

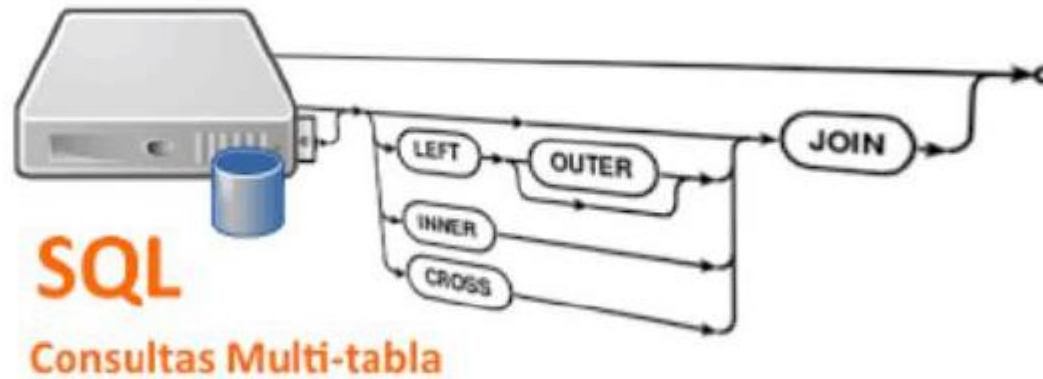
SESIÓN
03

CONSULTAS MULTITABLAS

- Operadores INNER JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN Y CROSS JOIN
- La función CASE
- Operador UNION

/ OBJETIVOS

En la mayoría de los casos, la recuperación de los datos que los usuarios necesitan para trabajar implica la lectura de muchas tablas para que la información así obtenida sea de utilidad para ellos. 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 el diseño de las instrucciones SELECT que nos permiten recuperar datos de varias 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.

/ EL OPERADOR INNER JOIN

Una combinación interna (INNER JOIN) es aquélla en la que los valores de las columnas que se están combinando se comparan mediante un operador de comparación.

Esta combinación interna se conoce como una combinación equivalente. Devuelve todas las columnas de ambas tablas y sólo devuelve las filas en las que haya un valor igual en la columna de la combinación.

Veamos el siguiente caso:

“Elabore una consulta donde se deban mostrar los campos razón social, dirección, teléfono y el Id de ubigeo del cliente y así como el departamento, provincia y distrito donde esta ubicado”

/ EL OPERADOR INNER JOIN

Para elaborar la consulta requerida, se deben recurrir a las tablas Tb_Cliente y Tb_Ubigeo y combinarlas en base a la igualdad de los campos Id_Ubigeo que tienen en común ambas tablas

```
SELECT C.Raz_soc_cli, C.Dir_cli, C.Tel_cli,C.Id_ubigeo,
U.Departamento, U.Provincia, U.Distrito
FROM Tb_Cliente C INNER JOIN Tb_Ubigeo U
ON U.Id_Ubigeo = C.Id_Ubigeo
GO
```

Results Messages

Raz_soc_cli	Dir_cli	Tel_cli	Id_ubigeo	Departamento	Provincia	Distrito
A & V METALEN PRODUCT S.A.C.	Jr. Ezekiel Dorsey 8665	957522407	040101	AREQUIPA	AREQUIPA	AREQUIPA
A + A SERVICIOS EMPRESARIALES SOCIEDAD ANONIMA CE...	Jr. Brielle Kirby 9237	937639972	040101	AREQUIPA	AREQUIPA	AREQUIPA
A & Z E.I.R.LTDA.	Jr. Thor Franklin 3434	916747506	040102	AREQUIPA	AREQUIPA	CAYMA
A Y M PALLETS S.A.C.	Av. Kevin Estes 6963	986244443	040102	AREQUIPA	AREQUIPA	CAYMA
A & Z CONSTRUCTORES S.A.C.	Jr. Gareth Anthony 9681	964098866	040103	AREQUIPA	AREQUIPA	CERRO COLORADO
A & S ASESORES EN SISTEMAS S.R.L.	Av. Jenette Parsons 3227	905363998	040103	AREQUIPA	AREQUIPA	CERRO COLORADO
A & S CONSULTORES Y EJECUTORES E.I.R.L.	Av. Montana Cannon 2615	996900870	040103	AREQUIPA	AREQUIPA	CERRO COLORADO
A & R TOURS SOCIEDAD COMERCIAL DE RESPONSABILIDA...	Av. Jonas Ayala 9823	965498504	040104	AREQUIPA	AREQUIPA	CHARACATO
A Y D ASOCIADOS S A C	Jr. Gemma Santos 8699	954573496	040105	AREQUIPA	AREQUIPA	CHIGUATA
A Y G AGENCIA DE VIAJES Y TURISMO S.A.C	Jr. Fulton Ratliff 6289	974185909	040105	AREQUIPA	AREQUIPA	CHIGUATA
A&N LOGISTIC SOLUTIONS S.R.L.	Jr. Jery Mckenzie 5196	970624170	040105	AREQUIPA	AREQUIPA	CHIGUATA
A & S INVERSIONES SULCA S.A.C.	Av. Claire Salinas 5445	938866722	040106	AREQUIPA	AREQUIPA	LA JOYA
A & R ASESORES S.A.C.	Jr. Dane Day 9627	906160727	040107	AREQUIPA	AREQUIPA	MIRAFLORES
A DIOS SEA LA GLORIA S.A.C.	Av. Andrew Woodward 3966	923025709	040107	AREQUIPA	AREQUIPA	MIRAFLORES
A & R LOGISTIC CARGO INTERNATIONAL S.A.C.	Av. Jackson Morse 8381	926847255	040108	AREQUIPA	AREQUIPA	MOLLEBAYA
A & R SERVICIOS Y PROYECTOS INDUSTRIALES E.I.R.L.	Av. Jaime Moon 6899	942203263	040108	AREQUIPA	AREQUIPA	MOLLEBAYA

/ EL OPERADOR INNER JOIN

```
/* Solucion : */  
SELECT C.Raz_soc_cli, C.Dir_cli, C.Tel_cli, C.Id_ubigeo,  
U.Departamento, U.Provincia, U.Distrito  
FROM Tb_Cliente C INNER JOIN Tb_Ubigeo U ON U.Id_Ubigeo = C.Id_Ubigeo  
GO
```

Como se observa, la combinación interna se hace en base a la igualdad de el Id_Ubigeo de ambas tablas.

También se hace necesario el uso de ALIAS para identificar a cada una de las tablas que participan en la consulta. En el ejemplo se emplearon el alias “C” para identificar a la tabla Tb_Cliente y el alias “U” para identificar a la tabla Tb_Ubigeo. Sino empleara los alias tendría que estar repitiendo el nombre de cada tabla cuando sea necesario, lo cual haría mas pesada la redacción de la consulta.

NOTA: No es obligatorio (pero si recomendable) colocar los alias antes de cada campo seleccionado de la consulta. Solo es obligatorio para el campo común en las 2 tablas.

/ LOS OPERADORES LEFT Y RIGHT OUTER JOIN

Las combinaciones internas sólo devuelven filas cuando hay una fila de ambas tablas, como mínimo, que coincide con la condición de la combinación. Las combinaciones internas eliminan las filas que no coinciden con alguna fila de la otra tabla. Sin embargo, las combinaciones externas devuelven todas las filas de una de las tablas o vistas mencionadas en la cláusula FROM, como mínimo, siempre que tales filas cumplan con alguna de las condiciones de búsqueda. Todas las filas se recuperarán de la tabla izquierda a la que se haya hecho referencia con una combinación externa izquierda (LEFT OUTER JOIN), y de la tabla derecha a la que se haya hecho referencia con una combinación externa derecha (RIGHT OUTER JOIN).

Veamos un caso con LEFT OUTER JOIN

“Elabore una consulta con todos los números de facturas, así como su fecha de facturación, estado , la razón social y RUC del cliente que la generó, INCLUYENDO A TODOS LOS CLIENTES, HAYAN O NO GENERADO FACTURAS. Ordene el resultado por numero de factura.”

/ LOS OPERADORES LEFT Y RIGHT OUTER JOIN

Para elaborar la consulta requerida empleando LEFT OUTER JOIN , se deben recurrir a las tablas Tb_Factura y Tb_Cliente en base a la igualdad del campo común ambas tablas , es decir Cod_cli.

```
SELECT F.NUM_FAC, F.FEC_FAC, F.EST_FAC, C.RAZ_SOC_CLI, C.RUC_CLI
FROM TB_CLIENTE C LEFT OUTER JOIN TB_FACTURA F
ON C.COD_CLI=F.COD_CLI
ORDER BY F.NUM_FAC
GO
```

NUM_FAC	FEC_FAC	EST_FAC	RAZ_SOC_CLI	RUC_CLI
NULL	NULL	NULL	A & M INDUSTRIAS GENERALES SOCIEDAD COMERCIAL DE ...	20480493037
NULL	NULL	NULL	FUMIGACIONES LOSSIO S.A.C.	20503899796
NULL	NULL	NULL	FUMIGADORA IDIAQUEZ E.I.R.LTDA.	20320612811
FA0001	2017-08-18 00:00:00.000	2	FINSETH	48632081112
FA0002	2017-10-29 00:00:00.000	3	COREFO	57201691887
FA0003	2017-11-03 00:00:00.000	2	SERVIEMSA	75012403231
FA0004	2017-11-08 00:00:00.000	2	CARDELI	26403158295
FA0005	2017-11-13 00:00:00.000	2	MEBA	50319542920
FA0006	2017-11-22 00:00:00.000	3	PROMINENT	43233519100
FA0007	2017-12-07 00:00:00.000	1	COREFO	57201691887
FA0008	2017-12-11 00:00:00.000	2	SUCERTE	62014503344
FA0009	2017-12-12 00:00:00.000	2	INTECH	34021824991
FA0010	2017-02-22 00:00:00.000	2	PAYET	70594032992
FA0011	2017-12-23 00:00:00.000	2	COREFO	57201691887
FA0012	2018-01-25 00:00:00.000	1	KADIA	22202915028
FA0013	2018-02-02 00:00:00.000	1	COREFO	57201691887

/ LOS OPERADORES LEFT Y RIGHT OUTER JOIN

```
SELECT F.NUM_FAC, F.FEC_FAC, F.EST_FAC, C.RAZ_SOC_CLI, C.RUC_CLI  
FROM TB_CLIENTE C LEFT OUTER JOIN TB_FACTURA F  
ON C.COD_CLI=F.COD_CLI  
ORDER BY F.NUM_FAC  
GO
```

Si hubiéramos empleado INNER JOIN solo se hubieran seleccionado las facturas de clientes que hayan generado por lo menos una factura. Como se empleo LEFT OUTER JOIN, SQL entendió que se seleccionen todos los registros de la tabla que esta a la izquierda del operador, es decir Tb_Cliente, tengan o no referencia en la tabla de la derecha del operador , o sea Tb_Factura.

Si aprecia en resultado, se visualiza a 3 clientes donde los campos Num_fac, Fec_fac y Est_fac son nulos. Eso quiere decir que esos 3 clientes no han generado factura alguna hasta el momento.

/ LOS OPERADORES LEFT Y RIGHT OUTER JOIN

```
⇒ SELECT F.NUM_FAC, F.FEC_FAC, F.EST_FAC, C.RAZ_SOC_CLI, C.RUC_CLI  
FROM TB_FACTURA F RIGHT OUTER JOIN TB_CLIENTE C  
ON C.COD_CLI=F.COD_CLI  
ORDER BY F.NUM_FAC  
GO
```

Se puede dar solución a la misma consulta empleando RIGHT OUTER JOIN, pero cambiando la posición de las tablas (operandos), colocando Tb_Cliente a la derecha del operador y Tb_Factura a la izquierda.

Como se puede ver , para los casos analizados es fundamental el concepto de llaves Foraneas y Primarias, dado que en base a estas se hace la combinación entre las tablas que forman parte de la consulta.

/ OPERADOR CROSS JOIN

Este operador es también conocido como el producto cartesiano entre los registros de 2 tablas, dado que establece la combinación de los registros de las tablas involucradas. A diferencia de INNER y OUTER JOIN, CROSS JOIN no necesita que las tablas operandos tengan un campo en común. Por ejemplo, a continuación se muestra una combinación total de los registros de la tabla Tb_UnidadMedida y Tb_Producto.

```
SELECT U.Id_UM, U.Des_UM, P.Cod_pro, P.Des_Pro, P.Pre_pro  
from Tb_UnidadMedida U CROSS JOIN Tb_Producto P  
order by U.Id_UM  
go
```

Si ejecuta la consulta podrá observarse que cada registro de la tabla de unidades de medida se combina con cada registro de la tabla de productos.

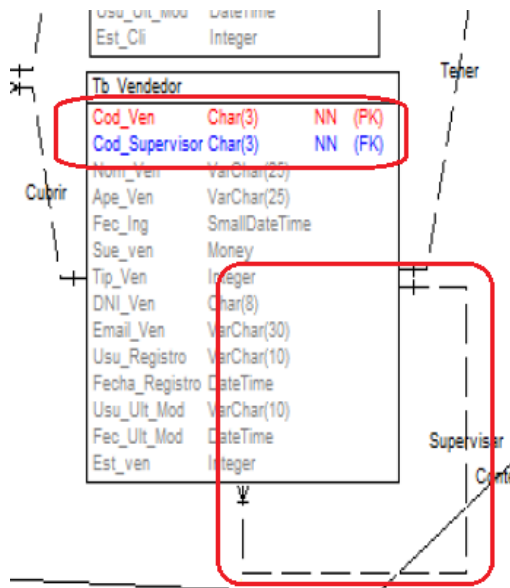
NOTA:

En escenarios reales, este operador no tiene mucho empleo, pero es posible que para algún caso en particular sea relevante su uso.

/ CASO ESPECIAL : JOIN RECURSIVO O AUTOJOIN

Este caso se da cuando una tabla se combina consigo misma, derivada de una relación recursiva definida desde el modelamiento de la base de datos.

En el caso de VentasLeon, este fenómeno se da en la tabla Tb_Vendedor:



dbo.Tb_Vendedor
Columns
Cod_Ven (PK, nchar(3), not null)
Nom_ven (nvarchar(25), null)
Ape_ven (nvarchar(25), null)
Sue_ven (money, null)
Fec_ing (datetime, null)
Tip_ven (int, null)
Dni_ven (nchar(8), null)
Email_ven (varchar(50), null)
Cod_Supervisor (FK, nchar(3), null)
Fec_Registro (datetime, null)
Usu_Registro (varchar(20), null)
Fec_Ult_Mod (datetime, null)
Usu_Ult_Mod (varchar(20), null)
Est_ven (int, null)

/ CASO ESPECIAL : JOIN RECURSIVO O AUTOJOIN

Como se aprecia, el código de supervisor (Cod_Supervisor) es la FK de la tabla, apuntando a la PK de su misma tabla, es decir Tb_Vendedor. Es por eso que se denomina AUTOJOIN o JOIN Recursivo. En si, las consultas se implementan con los operadores JOIN explicados, por lo que no se vaya a pensar que existe un operador AUTOJOIN, sino que se emplea este termino para identificar este fenómeno o caso especial. Veamos un ejemplo:

“Seleccione los nombres y apellidos de los vendedores con el nombre y apellido de su respectivo supervisor”

/ CASO ESPECIAL : JOIN RECURSIVO O AUTOJOIN

```
/*Seleccione los nombres y apellidos de los vendedores con el nombre y apellido  
de su respectivo supervisor*/  
SELECT V1.Cod_ven, V1.ape_ven+' '+ V1.nom_ven as ApeNomVendedor,  
       V2.ape_ven+' '+ V2.nom_ven as ApeNomSupervisor  
FROM Tb_Vendedor V1 INNER JOIN Tb_Vendedor V2 ON V1.Cod_Supervisor = V2.Cod_ven  
go
```

175 %

Results Messages

	Cod_ven	ApeNomVendedor	ApeNomSupervisor
24	V24	Paredes Perales Nathalie	Palacios Jose
25	V25	Ramos Llerena Ricardo	Vega Julio
26	V26	Querales Soto Alfonso	Abad Lopez Eva Maria
27	V27	Alvarez Ramos Pedro	Masias Juana
28	V28	Quispe Mayra	Zavala Eduardo
29	V29	Perez Sanchez Esmer...	Zavala Eduardo
30	V30	Flores Pilares Hemesto	Masias Juana
31	V31	Fuentes Rosales Alvaro	Zavala Eduardo
32	V32	Romero Gavallio Jean	Masias Juana
33	V33	Piemonte Rosales Willi...	Abad Lopez Eva Maria
34	V34	Palomino Puentes Miriam	Vega Julio
35	V35	Aguilar Paz Ruben	Martinez Felipe
36	V36	Soto Campos Javier	Masias Juana
37	V37	Aguila Gavidia Natalia	Vega Julio
38	V38	Ramirez Tapia Juana	Zavala Eduardo
39	V39	Vigil Neto Aurelio	Abad Lopez Eva Maria

Como se aprecia el INNER JOIN se hace con la misma tabla de Vendedor en base a la igualdad del Cod_Ven (PK) con el Cod_Supervisor (FK). Los alias juegan un rol importante dado que el alias V1 identifica a los vendedores supervisados y el alias V2 a los supervisores.

/ La expression condicional CASE:

CASE es una de las más potentes y más complejas expresiones incorporadas en Transact-SQL. Debido a su nombre, esta expresión es regularmente confundida con la sentencia CASE disponible en otros lenguajes. En SQL Server, el propósito de la expresión CASE es devolver siempre una expresión. No tiene la intención de efectuar un control de flujo, es por eso que no la llamamos instrucción sino expresión CASE. Veamos un ejemplo :

“Seleccione el código , razón social , ruc y una descripción literal del estado del cliente si sabe que el est_cli 1 es Activo y el est_cli 0 es Inactivo.”

/ La expression condicional CASE:

```
Select cod_cli, raz_soc_cli , ruc_cli,  
case est_cli  
    when 1 then 'Activo'  
    when 0 then 'Inactivo'  
end as Estado  
from tb_cliente  
go
```

175 %

Results Messages

	cod_cli	raz_soc_cli	ruc_cli	est_cli	Estado
1	C001	FINSETH	48632081112	1	Activo
2	C002	ORBI	57031642221	0	Inactivo
3	C003	SERVIEMSA	75012403231	1	Activo
4	C004	ISSA	46720159214	1	Activo
5	C005	MASS	83175942998	0	Inactivo
6	C006	BERKER	54890124200	1	Activo
7	C007	FIDENZA	16204790012	1	Activo
8	C008	INTECH	34021824991	1	Activo
9	C009	PROMINENT	43233519100	1	Activo
10	C010	LANDU	30405261192	1	Activo
11	C011	EL ALIP	70245201222	1	Activo

Query executed successfully. LAPTOP

El uso adecuado de la expresión CASE nos permita definir campos tipo “flags” como en este caso el campo “est_cli”, donde solo con un valor 1 o 0 podemos definir el status del cliente y que mediante CASE podemos incluir como resultado de la consulta una mejor descripción de dicho status.

/ El operador UNION:

Este operador permite unir el resultado de 2 o mas Select, siempre y cuando estos resultados sean compatibles de ser reunidos. La compatibilidad de ser reunidos en un solo resultado depende de los campos seleccionados en cada uno de los Select que van a participar de la consulta

Veamos un ejemplo :

“ Seleccione los códigos de productos, descripción y precio de aquellos productos con stock actual mayor al stock mínimo y con un precio entre 10 y 50 en unión con los productos con un precio entre 60 y 100 que sean nacionales”

/ El operador UNION:

```
[-] Select cod_pro,des_pro, pre_pro
    from Tb_Producto
    where (pre_pro between 10 and 50) and stk_act>stk_min
    union
    Select cod_pro,des_pro, pre_pro
    from Tb_Producto
    where (pre_pro between 60 and 100) and importado=0
go
```

Como se aprecia, ambos select tienen la misma cabecera, lo que los hace compatibles a ser reunidos en un solo resultado.

/ RESUMEN

/ RESUMEN

- Hemos aprendido a realizar consultas multitablas, gracias a las posibles combinaciones que se pueden realizar entre ellas.
- Identificamos el rol importante que tienen las llaves foraneas (FK) para la implementacion de consultas multitablas coherentes.
- Aprendimos a emplear la expresion CASE para poder evaluar valores de un campo y poder establecer resultados basados en dicha evaluacion.
- Conocimos que es posible unir 2 o mas resultados de Select, siempre y cuando las estructuras de los resultados a reunir sean compatibles.

