado en la sección E (PROCEDIMIENTO). Corte y sólo presente el área de trabajo donde aparece la instrucción y el resultado obtenido, no incluya el explorador de objetos, el menú de opciones ni la barra de herramientas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

1. **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

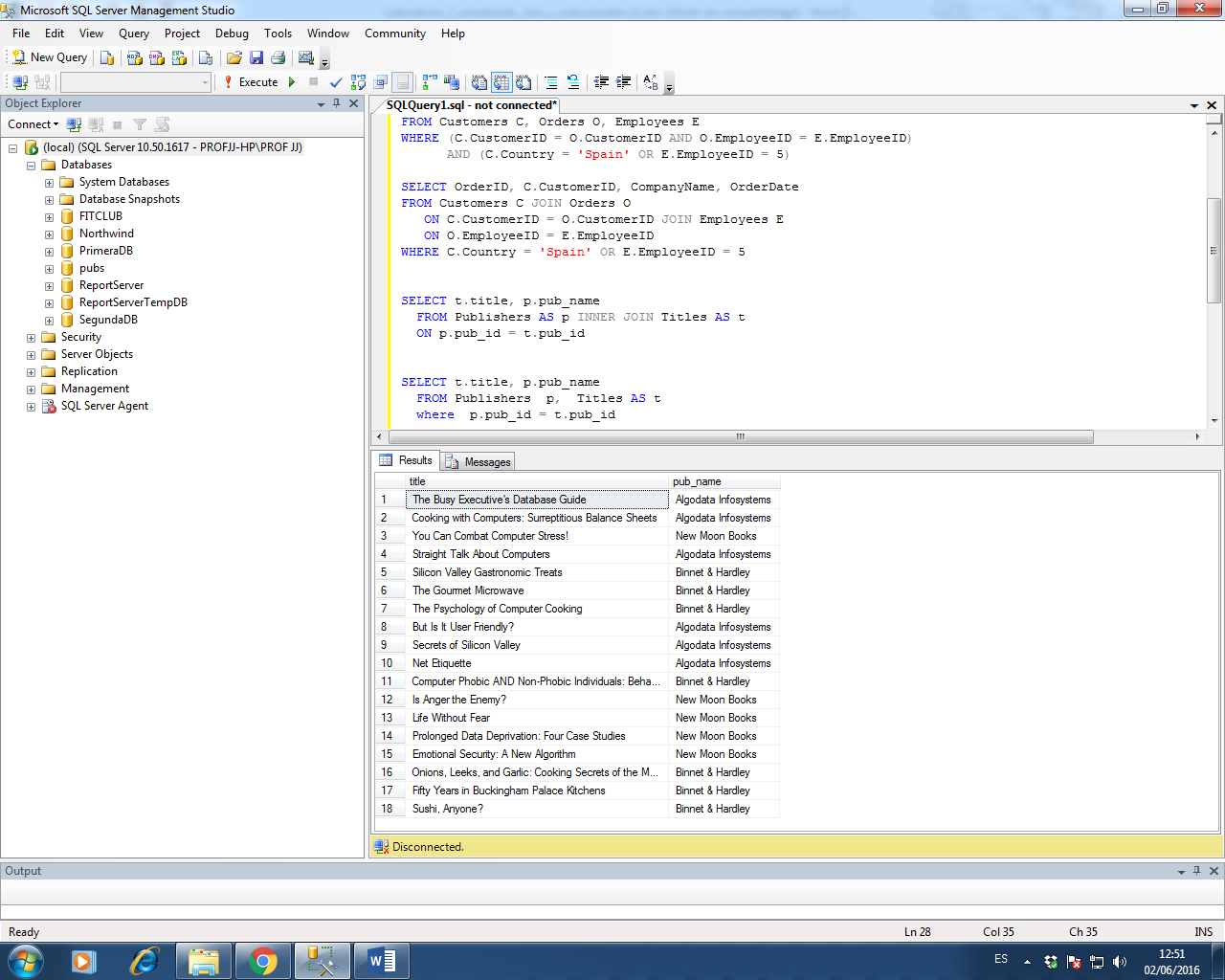
ON p.pub\_id = t.pub\_id

El comando SELECT recupera el nombre del libro (tiltle) 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.

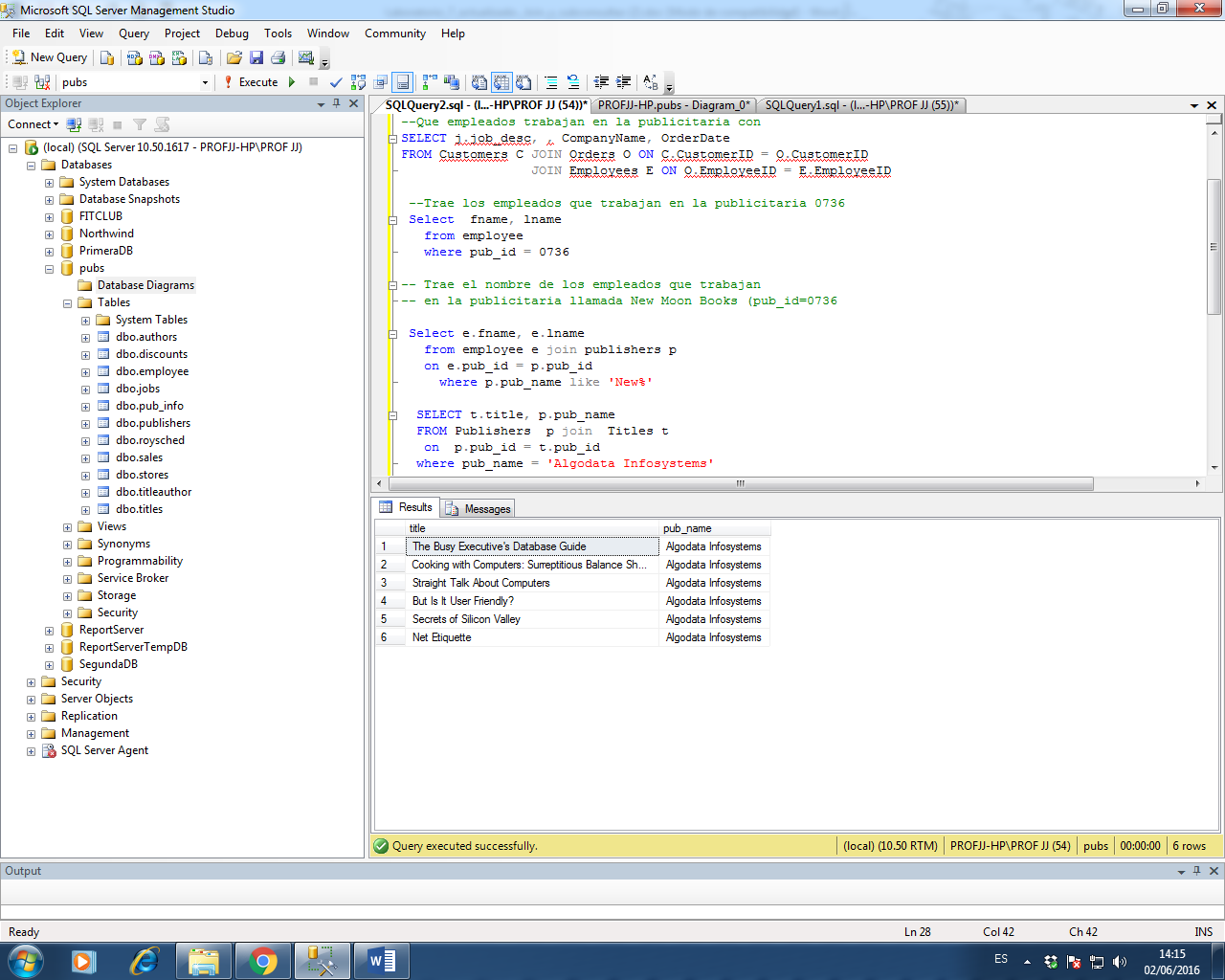
Otra forma de **hacer lo mismo** es indicando las tablas en el from separadas por comas y colocar en el where los criterios de coincidencia que debemos validar (PK y FK)

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 FR porque no teníamos ninguna condición adicional que validar.

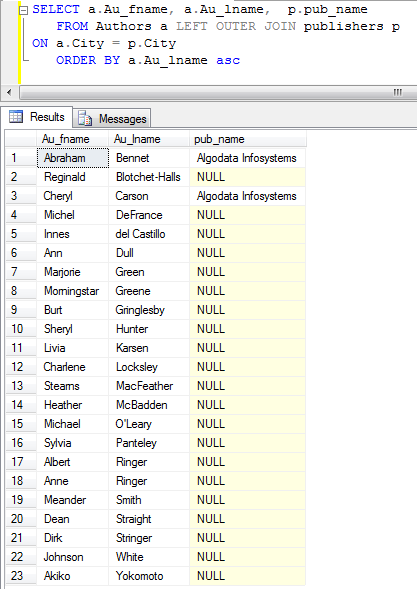


Si particularmente sólo quisiéramos los títulos de libros de Algodata System, tendríamos que usar el **join** para definir qué tablas está usando, el **on** para unir las llaves y el **where** para la condición que desea.

**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**

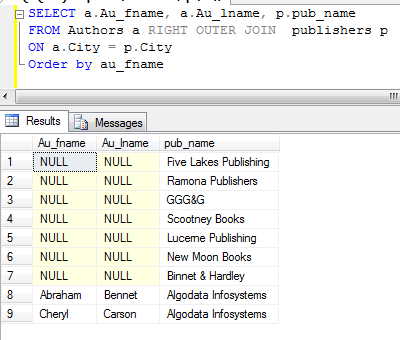
**ON a.City = p.City**

**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

**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. **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**

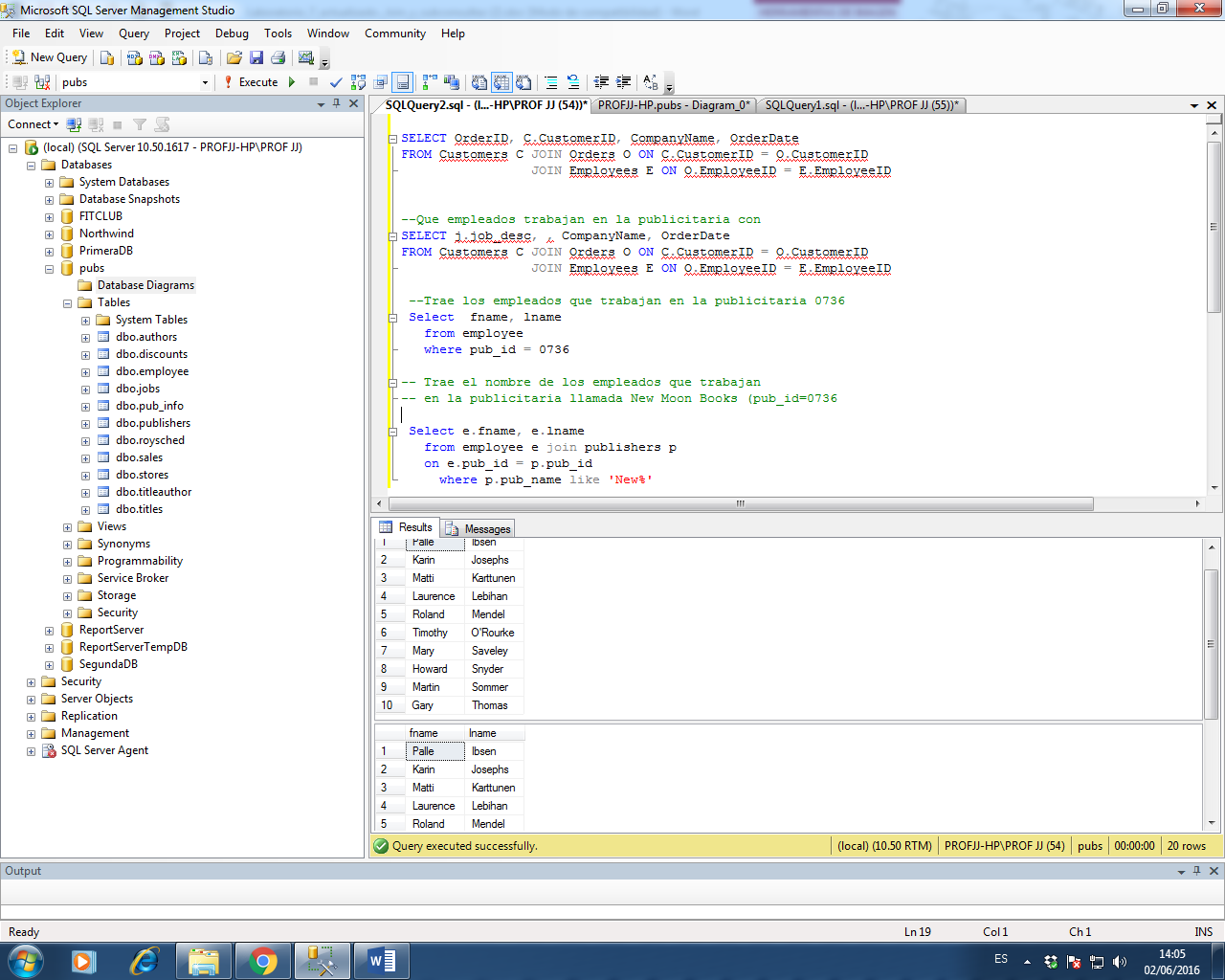
**ON a.City = p.City**

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.

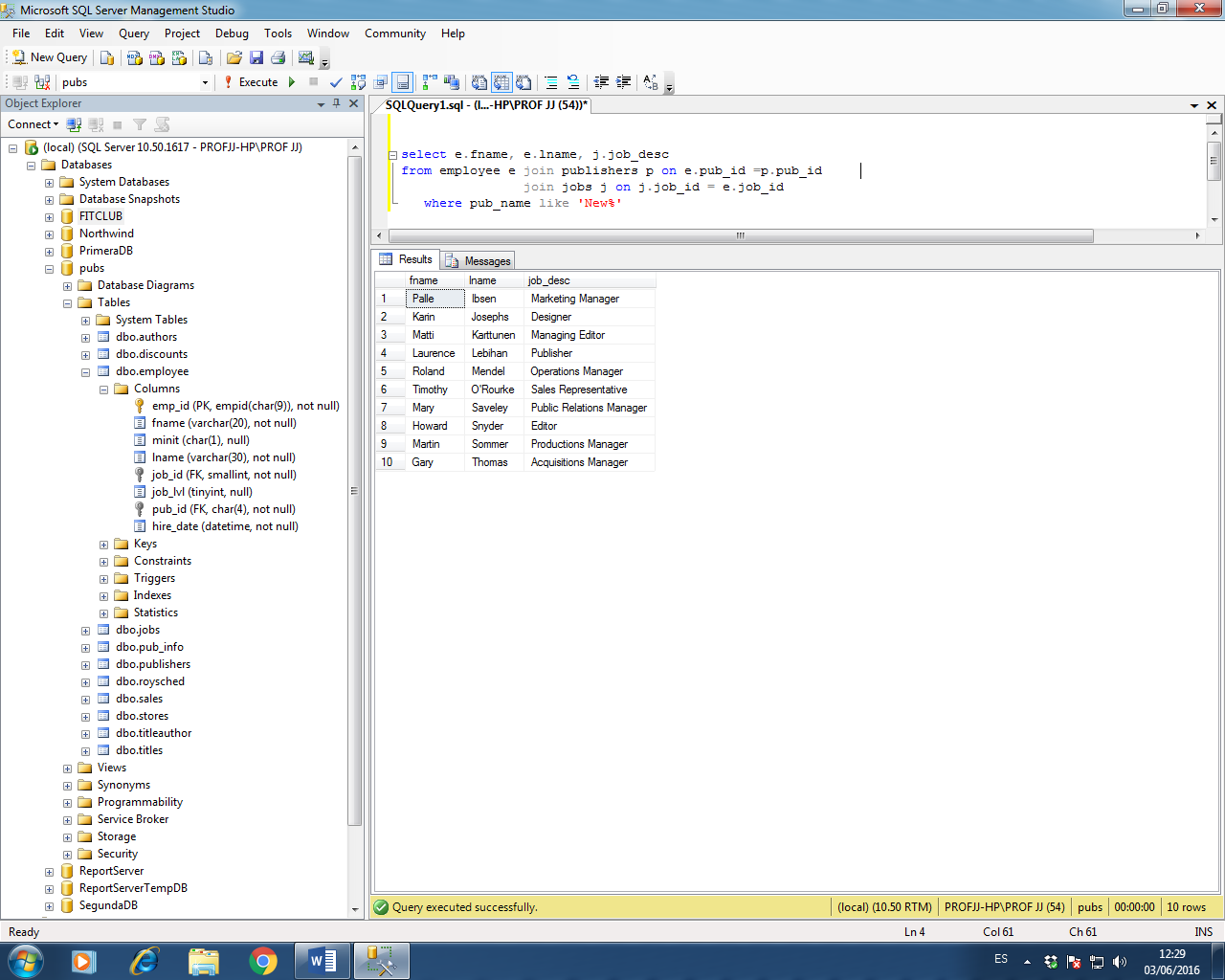
Si un editor no se encuentra en la misma ciudad del autor, se retorna un valor nulo en la columna del nombre. Cuando se cumple la condición todas las columnas del conjunto de resultados tendrán un valor. Ejecute esta instrucción en el analizador de consulta.

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 tabal 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)



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

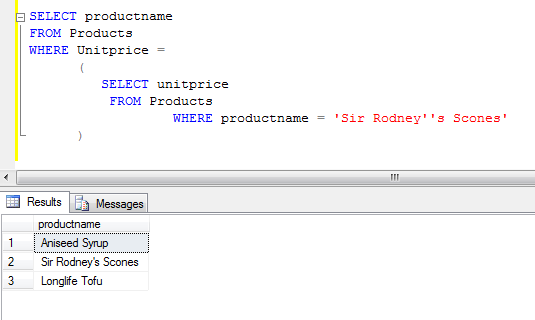


Note que unimos primero dos tablas y colocamos en el ON, qué campos las relacionan. Luego otro join para unir la tercera tabla y con el on el campo a través de la cual las relacionó.

**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 valorsimple 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>)

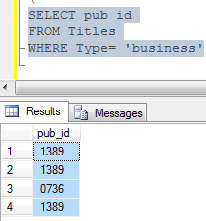
**Subconsultas que se usan con IN o con NOT IN**

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**

**(**

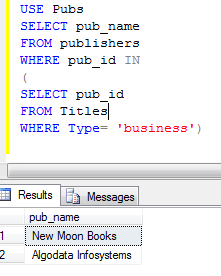
**SELECT pub\_id**

**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.

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.

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 todos aquellas filas cuyos valores en la columna de comparación (pub\_id en este caso) no se encuentren del conjunto de resultados de la subconsulta.

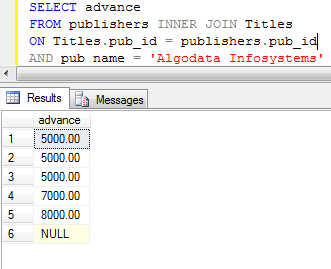
**Subconsultas que son usadas con operadores de comparación**

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**

**(**

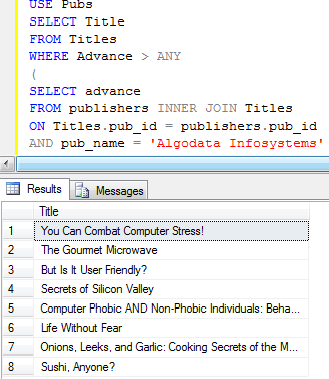
**SELECT advance**

**FROM publishers INNER JOIN Titles**

**ON Titles.pub\_id = publishers.pub\_id**

**AND pub\_name = 'Algodata Infosystems'**

**)**

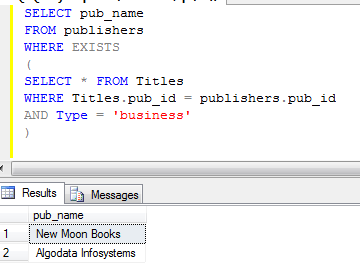
Este comando encuentra los títulos que reciben un adelanto mayor que el mínimo adelanto pagado por Algodata Infosystems.

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.

**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'**)

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 (Titulos.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.

1. **RECURSOS:**

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

1. **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'

**Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente**

1. **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'

**Tabla

Descripción generada automáticamente**

1. **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

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Ahora usando la BD Northwind**

1. **Liste el nombre del producto (productname) y el nombre de la compañía que lo suple (companyname).**

Select p.[ProductName], s.[CompanyName]

from [dbo].[Products]as p inner join [dbo].[Suppliers] as s

on p.[SupplierID] = s.[SupplierID]

**Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente**

**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')

**Tabla

Descripción generada automáticamente**

1. **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.*

1. **BIBLIOGRAFÍA:**

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

1. **RÚBRICA:**

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