

Práctica A: Recuperación de datos y transformación de conjuntos de resultados

Objetivos

Después de realizar esta práctica, el alumno será capaz de:

- Ejecutar consultas en bases de datos con la instrucción **SELECT**.
- Ordenar los datos y eliminar los valores duplicados de un conjunto de resultados.
- Dar formato al conjunto de resultados con alias de columnas y literales.

Requisitos previos

- El archivo lab051.zip contiene los archivos de comandos necesarios para la realización de la práctica, así como las correspondientes soluciones.
- Para la realización de las prácticas es necesario crear la base de datos **library**, ejecutando el script **bldlib.cmd** del directorio **Library**.

Ejercicio 1

Recuperación de datos

En este ejercicio, va a seleccionar datos específicos de tablas de la base de datos **library**. La carpeta Soluciones contiene las secuencias de comandos completas para este ejercicio.

✍ Para seleccionar columnas específicas

En este procedimiento, va a escribir y ejecutar una instrucción **SELECT** que obtenga las columnas **title** y **title_no** de la tabla **title**.

1. Abra el Analizador de consultas de SQL Server y, si así se le pide, inicie una sesión en el servidor local con autenticación de Microsoft Windows®.
2. En la lista **BD**, haga clic en **library**.
3. Escriba y ejecute una instrucción **SELECT** que obtenga las columnas **title** y **title_no** de la tabla **title**. **Answers_Columns.sql** es el archivo de comandos completo correspondiente a este paso.

```
USE library
SELECT title, title_no
FROM title
GO
```

4. Guarde la instrucción **SELECT** como texto ANSI con la extensión de nombre de archivo **.sql**.
5. Guarde el conjunto de resultados con la extensión de nombre de archivo **.rpt**.

Resultado

Su resultado debería ser similar al siguiente conjunto de resultados parciales.

Title	title_no
-------	----------

Last of the Mohicans	1
The Village Watch-Tower	2
Self Help; Conduct & Perseverance	3
Songs of a Savoyard	4

.

.

.

.

✍ Para seleccionar filas con un operador de comparación

En este procedimiento, va a escribir y ejecutar una instrucción SELECT que obtenga datos de filas específicas mediante una cláusula WHERE con un operador de comparación. Answer_Comparison.sql es la secuencia de comandos completa de este procedimiento.

- Escriba y ejecute una instrucción SELECT que obtenga el título número 10 de la tabla **title**.

Puede ejecutar el procedimiento almacenado del sistema **sp_help** con la tabla **title** para averiguar los nombres correctos de las columnas.

```
USE library
SELECT title
FROM title
WHERE title_no = 10
GO
```

Resultado

Su resultado debería ser similar al siguiente conjunto de resultados.

title

The Night-Born
(1 fila afectada)

✍ Para seleccionar filas con un intervalo

En este procedimiento, va a escribir y ejecutar una instrucción SELECT que obtenga datos de filas específicas mediante una cláusula WHERE con un intervalo. La secuencia de comandos completa para este procedimiento es Answer_Range.sql.

- Escriba y ejecute una instrucción SELECT que obtenga los números de miembro y las penalizaciones calculadas (fine_assessed) de la tabla **loanhist** para todos los miembros con penalizaciones comprendidas entre 8,00 y 9,00 dólares.

Puede ejecutar el procedimiento almacenado del sistema **sp_help** con la tabla **loanhist** para averiguar los nombres de columna correctos.

```
USE library
SELECT member_no, fine_assessed
FROM loanhist
WHERE (fine_assessed BETWEEN $8.00 AND $9.00)
GO
```

Resultado

Su resultado debería ser similar al siguiente conjunto de resultados parciales. El número de filas que se devuelven puede variar.

member_no	fine_assessed
7399	9
7399	9
7399	9
7399	9
.	.
.	.
.	.
(312 filas afectadas)	

✍ Para seleccionar filas con una lista de valores

En este procedimiento, va a escribir y ejecutar una instrucción **SELECT** que obtenga datos de filas específicas mediante una cláusula **WHERE** que contiene una lista de valores. La secuencia de comandos completa para este procedimiento es `Answer_InList.sql`.

- Escriba y ejecute una instrucción **SELECT** que obtenga de la tabla **title** el número de título y el autor de todos los libros de Charles Dickens o Jane Austen. Utilice el operador **IN** como parte de la instrucción **SELECT**.

```
USE library
SELECT author, title_no
FROM title
WHERE author IN ('Charles Dickens','Jane Austen')
GO
```

Resultado

Su resultado debería ser similar al siguiente conjunto de resultados.

author	title_no
Jane Austen	27
Charles Dickens	30
Charles Dickens	31
Jane Austen	41
Jane Austen	43

(5 filas afectadas)

✍ Para seleccionar filas con una comparación de cadenas de caracteres

En este procedimiento, va a escribir y ejecutar una instrucción **SELECT** para obtener los datos de las filas que contengan una cadena de caracteres similar a una cadena dada. `Answer_String.sql` es la secuencia de comandos completa de este procedimiento.

- Escriba y ejecute una instrucción **SELECT** que obtenga de la tabla **title** el número de título y el título de todas las filas cuyo título contenga la cadena de caracteres “adventures”. Utilice el operador **LIKE** en la consulta.

```
USE library
SELECT title_no, title
FROM title
WHERE title LIKE ('%Adventures%')
GO
```

Resultado

Su resultado debería ser similar al siguiente conjunto de resultados.

title_no	title
-----------------	--------------

26	The Adventures of Robin Hood
44	Adventures of Huckleberry Finn

(2 filas afectadas)

✍ Para seleccionar filas que contienen valores NULL

En este procedimiento, va a escribir y ejecutar una instrucción SELECT para obtener datos de filas específicas mediante una cláusula WHERE que busca valores NULL. Answer_IsNull.sql es la secuencia de comandos completa de este procedimiento.

- Escriba y ejecute una instrucción SELECT que obtenga el número de miembro, la penalización calculada y la penalización que se debe pagar en los préstamos con penalizaciones impagadas de la tabla **loanhist**. Recupere las filas con penalizaciones en la columna **fine_assessed** y con valores NULL en la columna **fine_paid**.

USE library

SELECT member_no, fine_assessed, fine_paid

FROM loanhist

WHERE (fine_assessed IS NOT NULL) AND (fine_paid IS NULL)

GO

Resultado

Su resultado debería ser similar al siguiente conjunto de resultados. El número de filas que se devuelven puede variar.

member_no	fine_assessed	fine_paid
4645	5.0000	NULL
4240	.0000	NULL
3821	1.0000	NULL
3389	9.0000	NULL

.

.

.

(1118 filas afectadas)

Ejercicio 2

Transformación de conjuntos de resultados

En este ejercicio, va a escribir y ejecutar consultas que cambien la presentación de los datos de un conjunto de resultados. Utilizará la palabra clave **DISTINCT** para eliminar las filas duplicadas y la palabra clave **ORDER BY** para ordenar el conjunto de resultados. Además, cambiará los nombres de columna y la presentación de los datos con alias y literales.

La carpeta Soluciones contiene las secuencias de comandos completas para este ejercicio.

✍ Para eliminar las filas duplicadas del conjunto de resultados

En este procedimiento, va a escribir y ejecutar una consulta en la tabla **adult** para obtener solamente las combinaciones únicas de ciudades y estados que se encuentren en el conjunto de resultados. `Answer_Duplicates.sql` es la secuencia de comandos completa de este procedimiento.

- Escriba y ejecute una consulta que obtenga todas las parejas únicas de ciudad y estado de la tabla **adult**. En el conjunto de resultados sólo debe aparecer una fila por cada pareja de ciudad y estado.

```
USE library
SELECT DISTINCT city, state
FROM adult
GO
```

Resultado

Su resultado debería ser similar al siguiente conjunto de resultados parciales.

City	state
Salt Lake City	UT
Atlanta	GA
Tallahassee	FL
Washington	DC
.	
.	
.	
(23 filas afectadas)	

✍ Para ordenar los datos

En este procedimiento, va a escribir y ejecutar una consulta que obtenga los títulos de la tabla **title** y los muestre en orden alfabético. `Answer_Sort.sql` es la secuencia de comandos completa de este procedimiento.

- Escriba y ejecute una consulta que obtenga una lista ordenada con todos los títulos de la tabla **title**.

```
USE library
SELECT title
FROM title
ORDER BY title
GO
```

Resultado

Su resultado debería ser similar al siguiente conjunto de resultados parciales.

Title

A Tale of Two Cities
 Adventures of Huckleberry Finn
 Ballads of a Bohemian
 Candide

.
 .
 .

(50 filas afectadas)

✎ **Para calcular datos, obtener valores calculados y utilizar un alias de columna**

En este ejercicio, va a escribir y ejecutar una consulta que obtenga las columnas **member_no**, **isbn** y **fine_assessed** de la tabla **loanhist** para todos los préstamos archivados que tengan un valor distinto de NULL en la columna **fine_assessed**. A continuación, en el conjunto de resultados va a crear una columna nueva que contenga el valor de la columna calculada **fine_assessed** multiplicado por dos; para esta columna utilizará el alias **double fine**.

Answer_Computed.sql es la secuencia de comandos completa de este procedimiento.

1. Escriba y ejecute una consulta que obtenga de la tabla **loanhist** las columnas **member_no**, **isbn** y **fine_assessed** de todos los préstamos archivados que tengan un valor distinto de NULL en la columna **fine_assessed**.
2. Cree una columna calculada que contenga el valor de la columna **fine_assessed** multiplicado por dos.
3. Utilice el alias de columna 'double fine' para la columna calculada. Debe especificar el alias entre comillas simples, ya que no se ajusta a las convenciones de denominación de objetos de SQL Server.

```
USE library
SELECT member_no, isbn, fine_assessed
      ,(fine_assessed * 2) AS 'double fine'
FROM loanhist
WHERE (fine_assessed IS NOT NULL)
GO
```

Resultado

Su resultado debería ser similar al siguiente conjunto de resultados. El número de filas que se devuelven puede variar.

member_no	isbn	fine_assessed	double fine
7399	101	9.0000	18.0000
6709	102	9.0000	18.0000
.			
.			
.			

(1560 filas afectadas)

✍ Para dar formato al conjunto de resultados de una columna con funciones de cadena

En este procedimiento, va a escribir y ejecutar una consulta que obtenga una lista con todos los miembros de la tabla **member** cuyo apellido sea Anderson. Dé formato al resultado para que se muestre en letras minúsculas y para incluir en él una columna de nombres de correo electrónico que constarán del nombre de cada miembro, la inicial de su segundo nombre y las dos primeras letras de su apellido.

Answer_Formatting.sql es la secuencia de comandos completa de este procedimiento.

1. Escriba y ejecute una consulta para generar una sola columna que contenga las columnas **firstname**, **middleinitial** y **lastname** de la tabla **member** para todos los miembros cuyo apellido sea Anderson.
2. Utilice el alias de columna **email_name**.
3. Modifique la consulta para obtener una lista de nombres de correo electrónico formados por el nombre, la inicial de su segundo nombre y las dos primeras letras del apellido en minúsculas. Utilice la función **SUBSTRING** para obtener partes de una columna de cadena. Utilice la función **LOWER** para pasar el resultado a minúsculas. Utilice también el operador de adición (+) para concatenar las cadenas.

USE library

```
SELECT LOWER(firstname + middleinitial  
          +SUBSTRING(lastname, 1, 2) ) AS email_name  
FROM member  
WHERE lastname = 'anderson'  
GO
```

Resultado

Su resultado debería ser similar al siguiente conjunto de resultados parciales. El número de filas que se devuelven puede variar.

email_name

Amyaan
Angelaan
Brianaan
Clairaan

.
. .
.

(390 filas afectadas)

✍ Para dar formato al conjunto de resultados de una columna mediante literales

En este procedimiento, va a dar formato al conjunto de resultados de una consulta para mejorar su legibilidad. Para ello, utilizará la función CONVERT y literales de cadena. Answer_Literals.sql es la secuencia de comandos completa de este procedimiento.

1. Escriba y ejecute una consulta que obtenga las columnas **title** y **title_no** de la tabla **title**. El conjunto de resultados debe ser una única columna con el formato siguiente:

```
the title is: Poems, title number 7
```

Esta consulta devuelve una sola columna basada en una expresión que concatena cuatro elementos:

- La constante de cadena **the title is:**
 - La columna **title.title**
 - La constante de cadena **title number**
 - La columna **title.title_no**
2. Utilice la función CONVERT para dar formato a la columna **title.title_no** y el operador de adición (+) para concatenar las cadenas de caracteres.

USE library

```
SELECT 'the title is: ' + title + ', title number ' +  
        CONVERT(char(6),title_no)  
FROM title  
GO
```

Resultado

Su resultado debería ser similar al siguiente conjunto de resultados parciales.

```
the title is: Last of the Mohicans, title number 1  
the title is: The Village Watch-Tower, title number 2  
the title is: Self Help; Conduct & Perseverance, title number 3  
the title is: Songs of a Savoyard, title number 4  
the title is: Fall of the House of Usher, title number 5  
.  
.  
.  
(50 filas afectadas)
```


Ejercicio 3

Uso de las funciones del sistema

En este ejercicio, va a recopilar información del sistema mediante las funciones del sistema. La carpeta Soluciones contiene las secuencias de comandos completas para este ejercicio.

✍ Para determinar el identificador de proceso del servidor

En este procedimiento, va a observar la actividad actual del servidor y a determinar la actividad que la sesión está generando. `Answer_SPID.sql` es la secuencia de comandos completa de este procedimiento.

1. Ejecute el procedimiento almacenado del sistema **sp_who**.

SQL Server muestra toda la actividad que se está produciendo en el servidor.

2. Para determinar qué actividad es la suya, ejecute la siguiente instrucción:

```
SELECT @@spid  
GO
```

El número identificador de proceso del servidor (spid) del proceso se devuelve en los resultados.

3. Ejecute otra vez el procedimiento almacenado del sistema **sp_who** con el número de spid como parámetro adicional. (En la siguiente instrucción, *n* representa el número de spid.)

```
EXEC sp_who n  
GO
```

Se muestra la actividad relacionada con su spid.

✍ Para recuperar información del entorno

En este procedimiento, va a determinar qué versión de SQL Server está ejecutando y va a recuperar información acerca de la conexión, el contexto de la base de datos y el servidor. Realizará estas tareas con las funciones del sistema.

La secuencia de comandos completa para este procedimiento es `Answer_Environment.sql`.

1. Ejecute la instrucción siguiente:

```
SELECT @@version  
GO
```

2. Ejecute la instrucción siguiente:

```
SELECT USER_NAME(), DB_NAME(), @@servername  
GO
```

⚡ Para recuperar metadatos

En este procedimiento, va a ejecutar varias consultas para devolver los metadatos de los objetos específicos de la base de datos mediante vistas de esquema de información. Recuerde que **information_schema** es un usuario predefinido de la base de datos que es propietario de las vistas de esquema de información. Answer_Metadata.sql es la secuencia de comandos completa de este procedimiento.

1. Ejecute la instrucción siguiente para devolver una lista de todas las tablas definidas por el usuario de una base de datos:

```
USE library
SELECT *
  FROM information_schema.tables
 WHERE table_type = 'base table'
GO
```

2. Ejecute la instrucción siguiente para devolver las columnas de la clave principal y de la clave externa de la tabla **orders**:

```
USE northwind
SELECT *
  FROM information_schema.key_column_usage
 WHERE table_name = 'orders'
GO
```