Capítulo VIII

Consultas: Casos Especiales

En este capítulo complementaremos el tema consultas de SQL Server mostrando casos especiales como las consultas DISTINCT, la creación de rankings, la creación de tablas a partir de las consultas, el uso de los operadores de conjuntos, entre otros.

1. CLÁUSULA DISTINCT

La cláusula DISTINCT establece que el conjunto de resultados de la consulta solo debe incluir filas únicas. Las filas duplicadas no se muestran como parte del resultado.

Sintáxis

```
SELECT [ DISTINCT ] …
FROM …
```

Ejercicio 8.1: Uso de DISTINCT

Se desea obtener un listado de los artículos enviados a las tiendas.

```
USE QhatuPERU
go

-- Artículos enviados a las tiendas
SELECT GUIA_DETALLE.CodArticulo,
     ARTICULO.DescripcionArticulo
FROM GUIA_DETALLE INNER JOIN ARTICULO
     ON GUIA_DETALLE.CodArticulo =
          ARTICULO.CodArticulo
ORDER BY 1
go
```

Resultados | Mensajes

	CodArticulo	DescripcionArticulo
1	1	CARAMELOS BASTON VIENA ARCOR
2	1	CARAMELOS BASTON VIENA ARCOR
3	1	CARAMELOS BASTON VIENA ARCOR
4	1	CARAMELOS BASTON VIENA ARCOR
5	1	CARAMELOS BASTON VIENA ARCOR
6	1	CARAMELOS BASTON VIENA ARCOR
7	1	CARAMELOS BASTON VIENA ARCOR
8	1	CARAMELOS BASTON VIENA ARCOR
9	1	CARAMELOS BASTON VIENA ARCOR
10	1	CARAMELOS BASTON VIENA ARCOR

El resultado muestra varios artículos más de una vez ya que éstos han salido del almacén en varias oportunidades. Para eliminar las filas duplicadas del resultado ejecute la consulta con la cláusula DISTINCT.

```
-- Eliminando filas duplicadas
SELECT DISTINCT GUIA_DETALLE.CodArticulo,
     ARTICULO.DescripcionArticulo
FROM GUIA_DETALLE INNER JOIN ARTICULO
     ON GUIA_DETALLE.CodArticulo =
          ARTICULO.CodArticulo
ORDER BY 1
go
```

Resultados | Mensajes

	CodArticulo	DescripcionArticulo
1	1	CARAMELOS BASTON VIENA ARCOR
2	2	CARAMELOS SURTIDO DE FRUTAS
3	3	CARAMELOS FRUTAS SURTIDA ARCOR
4	4	CARAMELOS FRUTAS MASTICABLES
5	6	FRUNA SURTIDA DONOFRIO
6	7	CHOCOLATE DOÑA PEPA FIELD
7	8	CHOCOLATE CUA CUA FIELD
8	9	MELLOWS FAMILIAR FIELD
9	10	WAFER CHOCOLATE FIELD
10	11	CHOCOLATE BARRA REGULAR

2. CLÁUSULA TOP

Especifica que el resultado de la consulta solo debe mostrar un tramo compuesto por las filas iniciales ó por un porcentaje del total de filas recuperadas. Cuando la cláusula TOP se utiliza junto con la cláusula ORDER BY es posible generar un ranking en base a los valores de una columna numérica.

Sintáxis

```
SELECT TOP (cantidad_filas_iniciales_a_mostrar)
     [ PERCENT ] [ WITH TIES ] …
FROM …
```

- **cantidad_filas_iniciales_a_mostrar** indica la cantidad ó porcentaje de filas a mostrar en el resultado.
- PERCENT indica que debe recuperarse el porcentaje de filas especificado por *cantidad_filas_iniciales_a_mostrar*.
- WITH TIES permite recuperar filas adicionales cuando en el último lugar del conjunto se produce un empate entre varias filas. Requiere que se utilice la cláusula ORDER BY.

Ejercicio 8.2: Uso de TOP

Se desea generar un ranking de los artículos más solicitados por las tiendas. Para ello, diseñe una consulta que muestre el total de unidades enviadas a las tiendas para cada artículo.

```
USE QhatuPERU
go

-- Ranking de los artículos más solicitados
-- por las tiendas
SELECT GUIA_DETALLE.CodArticulo,
     ARTICULO.DescripcionArticulo,
     SUM(GUIA_DETALLE.CantidadEnviada)
          AS TotalUnidades
FROM GUIA_DETALLE INNER JOIN ARTICULO
     ON GUIA_DETALLE.CodArticulo =
          ARTICULO.CodArticulo
GROUP BY GUIA_DETALLE.CodArticulo,
     ARTICULO.DescripcionArticulo
ORDER BY TotalUnidades DESC
go
```

Resultados | Mensajes

	CodArticulo	DescripcionArticulo	TotalUnidades
1	66	JABON ROSAS Y LIMON ROSADO	20000
2	65	JABON ROSAS Y LIMON BLANCO	20000
3	130	DETERGENTE LIMON INVICTO	15000
4	127	DETERGENTE PODER LIMON ACE	7500
5	124	DETERGENTE C/BLANQUEADOR ARIEL	7500
6	126	DETERGENTE LIMON ECOLOGICO MAGIA BLANCA	7500
7	132	DETERGENTE LIMON OPAL	7500
8	125	DETERGENTE LIMON ARIEL	7500
9	64	JABON DOVE BLANCO	7000
10	74	P.H. BLANCO SUAVE (ROJA)	4000

Modifique la consulta para que muestre los 5 artículos más solicitados.

```
-- Ranking de los 5 primeros
SELECT TOP 5 GUIA_DETALLE.CodArticulo,
     ARTICULO.DescripcionArticulo,
     SUM(GUIA_DETALLE.CantidadEnviada)
          AS TotalUnidades
FROM GUIA_DETALLE INNER JOIN ARTICULO
     ON GUIA_DETALLE.CodArticulo =
          ARTICULO.CodArticulo
GROUP BY GUIA_DETALLE.CodArticulo,
     ARTICULO.DescripcionArticulo
ORDER BY TotalUnidades DESC
go
```

Resultados | Mensajes

	CodArticulo	DescripcionArticulo	TotalUnidades
1	66	JABON ROSAS Y LIMON ROSADO	20000
2	65	JABON ROSAS Y LIMON BLANCO	20000
3	130	DETERGENTE LIMON INVICTO	15000
4	127	DETERGENTE PODER LIMON ACE	7500
5	124	DETERGENTE C/BLANQUEADOR ARIEL	7500

Observe que el 4to y 5to lugares están empatados. Modifique la consulta para que verifique si hay más artículos empatados con 7500 unidades despachadas.

```
-- Ranking de los 5 primeros con empates
SELECT TOP 5 WITH TIES GUIA_DETALLE.CodArticulo,
     ARTICULO.DescripcionArticulo,
     SUM(GUIA_DETALLE.CantidadEnviada)
          AS TotalUnidades
FROM GUIA_DETALLE INNER JOIN ARTICULO
     ON GUIA_DETALLE.CodArticulo =
          ARTICULO.CodArticulo
GROUP BY GUIA_DETALLE.CodArticulo,
     ARTICULO.DescripcionArticulo
ORDER BY TotalUnidades DESC
go
```

Resultados | Mensajes

	CodArticulo	DescripcionArticulo	TotalUnidades
1	66	JABON ROSAS Y LIMON ROSADO	20000
2	65	JABON ROSAS Y LIMON BLANCO	20000
3	130	DETERGENTE LIMON INVICTO	15000
4	127	DETERGENTE PODER LIMON ACE	7500
5	124	DETERGENTE C/BLANQUEADOR ARIEL	7500
6	126	DETERGENTE LIMON ECOLOGICO MAGIA BLANCA	7500
7	132	DETERGENTE LIMON OPAL	7500
8	125	DETERGENTE LIMON ARIEL	7500

Observe que hay 5 artículos empatados en el último lugar de este ranking de los 5 primeros.

Modifique la consulta para que muestre el quinto superior de todos los artículos enviados a las tiendas.

```
-- Quinto superior
SELECT TOP 20 PERCENT GUIA_DETALLE.CodArticulo,
     ARTICULO.DescripcionArticulo,
     SUM(GUIA_DETALLE.CantidadEnviada)
          AS TotalUnidades
FROM GUIA_DETALLE INNER JOIN ARTICULO
```

```
     ON GUIA_DETALLE.CodArticulo =
          ARTICULO.CodArticulo
GROUP BY GUIA_DETALLE.CodArticulo,
     ARTICULO.DescripcionArticulo
ORDER BY TotalUnidades DESC
go
```

Resultados | Mensajes

	CodArticulo	DescripcionArticulo	TotalUnidades
1	66	JABON ROSAS Y LIMON ROSADO	20000
2	65	JABON ROSAS Y LIMON BLANCO	20000
3	130	DETERGENTE LIMON INVICTO	15000
4	127	DETERGENTE PODER LIMON ACE	7500
5	124	DETERGENTE C/BLANQUEADOR ARIEL	7500
6	126	DETERGENTE LIMON ECOLOGICO MAGIA BLANCA	7500
7	132	DETERGENTE LIMON OPAL	7500
8	125	DETERGENTE LIMON ARIEL	7500
9	64	JABON DOVE BLANCO	7000
10	74	P.H. BLANCO SUAVE (ROJA)	4000
11	68	PASTA DENTAL KOLYNOS SUPER BLANCO	4000
12	129	DETERGENTE LIMON ARIEL	3750
13	106	DORINA CLASICA	1700
14	101	MANTEQUILLA LAIVE C/SAL	1700

Verifique si hay más artículos empatados en el puesto 14.

```
-- Quinto superior
SELECT TOP 20 PERCENT WITH TIES
     GUIA_DETALLE.CodArticulo,
     ARTICULO.DescripcionArticulo,
     SUM(GUIA_DETALLE.CantidadEnviada)
          AS TotalUnidades
FROM GUIA_DETALLE INNER JOIN ARTICULO
     ON GUIA_DETALLE.CodArticulo =
          ARTICULO.CodArticulo
GROUP BY GUIA_DETALLE.CodArticulo,
     ARTICULO.DescripcionArticulo
ORDER BY TotalUnidades DESC
go
```

Resultados | Mensajes

	CodArticulo	DescripcionArticulo	TotalUnidades
1	66	JABON ROSAS Y LIMON ROSADO	20000
2	65	JABON ROSAS Y LIMON BLANCO	20000
3	130	DETERGENTE LIMON INVICTO	15000
4	127	DETERGENTE PODER LIMON ACE	7500
5	124	DETERGENTE C/BLANQUEADOR ARIEL	7500
6	126	DETERGENTE LIMON ECOLOGICO MAGIA BLANCA	7500
7	132	DETERGENTE LIMON OPAL	7500
8	125	DETERGENTE LIMON ARIEL	7500
9	64	JABON DOVE BLANCO	7000
10	74	P.H. BLANCO SUAVE (ROJA)	4000
11	68	PASTA DENTAL KOLYNOS SUPER BLANCO	4000
12	129	DETERGENTE LIMON ARIEL	3750
13	106	DORINA CLASICA	1700
14	101	MANTEQUILLA LAIVE C/SAL	1700
15	103	MANTEQUILLA FERM C/SAL	1700
16	105	MARGARINA ASTRA	1700

Hay 4 artículos empatados con 1700 unidades enviadas en el último lugar de este ranking del quinto superior.

3. CLÁUSULAS OFFSET Y FETCH

Estas cláusulas permiten recuperar un intervalo específico de filas del resultado de la consulta.

Sintáxis

```
… ORDER BY …
[ OFFSET cantidad_filas_iniciales_a_excluir
     ROW | ROWS ]
[ FETCH FIRST | NEXT cantidad_filas_a_mostrar
     ROW | ROWS ONLY ]
```

- **cantidad_filas_iniciales_a_excluir** especifica el número de filas iniciales que no se mostrarán en el resultado de la consulta.

- **cantidad_filas_a_mostrar** especifica el número de filas que se mostrará en el resultado de la consulta luego de procesar la cláusula OFFSET.

Ejercicio 8.3: Uso de las cláusulas OFFSET y FETCH

Para ilustrar el uso de OFFSET y FETCH seguiremos revisando el ranking de los artículos despachados a las tiendas.

```
USE QhatuPERU
go

-- Ranking de los artículos más solicitados
-- por las tiendas
SELECT GUIA_DETALLE.CodArticulo,
     ARTICULO.DescripcionArticulo,
     SUM(GUIA_DETALLE.CantidadEnviada)
          AS TotalUnidades
FROM GUIA_DETALLE INNER JOIN ARTICULO
     ON GUIA_DETALLE.CodArticulo =
          ARTICULO.CodArticulo
GROUP BY GUIA_DETALLE.CodArticulo,
     ARTICULO.DescripcionArticulo
ORDER BY TotalUnidades DESC
go
```

Resultados | Mensajes

	CodArticulo	DescripcionArticulo	TotalUnidades
1	66	JABON ROSAS Y LIMON ROSADO	20000
2	65	JABON ROSAS Y LIMON BLANCO	20000
3	130	DETERGENTE LIMON INVICTO	15000
4	127	DETERGENTE PODER LIMON ACE	7500
5	124	DETERGENTE C/BLANQUEADOR ARIEL	7500
6	126	DETERGENTE LIMON ECOLOGICO MAGIA BLANCA	7500
7	132	DETERGENTE LIMON OPAL	7500
8	125	DETERGENTE LIMON ARIEL	7500
9	64	JABON DOVE BLANCO	7000
10	74	P.H. BLANCO SUAVE (ROJA)	4000

Modifique la consulta para que el resultado muestre el ranking a partir del puesto 11.

```
-- Ranking a partir del puesto 11
SELECT GUIA_DETALLE.CodArticulo,
     ARTICULO.DescripcionArticulo,
     SUM(GUIA_DETALLE.CantidadEnviada)
          AS TotalUnidades
FROM GUIA_DETALLE INNER JOIN ARTICULO
     ON GUIA_DETALLE.CodArticulo =
          ARTICULO.CodArticulo
GROUP BY GUIA_DETALLE.CodArticulo,
     ARTICULO.DescripcionArticulo
ORDER BY TotalUnidades DESC
     OFFSET 10 ROWS
go
```

Resultados | Mensajes

	CodArticulo	DescripcionArticulo	TotalUnidades
1	68	PASTA DENTAL KOLYNOS SUPER BLANCO	4000
2	129	DETERGENTE LIMON ARIEL	3750
3	106	DORINA CLASICA	1700
4	101	MANTEQUILLA LAIVE C/SAL	1700
5	103	MANTEQUILLA FERM C/SAL	1700
6	105	MARGARINA ASTRA	1700
7	116	SPRITE RETORNABLE	1500
8	117	TRIPLE DIET NO RETORNABLE	1500
9	111	INCA KOLA DIET	1500
10	114	SPRITE CONTOUR	1500

Ahora, permita que el resultado muestre los 5 artículos que se encuentran después del puesto 10.

```
-- Ranking del puesto 11 al puesto 15
SELECT GUIA_DETALLE.CodArticulo,
     ARTICULO.DescripcionArticulo,
     SUM(GUIA_DETALLE.CantidadEnviada)
          AS TotalUnidades
FROM GUIA_DETALLE INNER JOIN ARTICULO
     ON GUIA_DETALLE.CodArticulo =
```

```
          ARTICULO.CodArticulo
GROUP BY GUIA_DETALLE.CodArticulo,
     ARTICULO.DescripcionArticulo
ORDER BY TotalUnidades DESC
     OFFSET 10 ROWS
          FETCH FIRST 5 ROWS ONLY
go
```

Resultados | Mensajes

	CodArticulo	DescripcionArticulo	TotalUnidades
1	68	PASTA DENTAL KOLYNOS SUPER BLANCO	4000
2	129	DETERGENTE LIMON ARIEL	3750
3	106	DORINA CLASICA	1700
4	101	MANTEQUILLA LAIVE C/SAL	1700
5	103	MANTEQUILLA FERM C/SAL	1700

4. CREACIÓN DE TABLA A PARTIR DE UNA CONSULTA

Usando la cláusula INTO es posible guardar el resultado de una consulta en una tabla nueva.

Sintáxis

```
SELECT …
INTO nombre_tabla_nueva …
```

- **nombre_tabla_nueva** especifica el nombre de la tabla que guardará el resultado de la consulta. Si la tabla ya existe, se producirá un error.

Ejercicio 8.4: Creación de una tabla con SELECT … INTO

Diseñar una consulta que genere el catálogo de artículos y lo guarde en la tabla **CatalogoArticulos**.

```
-- Crear una tabla con columnas
-- de distintas tablas origen
SELECT ARTICULO.CodArticulo,
     ARTICULO.DescripcionArticulo,
     ARTICULO.Presentacion,
     ARTICULO.PrecioProveedor,
     LINEA.NomLinea, PROVEEDOR.NomProveedor
INTO CatalogoArticulos
FROM ARTICULO INNER JOIN LINEA
     ON ARTICULO.CodLinea = LINEA.CodLinea
INNER JOIN PROVEEDOR
     ON ARTICULO.CodProveedor =
          PROVEEDOR.CodProveedor
go

SELECT * FROM CatalogoArticulos
```

Resultados | Mensajes

	CodArticulo	DescripcionArticulo	Presentacion	PrecioProveedor	NomLinea	NomProveedor
1	1	CARAMELOS BASTON VIENA ARCOR	PAQUETE 454 GR	1,50	GOLOSINAS	GOLOSINAS Y ANTOJOS
2	2	CARAMELOS SURTIDO DE FRUTAS	PAQUETE 450 GR	1,00	GOLOSINAS	DISTRIBUIDORA DE GOLOSINAS FENIX
3	3	CARAMELOS FRUTAS SURTIDA ARCOR	PAQUETE 520 GR	1,50	GOLOSINAS	GOLOSINAS Y ANTOJOS
4	4	CARAMELOS FRUTAS MASTICABLES	PAQUETE 454 GR	1,30	GOLOSINAS	GOLOSINAS Y ANTOJOS
5	5	CHUPETES LOLY AMBROSOLI	KILOGRAMO	1,20	GOLOSINAS	DISTRIBUIDORA DE GOLOSINAS FENIX
6	6	FRUNA SURTIDA DONOFRIO	PAQUETE X 24 UNIDADES	1,80	GOLOSINAS	DISTRIBUIDORA DE GOLOSINAS FENIX
7	7	CHOCOLATE DOÑA PEPA FIELD	PAQUETE X 6 UNIDADES	2,20	GOLOSINAS	DISTRIBUIDORA DE GOLOSINAS FENIX
8	8	CHOCOLATE CUA CUA FIELD	PAQUETE X 6 UNIDADES	1,60	GOLOSINAS	DISTRIBUIDORA DE GOLOSINAS FENIX
9	9	MELLOWS FAMILIAR FIELD	PAQUETE 454 GR	2,10	GOLOSINAS	DISTRIBUIDORA DE GOLOSINAS FENIX
10	10	WAFER CHOCOLATE FIELD	PAQUETE X 9 UNIDADES	0,70	GOLOSINAS	DISTRIBUIDORA DE GOLOSINAS FENIX

4.1. Uso de la función IDENTITY

La función IDENTITY permite definir una columna con la propiedad IDENTITY cuando la tabla se crea con SELECT … INTO.

Ejercicio 8.5: Creación de una columna IDENTITY en una tabla creada con SELECT … INTO

Guardar el ranking creado por la consulta del ejercicio 8.3 en una tabla de nombre **RankingArticulos**. Esta tabla deberá tener una columna IDENTITY que indique la posición de cada artículo en el ranking.

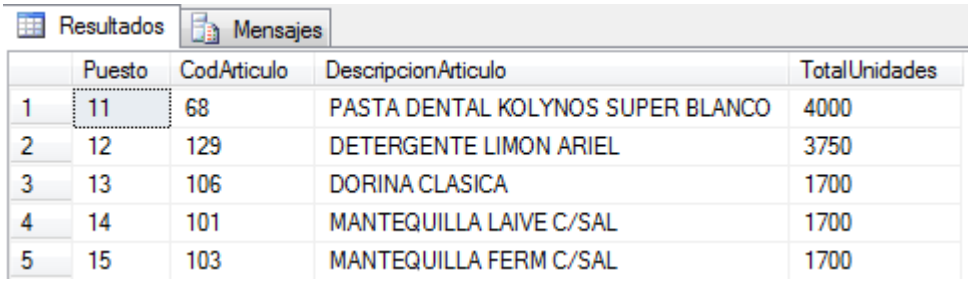
```
USE QhatuPERU
```

```
go

-- Ranking de los artículos
-- más solicitados por las tiendas
-- guardado en la tabla RankingArticulos
SELECT IDENTITY(integer, 1, 1) AS Puesto,
     GUIA_DETALLE.CodArticulo,
     ARTICULO.DescripcionArticulo,
     SUM(GUIA_DETALLE.CantidadEnviada)
          AS TotalUnidades
INTO RankingArticulos
FROM GUIA_DETALLE INNER JOIN ARTICULO
     ON GUIA_DETALLE.CodArticulo =
          ARTICULO.CodArticulo
GROUP BY GUIA_DETALLE.CodArticulo,
     ARTICULO.DescripcionArticulo
ORDER BY TotalUnidades DESC
go
```

Consulte el ranking para que muestre los puestos del 11 al 15.

```
-- Mostrar los puestos del 11 al 15 en el ranking
SELECT * FROM RankingArticulos
ORDER BY Puesto
     OFFSET 10 ROWS
          FETCH NEXT 5 ROWS ONLY
go
```



Resultados | Mensajes

	Puesto	CodArticulo	DescripcionArticulo	TotalUnidades
1	11	68	PASTA DENTAL KOLYNOS SUPER BLANCO	4000
2	12	129	DETERGENTE LIMON ARIEL	3750
3	13	106	DORINA CLASICA	1700
4	14	101	MANTEQUILLA LAIVE C/SAL	1700
5	15	103	MANTEQUILLA FERM C/SAL	1700

5. OPERADORES DE CONJUNTOS: UNION, EXCEPT, INTERSECT

Los operadores de conjuntos permiten combinar los resultados de dos ó más consultas en un solo conjunto de resultados. Microsoft SQL Server proporciona 3 operadores para dichas operaciones: UNION, EXCEPT e INTERSECT.

5.1. Operador UNION

Este operador reúne los resultados de dos consultas en un solo conjunto de resultados.

Sintáxis

```
Consulta1
UNION [ ALL ]
Consulta2
```

- ALL establece que el conjunto resultante debe incluir todas las filas de ambas consultas incluso las duplicadas.

Ejercicio 8.6: Uso del operador UNION

Se desea obtener un listado de las fechas en las que se ha producido movimientos en el almacén de QhatuPERU.

```
USE QhatuPERU
go

SELECT COUNT(DISTINCT FechaIngreso)
     FROM ORDEN_COMPRA
go
-- 16 fechas distintas para FechaIngreso

SELECT COUNT(DISTINCT FechaSalida)
     FROM GUIA_ENVIO
go
-- 107 fechas distintas para FechaSalida
```

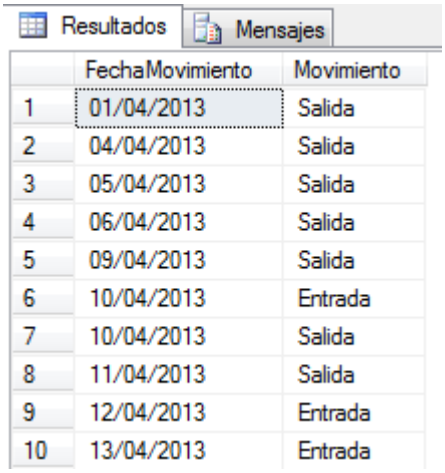
```
-- Uso de UNION ALL
-- Todas las fechas en las que hubo
-- ingreso y/o salida
-- incluyendo las filas duplicadas
SELECT CONVERT(CHAR(10), FechaIngreso, 103)
     AS FechaMovimiento,
     'Entrada' AS Movimiento
FROM ORDEN_COMPRA
UNION ALL
SELECT CONVERT(CHAR(10), FechaSalida, 103),
     'Salida' AS Movimiento
FROM GUIA_ENVIO
ORDER BY FechaMovimiento
go
-- 125 movimientos
```

Resultados | Mensajes

	FechaMovimiento	Movimiento
1	01/04/2013	Salida
2	01/04/2013	Salida
3	01/04/2013	Salida
4	01/04/2013	Salida
5	01/04/2013	Salida
6	01/04/2013	Salida
7	04/04/2013	Salida
8	04/04/2013	Salida
9	04/04/2013	Salida
10	04/04/2013	Salida

```
-- UNION
-- Todas las fechas en las que hubo
-- ingreso y/o salida
-- sin incluir las filas duplicadas
SELECT CONVERT(CHAR(10), FechaIngreso, 103)
     AS FechaMovimiento,
     'Entrada' AS Movimiento
FROM ORDEN_COMPRA
UNION
SELECT CONVERT(CHAR(10), FechaSalida, 103),
     'Salida' AS Movimiento
```

```
FROM GUIA_ENVIO
ORDER BY FechaMovimiento
go
-- 21 combinaciones fecha-movimiento
```



Resultados | Mensajes

	FechaMovimiento	Movimiento
1	01/04/2013	Salida
2	04/04/2013	Salida
3	05/04/2013	Salida
4	06/04/2013	Salida
5	09/04/2013	Salida
6	10/04/2013	Entrada
7	10/04/2013	Salida
8	11/04/2013	Salida
9	12/04/2013	Entrada
10	13/04/2013	Entrada

5.2. Operador EXCEPT

Este operador muestra en el conjunto de resultados las filas que solo están presentes en la primera consulta, y no se encuentran en la segunda consulta.

Sintáxis

```
Consulta1
EXCEPT
Consulta2
```

Ejercicio 8.7: Uso del operador EXCEPT

Se desea obtener un listado de las fechas en las que solo se ha ejecutado movimientos de entrada en el almacén de QhatuPERU.

```
SELECT COUNT(DISTINCT FechaIngreso)
     FROM ORDEN_COMPRA
go
-- 16 fechas distintas para FechaIngreso
```

```
SELECT COUNT(DISTINCT FechaSalida)
     FROM GUIA_ENVIO
go
-- 107 fechas distintas para FechaSalida

-- EXCEPT
-- Todas las fechas en las que hubo entradas,
-- pero no hubo salidas
SELECT CONVERT(CHAR(10), FechaIngreso, 103)
     AS FechaMovimiento,
     'Entrada' AS Movimiento
FROM ORDEN_COMPRA
EXCEPT
SELECT CONVERT(CHAR(10), FechaSalida, 103),
     'Salida' AS Movimiento
FROM GUIA_ENVIO
ORDER BY FechaMovimiento
go
-- 4 fechas en las que solo hubo entradas
```

Resultados | Mensajes

	FechaMovimiento	Movimiento
1	10/04/2013	Entrada
2	12/04/2013	Entrada
3	13/04/2013	Entrada
4	15/04/2013	Entrada

5.3. Operador Intersect

Este operador muestra en el conjunto de resultados únicamente las filas que se encuentran en las dos consultas sobre las que opera.

Sintáxis

```
Consulta1
INTERSECT
Consulta2
```

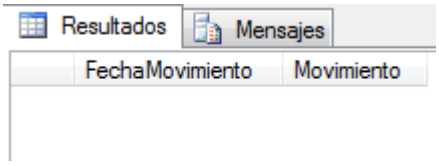
Ejercicio 8.8: Uso del operador INTERSECT

Se desea obtener un listado de las fechas en las que se ha ejecutado movimientos de entrada, y también movimientos de salida.

```
SELECT COUNT(DISTINCT FechaIngreso)
     FROM ORDEN_COMPRA
go
-- 16 fechas distintas para FechaIngreso

SELECT COUNT(DISTINCT FechaSalida)
     FROM GUIA_ENVIO
go
-- 107 fechas distintas para FechaSalida

-- INTERSECT
-- Fechas en las que hubo entrada y salida
SELECT CONVERT(CHAR(10), FechaIngreso, 103)
     AS FechaMovimiento,
     'Entrada' AS Movimiento
FROM ORDEN_COMPRA
INTERSECT
SELECT CONVERT(CHAR(10), FechaSalida, 103),
     'Salida' AS Movimiento
FROM GUIA_ENVIO
ORDER BY 1
go
-- 0 filas
```



6. EXPRESIÓN CASE

Se utiliza para evaluar una lista de expresiones lógicas y tomar alguna acción en función a los resultados.

6.1. Expresión CASE simple

Compara una expresión contra un conjunto de expresiones simples para determinar la acción a seguir.

Sintáxis

```
CASE expresión_de_entrada
     WHEN expresión_a_evaluar1
          THEN acción_a_seguir1,
     WHEN expresión_a_evaluar2
          THEN acción_a_seguir2,
     …
     [ ELSE
          acción_cuando_ninguna_anterior_cumple ]
END
```

Ejercicio 8.9: Uso de expresión CASE simple

```
USE QhatuPERU
go

SELECT NomProveedor, Ciudad,
     Ubicacion =
          CASE Ciudad
               WHEN 'Lima' THEN 'Capital'
               WHEN 'Callao' THEN 'Puerto'
               ELSE 'Provincias'
          END
FROM PROVEEDOR
go
```

En este ejercicio, la expresión CASE evalúa el contenido de la columna **Ciudad** de la tabla PROVEEDOR, y según su valor almacena la cadena **'Capital'**, **'Puerto'** ó **'Provincias'** en la columna **Ubicación** del resultado de la consulta.

Resultados | Mensajes

	NomProveedor	Ciudad	Ubicacion
1	LACTEOS DEL CENTRO	HUANCAYO	Provincias
2	DISTRIBUIDORA ALEMANA	LIMA	Capital
3	EMBUTIDOS EL GORDITO	CALLAO	Puerto
4	DISTRIBUIDORA NANDO	CALLAO	Puerto
5	DISTRIBUIDORA ALBRICIAS	LIMA	Capital
6	DISTRIBUIDORA DEL HOGAR	LIMA	Capital
7	PAPELERA PACHACAMAC	CAÑETE	Provincias
8	DISTRIBUIDORA SAN ANTONIO	LIMA	Capital
9	EMBOTELLADORA LA PREFERIDA	TRUJILLO	Provincias
10	DROGUERIA MAHAN	AREQUIPA	Provincias

6.2. Expresión CASE de búsqueda

En este caso, la expresión CASE evalúa un conjunto de expresiones lógicas, y determina para cada una de ellas una acción a seguir.

Sintáxis

```
CASE
     WHEN expresión_lógica1 THEN acción_a_seguir1
     WHEN expresión_lógica1 THEN acción_a_seguir2
     …
    [ELSE acción_cuando_ninguna_anterior_cumple]
END
```

Ejercicio 8.10: Uso de expresión CASE de búsqueda

```
USE QhatuPERU
go

SELECT CodArticulo, DescripcionArticulo,
     'Acción a tomar' = CASE
          WHEN StockActual <= StockMinimo
               THEN 'URGENTE: Colocar pedido'
          WHEN StockActual < StockMinimo * 1.1
               THEN 'Stock cerca al mínimo'
          ELSE 'Stock adecuado'
     END
```

```
FROM ARTICULO
go
```

Resultados | Mensajes

	CodArticulo	DescripcionArticulo	Acción a tomar
23	23	FUDGE SHOPPE DELUXE GRAHAMS	Stock adecuado
24	24	FUDGE SHOPPE STICKS KEEB	Stock adecuado
25	25	GALLETAS DELICE	Stock adecuado
26	26	JAMONADA LAIVE	Stock cerca al mínimo
27	27	JAMONADA ESPECIAL LA SEGOVIANA	URGENTE: Colocar pedido
28	28	JAMONADA POLACA OTTO KUNZ	Stock adecuado
29	29	JAMONADA DE POLLO SAN FERNANDO	Stock cerca al mínimo
30	30	JAMONADA ESPECIAL OTTO KUNZ	Stock adecuado
31	31	JAMON INGLES SAN FERNANDO	Stock adecuado
32	32	JAMON INGLES LAIVE	Stock adecuado

En este ejercicio se compara para cada producto, el **StockActual** contra el **StockMinimo**, y de acuerdo con el resultado se coloca la cadena correspondiente en la columna **Acción a tomar** del resultado de la consulta.

7. OPERADOR PIVOT

Permite crear una tabla de doble entrada haciendo que los valores que se muestran como filas en un conjunto de datos, se puedan mostrar como títulos de columna.

Sintáxis

```
SELECT * FROM
(
     Consulta Origen:
     consulta con la data a agregar y pivotear
) AS alias1
PIVOT( función_agregación( columna_numérica )
     FOR columna_dimensión
     IN ( lista_valores_columna_dimensión ) )
     AS alias2
```

- **consulta_origen** es la consulta que tiene los datos que normalmente se muestran como filas, y se desea consolidar y pivotear.
- **columna_numérica** es la columna cuyos valores se desea mostrar en una tabla de doble entrada.
- **columna_dimensión** es la columna cuyos valores en una consulta normal se ven como filas, y que en la consulta PIVOT se desea ver como columnas.
- **lista_valores_columna_dimensión** son los valores de *columna_dimensión* a mostrar como columnas en la tabla de doble entrada.

Ejercicio 8.11: Uso del operador PIVOT

Se tiene la siguiente consulta que muestra todos los despachos enviados a las tiendas especificando la línea del artículo enviado, el año y mes de envío, y el monto de lo enviado.

```
USE QhatuPERU
go

-- Consulta que muestra el monto despachado
-- de todos los envíos a las tiendas.
SELECT LINEA.NomLinea AS Linea,
CAST(YEAR(GUIA_ENVIO.FechaSalida) AS CHAR(4))
     + '-' +
     DATENAME(MONTH, GUIA_ENVIO.FechaSalida)
     AS Periodo,
     GUIA_DETALLE.CantidadEnviada *
          GUIA_DETALLE.PrecioVenta AS Monto
FROM LINEA INNER JOIN ARTICULO
     ON LINEA.CodLinea = ARTICULO.CodLinea
INNER JOIN GUIA_DETALLE
     ON ARTICULO.CodArticulo =
          GUIA_DETALLE.CodArticulo
INNER JOIN GUIA_ENVIO
     ON GUIA_DETALLE.NumGuia = GUIA_ENVIO.NumGuia
go
```

Resultados | Mensajes

	Linea	Periodo	Monto
1	GOLOSINAS	2013-Marzo	45,00
2	GOLOSINAS	2013-Marzo	30,00
3	GOLOSINAS	2013-Marzo	45,00
4	GOLOSINAS	2013-Marzo	39,00
5	GOLOSINAS	2013-Marzo	81,00
6	GOLOSINAS	2013-Marzo	82,50
7	GOLOSINAS	2013-Marzo	60,00
8	GOLOSINAS	2013-Marzo	63,00
9	GOLOSINAS	2013-Marzo	21,00
10	GOLOSINAS	2013-Marzo	30,00

Se desea un reporte que consolide el monto enviado por **Linea** y por **Periodo**, solo que los periodos deben mostrarse como columnas, tal como se observa a continuación.

Resultados | Mensajes

	Linea	2013-Marzo	2013-Abril
1	EMBUTIDOS	24312,00	36468,00
2	GOLOSINAS	4872,00	7308,00
3	HIGIENE PERSONAL	30938,00	46407,00
4	LACTEOS	9930,00	11171,25
5	LICORES Y GASEOSAS	26114,00	22849,75
6	LIMPIEZA	84120,00	73605,00

```
SELECT * FROM
(
SELECT LINEA.NomLinea AS Linea,
CAST(YEAR(GUIA_ENVIO.FechaSalida) AS CHAR(4))
     + '-' +
     DATENAME(MONTH, GUIA_ENVIO.FechaSalida)
          AS Periodo,
     GUIA_DETALLE.CantidadEnviada *
          GUIA_DETALLE.PrecioVenta AS Monto
FROM LINEA INNER JOIN ARTICULO
     ON LINEA.CodLinea = ARTICULO.CodLinea
INNER JOIN GUIA_DETALLE
```

```
     ON ARTICULO.CodArticulo =
          GUIA_DETALLE.CodArticulo
INNER JOIN GUIA_ENVIO
     ON GUIA_DETALLE.NumGuia = GUIA_ENVIO.NumGuia
) AS Origen
PIVOT (SUM(Monto) FOR Periodo
     IN ( [2013-Marzo], [2013-Abril] ))
     AS Destino
go
```

8. COMMON TABLE EXPRESSION (CTE)

Un CTE es un conjunto de resultados temporal derivado de una consulta que puede ser utilizado como una tabla derivada. Su comportamiento es muy similar al de una vista. Una CTE puede contener referencias a ella misma, por lo que permite diseñar consultas recursivas.

Sintáxis

```
WITH nombre_CTE( lista_columnas )
AS
(
SELECT_que_puebla_la_CTE
)
SELECT_que_muestra_la_CTE
```

- **SELECT_que_puebla_la_CTE** es la consulta que carga los datos en la CTE.
- **SELECT_que_muestra_la_CTE** es la consulta que muestra la data cargada en la CTE.

Ejercicio 8.12: Uso de Common Table Expression

Crear una CTE que contenga el proveedor, fecha y monto de cada una de las órdenes de envío. Luego consultarla para que muestre las órdenes del 11 de abril del 2013.

```
USE QhatuPERU
go
```

```
WITH ComprasCTE(Orden, Fecha, Proveedor, Monto)
AS
(
SELECT ORDEN_COMPRA.NumOrden,
CONVERT(CHAR(10), ORDEN_COMPRA.FechaOrden, 102),
     PROVEEDOR.NomProveedor,
     SUM(ORDEN_DETALLE.CantidadRecibida *
          ORDEN_DETALLE.PrecioCompra)
FROM ORDEN_COMPRA INNER JOIN ORDEN_DETALLE
     ON ORDEN_COMPRA.NumOrden =
          ORDEN_DETALLE.NumOrden
INNER JOIN ARTICULO
     ON ORDEN_DETALLE.CodArticulo =
          ARTICULO.CodArticulo
INNER JOIN PROVEEDOR
     ON ARTICULO.CodProveedor =
          PROVEEDOR.CodProveedor
GROUP BY ORDEN_COMPRA.NumOrden,
     CONVERT(CHAR(10),
     ORDEN_COMPRA.FechaOrden, 102),
     PROVEEDOR.NomProveedor
)
SELECT * FROM ComprasCTE
     WHERE Fecha = '2013.04.11'
go
```

Resultados | Mensajes

	Orden	Fecha	Proveedor	Monto
1	5	2013.04.11	DISTRIBUIDORA DE GOLOSINAS FENIX	12525,00
2	6	2013.04.11	GOLOSINAS Y ANTOJOS	4700,00

Ejercicio 8.13: Consultas recursivas usando CTE

Ejecute las siguientes instrucciones (la tabla TRABAJADOR fue creada en el capítulo anterior):

```
USE QhatuPERU
go
```

```
SELECT * FROM Trabajador
go
```

Resultados | Mensajes

	idTrabajador	Apellidos	Jefe
1	101	Camacho Saravia	102
2	102	Ardiles Soto	NULL
3	103	Sánchez Aliaga	101
4	104	Castro Avila	101
5	105	Vilchez Santos	102
6	106	Juárez Pinto	105
7	107	Umunaga Tapia	101

El árbol de jerarquías de los trabajadores tiene la siguiente estructura:

```
Nivel 0         Nivel 1         Nivel 2
Ardiles Soto
                Camacho Saravia
                                Sánchez Aliaga
                                Castro Avila
                                Urrunaga Tapia
                Vilchez Santos
                                Juárez Pinto
```

Mediante una consulta recursiva a una CTE obtener un listado de los trabajadores que muestre en qué nivel de la jerarquía se encuentra cada uno.

```
WITH EmpleadosCTE(Codigo, Nombre, Nivel)
AS
(
-- miembro ancla - consulta no recursiva
SELECT idTrabajador, Apellidos, 0 AS Nivel
FROM Trabajador
WHERE Jefe IS NULL
UNION ALL
-- miembro recursivo - consulta recursiva
SELECT t.idTrabajador, t.Apellidos, Nivel+1
FROM Trabajador t INNER JOIN EmpleadosCTE e
     ON t.Jefe = e.Codigo
```

```
)
-- consulta externa
SELECT * FROM EmpleadosCTE
go
```

Resultados | Mensajes

	Codigo	Nombre	Nivel
1	102	Ardiles Soto	0
2	101	Camacho Saravia	1
3	105	Vilchez Santos	1
4	106	Juárez Pinto	2
5	103	Sánchez Aliaga	2
6	104	Castro Avila	2
7	107	Urrunaga Tapia	2

Una CTE recursiva es construida desde al menos dos consultas. Una, es una consulta no recursiva conocida como el miembro ancla. La segunda, es la consulta recursiva conocida como el miembro recursivo. Estas consultas van separadas por el operador UNION ALL.

La CTE es inicialmente cargada con el resultado de la consulta del miembro ancla. A continuación, el operador UNION ALL combina el resultado de la primera consulta con el resultado de la consulta recursiva.

9. EJERCICIOS PROPUESTOS

Para los siguientes ejercicios debe utilizar las bases de datos RH y EDUCA. Los scripts para crearlas en su servidor los encontrará en el CD que acompaña al libro.

SELECT … INTO

1. (RH) Crear una tabla de nombre PLANILLA que contenga la siguiente información:

- Código de departamento
- Nombre del departamento
- Importe de planilla
- Importe de planilla proyectada con un aumento de 15%

Operador UNION

2. (EDUCA) En una sola consulta combine los datos de los alumnos y profesores. El resultado debe mostrar una columna adicional cuyo valor indica si los datos del registro corresponden a un Alumno o a un Profesor.

Operador EXCEPT

3. (EDUCA) Se necesita un listado de los alumnos (solo el código) que no se han matriculado en el curso **SQL Server Implementación**.

Operador INTERSECT

4. (EDUCA) Se necesita un listado de los alumnos (solo el código) que están matriculados en los cursos **SQL Server Implementación** y **SQL Server Administración**.

Cláusula TOP

5. (RH) Escriba una consulta para averiguar quiénes son los trabajadores que tienen el sueldo más bajo.

6. (EDUCA) Escriba una consulta para averiguar quiénes son los alumnos con la menor nota.

Cláusula DISTINCT

7. (EDUCA) Escriba una consulta que muestre el listado de los profesores.

Función CASE

8. (EDUCA) Escriba una consulta que califique la nota de cada alumno según el siguiente cuadro:

Nota	Calificación
[0, 10>	Malo
[10, 14>	Regular
[14, 18>	Bueno
[18,20]	Excelente

Operador PIVOT

9. (EDUCA) Encontrar el ingreso por mes de cada curso.
10. (EDUCA) Encontrar el ingreso por trimestre de cada año.
11. (RH) Encontrar la cantidad de empleados que han ingresado por trimestre en cada año.

Common Table Expression (CTE)

12. (RH) Encontrar el empleado que tiene el menor salario por departamento.
13. (EDUCA) Encontrar el alumno con la mejor nota por curso.