**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

1. **TÍTULO** **DE LA EXPERIENCIA:**

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

1. **TEMAS :**
2. Concatenar dos o más atributos (Operador +)
3. Control de filas duplicadas (distinct)
4. Salida ordenada ascendente o descendentemente (order by)
5. Salida controlada por clausula Where
6. **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.

1. **METODOLOGÍA:**

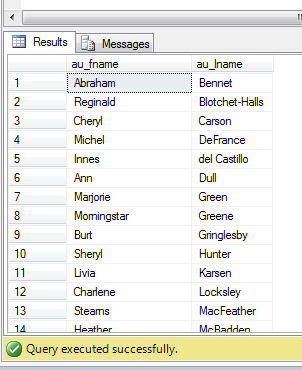
Para presentar el informe de los resultados obtenidos, haga captura de pantalla desde el SQL Server mostrando la instrucción y el resultado de la consulta generada por la misma. Coloque además el código usado en forma de texto de forma directa y no como parte de la captura de pantalla.

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.

1. **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

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.

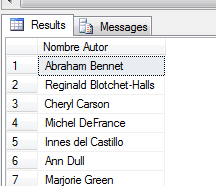
*El literal puede ser cualquier carácter, expresión o número, incluido entre las comillas*.

**select au\_fname + ' ' + au\_lname 'Nombre Autor'**

**from authors**

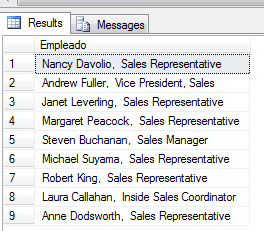
Literal Blanco adicionado para que no imprima pegado el nombre y el apellido

El **+** es el operador que realiza la concatenación

****

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 operaciòn 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.**

**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.

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

**select** ContactTitle

**from** Customers

**(Observe que se presentan campos repetidos )**

1. **Ahora adicione la clausula DISTINCT o distinct después del select**

**select distinct** ContactTitle

**from** Customers

**¿Qué ocurre? Imprima y comente sus resultados.**

1. **Realice y muestre la siguiente consulta**

**select** ContactTitle, Country

**from** Customers

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

El DISTICT puede aplicarse a multiples columnas, en cuyo caso: **explique qué suscede.**

1. **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.

SELECT [ \* ó columna ó grupo de columnas]

FROM Nombre\_Tabla

ORDER BY columna [asc ó desc]

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

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

**select \***

**from** Employees

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

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

select \*

from Employees

order by LastName

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

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

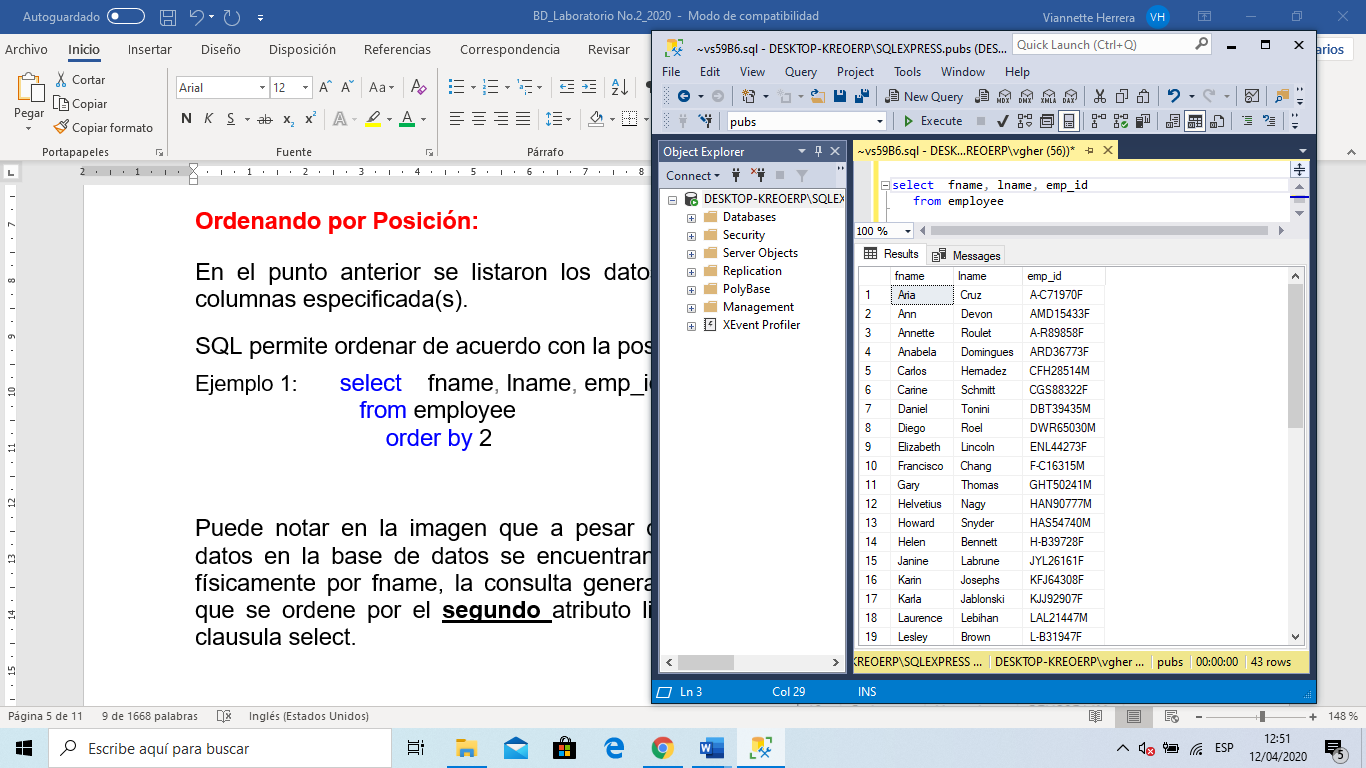
***order by nombre\_columna\_a\_ordenar desc***

***Anote sus resultados***

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

**Active la base de datos PUBS**

**Ordenando por Posición:**

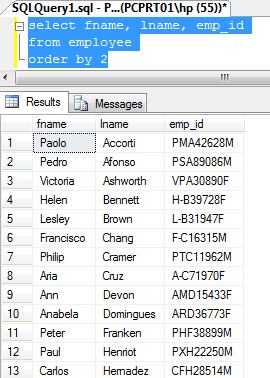
En el punto anterior se listaron los datos ordenados de acuerdo con alguna columna o columnas especificada(s).

Veamos el resultado de la consulta

select fname, lname, emp\_id

from employee

*A pesar de que no se 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 mustra ordenado)*

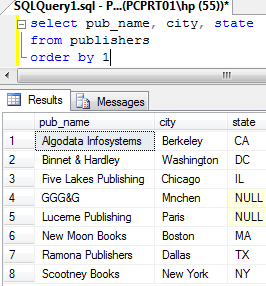
SQL Server nos permite ordenar de acuerdo con la posición que ocupe el campo en la lista Select.

Ejemplo 1: select fname, lname, emp\_id

from employee

order by 2

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.

1. **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**

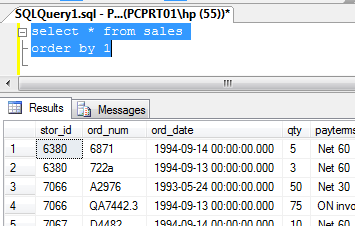
**obtenido.**

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

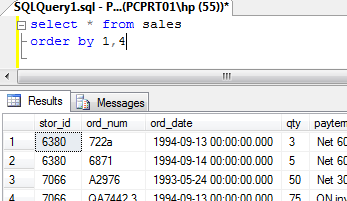
**Ordenando por Varias Columnas:**

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.

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.

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

1. **Ahora repita la consulta del punto 12, pero ordene por los atributos title\_id y stor\_id. Comente sus resultados**
2. **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.

**SELECT expresión**

**FROM tabla**

**WHERE condición (es)**

**[Order by]**

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

El formato es el siguiente:

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

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

**En este laboratorio veremos los operadores lógicos:**

1. **Liste todos los campos de la tabla authors. ¿Cuántos autores despliega?**
2. **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**

1. **Ahora liste sólo los autores que viven en la ciudad de Oakland.**
2. **Despliegue la tabla roysched. Muestre los resultados de la consulta que le permiten obtener los datos de un hirange menor que 4000.**
3. **Muestre los resultados de las consultas que le permiten obtener los datos de un hirange mayor que 30000.**
4. **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, curso de Base de Datos I.

1. **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 | select [FirstName]+ ', ' + [LastName]+ ', ' + [Title] 'Empleado'  from [dbo].[Employees]    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. | *5* |
| *2* | select city from Employees | select distinct ContactTitle  from Customers    Acá podremos observar las ciudades en las que habitan cada uno de los Empleados de la organización ficticia en la Base de Datos. | *5* |
| *3* | select DISTINCT city from Employees | select distinct ContactTitle  from Customers | *5* |
| **Explicación**: 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. |
| *4* | select Title, City, Country from Employee | **select** ContactTitle, Country  **from** Customers | *5* |
| *5* | Adicione la clausula DISTINCT al pto 4 | select distinct ContactTitle, Country  from Customers | *5* |
| **Explique**: 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. |
| *6* | select \* from Employees | select \*  from Employees | *5* |
| *7* | select \* from Employees  order by LastName | select \* from Employees  order by LastName | *5* |
| **¿Qué les ocurrió a los datos? ¿Cómo se ordenaron?**  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. |
| *8* | Ordene los datos del punto 7 de manera descendente utilizando el order by nombre\_columna\_a\_ordenar desc  Anote sus resultados | select \* from Employees  order by 1 desc    Para ordenar los resultados de manera descendente, se procedió a utilizar order by 1 desc. | *5* |
|  |  |  |  |
| *9* | Realice la consulta de todos los campos de la tabla Customers de la BD Northwind, pero muéstrela ordenada ascendentemente por ContactTitle | Select \* from [dbo].[Customers]  order by 4 asc    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. | *5* |
| *10* | Liste stor\_name, address, city, state de la tabla stores ordenados por la tercera posición | Select [stor\_name]'Nombre de tienda',  [stor\_address]'Dirección',  [city]'Ciudad',  [state]'Estado'  from [dbo].[stores]  order by 4    *Como se observa la columna 3 fue la que se dio preferencia de ordenamiento de forma ascendente.* | *5* |
| *11* | Liste todos los atributos de la tabla publishers y aplique la ordenación por la posición 4. | Select \*  From [dbo].[publishers]  order by 4 | *5* |
| **Explique**: 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. |
| *12* | Muestre de la tabla sales los atributos stor\_id, ord\_num, title\_id, qty , ordenados por el atributo title\_id. | Select [stor\_id],  [ord\_num],  [title\_id],  [qty]  From [dbo].[sales]  Order by 3 | *5* |
| *13* | Repita la consulta del punto 12, pero ordene por los atributos title\_id y stor\_id. | Select [stor\_id],  [ord\_num],  [title\_id],  [qty]  From [dbo].[sales]  Order by 3,1 | *5* |
| *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.* |
| *14* | Realice dos consultas ordenadas por más de una columna. | Select [phone],  [address],  [City]  From [dbo].[authors]  order by 1,3 | *5* |
| Select [phone],  [city],  [state]  From [dbo].[authors]  order by 2, 3 |
| *15* | Liste todos los campos de la tabla authors | Select\*  from [dbo].[authors] | *5* |
| ¿Cuántos autores despliega?  R/. En total, la Base de Datos Pubs, despliega 23 autores que están organizados en 23 filas, al momento de hacer el query, respectivamente. |
| *16* | Muestre los autores cuyo apellido au\_lname = 'Ringer' |  | *5* |
| Select\*  from [dbo].[authors]  where au\_lname = 'Ringer'  *¿Qué paso?*  *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.* |
| Where au\_lname = 'RINGER' | Select\*  from [dbo].[authors]  where au\_lname = 'RINGER' |
| *¿Qué paso, que datos trajo?*  *R/. Al momento de colocarlo todo en mayúscula, no existió ningún tipo de cambio, ya que la función where no es CAPS sensitive.* |
| *17* | Liste los autores que viven en la ciudad de Oakland. | Select \*  from authors  where city = 'Oakland' | *5* |
| *18* | Tabla roysched, hirange menor que 4000. | Select [hirange]  from [dbo].[roysched]  where [hirange] < 4000 | *5* |
| *19* | hirange mayor que 30000. | Select [hirange]  from [dbo].[roysched]  where [hirange] > 30000 | *5* |
| *20* | hirange mayor e igual a 40000. | Select [hirange]  from [dbo].[roysched]  where [hirange] >= 40000 | *5* |

1. **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++, étc”, 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.

1. **BIBLIOGRAFIA:**

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

1. **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.