



**Universidad Tecnológica de Panamá**  
**Facultad de Ingeniería de Sistemas Computacionales**  
**BASE DE DATOS I**  
**Laboratorio N°4**



**Facilitador:** Víctor A. Fuentes T.

**Estudiante:** Johel Heraclio Batista Cárdenas

**Cédula:** 8-914-587

**Grupo:** 1IF-131

**A. TÍTULO DE LA EXPERIENCIA:**

Laboratorio No.4. Concatenar, Control de filas duplicadas, Ordenación y salida controlada

**B. TEMAS :**

- I. Concatenar dos o más atributos (Operador +)
- II. Control de filas duplicadas (distinct)
- III. Salida ordenada ascendente o descendentemente (order by)
- IV. Salida controlada por cláusula Where

**C. OBJETIVO(S):**

- Realizar consultas con la cláusula Select, utilizando el operador de concatenación +, para visualizar la salida concatenada
- Utilizar la cláusula distinct en el select para omitir las filas duplicadas en una consulta
- Presentar salida ordenada con el uso de la cláusula order by en la sentencia select
- Controlar la salida de un select con el uso de la cláusula where.

**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. Coloque además el código usado en forma de texto de forma directa y no como parte de la captura de pantalla.

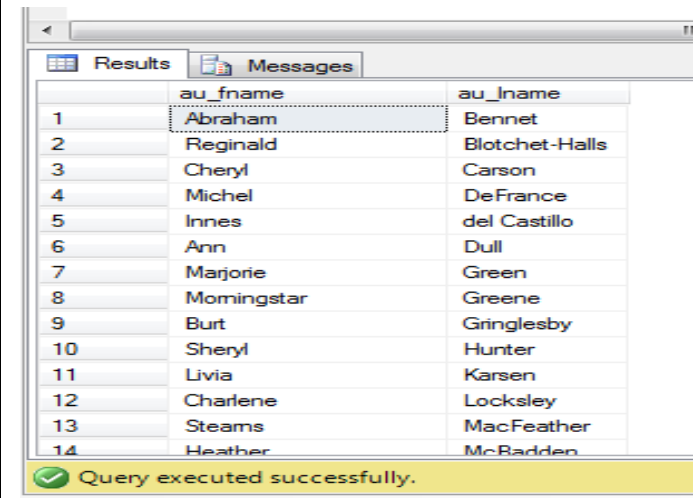
Copie estas capturas de pantalla en la sección G (RESULTADOS) de esta guía, en según el número mostrado en la sección E (PROCEDIMIENTO). Corte y sólo presente el área de trabajo donde aparece el resultado obtenido, no incluya el explorador de objetos, el menú de opciones ni la barra de herramientas estándar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

## E. PROCEDIMIENTO O ENUNCIADO DE LA EXPERIENCIA

### Operador de Concatenación + y salida de literal ''

Muchas veces deseamos que nuestra salida se presente en un formato más elegante o legible para el usuario final. Note que al desplegar el nombre y el apellido de la tabla Authors de la BD PUBS, los campos se presentan como columnas.

```
select  au_fname,  
        au_lname  
from    authors
```



	au_fname	au_lname
1	Abraham	Bennet
2	Reginald	Blotchet-Halls
3	Cheryl	Carson
4	Michel	DeFrance
5	Innes	del Castillo
6	Ann	Dull
7	Marjorie	Green
8	Momingstar	Greene
9	Burt	Gringlesby
10	Sheryl	Hunter
11	Livia	Karsen
12	Charlene	Locksley
13	Steams	MacFeather
14	Heather	McBadden

Podemos lograr que estos dos campos se presenten como si hubieran sido almacenados como un nombre completo utilizando el operador de **concatenación +**, y además podemos **incluir entre comilla simple algún carácter**, en este caso un espacio en blanco, para que la salida no se visualice pegada.

*El literal puede ser cualquier carácter, expresión o número, incluido entre las comillas.*

```
select  au_fname + ' ' + au_lname 'Nombre Autor'  
from    authors
```

Literal Blanco adicionado para que no imprima pegado el nombre y el apellido

El + es el operador que realiza la concatenación

Results		Messages
	Nombre Autor	
1	Abraham Bennet	
2	Reginald Blotchet-Halls	
3	Cheryl Carson	
4	Michel DeFrance	
5	Innes del Castillo	
6	Ann Dull	
7	Marjorie Green	

Note la diferencia en la salida. Ahora se despliega el nombre del autor como si estuviera almacenado como uno solo.

Observación: El operador + para la operación de concatenación es propio del SQL SERVER, otros sistemas de gestión usan el caracter & (ampersand) como operador rápido para concatenar. El estándar del SQL es la función de cadena CONCAT (esta será vista en BD II cuando vean las funciones de cadena)

### 1. Utilizando la base de datos Northwind realice la consulta que produce como resultado la siguiente salida.

Results		Messages
	Empleado	
1	Nancy Davolio, Sales Representative	
2	Andrew Fuller, Vice President, Sales	
3	Janet Leverling, Sales Representative	
4	Margaret Peacock, Sales Representative	
5	Steven Buchanan, Sales Manager	
6	Michael Suyama, Sales Representative	
7	Robert King, Sales Representative	
8	Laura Callahan, Inside Sales Coordinator	
9	Anne Dodsworth, Sales Representative	

Utilice los campos **FirstName**, **LastName** y **Title** de la tabla **Employees**

## Cláusula DISTINCT

Es frecuente que al usar la cláusula Select, se presenten filas repetidas.

2. Realice la siguiente consulta a la base de datos Northwind y muestre la salida:

```
select ContactTitle
from Customers
```

(Observe que se presentan campos repetidos )

3. Ahora adicione la clausula DISTINCT o distinct después del select

```
select distinct ContactTitle
from Customers
```

¿Qué ocurre? Imprima y comente sus resultados.

4. Realice y muestre la siguiente consulta

```
select ContactTitle, Country
from Customers
```

Observe que hay salidas repetidas, ejemplo Owner Mexico y Owner Francia.

El DISTINCT puede aplicarse a multiples columnas, en cuyo caso: **explique qué sucede.**

5. Muestre y explique qué pasa al utilizar DISTINCT .

```
select distinct ContactTitle, Country
from Customers
```

## Cláusula ORDER BY:

Permite ordenar los datos listados. No cambia el orden de los mismos en la BD.

<pre>SELECT      [ *  ó  columna  ó  grupo de columnas] FROM        Nombre_Tabla ORDER BY    columna [asc  ó desc]</pre>
--

Si no se especifica asc ó desc, **asume ordenar ascendentemente**

**6. Usando la BD Northwind, realice la siguiente consulta**

```
select *  
from Employees
```

Note el orden de los datos en la columna LastName. **Muestre sus resultados**

**7. Ahora utilicemos la clausula ORDER BY para ordenar estos datos**

```
select *  
from Employees  
order by LastName
```

**¿Qué le ocurrió a los datos? ¿Cómo se ordenaron?**

**8. Ordene los datos del punto 7 de manera descendente utilizando el**

***order by nombre\_columna\_a\_ordenar desc***

***Anote sus resultados***

**9. Realice la consulta de todos los campos de la tabla Customers de la BD Northwind, pero muéstrela ordenada ascendentemente por ContactTitle**

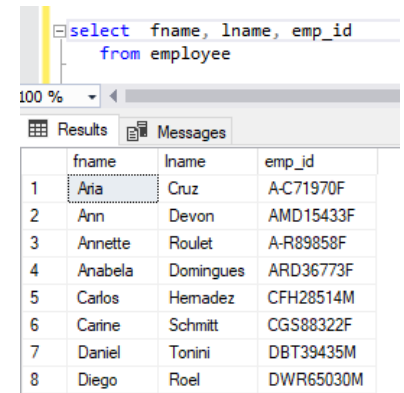
## Active la base de datos PUBS

### Ordenando por Posición:

En el punto anterior se listaron los datos ordenados de acuerdo con alguna columna o columnas especificada(s).

Veamos el resultado de la consulta

```
select  fname, lname, emp_id
from employee
```



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

```
select  fname, lname, emp_id
from employee
```

The results pane displays the following data:

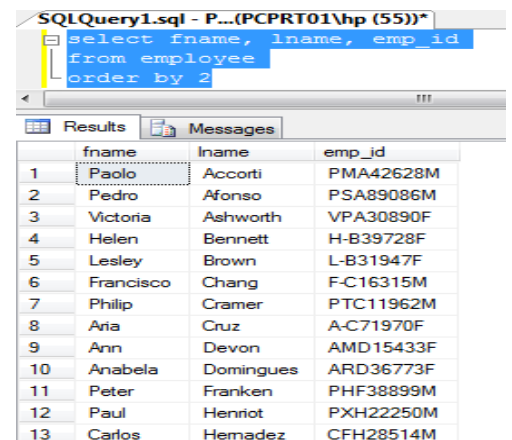
	fname	lname	emp_id
1	Aria	Cruz	A-C71970F
2	Ann	Devon	AMD15433F
3	Annette	Roulet	A-R89858F
4	Anabela	Domingues	ARD36773F
5	Carlos	Hemadez	CFH28514M
6	Carine	Schmitt	CGS88322F
7	Daniel	Tonini	DBT39435M
8	Diego	Roel	DWR65030M

*A pesar de que no se realiza ningún order by, observe que los campos se listan ordenados por el primer campo listado fname. (Esto se debe a que para esta tabla existe un índice que previamente ordena por fname y lname, lo que hará que aún cuando no se pida, se muestra ordenado)*

SQL Server nos permite ordenar de acuerdo con la posición que ocupe el campo en la lista Select.

Ejemplo 1: 

```
select  fname, lname, emp_id
from employee
order by 2
```



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

```
select  fname, lname, emp_id
from employee
order by 2
```

The results pane displays the following data:

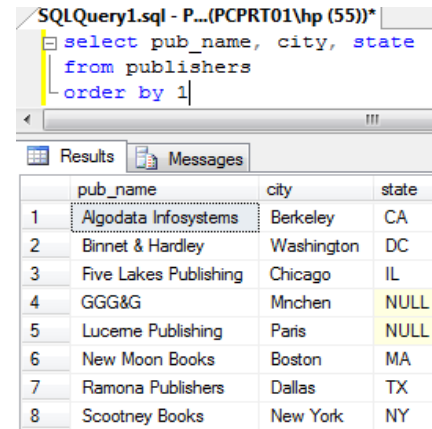
	fname	lname	emp_id
1	Paolo	Accorti	PMA42628M
2	Pedro	Afonso	PSA89086M
3	Victoria	Ashworth	VPA30890F
4	Helen	Bennett	H-B39728F
5	Lesley	Brown	L-B31947F
6	Francisco	Chang	F-C16315M
7	Philip	Cramer	PTC11962M
8	Aria	Cruz	A-C71970F
9	Ann	Devon	AMD15433F
10	Anabela	Domingues	ARD36773F
11	Peter	Franken	PHF38899M
12	Paul	Henriot	PXH22250M
13	Carlos	Hemadez	CFH28514M

Puede notar en la imagen que a pesar de que el primer campo listado es fname (ver la consulta previa mostrada), la consulta generada provoca que se ordene por el **segundo** atributo listado en la clausula select.

Ejemplo 2:

```
select pub_name, city, state
from publishers
order by 1
```

Note ahora que la salida esta ordenada de acuerdo al primer campo listado, es decir pub\_name.



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

```
SQLQuery1.sql - P...(PCPRT01\hp (55))
select pub_name, city, state
from publishers
order by 1
```

Below the query window, the 'Results' tab is active, displaying a table with 8 rows and 3 columns: pub\_name, city, and state. The rows are ordered by pub\_name.

	pub_name	city	state
1	Algodata Infosystems	Berkeley	CA
2	Binnet & Hardley	Washington	DC
3	Five Lakes Publishing	Chicago	IL
4	G&G	Mnchen	NULL
5	Luceme Publishing	Paris	NULL
6	New Moon Books	Boston	MA
7	Ramona Publishers	Dallas	TX
8	Scotney Books	New York	NY

**10. Liste los atributos stor\_name, address, city, state de la tabla stores ordenados de acuerdo a la tercera posición. Muestre la consulta realizada y el resultado obtenido.**

**11. Liste todos los atributos de la tabla publishers y aplique la ordenación por posición 4. Comente el resultado.**

### Ordenando por Varias Columnas:

Al hacer las consultas, se puede ordenar por más de una columna colocando seguido al **order by**, el orden de prioridad de la ordenación a realizar.

Cuando ordenamos por un grupo de columnas, es decir, cuando en el **order by** tenemos más de una columna, el sistema de gestión ordena la primera columna y luego de ordenar esta, procede a ordenar la segunda. Esto hace que sólo visualizaremos el orden de la segunda columna cuando las primeras sean iguales.



Note en este ejemplo en el que se ha ordenado sólo por la columna stor\_id, el orden en que aparecen las dos primeras tuplas listadas.

SQLQuery1.sql - P...(PCPRT01\hp (55))\*

```
select * from sales
order by 1
```

Results Messages

	stor_id	ord_num	ord_date	qty	payterms
1	6380	6871	1994-09-14 00:00:00.000	5	Net 60
2	6380	722a	1994-09-13 00:00:00.000	3	Net 60
3	7066	A2976	1993-05-24 00:00:00.000	50	Net 30
4	7066	QA7442.3	1994-09-13 00:00:00.000	75	ON invc
5	7067	DAAR2	1994-09-14 00:00:00.000	10	Net 60

Ahora al ordenar por dos campos, e incluir el order by stor\_id y qty, se ordena inicialmente por el primer campo (stor\_id 6380) pero el segundo campo definido como criterio de ordenación (qty) se aplica dentro de la primera ordenación realizada. Es decir que sólo veremos ordenados las segundas columnas definidas cuando la primera es igual. Si ordenamos por tres campos, veremos la ordenación del tercer campo, cuando el valor del primer campo y segundo campo sean exactamente iguales.

SQLQuery1.sql - P...(PCPRT01\hp (55))\*

```
select * from sales
order by 1, 4
```

Results Messages

	stor_id	ord_num	ord_date	qty	payterm
1	6380	722a	1994-09-13 00:00:00.000	3	Net 60
2	6380	6871	1994-09-14 00:00:00.000	5	Net 60
3	7066	A2976	1993-05-24 00:00:00.000	50	Net 30
4	7066	QA7442.3	1994-09-13 00:00:00.000	75	ON inv

**12.Despliegue de la tabla sales los atributos stor\_id, ord\_num, title\_id, qty y ordene por el atributo title\_id. Observe el orden.**

**13.Ahora repita la consulta del punto 12, pero ordene por los atributos title\_id y stor\_id. Comente sus resultados**

**14.Realice dos consultas ordenadas por más de una columna. Muestre sus resultados. Seleccione los campos y tablas de la base de datos NORTWIND. Recuerde que cada Select es una consulta.**

## Limitando la Selección de Columnas:

Se pueden restringir los resultados a través de la cláusula WHERE. Esta cláusula añade una condición a la salida, la cual debe cumplirse para poder mostrarse.

El formato es el siguiente:

```
SELECT      expresión
FROM        tabla
WHERE       condición (es)
[Order by]
```

*El order by siempre es la última instrucción del select.*

Los operadores de condición se dividen en dos categorías: **Operadores Lógicos y Operadores del SQL.**

Operador Lógico	Significado
=	Igual que
<	Menor
<=	Menor e igual
>	Mayor
>=	Mayor e igual

En este laboratorio veremos los operadores lógicos:

15. Liste todos los campos de la tabla authors. ¿Cuántos autores despliega?

16. Despliegue la tabla authors los autores cuyo apellido es Ringer

Select \*

```
from authors
where au_lname = 'Ringer' Sólo se omite la comilla si el dato es numérico
```

¿Qué se obtiene?

Repita la instrucción pero ahora coloque en mayúscula cerrada su consulta.

```
Select *
from authors
where au_lname = 'RINGER' SQL Server no discrimina entre mayúscula y minúscula
```

17. Ahora liste sólo los autores que viven en la ciudad de Oakland.

18. Despliegue la tabla roysched. Muestre los resultados de la consulta que le permiten obtener los datos de un hirange menor que 4000.

19. Muestre los resultados de las consultas que le permiten obtener los datos de un hirange mayor que 30000.

20. Muestre los resultados de las consultas que le permiten obtener los datos de un hirange mayor e igual a 40000.

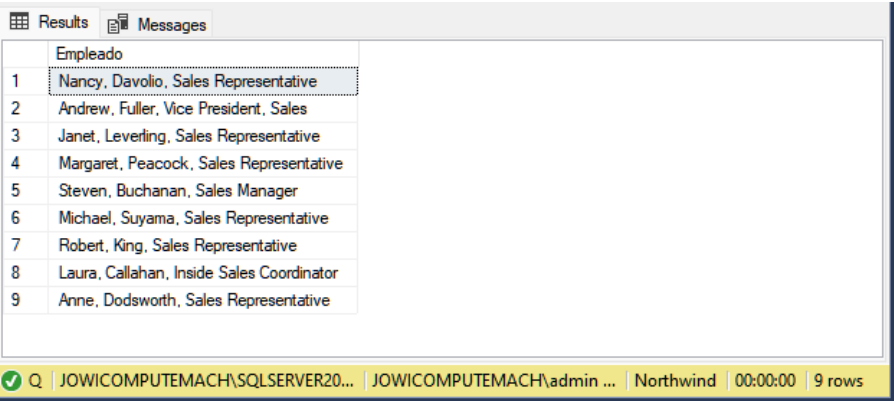
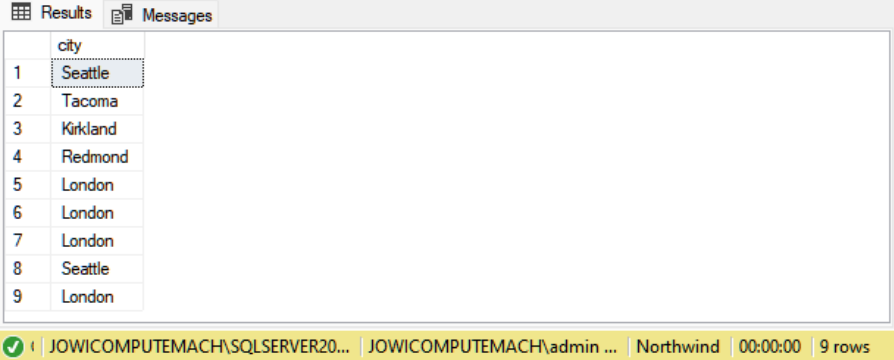
#### RECURSOS:

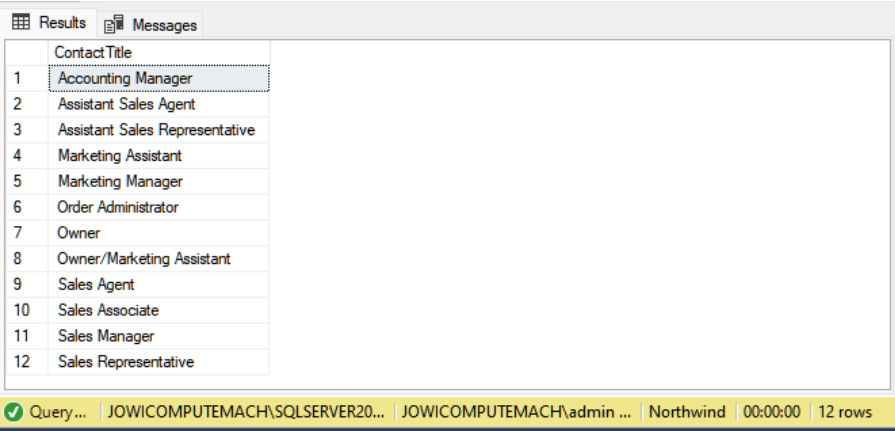
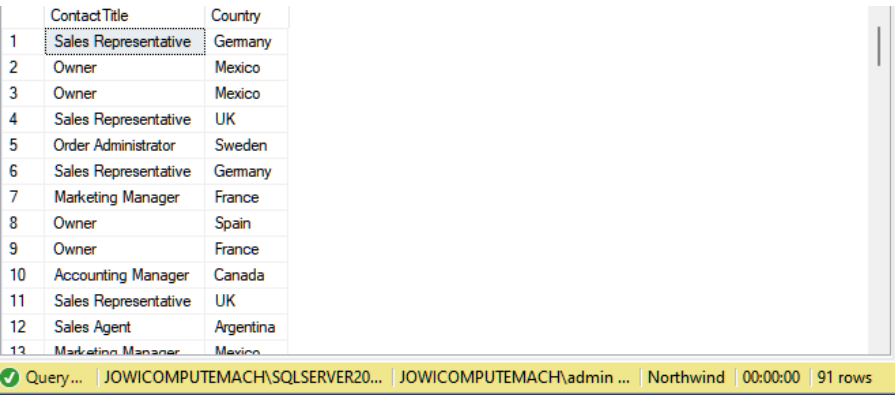
Computador con acceso a internet, Software SQL Server, acceso a plataforma [ecampus.utp.ac.pa/moodle](http://ecampus.utp.ac.pa/moodle), curso de Base de Datos I.

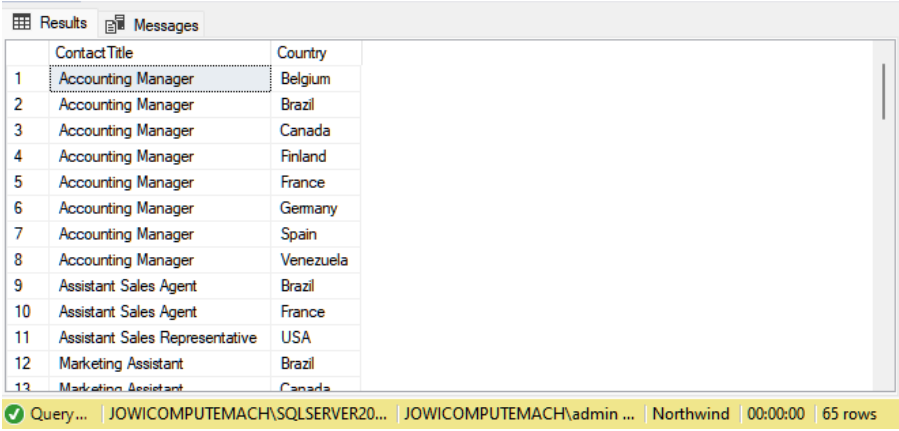
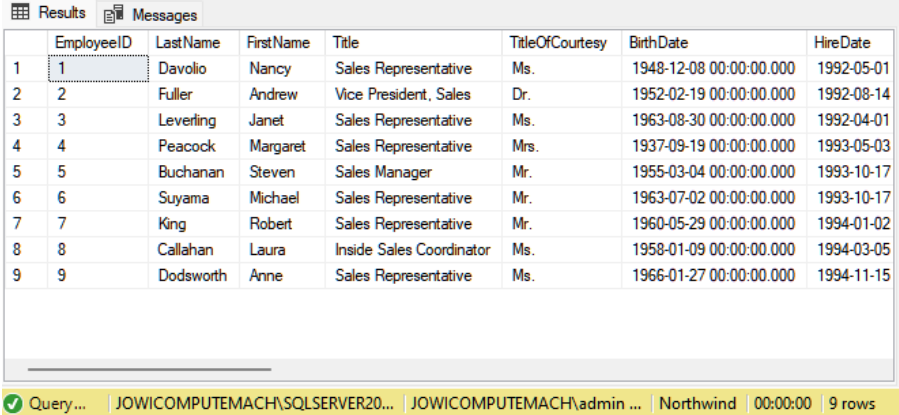
#### F. RESULTADOS:

*En esta sección Usted colocará las capturas de pantalla que muestran los resultados de los procesos realizados en el punto anterior.*

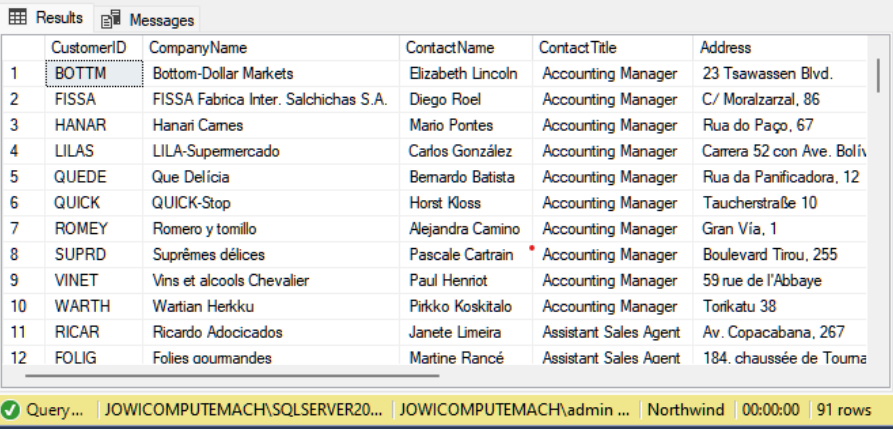
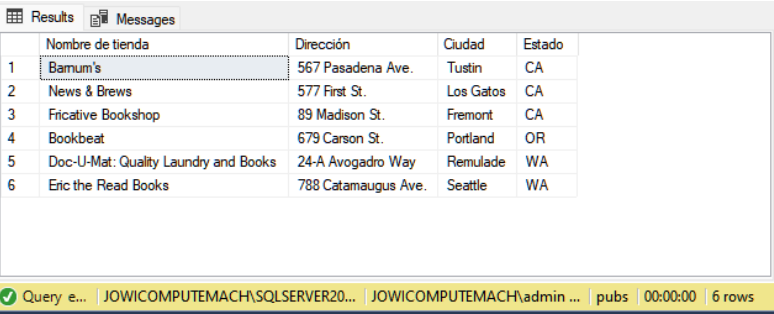
*En el caso de preguntas con respuestas escritas, las mismas deben colocarse abajo en el cuadro provisto.*

No	CONSULTA	RESULTADO	PTOS
1	Concatenar CompanyName, City de la tabla Customers	<pre>select [FirstName]+ ', ' + [LastName]+ ', ' + [Title] 'Empleado' from [dbo].[Employees]</pre>  <p>Aquí se puede observar, cómo al momento de concatenar diferentes cadenas de caracteres, se logró obtener el primer nombre y apellido de cada uno de los empleados de la organización, seguidos por una coma en la que se colocó su cargo.</p>	5
2	select city from Employees	<pre>select distinct ContactTitle from Customers</pre>  <p>Acá podremos observar las ciudades en las que habitan cada uno de los Empleados de la organización ficticia en la Base de Datos.</p>	5

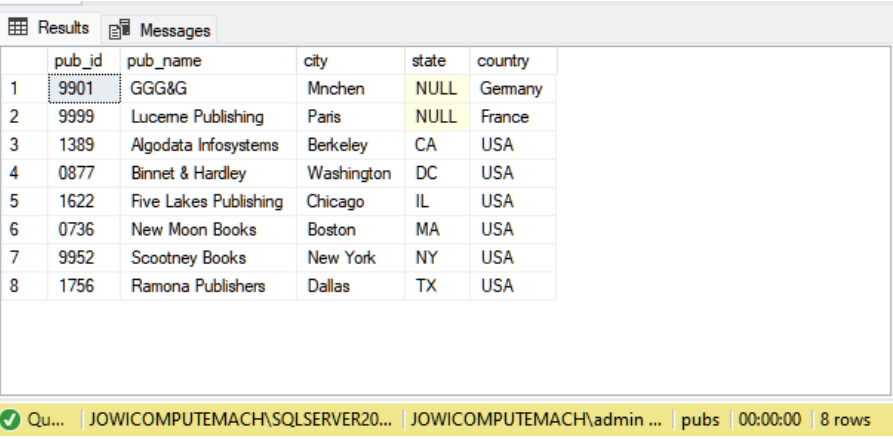
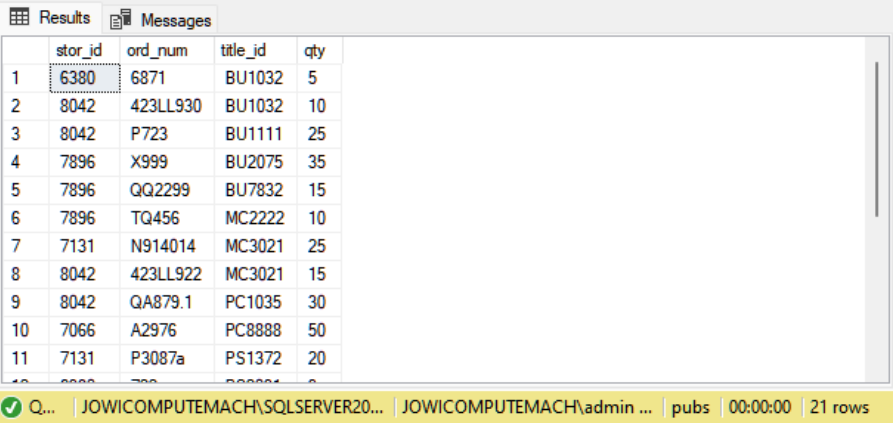
3	<p>select DISTINCT city from Employees</p>	<pre>select distinct ContactTitle       from Customers</pre>  <p><b>Explicación:</b> Es importante ver que, en este caso, a pesar de que múltiples empleados pueden tener el mismo cargo, estos no se repiten entre sí, dando solamente los cargos únicos que pueden existir dentro de la organización.</p>	5
4	<p>select Title, City, Country from Employee</p>	<pre>select ContactTitle, Country       from Customers</pre> 	5

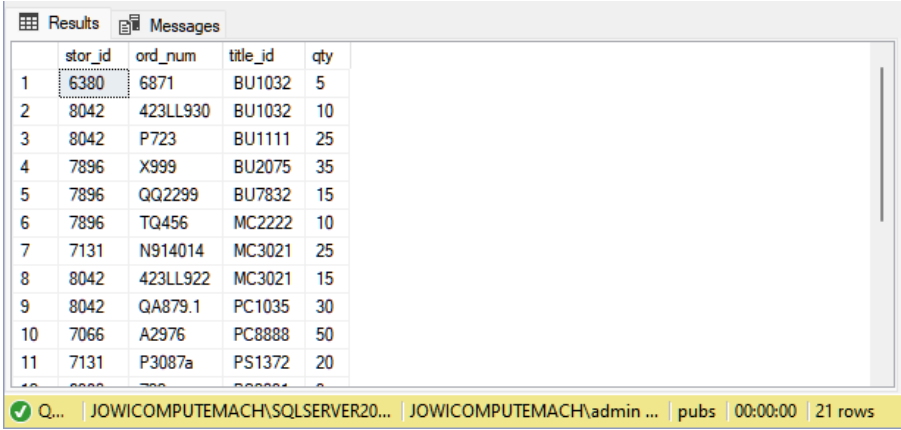
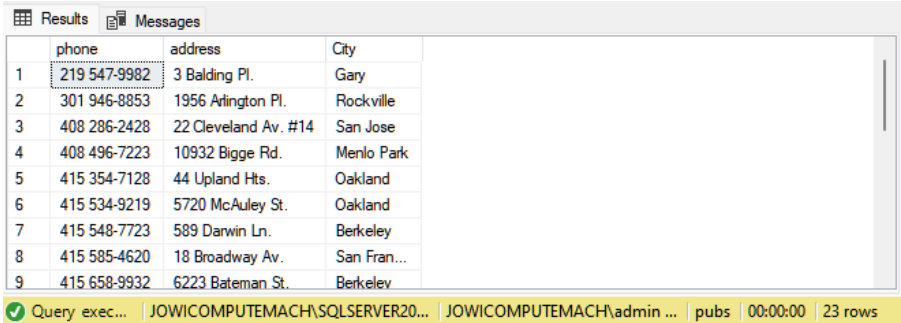
5	Adicione la clausula DISTINCT al pto 4	<pre>select distinct ContactTitle, Country from Customers</pre>  <p><b>Explicue:</b> En la consulta cuatro (4), aparecen los Títulos de cada uno de los Clients, pero estos se encuentran completamente desordenados y se repiten en sobremanera, ahora utilizaremos la función distinct, con los que los atributos nos van a aparecer en orden y no se repetirán datos.</p>	5
6	select * from Employees	<pre>select * from Employees</pre> 	5
7		<pre>select * from Employees order by LastName</pre>	5

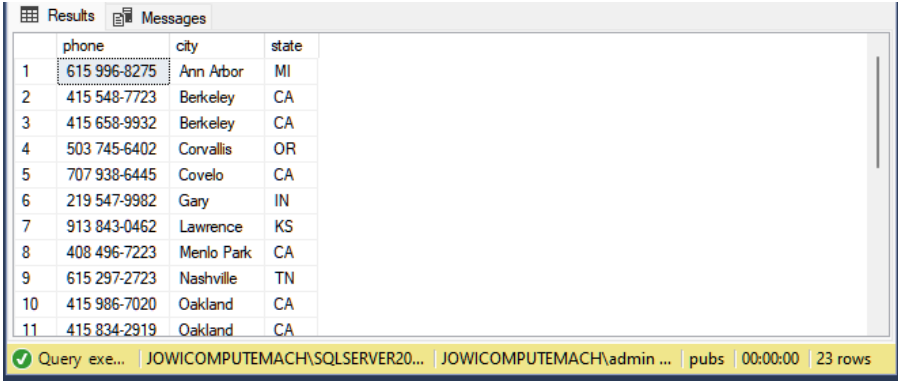
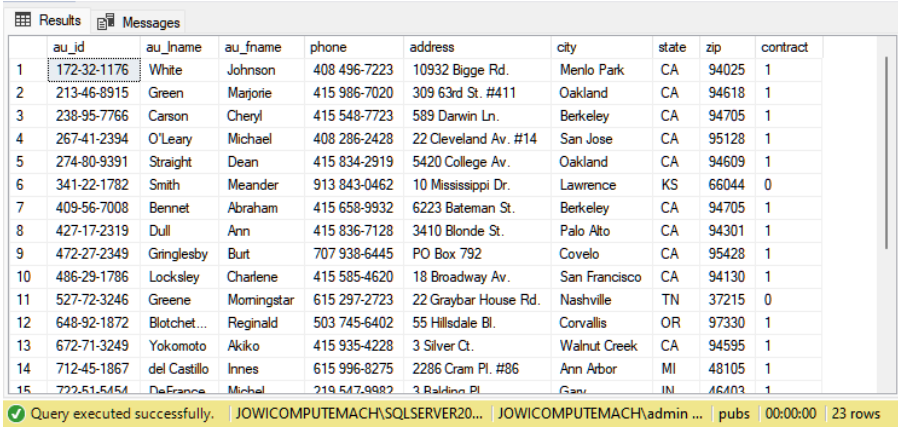
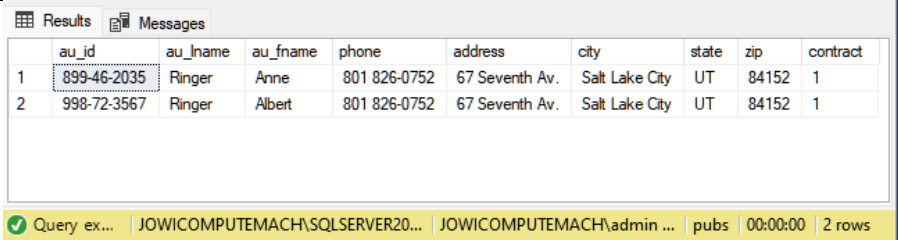
	<p>select * from Employees</p> <p>order by LastName</p>	<div><div>ResultsMessages</div><table><thead><tr><th></th><th>EmployeeID</th><th>LastName</th><th>FirstName</th><th>Title</th><th>TitleOfCourtesy</th><th>BirthDate</th><th>HireDate</th></tr></thead><tbody><tr><td>1</td><td>5</td><td>Buchanan</td><td>Steven</td><td>Sales Manager</td><td>Mr.</td><td>1955-03-04 00:00:00.000</td><td>1993-10-17</td></tr><tr><td>2</td><td>8</td><td>Callahan</td><td>Laura</td><td>Inside Sales Coordinator</td><td>Ms.</td><td>1958-01-09 00:00:00.000</td><td>1994-03-05</td></tr><tr><td>3</td><td>1</td><td>Davolio</td><td>Nancy</td><td>Sales Representative</td><td>Ms.</td><td>1948-12-08 00:00:00.000</td><td>1992-05-01</td></tr><tr><td>4</td><td>9</td><td>Dodsworth</td><td>Anne</td><td>Sales Representative</td><td>Ms.</td><td>1966-01-27 00:00:00.000</td><td>1994-11-15</td></tr><tr><td>5</td><td>2</td><td>Fuller</td><td>Andrew</td><td>Vice President, Sales</td><td>Dr.</td><td>1952-02-19 00:00:00.000</td><td>1992-08-14</td></tr><tr><td>6</td><td>7</td><td>King</td><td>Robert</td><td>Sales Representative</td><td>Mr.</td><td>1960-05-29 00:00:00.000</td><td>1994-01-02</td></tr><tr><td>7</td><td>3</td><td>Leverling</td><td>Janet</td><td>Sales Representative</td><td>Ms.</td><td>1963-08-30 00:00:00.000</td><td>1992-04-01</td></tr><tr><td>8</td><td>4</td><td>Peacock</td><td>Margaret</td><td>Sales Representative</td><td>Mrs.</td><td>1937-09-19 00:00:00.000</td><td>1993-05-03</td></tr><tr><td>9</td><td>6</td><td>Suyama</td><td>Michael</td><td>Sales Representative</td><td>Mr.</td><td>1963-07-02 00:00:00.000</td><td>1993-10-17</td></tr></tbody></table><div>Query...JOWICOMPUTEMACH\SQLSERVER20...JOWICOMPUTEMACH\admin...Northwind00:00:009 rows</div><p>¿Qué les ocurrió a los datos? ¿Cómo se ordenaron?</p><p>R/. En la pregunta seis (6), los datos se encontraban ordenados según el EmployeeID, pero ahora se estableció una consulta a través de la cual estos fuesen ordenados según el LastName.</p></div>		EmployeeID	LastName	FirstName	Title	TitleOfCourtesy	BirthDate	HireDate	1	5	Buchanan	Steven	Sales Manager	Mr.	1955-03-04 00:00:00.000	1993-10-17	2	8	Callahan	Laura	Inside Sales Coordinator	Ms.	1958-01-09 00:00:00.000	1994-03-05	3	1	Davolio	Nancy	Sales Representative	Ms.	1948-12-08 00:00:00.000	1992-05-01	4	9	Dodsworth	Anne	Sales Representative	Ms.	1966-01-27 00:00:00.000	1994-11-15	5	2	Fuller	Andrew	Vice President, Sales	Dr.	1952-02-19 00:00:00.000	1992-08-14	6	7	King	Robert	Sales Representative	Mr.	1960-05-29 00:00:00.000	1994-01-02	7	3	Leverling	Janet	Sales Representative	Ms.	1963-08-30 00:00:00.000	1992-04-01	8	4	Peacock	Margaret	Sales Representative	Mrs.	1937-09-19 00:00:00.000	1993-05-03	9	6	Suyama	Michael	Sales Representative	Mr.	1963-07-02 00:00:00.000	1993-10-17	
	EmployeeID	LastName	FirstName	Title	TitleOfCourtesy	BirthDate	HireDate																																																																												
1	5	Buchanan	Steven	Sales Manager	Mr.	1955-03-04 00:00:00.000	1993-10-17																																																																												
2	8	Callahan	Laura	Inside Sales Coordinator	Ms.	1958-01-09 00:00:00.000	1994-03-05																																																																												
3	1	Davolio	Nancy	Sales Representative	Ms.	1948-12-08 00:00:00.000	1992-05-01																																																																												
4	9	Dodsworth	Anne	Sales Representative	Ms.	1966-01-27 00:00:00.000	1994-11-15																																																																												
5	2	Fuller	Andrew	Vice President, Sales	Dr.	1952-02-19 00:00:00.000	1992-08-14																																																																												
6	7	King	Robert	Sales Representative	Mr.	1960-05-29 00:00:00.000	1994-01-02																																																																												
7	3	Leverling	Janet	Sales Representative	Ms.	1963-08-30 00:00:00.000	1992-04-01																																																																												
8	4	Peacock	Margaret	Sales Representative	Mrs.	1937-09-19 00:00:00.000	1993-05-03																																																																												
9	6	Suyama	Michael	Sales Representative	Mr.	1963-07-02 00:00:00.000	1993-10-17																																																																												
8	<p>Ordene los datos del punto 7 de manera descendente utilizando el order by nombre_columna_ordenar desc</p> <p>Anote sus resultados</p>	<div><div>ResultsMessages</div><table><thead><tr><th></th><th>EmployeeID</th><th>LastName</th><th>FirstName</th><th>Title</th><th>TitleOfCourtesy</th><th>BirthDate</th><th>HireDate</th></tr></thead><tbody><tr><td>1</td><td>9</td><td>Dodsworth</td><td>Anne</td><td>Sales Representative</td><td>Ms.</td><td>1966-01-27 00:00:00.000</td><td>1994-11-15</td></tr><tr><td>2</td><td>8</td><td>Callahan</td><td>Laura</td><td>Inside Sales Coordinator</td><td>Ms.</td><td>1958-01-09 00:00:00.000</td><td>1994-03-05</td></tr><tr><td>3</td><td>7</td><td>King</td><td>Robert</td><td>Sales Representative</td><td>Mr.</td><td>1960-05-29 00:00:00.000</td><td>1994-01-02</td></tr><tr><td>4</td><td>6</td><td>Suyama</td><td>Michael</td><td>Sales Representative</td><td>Mr.</td><td>1963-07-02 00:00:00.000</td><td>1993-10-17</td></tr><tr><td>5</td><td>5</td><td>Buchanan</td><td>Steven</td><td>Sales Manager</td><td>Mr.</td><td>1955-03-04 00:00:00.000</td><td>1993-10-17</td></tr><tr><td>6</td><td>4</td><td>Peacock</td><td>Margaret</td><td>Sales Representative</td><td>Mrs.</td><td>1937-09-19 00:00:00.000</td><td>1993-05-03</td></tr><tr><td>7</td><td>3</td><td>Leverling</td><td>Janet</td><td>Sales Representative</td><td>Ms.</td><td>1963-08-30 00:00:00.000</td><td>1992-04-01</td></tr><tr><td>8</td><td>2</td><td>Fuller</td><td>Andrew</td><td>Vice President, Sales</td><td>Dr.</td><td>1952-02-19 00:00:00.000</td><td>1992-08-14</td></tr><tr><td>9</td><td>1</td><td>Davolio</td><td>Nancy</td><td>Sales Representative</td><td>Ms.</td><td>1948-12-08 00:00:00.000</td><td>1992-05-01</td></tr></tbody></table><div>Query...JOWICOMPUTEMACH\SQLSERVER20...JOWICOMPUTEMACH\admin...Northwind00:00:009 rows</div><p>Para ordenar los resultados de manera descendente, se procedió a utilizar order by 1 desc.</p></div>		EmployeeID	LastName	FirstName	Title	TitleOfCourtesy	BirthDate	HireDate	1	9	Dodsworth	Anne	Sales Representative	Ms.	1966-01-27 00:00:00.000	1994-11-15	2	8	Callahan	Laura	Inside Sales Coordinator	Ms.	1958-01-09 00:00:00.000	1994-03-05	3	7	King	Robert	Sales Representative	Mr.	1960-05-29 00:00:00.000	1994-01-02	4	6	Suyama	Michael	Sales Representative	Mr.	1963-07-02 00:00:00.000	1993-10-17	5	5	Buchanan	Steven	Sales Manager	Mr.	1955-03-04 00:00:00.000	1993-10-17	6	4	Peacock	Margaret	Sales Representative	Mrs.	1937-09-19 00:00:00.000	1993-05-03	7	3	Leverling	Janet	Sales Representative	Ms.	1963-08-30 00:00:00.000	1992-04-01	8	2	Fuller	Andrew	Vice President, Sales	Dr.	1952-02-19 00:00:00.000	1992-08-14	9	1	Davolio	Nancy	Sales Representative	Ms.	1948-12-08 00:00:00.000	1992-05-01	5
	EmployeeID	LastName	FirstName	Title	TitleOfCourtesy	BirthDate	HireDate																																																																												
1	9	Dodsworth	Anne	Sales Representative	Ms.	1966-01-27 00:00:00.000	1994-11-15																																																																												
2	8	Callahan	Laura	Inside Sales Coordinator	Ms.	1958-01-09 00:00:00.000	1994-03-05																																																																												
3	7	King	Robert	Sales Representative	Mr.	1960-05-29 00:00:00.000	1994-01-02																																																																												
4	6	Suyama	Michael	Sales Representative	Mr.	1963-07-02 00:00:00.000	1993-10-17																																																																												
5	5	Buchanan	Steven	Sales Manager	Mr.	1955-03-04 00:00:00.000	1993-10-17																																																																												
6	4	Peacock	Margaret	Sales Representative	Mrs.	1937-09-19 00:00:00.000	1993-05-03																																																																												
7	3	Leverling	Janet	Sales Representative	Ms.	1963-08-30 00:00:00.000	1992-04-01																																																																												
8	2	Fuller	Andrew	Vice President, Sales	Dr.	1952-02-19 00:00:00.000	1992-08-14																																																																												
9	1	Davolio	Nancy	Sales Representative	Ms.	1948-12-08 00:00:00.000	1992-05-01																																																																												

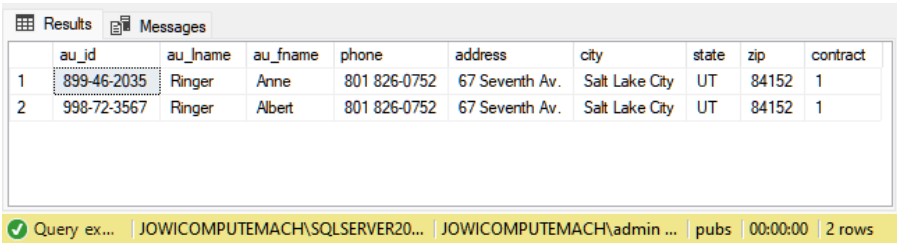
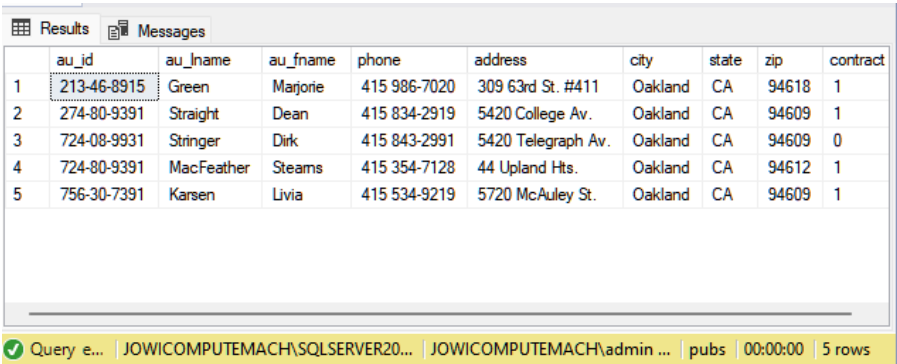
9	<p>Realice la consulta de todos los campos de la tabla Customers de la BD Northwind, pero muéstrela ordenada ascendente por ContactTitle</p>	<pre>Select * from [dbo].[Customers]  order by 4 asc</pre>  <p>Se procede a ordenar los datos a través de order by 4 asc, lo cual correspondía al ordenamiento con el espacio de la tabla ContactTitle, lo que permite que ahora se pueda apreciar de manera ascendente.</p>	5
10	<p>Liste stor_name, address, city, state de la tabla stores ordenados por la tercera posición</p>	<pre>Select [stor_name] 'Nombre de tienda',        [stor_address] 'Dirección',        [city] 'Ciudad',        [state] 'Estado' from [dbo].[stores]  order by 4</pre>  <p>Como se observa la columna 3 fue la que se dio preferencia de ordenamiento de forma ascendente.</p>	5

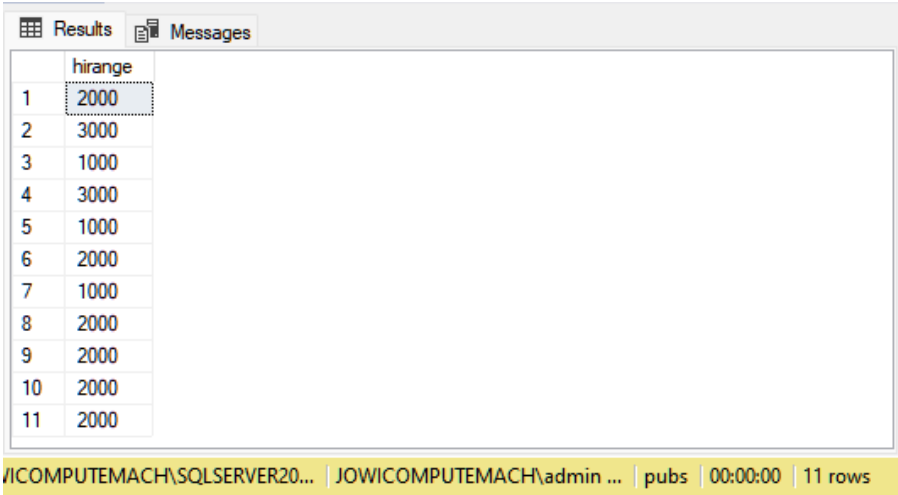
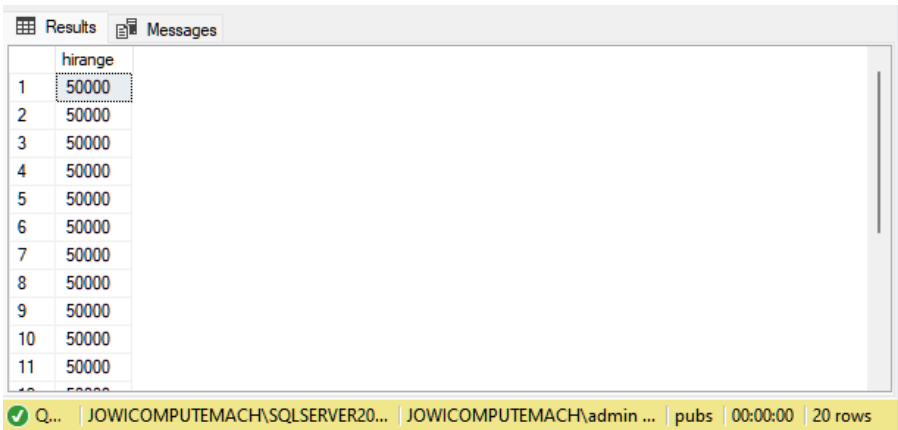


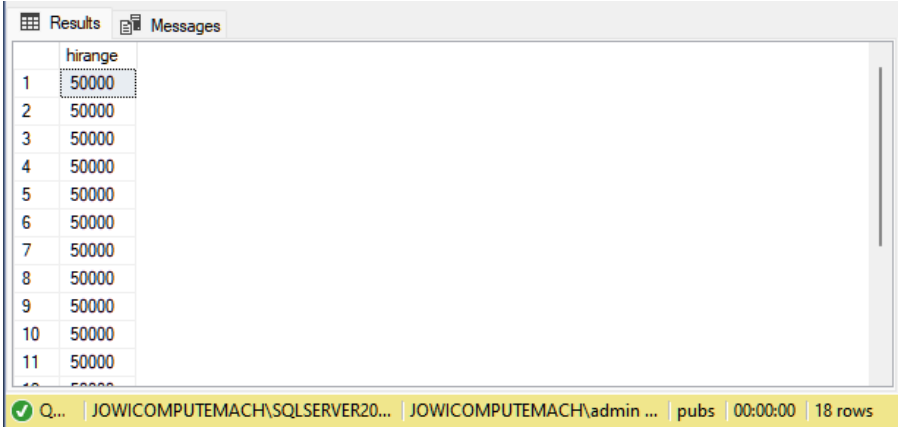
11	<p>Liste todos los atributos de la tabla publishers y aplique la ordenación por la posición 4.</p>	<pre> Select *     From [dbo].[publishers]     order by 4         </pre>  <p><b>Explique:</b> Lo que se ve en la columna de estate, es que primeramente se coloca los null que no se rellenaron y luego se procede a ordenar de manera ascendente.</p>	5
12	<p>Muestre de la tabla sales los atributos stor_id, ord_num, title_id, qty, ordenados por el atributo title_id.</p>	<pre> Select [stor_id],         [ord_num],         [title_id],         [qty]     From [dbo].[sales]     Order by 3         </pre> 	5

13	Repita la consulta del punto 12, pero ordene por los atributos title_id y stor_id.	<pre>Select [stor_id],         [ord_num],         [title_id],         [qty] From [dbo].[sales]  Order by 3,1</pre>  <p>Explique: Al poner primeramente la columna 1 y luego la columna 3, Se refleja que primeramente ordena los números de la columna 1 y en la columna 3 lo ordena a medida que se repiten los números, como se puede observar.</p>	5
14	Realice dos consultas ordenadas por más de una columna.	<pre>Select [phone],         [address],         [City] From [dbo].[authors] order by 1,3</pre> 	5

		<pre>Select [phone],         [city],         [state] From [dbo].[authors] order by 2, 3</pre> 	
15	Liste todos los campos de la tabla authors	<pre>Select* from [dbo].[authors]</pre>  <p>¿Cuántos autores despliega?</p> <p>R/. En total, la Base de Datos Pubs, despliega 23 autores que están organizados en 23 filas, al momento de hacer el query, respectivamente.</p>	5
16	Muestre los autores cuyo apellido		5

	<p>au_lname = 'Ringer'</p>	<pre>Select* from [dbo].[authors] where au_lname = 'Ringer'</pre> <p>¿Qué paso?</p> <p>R./. Cuando utilizamos la función where dentro de SQL, el query va a ser que exclusivamente sean llamados aquellos autores que su au_lname sea Ringer, para este caso muy en particular.</p>	
	<p>Where au_lname = 'RINGER'</p>	<pre>Select* from [dbo].[authors] where au_lname = 'RINGER'</pre>  <p>¿Qué paso, que datos trajo?</p> <p>R./. Al momento de colocarlo todo en mayúscula, no existió ningún tipo de cambio, ya que la función where no es CAPS sensitive.</p>	
17	<p>Liste los autores que viven en la ciudad de Oakland.</p>	<pre>Select * from authors where city = 'Oakland'</pre> 	5

18	Tabla roysched, hirange menor que 4000.	<pre>Select [hirange] from [dbo].[roysched] where [hirange] &lt; 4000</pre> 	5
19	hirange mayor que 30000.	<pre>Select [hirange] from [dbo].[roysched] where [hirange] &gt; 30000</pre> 	5

20	hirange mayor e igual a 40000.	<pre>Select [hirange] from [dbo].[roysched] where [hirange] &gt;= 40000</pre> 	5
----	--------------------------------------	--	---

## G. CONSIDERACIONES FINALES:

R/. Algo que me llamó supremamente la atención en este Laboratorio #4, es la capacidad que tiene Microsoft SQL Server para poder, así como otros lenguajes de programación más tradicionales como “Python, C, C++, etc”, de realizar concatenaciones entre datos de tipo entero (int), decimales (double o float) con datos de tipo cadena de caracteres), lo que permite presentar resultados a las consultas que se nos están solicitando, de una manera muchísimo más rápida y a la vez dinámica, lo que ayuda muchísimo para futuras implementaciones donde se desee imprimir en pantalla cierto dato almacenado.

## H. BIBLIOGRAFIA:

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

## I. RÚBRICA:

- Cada uno de los problemas es ponderado con 5 puntos, los cuales hacen un total de 100 puntos.

**Fecha de entrega:** indicada por el profesor.