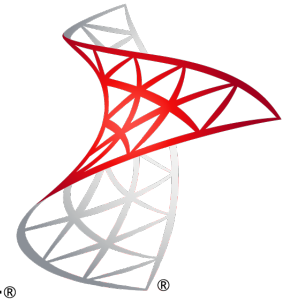
UCR

Tema 4:

Ordenamiento y Filtrado de Datos



Microsoft® SQL Server® 2012

Programa de Asesorías, Actualización y Capacitación Computacional

Alexandra Martínez y Diego Quirós

UCR

Agenda

1. Ordenamiento de datos con ORDER BY
2. Filtrado de datos con cláusula WHERE
3. Filtrado mediante la opción TOP
4. Filtrado mediante la opción OFFSET-FETCH
5. Trabajar con valores desconocidos y perdidos

UCR

4.1 ORDENAMIENTO DE DATOS CON ORDER BY

UCR

ORDER BY: Ejemplos

Ordenamiento por nombre de columna

```
SELECT SalesOrderID, CustomerID, OrderDate
FROM   Sales.SalesOrderHeader
ORDER BY OrderDate;
```

Ordenamiento por alias de columna

```
SELECT SalesOrderID, CustomerID,
       YEAR(OrderDate) AS OrderYear
FROM   Sales.SalesOrderHeader
ORDER BY OrderYear;
```

Ordenamiento descendente

```
SELECT SalesOrderID, CustomerID, OrderDate
FROM   Sales.SalesOrderHeader
ORDER BY OrderDate DESC;
```

UCR

Ejercicios 4.1

a) Suponga que usted necesita generar un reporte donde se muestren las columnas *Group* y *Name* de la tabla *Sales.SalesTerritory*, así como todas las filas de esa tabla

   i. ¿Qué consulta usaría si tuviera que generar el reporte ordenado ascendentemente por la columna *Group*?

   ii. ¿Qué consulta usaría si tuviera que generar el reporte ordenado ascendentemente por la columna *Group* y descendentemente por la columna *Name*?

   iii. ¿Qué consulta usaría si tuviera que generar el reporte ordenado descendentemente por ambas columnas (*Group* y *Name*)?

UCR

Ejercicios 4.1

b) Suponga que usted necesita hacer una consulta donde se muestren las columnas *Color* y *ListPrice* de la tabla *Production.Product*, ordenadas ascendentemente por *Color* y aquellas filas que tienen el mismo valor para *Color*, se ordenen ascendentemente por *ListPrice*

i. ¿Cómo escribiría esta consulta?

ii. ¿Cómo quedan ordenadas las filas que contienen valores NULL?

iii. Si solo se ordenara el resultado por *Color* ¿qué sucedería con las filas que tienen el mismo valor para *Color* (en qué orden se mostrarían)?

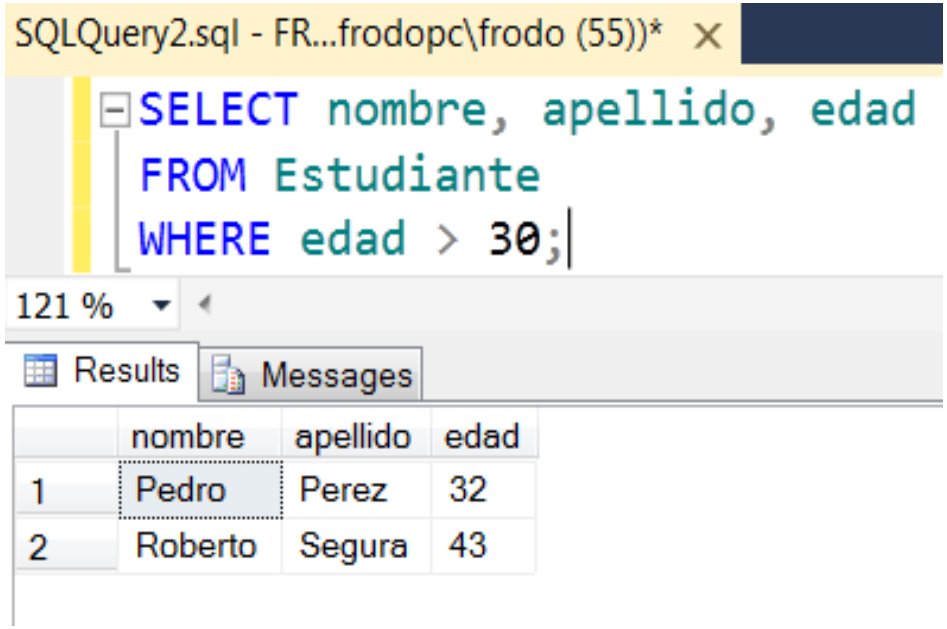
UCR

4.2 FILTRADO DE DATOS CON CLÁUSULA WHERE

UCR

Filtrado mediante WHERE

- La cláusula WHERE especifica predicados para filtrar las filas devueltas en el resultado



UCR

Uso de Predicados para Filtrar

- La cláusula WHERE usa predicados, que se expresan como condiciones lógicas
- Las filas para las que el predicado evalúa a **TRUE** son aceptadas y salen en el resultado de la consulta
- Las filas para las que el predicado evalúa a **FALSE** o **UNKNOWN** se dejan por fuera (no salen en resultado)
- Un predicado evalúa a **UNKNOWN** si uno de los valores que está siendo comparado es **NULL**

UCR

WHERE: Ejemplos

Devuelve clientes de territorio 6 (filtrado por operador =)

```
SELECT CustomerID, TerritoryID
FROM   Sales.Customer
WHERE  TerritoryID = 6;
```

Dvuelve clientes de territorios mayores o iguales a 5 (filtrado por operador >=)

```
SELECT CustomerID, TerritoryID
FROM   Sales.Customer
WHERE  TerritoryID >= 5;
```

Los predicados pueden incluir operadores de comparación tales como =, >, <, <=, >=, y <> (distinto).

UCR

WHERE no ve Alias

- La cláusula WHERE no puede “ver” alias que hayan sido declarados en la cláusula SELECT

- ¿Por qué?

  - De acuerdo al orden de procesamiento lógico de una consulta, para el momento en que el WHERE es procesado, el SELECT aún no se ha procesado y por lo tanto los alias aún no existen

UCR

Combinación de Predicados

- Es posible combinar predicados para crear filtros más avanzados usando los operadores lógicos AND y OR
- Se pueden negar predicados con el operador NOT
  - La negación de TRUE y FALSE es FALSE y TRUE, respectivamente
  - La negación de UNKNOWN sigue siendo UNKNOWN
- Precedencia de operadores lógicos: NOT, AND, OR

UCR

Ejemplos que combinan predicados

Devuelve clientes de territorio 6 o 7 cuya tienda no es 1972

```
SELECT CustomerID, TerritoryID, StoreID
FROM   Sales.Customer
WHERE  (TerritoryID=6 OR TerritoryID=7)
       AND StoreID <> 1972;
```

Devuelve clientes cuya tienda sea mayor o igual que 1000 y menor o igual que 1200 (rango)

```
SELECT CustomerID, TerritoryID, StoreID
FROM   Sales.Customer
WHERE  StoreID >= 1000 AND StoreID <= 1200;
```

UCR

Predicado IN

- El predicado IN permite verificar si un valor (o expresión escalar) es igual a al menos uno de los elementos de un conjunto

```
SELECT CustomerID, TerritoryID
FROM   Sales.Customer
WHERE  TerritoryID IN (2,5,6);
```

UCR

Predicado BETWEEN

- El predicado BETWEEN permite verificar si un valor se encuentra dentro de un rango dado
- Incluye los dos valores especificados en los extremos (bordes) del rango

```
SELECT CustomerID, TerritoryID, StoreID
FROM   Sales.Customer
WHERE  StoreID BETWEEN 1000 AND 1200;
```

UCR

Predicado LIKE

- El predicado LIKE permite verificar si una hilera de caracteres satisface (cumple) un patrón especificado
- el símbolo % representa cero o más caracteres y el símbolo _ representa exactamente un caracter

```
SELECT ProductID, Name
FROM   Production.Product
WHERE  Name LIKE 'M%R_';
```

UCR

Ejercicios 4.2

a) Suponga que usted escribe la siguiente consulta:

```
SELECT CustomerID, TerritoryID AS Lugar
FROM   Sales.Customer
WHERE  Lugar = 7;
```

Y la consulta falla cuando se ejecuta en SSMS.

- Explique por qué falla la consulta.

- Modifique el código de la consulta para que pueda ser ejecutado exitosamente.

UCR

Ejercicios 4.2

b) Suponga que usted necesita mostrar el ID de la oferta, la descripción y la categoría de aquellas ofertas cuya categoría es *Reseller* o *Customer*. Esta información está disponible en la tabla *Sales.SpecialOffer.*

i. Ofrezca dos soluciones (consultas) equivalentes para este problema

ii. ¿Cómo cambiaría sus soluciones para mostrar solo aquellas ofertas cuya categoría no sea *Reseller*?

UCR

Ejercicios 4.2

c) Escriba una consulta que liste el nombre y apellido de las personas (tabla *Person.Person*) cuyo apellido inicia con la letra C.

d) Escriba una consulta que liste el nombre y apellido de las personas cuyo apellido tiene cuatro letras.

e) Escriba una consulta que liste el nombre y apellido de las personas (tabla *Person.Person*) cuyo apellido inicia con las letras T,U,V,W,X,Y o Z y termina con la letra N.

f) Escriba una consulta que liste el nombre y apellido de las personas (tabla *Person.Person*) cuyo apellido no contiene la letra A.

4.3 FILTRADO MEDIANTE LA OPCIÓN TOP

UCR

Filtrado mediante opción TOP

- La opción TOP filtra un cierto número o porcentaje de las filas resultantes de una consulta con base en un ordenamiento especificado

- TOP se basa en la especificación del orden de las tuplas para hacer su función (siempre debe ir acompañado de la cláusula ORDER BY)

UCR

Tres versiones de TOP

- TOP (***n***)

  Muestra sólo las primeras ***n*** filas del resultado

- TOP (***n*** **PERCENT**)

  Muestra sólo el primer ***n*%** de las filas (redondeado hacia arriba) del resultado

- TOP (***n***) **WITH TIES**

  Muestra las primeras ***n*** filas con valor distinto en el campo de ordenamiento

UCR

TOP: Ejemplos

Despliega las 20 primeras filas

```
SELECT TOP (20) SalesOrderID, OrderDate
FROM Sales.SalesOrderHeader
ORDER BY OrderDate ASC;
```

Despliega los primeros 20 montos distintos

```
SELECT TOP (20) WITH TIES SalesOrderID, OrderDate
FROM Sales.SalesOrderHeader
ORDER BY OrderDate ASC;
```

Despliega el 5% de las filas

```
SELECT TOP (5) PERCENT SalesOrderID, CustomerID, TotalDue
FROM Sales.SalesOrderHeader
ORDER BY TotalDue DESC;
```

UCR

Ejercicios 4.3

a) Suponga que usted debe escribir una consulta sobre la tabla *Production.Product* que devuelva los cinco productos más caros de subcategoría 1. La consulta debe mostrar las columnas *ProductID*, *ListPrice* y *ProductSubcategoryID.*

   i. ¿Cuál sería el código de la consulta?

   ii. ¿Cómo cambiaría la consulta y el resultado si tuviera considerar empates?

   iii. ¿Cómo cambiaría la consulta y el resultado si tuviera que devolver el 5% de las filas en lugar de las primeras 5?

UCR

4.4 FILTRADO MEDIANTE LA OPCIÓN OFFSET-FETCH

UCR

Filtrado mediante opción OFFSET-FETCH

- La opción OFFSET-FETCH filtra datos con base en un número especificado de filas y un orden (similar a TOP)
- A diferencia de TOP, OFFSET-FETCH es estándar y tiene la capacidad de saltar (desplazarse)
- Útil para paginación
- Esta opción va después de la cláusula ORDER BY (de hecho es necesario que exista un ORDER BY para poder usar la opción OFFSET-FETCH)

UCR

Opción OFFSET

- Especifica cuántas filas se desean saltar
- OFFSET no requiere de una cláusula FETCH

```
SELECT   *
FROM     Production.Product
ORDER BY ListPrice
OFFSET 30 ROWS;
```

- OFFSET 0 no se salta ninguna fila

```
SELECT   *
FROM     Production.Product
ORDER BY ListPrice
OFFSET 0 ROWS;
```

UCR

Opción FETCH

- Especifica cuántas filas se desean recuperar
- FETCH es opcional
- FETCH requiere de una cláusula OFFSET

```
SELECT   *
FROM     Production.Product
ORDER BY ListPrice
OFFSET 10 ROWS
FETCH NEXT 20 ROWS ONLY;
```

```
SELECT   *
FROM     Production.Product
ORDER BY ListPrice
OFFSET 0 ROWS
FETCH NEXT 25 ROWS ONLY;
```

UCR

Variantes válidas

- En las cláusulas OFFSET Y FETCH es posible escribir de forma intercambiable:
  - ROW o ROWS
    - ✓ FETCH NEXT 1 **ROW**
    - ✓ FETCH NEXT 10 ROWS
  - NEXT y FIRST
    - ✓ FETCH **FIRST** 20 ROW
    - ✓ FETCH NEXT 20 ROWS

UCR

Ejercicios 4.4

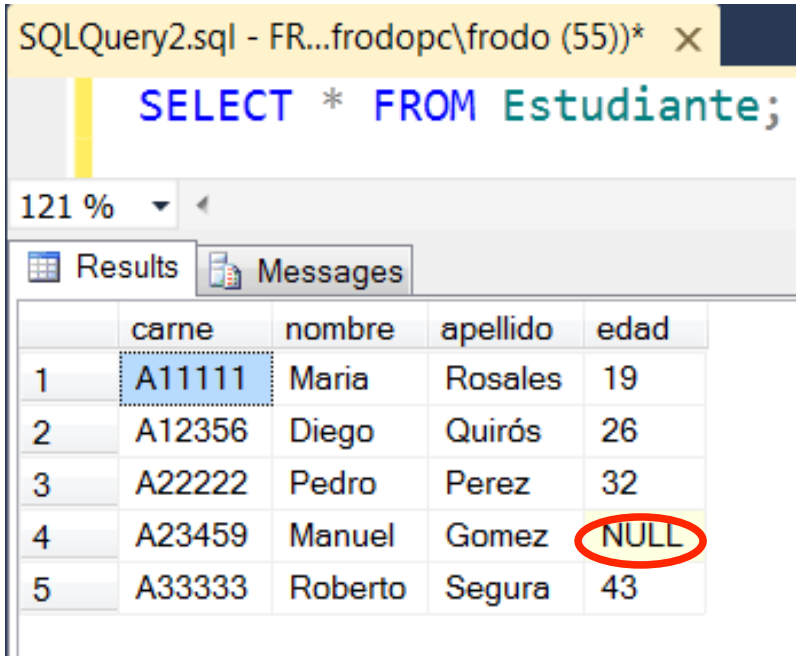
a) Escriba una consulta que recupere el nombre y apellido de las primeras 50 personas de la tabla *Person.Person*, ordenadas por Apellido y Nombre.

b) Escriba una consulta que recupere el nombre y apellido de las siguientes 50 personas (51 a 100) de la tabla *Person.Person*, ordenadas por Apellido y Nombre.

4.5 TRABAJAR CON VALORES DESCONOCIDOS Y PERDIDOS

Valores NULL

- Es posible que una consulta devuelva valores NULL.



	carne	nombre	apellido	edad
1	A11111	Maria	Rosales	19
2	A12356	Diego	Quirós	26
3	A22222	Pedro	Perez	32
4	A23459	Manuel	Gomez	NULL
5	A33333	Roberto	Segura	43

Los valores **NULL** aparecen cuando **no** se ingresa información para una columna en específico.

Los valores **NULL** representan información inexistente o desconocida

UCR

Manejo de valores NULL

- Los valores NULL son manejados de forma distinta por diferentes componentes de SQL Server:
  - Los filtros en consultas (ON, WHERE, HAVING) dejan por fuera valores UNKNOWN
  - Las restricciones CHECK aceptan valores UNKNOWN
  - Las operaciones GROUP BY y DISTINCT tratan los valores NULL como iguales

UCR

Ejemplo: Manejo de NULL en filtros

- Ej.: predicado ***salario > 100***
  - Evalúa a TRUE cuando el salario es 5000
  - Evalúa a FALSE cuando el salario es 80
  - Evalúa a UNKNOWN cuando el salario es NULL
- Si este predicado fuera el filtro en un WHERE…
  - Las filas para las cuales el predicado evalúe a TRUE se devolverían como parte del resultado
  - Las filas para las cuales el predicado evalúe a FALSE o UNKNOWN se descartarían

UCR

Resultado de ¿NULL = NULL?

- Uno de los tratamientos más interesantes de los valores NULL es cuando se comparan dos de ellos (NULL=NULL): su resultado es **UNKNOWN**
- ¿Por qué?
  - NULL representa un valor ausente o desconocido, por lo que realmente no se puede saber si un valor desconocido es igual a otro

UCR

¿Cómo saber si un valor es NULL?

- SQL provee los predicados IS NULL e IS NOT NULL
- Estos predicados deben usarse cuando se quiera probar (veriicar) si un valor es NULL o no

```
SELECT CustomerID, StoreID, TerritoryID
FROM   Sales.Customer
WHERE  StoreID IS NULL
ORDER BY TerritoryID
```

UCR

Ejercicios 4.5

a) Suponga que usted necesita escribir una consulta que devuelva las órdenes de compra que no tienen una tarjeta de crédito asociada. Esta consulta debe mostrar las columnas *SalesOrderID*, *CustomerID* y *CreditCardID.*

   i. ¿Cuál sería el código de la consulta?

   ii. ¿Cómo cambiaría la consulta si tuviera que devolver las órdenes que sí tienen una tarjeta de crédito asociada?

UCR

Ejercicios 4.5

b) Suponga que un usuario le pide buscar todos los productos de la tabla *Production.Product* cuyo *Color* no es blanco (‘*White*’).

i. Escriba una posible solución

ii. ¿Cuántos productos encontró su consulta?

iii. ¿Cuántos productos hay en total en la tabla *Product*?

iv. ¿Cuántos productos son de color blanco?

v. ¿Por qué el resultado de *iii* – *iv* no es igual a *ii* ?

vi. Ofrezca una solución alternativa tal que *iii* – *iv* = *ii*

UCR

Referencias

- Ben-Gan, I. Microsoft SQL Server 2012: T-SQL Fundamentals. Microsoft Press, O'Reilly Media, 2012.
- Ben-Gan, I., Sarka D. y Talmage, R. Querying Microsoft SQL Server 2012: Training Kit. Microsoft Press, O'Reilly Media, 2012.
- Elmasri R. y Navathe S. Fundamentos de Sistemas de Bases de Datos, 5ta ed. Pearson-Addison Wesley, 2007.
- Microsoft Virtual Academy. http://www.microsoftvirtualacademy.com