Capítulo VII

Consultas a más de una tabla

En este capítulo conoceremos cómo diseñar consultas cuyo resultado implica leer datos de más de una tablas: las combinaciones ó joins y las subconsultas correlacionadas, así como las subconsultas simples.

Por lo general, la generación de reportes útiles y fáciles de entender para los usuarios requiere que la consulta lea los datos de varias tablas. En este capítulo veremos cómo diseñar instrucciones SELECT que permitan recuperar datos de múltiples tablas en un solo conjunto de resultados.

1. COMBINACIONES Ó JOINS

Una combinación, join ó consulta correlacionada es la consulta que muestra columnas de dos tablas ó conjuntos de filas y las entrega en un único conjunto de resultados. Típicamente, la combinación se lleva a cabo relacionando valores comunes en los dos conjuntos de resultados, tales como los valores de clave primaria y clave foránea.

Sintáxis

```
SELECT lista_columnas
FROM tabla1 | conjunto_de_filas1
tipo_join JOIN tabla2 | conjunto_de_filas2
ON condición_del_join
```

- **lista_columnas** es la lista de columnas a mostrar en el resultado de la consulta. Se recomienda que cada columna sea calificada con el alias de la tabla a la cual pertenece.
- **tipo_join** indica si el join es interior (INNER), exterior (OUTER) ó irrestricto (CROSS).
- **condición_del_join** es una expresión que indica en base a qué columnas de cada una de las tablas se establece la relación entre ellas.

Una combinación (join) puede ser de cualquiera de los siguientes tipos:

- inner join
- outer join
    - left outer join
    - right outer join
    - full outer join
- cross join
- autojoin

2. INNER JOIN

Un **inner join** es la consulta correlacionada cuyo resultado muestra todas las filas que están relacionadas entre dos tablas ó conjuntos de filas.

Ejercicio 7.1: Uso de INNER JOIN

Se desea obtener un listado de los artículos que comercializa **QhatuPERU**. La lista debe mostrar: el código del artículo, su descripción, su presentación, su precio unitario, y quién es el proveedor del artículo.

Ejecute las siguientes instrucciones:

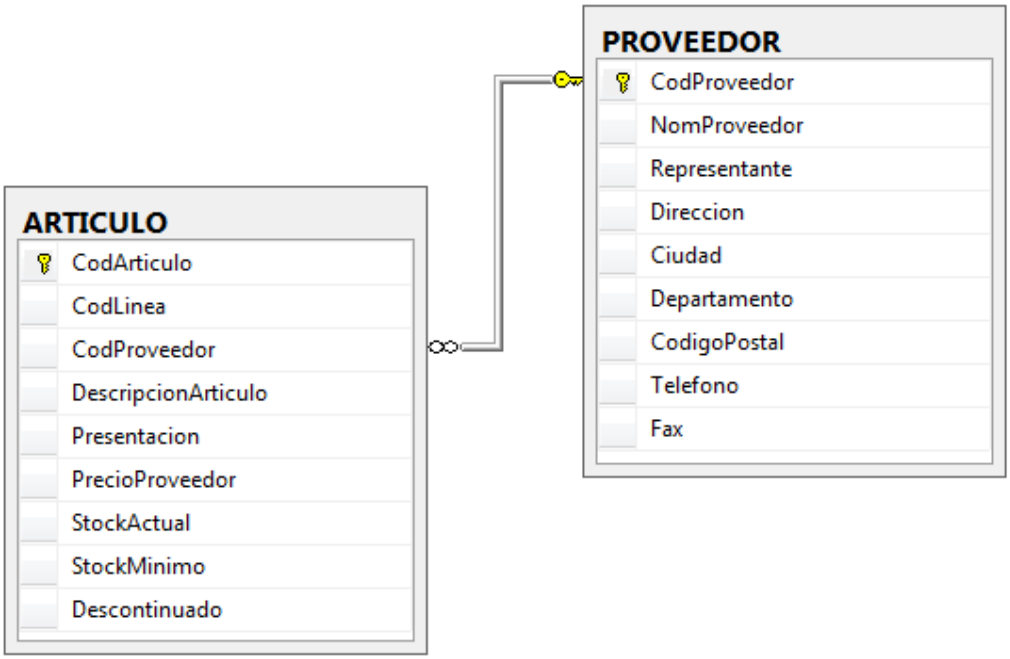
```
USE QhatuPERU
go

SELECT CodArticulo, DescripcionArticulo,
     Presentacion, PrecioProveedor, CodProveedor
FROM ARTICULO
go
```

Resultados | Mensajes

	CodArticulo	DescripcionArticulo	Presentacion	PrecioProveedor	CodProveedor
1	1	CARAMELOS BASTON VIENA ARCOR	PAQUETE 454 GR	1,50	14
2	2	CARAMELOS SURTIDO DE FRUTAS	PAQUETE 450 GR	1,00	15
3	3	CARAMELOS FRUTAS SURTIDA ARCOR	PAQUETE 520 GR	1,50	14
4	4	CARAMELOS FRUTAS MASTICABLES	PAQUETE 454 GR	1,30	14
5	5	CHUPETES LOLY AMBROSOLI	KILOGRAMO	1,20	15
6	6	FRUNA SURTIDA DONOFRIO	PAQUETE X 24 UNIDADES	1,80	15
7	7	CHOCOLATE DOÑA PEPA FIELD	PAQUETE X 6 UNIDADES	2,20	15
8	8	CHOCOLATE CUA CUA FIELD	PAQUETE X 6 UNIDADES	1,60	15
9	9	MELLOWS FAMILIAR FIELD	PAQUETE 454 GR	2,10	15
10	10	WAFER CHOCOLATE FIELD	PAQUETE X 9 UNIDADES	0,70	15

El resultado de la consulta muestra al proveedor del artículo (**CodProveedor**), pero no es muy cómodo ó útil para el usuario final ya que no muestra, por ejemplo, el nombre del proveedor. Este último dato, no es posible obtenerlo de la tabla ARTICULO. El dato se encuentra en la tabla PROVEEDOR. Felizmente hay

una relación entre el artículo y su proveedor tal como se muestra en el diagrama siguiente:



Si para un artículo conocemos su **CodProveedor**, a través de la relación

```
ARTICULO.CodProveedor = PROVEEDOR.CodProveedor
```

podemos conocer no solo el nombre del proveedor, sino también sus demás datos.

Vamos a modificar la consulta anterior para que usando la relación existente combine los datos de las tablas ARTICULO y PROVEEDOR.

```
SELECT ARTICULO.CodArticulo,
     ARTICULO.DescripcionArticulo,
     ARTICULO.Presentacion,
     ARTICULO.PrecioProveedor,
     ARTICULO.CodProveedor,
     PROVEEDOR.NomProveedor,
     PROVEEDOR.Departamento
```

```
FROM ARTICULO INNER JOIN PROVEEDOR
     ON ARTICULO.CodProveedor =
          PROVEEDOR.CodProveedor
ORDER BY ARTICULO.CodArticulo
go
```

Resultados | Mensajes

	CodArticulo	DescripcionArticulo	Presentacion	PrecioProveedor	CodProveedor	NomProveedor	Departament
1	1	CARAMELOS BASTON VIENA ARCOR	PAQUETE 454 GR	1,50	14	GOLOSINAS Y ANTOJOS	AREQUIPA
2	2	CARAMELOS SURTIDO DE FRUTAS	PAQUETE 450 GR	1,00	15	DISTRIBUIDORA DE GOLOSINAS FENIX	LIMA
3	3	CARAMELOS FRUTAS SURTIDA ARCOR	PAQUETE 520 GR	1,50	14	GOLOSINAS Y ANTOJOS	AREQUIPA
4	4	CARAMELOS FRUTAS MASTICABLES	PAQUETE 454 GR	1,30	14	GOLOSINAS Y ANTOJOS	AREQUIPA
5	5	CHUPETES LOLY AMBROSOLI	KILOGRAMO	1,20	15	DISTRIBUIDORA DE GOLOSINAS FENIX	LIMA
6	6	FRUNA SURTIDA DONOFRIO	PAQUETE X 24 UNIDADES	1,80	15	DISTRIBUIDORA DE GOLOSINAS FENIX	LIMA
7	7	CHOCOLATE DOÑA PEPA FIELD	PAQUETE X 6 UNIDADES	2,20	15	DISTRIBUIDORA DE GOLOSINAS FENIX	LIMA
8	8	CHOCOLATE CUA CUA FIELD	PAQUETE X 6 UNIDADES	1,60	15	DISTRIBUIDORA DE GOLOSINAS FENIX	LIMA
9	9	MELLOWS FAMILIAR FIELD	PAQUETE 454 GR	2,10	15	DISTRIBUIDORA DE GOLOSINAS FENIX	LIMA
10	10	WAFER CHOCOLATE FIELD	PAQUETE X 9 UNIDADES	0,70	15	DISTRIBUIDORA DE GOLOSINAS FENIX	LIMA

Como el resultado de la consulta debe mostrar la relación que hay entre cada artículo y su correspondiente proveedor, la consulta debe ser de tipo INNER JOIN. Observe que en la consulta, cada columna ha sido "calificada", es decir, se ha indicado de qué tabla debe ser leída. Si bien no es obligatorio calificar todas las columnas que participan en la consulta, se recomienda hacerlo ya que es una buena práctica que facilita el comprender la lógica de la misma.

2.1. Uso de alias para hacer referencia a las tablas

Un alias es un nombre alterno con el que se hace referencia a una tabla y que permite simplificar la escritura de las consultas. Por ejemplo, para la tabla ARTICULO podemos definir que su alias es ART, de modo que cuando en la consulta deseamos hacer referencia a la tabla usamos su alias ART en vez de su nombre.

Ejercicio 7.2: Uso de alias

En la consulta del ejercicio anterior

```
SELECT ARTICULO.CodArticulo,
     ARTICULO.DescripcionArticulo,
     ARTICULO.Presentacion,
     ARTICULO.PrecioProveedor,
```

```
     ARTICULO.CodProveedor,
     PROVEEDOR.NomProveedor,
     PROVEEDOR.Departamento
FROM ARTICULO INNER JOIN PROVEEDOR
     ON ARTICULO.CodProveedor =
          PROVEEDOR.CodProveedor
ORDER BY ARTICULO.CodArticulo
go
```

cada columna se ha calificado con el nombre de la tabla en la que se lee la columna.

Si bien calificar las columnas hace que la consulta sea más fácil de entender para el usuario, hacerlo no es obligatorio, salvo en los casos en los que no hacerlo puede conducir a ambigüedad en la lectura de la instrucción.

Por ejemplo, la ejecución de la consulta

```
SELECT CodArticulo, DescripcionArticulo,
     Presentacion, PrecioProveedor,
     CodProveedor, NomProveedor,
     Departamento
FROM ARTICULO INNER JOIN PROVEEDOR
     ON ARTICULO.CodProveedor =
PROVEEDOR.CodProveedor
ORDER BY CodArticulo
go
```

genera el error 209

```
Mens. 209, Nivel 16, Estado 1, Línea 3
El nombre de columna 'CodProveedor' es ambiguo.
```

debido a que la columna CodProveedor mencionada en la instrucción SELECT existe tanto en la tabla ARTICULO como en la tabla PROVEEDOR, por lo que el motor de datos no sabe de qué tabla debe leer la columna. No ocurre lo mismo con el resto de identificadores de columna porque éstos solo existen en una de las tablas.

Para evitar que el error se produzca, la consulta se debe escribir como sigue

```
SELECT CodArticulo, DescripcionArticulo,
     Presentacion, PrecioProveedor,
     ARTICULO.CodProveedor, NomProveedor,
     Departamento
FROM ARTICULO INNER JOIN PROVEEDOR
     ON ARTICULO.CodProveedor =
PROVEEDOR.CodProveedor
ORDER BY CodArticulo
go
```

También es posible simplificar la digitación de los nombres de tabla definiendo un alias para cada tabla. El alias es una cadena corta que podemos definir para hacer referencia a la tabla con un nombre corto.

```
SELECT Art.CodArticulo, Art.DescripcionArticulo,
     Art.Presentacion, Art.PrecioProveedor,
     Art.CodProveedor, Prov.NomProveedor,
     Prov.Departamento
FROM ARTICULO Art INNER JOIN PROVEEDOR Prov
     ON Art.CodProveedor = Prov.CodProveedor
ORDER BY Art.CodArticulo
go
```

Ya sea que utilice ó no los alias para tablas, la recomendación al escribir una consulta JOIN es que califique todas las columnas ya que así es más fácil de entender para el usuario que no ha escrito la consulta.

Ejercicio 7.3: Cabecera y detalle de la Guía de Envío X

Escriba una consulta que muestre los datos de la cabecera de la guía de envío número 27, y además su detalle.

```
SELECT GUIA_ENVIO.NumGuia,
     GUIA_ENVIO.FechaSalida,
     GUIA_ENVIO.CodTienda,
     GUIA_DETALLE.CodArticulo,
     GUIA_DETALLE.PrecioVenta,
     GUIA_DETALLE.CantidadEnviada
```

```
FROM GUIA_ENVIO INNER JOIN GUIA_DETALLE
     ON GUIA_ENVIO.NumGuia = GUIA_DETALLE.NumGuia
WHERE GUIA_ENVIO.NumGuia = 27
go
```

Resultados | Mensajes

	NumGuia	FechaSalida	CodTienda	CodArticulo	PrecioVenta	CantidadEnviada
1	27	2013-03-25 20:38:04.813	2	124	3,00	500
2	27	2013-03-25 20:38:04.813	2	125	3,00	500
3	27	2013-03-25 20:38:04.813	2	126	2,25	500
4	27	2013-03-25 20:38:04.813	2	127	2,70	500
5	27	2013-03-25 20:38:04.813	2	129	5,64	250
6	27	2013-03-25 20:38:04.813	2	130	1,80	1000
7	27	2013-03-25 20:38:04.813	2	132	2,10	500
8	27	2013-03-25 20:38:04.813	2	133	17,65	10
9	27	2013-03-25 20:38:04.813	2	134	17,40	10
10	27	2013-03-25 20:38:04.813	2	135	17,40	10
11	27	2013-03-25 20:38:04.813	2	136	25,55	10

Ejercicio 7.4: Cabecera y monto total de la Guía de Envío X

Modifique la consulta del ejercicio anterior para que muestre los datos de la cabecera de la guía de envío número 27 y además su monto total.

```
SELECT GUIA_ENVIO.NumGuia,
     GUIA_ENVIO.FechaSalida,
     GUIA_ENVIO.CodTienda,
     SUM(GUIA_DETALLE.CantidadEnviada *
          GUIA_DETALLE.PrecioVenta) AS MontoTotal
FROM GUIA_ENVIO INNER JOIN GUIA_DETALLE
     ON GUIA_ENVIO.NumGuia = GUIA_DETALLE.NumGuia
GROUP BY GUIA_ENVIO.NumGuia,
     GUIA_ENVIO.FechaSalida,
     GUIA_ENVIO.CodTienda
HAVING GUIA_ENVIO.NumGuia = 27
go
```

Resultados | Mensajes

	NumGuia	FechaSalida	CodTienda	MontoTotal
1	27	2013-03-25 20:38:04.813	2	10515,00

Ejercicio 7.5: JOIN de 3 tablas

Se desea generar el catálogo de artículos de QhatuPERU que debe mostrar:

- línea del artículo (tabla LINEA, columna NomLinea)
- código del artículo (tabla ARTICULO, columna CodArticulo)
- descripción del artículo (tabla ARTICULO, columna DescripcionArticulo)
- presentación del artículo (tabla ARTICULO, columna Presentacion)
- precio unitario del artículo (tabla ARTICULO, columna PrecioProveedor)
- nombre del proveedor (tabla PROVEEDOR, columna NomProveedor)

Vamos a construir la consulta paso a paso, de modo que aprovechemos la característica "autocompletar" del editor de consultas de SQL Server Management Studio.

En el editor digite SELECT y luego la cláusula FROM que define el primer join entre las tablas LINEA y ARTICULO:

SQLQuery5.sql - S...\SMATSUKAWA (52))*

```
USE QhatuPERU
go

SELECT
FROM LINEA INNER JOIN ARTICULO
    ON LINEA.CodLinea = ARTICULO.
```

CodArticulo
CodLinea — columna CodLinea(int, not nul
CodProveedor
Descontinuado
DescripcionArticulo
PrecioProveedor
Presentacion
StockActual
StockMinimo

A continuación especifique en el SELECT las columnas que provienen de las tablas LINEA y ARTICULO en el orden que se desea mostrar en el resultado.

SQLQuery5.sql - S...\SMATSUKAWA (52))*

```
USE QhatuPERU
go

SELECT LINEA.NomLinea, ARTICULO.CodArticulo,
    ARTICULO.DescripcionArticulo,
    ARTICULO.Presentacion, ARTICULO.PrecioProveedor
FROM LINEA INNER JOIN ARTICULO
    ON LINEA.CodLinea = ARTICULO.CodLinea
```

Si desea puede ejecutar la consulta a modo de comprobación.

Resultados | Mensajes

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

Ahora añada el segundo JOIN, que combina el resultado anterior con la tabla PROVEEDOR.

SQLQuery5.sql - S...\SMATSUKAWA (52))*

```sql
USE QhatuPERU
go

SELECT LINEA.NomLinea, ARTICULO.CodArticulo,
    ARTICULO.DescripcionArticulo,
    ARTICULO.Presentacion, ARTICULO.PrecioProveedor
FROM LINEA INNER JOIN ARTICULO
    ON LINEA.CodLinea = ARTICULO.CodLinea
INNER JOIN PROVEEDOR
    ON ARTICULO.CodProveedor = PROVEEDOR.
```

columna CodProveedor(PK, int, not null)

- Ciudad
- CodigoPostal
- CodProveedor
- Departamento
- Direccion
- Fax
- NomProveedor
- Representante
- Telefono

Añada al SELECT las columnas a leer de la tabla PROVEEDOR.

SQLQuery5.sql - S...\SMATSUKAWA (52))*

```sql
USE QhatuPERU
go

SELECT LINEA.NomLinea, ARTICULO.CodArticulo,
    ARTICULO.DescripcionArticulo,
    ARTICULO.Presentacion, ARTICULO.PrecioProveedor,
    PROVEEDOR.NomProveedor
FROM LINEA INNER JOIN ARTICULO
    ON LINEA.CodLinea = ARTICULO.CodLinea
INNER JOIN PROVEEDOR
    ON ARTICULO.CodProveedor = PROVEEDOR.CodProveedor
```

Ejecute la consulta.

Resultados | Mensajes

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

Ejercicio 7.6: Monto total enviado a cada tienda

Escriba una consulta que muestre el monto total enviado a cada tienda.

```
SELECT TIENDA.CodTienda, TIENDA.Distrito,
     SUM(GUIA_DETALLE.CantidadEnviada *
          GUIA_DETALLE.PrecioVenta)
     AS MontoTotal
FROM TIENDA INNER JOIN GUIA_ENVIO
     ON TIENDA.CodTienda = GUIA_ENVIO.CodTienda
INNER JOIN GUIA_DETALLE
     ON GUIA_ENVIO.NumGuia = GUIA_DETALLE.NumGuia
GROUP BY TIENDA.CodTienda, TIENDA.Distrito
go
```

Resultados | Mensajes

	CodTienda	Distrito	MontoTotal
1	1	MIRAFLORES	76363,75
2	2	PUEBLO LIBRE	76363,75
3	3	CALLAO	75122,50
4	4	LOS OLIVOS	75122,50
5	5	BREÑA	75122,50

Ejercicio 7.7: Total de unidades despachadas por mes del artículo X

Escriba una consulta que muestre el total de unidades mensuales despachadas del artículo 37.

```
SELECT YEAR(GUIA_ENVIO.FechaSalida) AS Año,
     MONTH(GUIA_ENVIO.FechaSalida) AS Mes,
     SUM(GUIA_DETALLE.CantidadEnviada) AS
          UnidadesEnviadas
FROM GUIA_ENVIO INNER JOIN GUIA_DETALLE
     ON GUIA_ENVIO.NumGuia = GUIA_DETALLE.NumGuia
WHERE GUIA_DETALLE.CodArticulo = 37
GROUP BY YEAR(GUIA_ENVIO.FechaSalida),
     MONTH(GUIA_ENVIO.FechaSalida)
go
```

Resultados | Mensajes

	Año	Mes	UnidadesEnviadas
1	2013	3	200
2	2013	4	300

3. OUTER JOIN

En ocasiones es necesario obtener un reporte que muestre qué filas de una tabla no tienen ninguna relación con otra tabla. En estas situaciones, podemos emplear una consulta outer join.

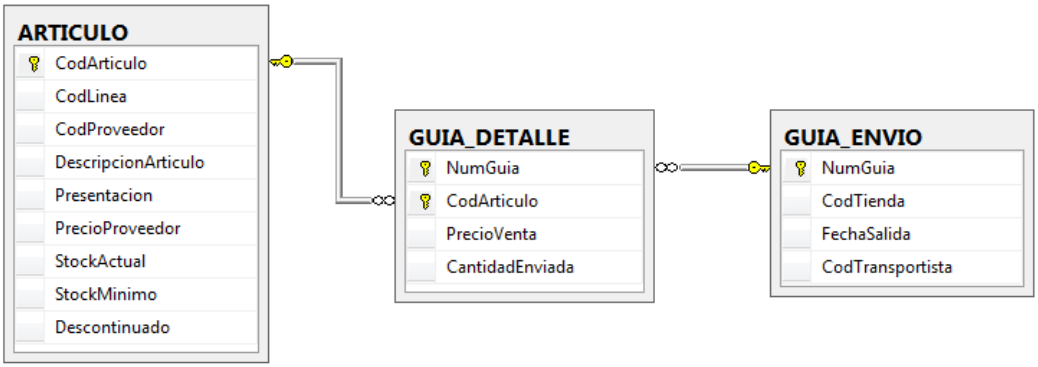
Un **outer join** es la consulta correlacionada que entrega todas las filas que están relacionadas, y además:

- las filas no relacionadas de la tabla izquierda (LEFT OUTER JOIN), ó
- las filas no relacionadas de la tabla derecha (RIGHT OUTER JOIN), ó
- las filas no relacionadas de ambas tablas (FULL OUTER JOIN)

Se considera como la tabla izquierda, a aquella que se menciona primero en la cláusula FROM.

Ejercicio 7.8: Uso de OUTER JOIN

En la base de datos **QhatuPERU** la tabla ARTICULO es una lista de todos los artículos comercializados por la empresa. Cuando un artículo se despacha a una tienda, el movimiento se registra en el detalle de una guía de envío. El siguiente diagrama muestra la relación entre las tablas ARTICULO y GUIA_DETALLE.



Se desea conocer si existen artículos que aún no han sido despachados ni una sola vez a las tiendas; es decir, artículos que están registrados en la tabla ARTICULO, pero que no tienen registro en la tabla GUIA_DETALLE.

La siguiente consulta muestra que artículos están registrados en la tabla GUIA_DETALLE; es decir, artículos que han sido enviados a las tiendas.

```
SELECT ARTICULO.CodArticulo,
     ARTICULO.DescripcionArticulo,
     GUIA_DETALLE.CantidadEnviada
FROM ARTICULO INNER JOIN GUIA_DETALLE
     ON ARTICULO.CodArticulo =
          GUIA_DETALLE.CodArticulo
ORDER BY ARTICULO.CodArticulo
go
```

Resultados | Mensajes

	CodArticulo	DescripcionArticulo	CantidadEnviada
1	1	CARAMELOS BASTON VIENA ARCOR	20
2	1	CARAMELOS BASTON VIENA ARCOR	20
3	1	CARAMELOS BASTON VIENA ARCOR	20
4	1	CARAMELOS BASTON VIENA ARCOR	20
5	1	CARAMELOS BASTON VIENA ARCOR	20
6	1	CARAMELOS BASTON VIENA ARCOR	20
7	1	CARAMELOS BASTON VIENA ARCOR	20
8	1	CARAMELOS BASTON VIENA ARCOR	20
9	1	CARAMELOS BASTON VIENA ARCOR	20
10	1	CARAMELOS BASTON VIENA ARCOR	20

El resultado muestra que en el caso del artículo **1**, éste se ha enviado en más de una ocasión a las tiendas. Observe que en este resultado no se muestra el artículo **5**.

Como el requerimiento es obtener una lista de los artículos que no se ha despachado a las tiendas, vamos a convertir la consulta anterior en una consulta OUTER JOIN. La consulta debe ser LEFT OUTER JOIN porque la tabla izquierda es la tabla ARTICULO, y lo que buscamos son registros de ARTICULO que no tengan relación con los registros de GUIA_DETALLE.

```
SELECT ARTICULO.CodArticulo,
     ARTICULO.DescripcionArticulo,
     GUIA_DETALLE.CantidadEnviada
FROM ARTICULO LEFT OUTER JOIN GUIA_DETALLE
     ON ARTICULO.CodArticulo =
          GUIA_DETALLE.CodArticulo
ORDER BY ARTICULO.CodArticulo
go
```

Resultados | Mensajes

	CodArticulo	DescripcionArticulo	CantidadEnviada
75	4	CARAMELOS FRUTAS MASTICABLES	20
76	4	CARAMELOS FRUTAS MASTICABLES	20
77	4	CARAMELOS FRUTAS MASTICABLES	20
78	4	CARAMELOS FRUTAS MASTICABLES	20
79	4	CARAMELOS FRUTAS MASTICABLES	20
80	4	CARAMELOS FRUTAS MASTICABLES	20
81	5	CHUPETES LOLY AMBROSOLI	NULL
82	6	FRUNA SURTIDA DONOFRIO	30
83	6	FRUNA SURTIDA DONOFRIO	30
84	6	FRUNA SURTIDA DONOFRIO	30

Por definición, una consulta OUTER JOIN muestra las filas relacionadas y las filas no relacionadas. Observe que para el artículo **5** el valor en la columna **CantidadEnviada** es NULL (dato desconocido ó dato no disponible).

Filtraremos el resultado de la consulta para que solo muestre los artículos cuyo valor en **CantidadEnviada** es NULL.

```
SELECT ARTICULO.CodArticulo,
     ARTICULO.DescripcionArticulo,
     GUIA_DETALLE.CantidadEnviada
FROM ARTICULO LEFT OUTER JOIN GUIA_DETALLE
     ON ARTICULO.CodArticulo =
          GUIA_DETALLE.CodArticulo
WHERE GUIA_DETALLE.CantidadEnviada IS NULL
ORDER BY ARTICULO.CodArticulo
go
```

Resultados | Mensajes

	CodArticulo	DescripcionArticulo	CantidadEnviada
1	5	CHUPETES LOLY AMBROSOLI	NULL
2	13	CHOCOLATE BARRA MILKY WAY	NULL
3	14	SNICKERS BAR KING SIZE	NULL
4	15	CHOCOLATE BARRA MILK DOVE	NULL
5	16	CHOCOLATE BARRA DARK DOVE	NULL
6	17	MILKY WAY BAR KING SIZE	NULL
7	18	GALLETAS CHIPS AHOY	NULL
8	19	GALLETAS TUAREG COSTA	NULL
9	20	GALLETAS VAINILLA COSTA	NULL
10	21	GALLETAS SURTIDAS BUTTER COOKIES	NULL

Hay 72 artículos que NO tienen registrada salida del almacén, que sumados a los 66 artículos que tienen salida registrada, nos da un total de 138 artículos.

Eliminamos de la consulta la columna **CantidadEnviada**. La consulta queda como sigue:

```
SELECT ARTICULO.CodArticulo,
     ARTICULO.DescripcionArticulo
FROM ARTICULO LEFT OUTER JOIN GUIA_DETALLE
     ON ARTICULO.CodArticulo =
          GUIA_DETALLE.CodArticulo
WHERE GUIA_DETALLE.CantidadEnviada IS NULL
ORDER BY ARTICULO.CodArticulo
go
```

Ejercicio 7.9: Reporte de unidades enviadas de cada artículo

Diseñe una consulta que muestre cuántas unidades se han enviado en total a las tiendas para cada uno de los artículos. El resultado debe mostrar dicho valor para todos los artículos.

```
SELECT ARTICULO.CodArticulo,
     ARTICULO.DescripcionArticulo,
     SUM(GUIA_DETALLE.CantidadEnviada)
          AS TotalEnviado
```

```
FROM GUIA_DETALLE INNER JOIN ARTICULO
     ON GUIA_DETALLE.CodArticulo =
          ARTICULO.CodArticulo
GROUP BY ARTICULO.CodArticulo,
     ARTICULO.DescripcionArticulo
ORDER BY ARTICULO.CodArticulo
go
```

Resultados | Mensajes

	CodArticulo	DescripcionArticulo	TotalEnviado
1	1	CARAMELOS BASTON VIENA ARCOR	400
2	2	CARAMELOS SURTIDO DE FRUTAS	400
3	3	CARAMELOS FRUTAS SURTIDA ARCOR	400
4	4	CARAMELOS FRUTAS MASTICABLES	400
5	6	FRUNA SURTIDA DONOFRIO	600
6	7	CHOCOLATE DOÑA PEPA FIELD	500
7	8	CHOCOLATE CUA CUA FIELD	500
8	9	MELLOWS FAMILIAR FIELD	400
9	10	WAFER CHOCOLATE FIELD	400
10	11	CHOCOLATE BARRA REGULAR	1000

Note que el resultado no muestra todos los artículos. Para que se muestren todos los artículos convierta la consulta INNER JOIN en una consulta OUTER JOIN, y para los artículos en los que **CantidadEnviada** es NULL que se muestre 0 (cero) como el valor en **TotalEnviado**.

```
SELECT ARTICULO.CodArticulo,
     ARTICULO.DescripcionArticulo,
     ISNULL(SUM(GUIA_DETALLE.CantidadEnviada), 0)
          AS TotalEnviado
FROM GUIA_DETALLE RIGHT OUTER JOIN ARTICULO
     ON GUIA_DETALLE.CodArticulo =
          ARTICULO.CodArticulo
GROUP BY ARTICULO.CodArticulo,
     ARTICULO.DescripcionArticulo
ORDER BY ARTICULO.CodArticulo
go
```

Resultados | Mensajes

	CodArticulo	DescripcionArticulo	TotalEnviado
1	1	CARAMELOS BASTON VIENA ARCOR	400
2	2	CARAMELOS SURTIDO DE FRUTAS	400
3	3	CARAMELOS FRUTAS SURTIDA ARCOR	400
4	4	CARAMELOS FRUTAS MASTICABLES	400
5	5	CHUPETES LOLY AMBROSOLI	0
6	6	FRUNA SURTIDA DONOFRIO	600
7	7	CHOCOLATE DOÑA PEPA FIELD	500
8	8	CHOCOLATE CUA CUA FIELD	500
9	9	MELLOWS FAMILIAR FIELD	400
10	10	WAFER CHOCOLATE FIELD	400

La consulta debe ser RIGHT OUTER JOIN porque la tabla ARTICULO se encuentra a la derecha y se desea mostrar todos los artículos: los que tienen relación con la tabla GUIA_DETALLE, y los que no la tienen.

4. Cross Join

Un **cross join** es la consulta correlacionada que combina cada una de las filas de una de las tablas con todas las filas de la otra tabla.

No es necesario que exista una columna común para ejecutar cross join. Esta consulta también se conoce como el **producto cartesiano** de dos tablas.

Ejercicio 7.10: Consulta CROSS JOIN

```
SELECT LINEA.Descripcion,
     ARTICULO.DescripcionArticulo
FROM LINEA CROSS JOIN ARTICULO
ORDER BY LINEA.Descripcion
go
```

Resultados | Mensajes

	Descripcion	DescripcionArticulo
1	DETERGENTES,DESINFECTANTES,ACCESORIOS	7 UP DESCARTABLE
2	DETERGENTES,DESINFECTANTES,ACCESORIOS	ACEITE BABY JOHNSONS
3	DETERGENTES,DESINFECTANTES,ACCESORIOS	ACEITE BABY JOHNSONS C/ALOE Y VIT. E
4	DETERGENTES,DESINFECTANTES,ACCESORIOS	ACEITE BABY JOHNSONS C/ALOE Y VIT. E
5	DETERGENTES,DESINFECTANTES,ACCESORIOS	ACEITE JOHNSONS
6	DETERGENTES,DESINFECTANTES,ACCESORIOS	ACEITE JOHNSONS CREMOSO
7	DETERGENTES,DESINFECTANTES,ACCESORIOS	ACEITE P/BEBES CHICCO
8	DETERGENTES,DESINFECTANTES,ACCESORIOS	ACEITE P/BEBES DR. ZAIDMAN
9	DETERGENTES,DESINFECTANTES,ACCESORIOS	ACEITE P/BEBES NINET
10	DETERGENTES,DESINFECTANTES,ACCESORIOS	CARAMELOS BASTON VIENA ARCOR

La tabla LINEA tiene 6 líneas:

```
SELECT COUNT(*) FROM LINEA
go
```

La tabla ARTICULO tiene 138 artículos:

```
SELECT COUNT(*) FROM ARTICULO
go
```

El resultado de la consulta CROSS JOIN entrega 6 x 138 = 828 filas.

5. AUTOJOIN

Es una consulta correlacionada en la que una tabla se combina consigo misma para obtener un nuevo conjunto de resultados.

Ejercicio 7.11: Consulta AUTOJOIN

Para ilustrar este caso vamos a crear una tabla que tenga una autorrelación; es decir, que tenga una clave foránea que apunte a la clave primaria de la misma tabla.

Ejecute las siguientes instrucciones para crear la tabla y cargar sus datos:

```
USE MarketPERU
go

-- Consulta autojoin,
-- creación de la tabla con autorelación
CREATE TABLE Trabajador(
   idTrabajador int PRIMARY KEY,
   Apellidos varchar(30) not null,
   Jefe int null )
go

ALTER TABLE Trabajador
   ADD CONSTRAINT fk_Trabajador_Trabajador
   FOREIGN KEY(Jefe)
   REFERENCES Trabajador
go

INSERT INTO Trabajador
   VALUES(102, 'Ardiles Soto', NULL)
INSERT INTO Trabajador
   VALUES(101, 'Camacho Saravia', 102)
INSERT INTO Trabajador
   VALUES(105, 'Vilchez Santos', 102)
INSERT INTO Trabajador
   VALUES(103, 'Sánchez Aliaga', 101)
INSERT INTO Trabajador
   VALUES(104, 'Castro Avila', 101)
INSERT INTO Trabajador
   VALUES(107, 'Urrunaga Tapia', 101)
INSERT INTO Trabajador
   VALUES(106, 'Juárez Pinto', 105)
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	Urunaga Tapia	101

La columna **Jefe** de la tabla **Trabajador** registra el código del jefe de un trabajador. Por ejemplo, el trabajador **101 (Camacho Saravia)** es el jefe de los trabajadores **103**, **104** y **107**.

Se desea crear una consulta que muestre una lista de trabajadores. La lista debe mostrar los **apellidos** del **jefe** de cada trabajador.

```
SELECT T1.idTrabajador, T1.apellidos,
   T2.apellidos AS Jefe
FROM Trabajador T1 INNER JOIN Trabajador T2
   ON T1.jefe = T2.idTrabajador
go
```

Resultados | Mensajes

	idTrabajador	apellidos	Jefe
1	101	Camacho Saravia	Ardiles Soto
2	103	Sánchez Aliaga	Camacho Saravia
3	104	Castro Avila	Camacho Saravia
4	105	Vilchez Santos	Ardiles Soto
5	106	Juárez Pinto	Vilchez Santos
6	107	Urunaga Tapia	Camacho Saravia

Note que el resultado muestra a todos los trabajadores con su respectivo jefe, pero el trabajador **102 (Ardiles Soto)** no aparece en la lista porque él no tiene **jefe**.

Modifique la consulta para que también se muestre al trabajador **102**.

```
SELECT T1.idTrabajador, T1.apellidos,
   T2.apellidos AS Jefe
FROM Trabajador T1 LEFT OUTER JOIN Trabajador T2
   ON T1.jefe = T2.idTrabajador
go
```

Resultados | Mensajes

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

6. SUBCONSULTAS

Una subconsulta es una declaración SELECT anidada dentro una sentencia SELECT, INSERT, UPDATE o DELETE o dentro de otra subconsulta.

Si la respuesta a un requerimiento de datos requiere la ejecución de una serie de pasos lógicos, utilice subconsultas para tratar de resolver el requerimiento con una sola sentencia.

Las subconsultas son de los tipos siguientes:

- Subconsulta que entrega un solo valor (1 fila, 1 columna)
- Subconsulta que entrega un conjunto de valores (varias filas, 1 columna)

Una subconsulta se especifica entre paréntesis, y se puede especificar en cualquier donde la sintáxis permite una expresión.

6.1. Subconsulta que entrega un solo valor (1 fila, 1 columna)

Cuando la subconsulta se especifica:

- en la lista de columnas del SELECT externo, ó
- en la cláusula WHERE del SELECT externo usando un operador relacional (test de comparación),

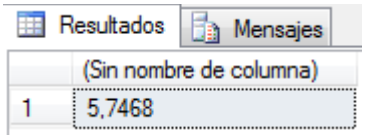
la subconsulta debe ser una que entregue un solo valor.

Ejercicio 7.12: Subconsulta definida en la lista de columnas del SELECT externo

Genere una consulta que entregue la lista de precios de todos los artículos, especificando en una columna adicional la diferencia entre el precio de cada artículo y el precio promedio de todos los artículos.

Primero, especifique la consulta que entrega el precio promedio de todos los artículos.

```
SELECT AVG(PrecioProveedor) FROM ARTICULO
go
```



Resultados | Mensajes

	(Sin nombre de columna)
1	5,7468

Ahora, escriba la consulta que entrega la lista de precios solicitada teniendo en cuenta la fórmula que determina la diferencia entre el precio de cada artículo y el precio promedio de todos los artículos.

```
SELECT CodArticulo, DescripcionArticulo,
   PrecioProveedor,
   Diferencia = PrecioProveedor -
      (SELECT AVG(PrecioProveedor) FROM ARTICULO)
FROM ARTICULO
go
```

Resultados | Mensajes

	CodArticulo	DescripcionArticulo	PrecioProveedor	Diferencia
1	1	CARAMELOS BASTON VIENA ARCOR	1,50	-4,2468
2	2	CARAMELOS SURTIDO DE FRUTAS	1,00	-4,7468
3	3	CARAMELOS FRUTAS SURTIDA ARCOR	1,50	-4,2468
4	4	CARAMELOS FRUTAS MASTICABLES	1,30	-4,4468
5	5	CHUPETES LOLY AMBROSOLI	1,20	-4,5468
6	6	FRUNA SURTIDA DONOFRIO	1,80	-3,9468
7	7	CHOCOLATE DOÑA PEPA FIELD	2,20	-3,5468
8	8	CHOCOLATE CUA CUA FIELD	1,60	-4,1468
9	9	MELLOWS FAMILIAR FIELD	2,10	-3,6468
10	10	WAFER CHOCOLATE FIELD	0,70	-5,0468

Ejercicio 7.13: Porcentaje despachado de cada artículo respecto al total despachado para la línea X

Escriba una consulta que determine el porcentaje de unidades despachadas de cada artículo de la línea 4 respecto al total despachado de la línea.

Primero, escriba la consulta que calcula el total despachado para la línea 4.

```
SELECT SUM(CantidadEnviada)
FROM GUIA_DETALLE INNER JOIN ARTICULO
   ON GUIA_DETALLE.CodArticulo =
      ARTICULO.CodArticulo
WHERE ARTICULO.CodLinea = 4
go
```

Resultados | Mensajes

	(Sin nombre de columna)
1	10625

Ahora, escriba la consulta que, utilizando la consulta anterior, presente el listado requerido.

```
SELECT ARTICULO.CodArticulo,
```

```
   ARTICULO.DescripcionArticulo,
   Despachado =
      ISNULL(SUM(GUIA_DETALLE.CantidadEnviada), 0),
   Porcentaje =
   CONVERT(float,
      ISNULL(SUM(GUIA_DETALLE.CantidadEnviada), 0))
         /
         (SELECT SUM(CantidadEnviada)
            FROM GUIA_DETALLE INNER JOIN ARTICULO
            ON GUIA_DETALLE.CodArticulo =
                 ARTICULO.CodArticulo
            WHERE ARTICULO.CodLinea = 4) * 100
FROM ARTICULO LEFT OUTER JOIN GUIA_DETALLE
   ON ARTICULO.CodArticulo =
      GUIA_DETALLE.CodArticulo
WHERE ARTICULO.CodLinea = 4
GROUP BY ARTICULO.CodArticulo,
   ARTICULO.DescripcionArticulo
go
```

Resultados | Mensajes

	CodArticulo	DescripcionArticulo	Despachado	Porcentaje
1	92	CREMA DE LECHE LAIVE	425	4
2	93	CREMA DE LECHE DUPRE	425	4
3	94	CREMA DE LECHE NESTLE	0	0
4	95	YOGURT GLORIA FRESA	850	8
5	96	YOGURT YOLEIT FRESA	0	0
6	97	YOGURT LAIVE FRESA	850	8
7	98	CREAM CHESSE LAIVE	0	0
8	99	CREMA DE QUESO LAIVE	0	0
9	100	QUESO CREMA MILKITO	425	4
10	101	MANTEQUILLA LAIVE C/SAL	1700	16

Ejercicio 7.14: Subconsulta definida en el WHERE del SELECT externo

Escriba una consulta que entregue una lista de los artículos que se despacharon en la fecha que se despachó la última salida del almacén. Tenga en cuenta que en dicha fecha se puede haber registrado más de una salida.

Primero, obtenga la fecha de la última salida

```
SELECT MAX(FechaSalida) FROM GUIA_ENVIO
go
```

Resultados | Mensajes

	(Sin nombre de columna)
1	2013-04-19 20:38:06.400

Ahora, utilizando adecuadamente la consulta anterior, escriba la consulta que responde al requerimiento especificado.

```
SELECT DISTINCT GUIA_DETALLE.CodArticulo,
   ARTICULO.DescripcionArticulo
FROM GUIA_DETALLE INNER JOIN ARTICULO
   ON GUIA_DETALLE.CodArticulo =
      ARTICULO.CodArticulo
INNER JOIN GUIA_ENVIO
   ON GUIA_DETALLE.NumGuia = GUIA_ENVIO.NumGuia
WHERE CONVERT(CHAR(10),
      GUIA_ENVIO.FechaSalida, 103) =
      (SELECT CONVERT(CHAR(10), MAX(FechaSalida),
         103) FROM GUIA_ENVIO)
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

Es necesario utilizar la función CONVERT() con las fechas a comparar, ya que al ser los datos de tipo **datetime**, se requiere eliminar la parte de la hora de las fechas para que la comparación sea solo de las fechas y no de la fecha y la hora.

6.2. Subconsulta que entrega un conjunto de valores (varias filas, 1 columna)

Cuando la subconsulta se define en la cláusula WHERE del SELECT externo utilizando el operador IN (test de pertenencia), puede ser una subconsulta que entrega un conjunto de valores.

Ejercicio 7.15: Test de pertenencia

Escriba una consulta que entregue una lista de los artículos que no registran envíos a las tiendas. Recuerde que este requerimiento fue resuelto líneas arriba utilizando una consulta OUTER JOIN.

```
SELECT CodArticulo, DescripcionArticulo
FROM ARTICULO
WHERE CodArticulo NOT IN
   (SELECT CodArticulo FROM GUIA_DETALLE)
ORDER BY CodArticulo
go
```

Resultados | Mensajes

	CodArticulo	DescripcionArticulo
1	5	CHUPETES LOLY AMBROSOLI
2	13	CHOCOLATE BARRA MILKY WAY
3	14	SNICKERS BAR KING SIZE
4	15	CHOCOLATE BARRA MILK DOVE
5	16	CHOCOLATE BARRA DARK DOVE
6	17	MILKY WAY BAR KING SIZE
7	18	GALLETAS CHIPS AHOY
8	19	GALLETAS TUAREG COSTA
9	20	GALLETAS VAINILLA COSTA
10	21	GALLETAS SURTIDAS BUTTER COOKIES

6.3. Subconsulta correlacionada

Se presenta cuando la consulta externa debe entregar datos a la consulta interna para que se pueda ejecutar.

- La consulta interna se evalúa repetidamente, una vez por cada fila de la consulta externa.
- Se puede definir en la cláusula WHERE de la consulta externa usando el operador EXISTS (Test de existencia).

Ejercicio 7.16: Test de existencia – Uso de EXISTS

Genere la lista de artículos que registran envíos a las tiendas.

```
SELECT ARTICULO.CodArticulo,
   ARTICULO.DescripcionArticulo
FROM ARTICULO
WHERE EXISTS
   (SELECT * FROM GUIA_DETALLE
      WHERE ARTICULO.CodArticulo =
         GUIA_DETALLE.CodArticulo)
ORDER BY ARTICULO.CodArticulo
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

7. 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.

Escriba las instrucciones SELECT para obtener:

1. (EDUCA) Un listado que incluya el nombre del curso con sus respectivos nombres de alumnos.
2. (EDUCA) Un reporte que muestre el nombre del alumno y la suma de todos sus pagos.
3. (EDUCA) Un reporte que muestre el nombre del curso y el importe de todos sus pagos.
4. (RH) Un reporte que muestre el nombre del departamento y el importe de su planilla.
5. (RH) La cantidad de trabajadores en cada ciudad.
6. (RH) Un listado de todos los departamentos y la cantidad de trabajadores en cada uno de ellos.
7. (EDUCA) La cantidad de alumnos matriculados en cada curso, debe incluir en el listado todos los cursos.
8. (EDUCA) Un reporte que para cada curso muestre la cantidad de alumnos matriculados y el importe recaudado por los pagos realizados por los alumnos.
9. (RH) Todas las posibles combinaciones entre las tabla departamento y cargo.
10. (RH) Un listado de los empleados con el respectivo nombre de su superior inmediato.

Utilizando subconsultas escriba las instrucciones para obtener:

11. (RH) Quiénes son los empleados que tienen el menor sueldo.
12. (RH) Un listado de los empleados mostrando su sueldo y la diferencia con el sueldo promedio de la empresa.

13. (RH) Un reporte que muestre las personas que tienen el menor sueldo por departamento.

14. (RH) Un listado que muestre la cantidad de empleados y el importe de la planilla por departamento.

15. (EDUCA) Un reporte que muestre los alumnos matriculados en el curso SQL Server Implementación.

16. (RH) Un reporte que muestre los empleados que ocupan alguna gerencia.

17. (RH) Un reporte que muestre a los empleados que laboran en Trujillo.

18. (RH) Un reporte que muestre a los empleados que no laboran en Lima.