

EXERCISES QUE TRABAJAREMOS EN EL CUE:

- EXERCISE 1: RELACIONES DE TABLAS.
- EXERCISE 2: REALIZANDO CONSULTAS MULTITABLAS.

EXERCISE 1: RELACIONES DE TABLAS.

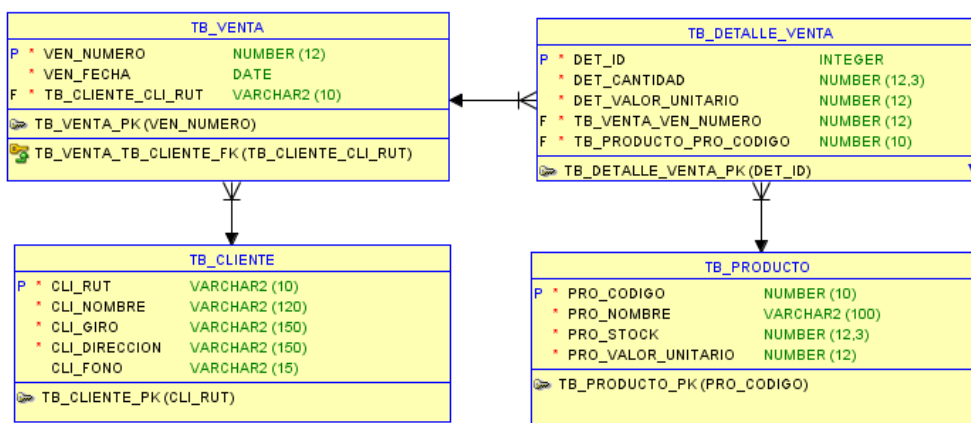
En la mayoría de las consultas intervienen más de una tabla, las cuales deben relacionarse para poder obtener las columnas requeridas, y así lograr el diseño esperado. Estas relaciones se logran gracias a las claves primarias (*PK*) y foráneas (*FK*).

Las operaciones más comunes que relacionan dos, o más tablas, son:



INNER JOIN ON

Supongamos el siguiente Modelo Relacional:



Donde se nos pide el siguiente listado del detalle de venta por año, ordenadas por fecha de venta.

Fecha de Venta	Nº de Venta	Total Venta

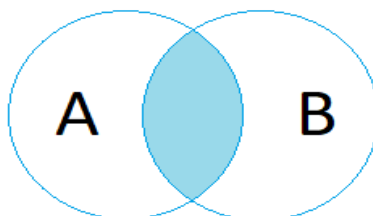
Para poder construir este listado o informe, debemos contar con el Modelo Relacional, y así diseñar o pensar nuestra consulta. Si revisamos dicho Modelo, se observará que los datos solicitados pueden ser obtenidos desde las tablas **TB_VENTA** y **TB_DETALLE_VENTA**, por lo que debemos relacionar ambas tablas mediante la relación indicada en el modelo. Por otra parte, las columnas del informe “Fecha de Venta” y “Nº de Venta”, se pueden obtener de forma directa, pero “Total Venta” no es directa, por lo que deben realizarse operaciones matemáticas para calcularla.

En primer lugar, construiremos una consulta que traiga todas las columnas que se requieren en el informe.

```
1 SELECT VEN_NUMERO, VEN_FECHA, DET_CANTIDAD, DET_VALOR_UNITARIO
2 FROM TB_VENTA AS V
3 INNER JOIN TB_DETALLE_VENTA AS DV ON V.VEN_NUMERO = DV.VEN_NUMERO
```

La sentencia **JOIN** relaciona **TB_VENTA** con **TB_DETALLE_VENTA**, a través del campo **VEN_NUMERO (PK)** en **TB_VENTA**, y el campo **VEN_NUMERO (FK)** en **TB_DETALLE_VENTA**. Esta relación significa que, para cada registro de **TB_VENTA**, se buscarán los registros de **TB_DETALLE_VENTA** que cumplan con la condición: **V.VEN_NUMERO = DV.VEN_NUMERO**.

Gráficamente, estamos obteniendo la intersección entre las dos tablas.



Luego, el subtotal de cada detalle de una venta lo podemos obtener de la siguiente forma:

```
1 SELECT VEN_NUMERO,  
2 VEN_FECHA,  
3 DET_CANTIDAD * DET_VALOR_UNITARIO AS SubTotal  
4 FROM TB_VENTA AS V  
5 INNER JOIN TB_DETALLE_VENTA AS DV ON V.VEN_NUMERO = DV.VEN_NUMERO
```

Ahora, debemos agrupar por los campos que se repiten, para poder obtener la suma de todos los subtotales con la función de agregado SUM, lo que representa el total de la venta. La consulta sería la siguiente:

```
1 SELECT VEN_NUMERO AS NumeroVenta,  
2 VEN_FECHA AS FechaVenta,  
3 SUM(DET_CANTIDAD * DET_VALOR_UNITARIO) AS TotalVenta  
4 FROM TB_VENTA AS V  
5 INNER JOIN TB_DETALLE_VENTA AS DV ON V.VEN_NUMERO = DV.VEN_NUMERO  
6 GROUP BY VEN_NUMERO, VEN_FECHA
```

Para finalizar, agregamos el filtro para clasificar y listar solo las ventas de un año.

```
1 SELECT VEN_NUMERO AS NumeroVenta,  
2 VEN_FECHA AS FechaVenta,  
3 SUM(DET_CANTIDAD * DET_VALOR_UNITARIO) AS TotalVenta  
4 FROM TB_VENTA AS V  
5 INNER JOIN TB_DETALLE_VENTA AS DV ON V.VEN_NUMERO = DV.VEN_NUMERO  
6 WHERE to_number(to_char(VEN_FECHA, 'YYYY')) = 2020  
7 GROUP BY VEN_NUMERO, VEN_FECHA
```

Al relacionar mediante la sentencia **JOIN**, se puede hacer por más de un campo, e incluso, con cualquier operador, no necesariamente con uno igual.

```
1 ...  
2 FROM A  
3 INNER JOIN B ON A.C1 = B.C2 AND A.C2 = B.C3  
4 ...
```

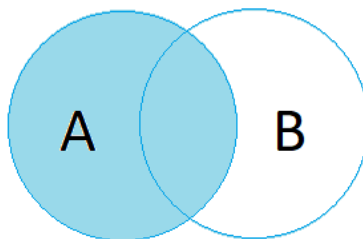
```
1 ...  
2 FROM A  
3 INNER JOIN B ON A.C1 = B.C2 OR A.C2 > B.C3  
4 ...
```

```
1 ...
2 FROM A
3 INNER JOIN B ON expresión
4 ...
```

LEFT JOIN ON

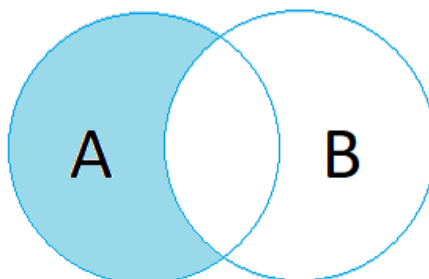
La sintaxis es idéntica a **INNER JOIN**, con la diferencia que se incluirán todos los elementos de la tabla, a la izquierda de la sentencia.

```
1 SELECT *
2 FROM A
3 LEFT JOIN B ON A.ColumnaN = B.ColumnaM
```



Agregando un criterio, podemos obtener la diferencia A – B:

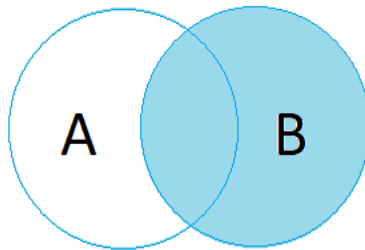
```
1 SELECT *
2 FROM A
3 LEFT JOIN B ON A.ColumnaN = B.ColumnaM
4 WHERE B.ColumnaM IS NULL
```



RIGHT JOIN ON

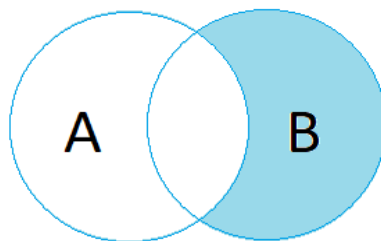
La sintaxis es idéntica a **INNER JOIN**, con la diferencia que se incluirán todos los elementos de la tabla, a la derecha de la sentencia.

```
1 SELECT *  
2 FROM A  
3 RIGHT JOIN B ON A.ColumnaN = B.ColumnaM
```

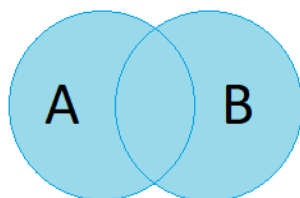


Agregando un criterio, podemos obtener la diferencia A – B:

```
1 SELECT *  
2 FROM A  
3 RIGHT JOIN B ON A.ColumnaN = B.ColumnaM  
4 WHERE B.ColumnaM IS NULL
```



UNIÓN



La palabra UNION permite añadir el resultado de un SELECT, a otro SELECT. Para ello, ambas instrucciones tienen que utilizar el mismo número y tipo de columnas.

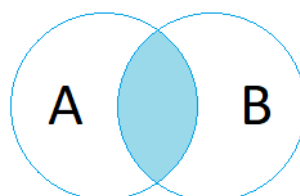
Ejemplo:

```
1 SELECT nombre FROM clientes
2 UNION
3 SELECT nombre FROM proveedores
```

El resultado serán los nombres de los clientes y proveedores que no se encuentren duplicados. La UNION retorna el resultado de las filas de ambas consultas que no se encuentren duplicados. Para el caso de UNION ALL, el funcionamiento es idéntico, pero éste no elimina los datos repetidos.

```
1 SELECT nombre FROM clientes
2 UNION ALL
3 SELECT nombre FROM proveedores
```

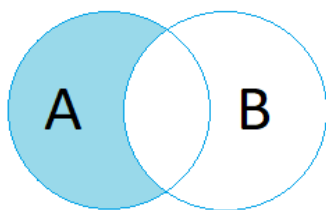
INTERSECCIÓN



La sentencia INTERSECT permite unir dos consultas SELECT, de modo que el resultado serán las filas que estén presentes en ambas consultas, es decir, la intercepción de ambas.

Ejemplo:

```
1 SELECT nombre FROM clientes
2 INTERSECT
3 SELECT nombre FROM proveedores
```

DIFERENCIA

Con MINUS también se combinan dos consultas SELECT, de forma que aparecerán los registros del primer SELECT que no estén presentes en el segundo.

Ejemplo:

```
1 SELECT nombre, marca
2 FROM productos
3 WHERE tipo=1
4 MINUS
5 SELECT nombre, marca
6 FROM productos
7 WHERE tipo=2
```

PRODUCTO CARTESIANO

Opera sobre dos o más tablas, efectuando un producto cartesiano del contenido de éstas, no siendo necesario que ambas posean la misma estructura, y devolviendo una nueva, cuyo contenido es todas las posibles combinaciones de las filas de una de las tablas.

```
1 SELECT ... FROM T1, T2, ..., Tn
```

Supongamos que tenemos las tablas **Trabajador** y **Comuna**, con los datos que se indican a continuación:

Trabajador	
Nombre	CodigoComuna
Juan	2
Pedro	1
Carlos	1

Comuna	
Codigo	NombreComuna
1	Santiago
2	Concepción

Si queremos realizar el producto cartesiano entre ambas tablas, podríamos desarrollar la siguiente operación:

```
1 SELECT Nombre, NombreComuna
2 FROM Trabajador, Comuna
3 ORDER BY Trabajador, Comuna
```

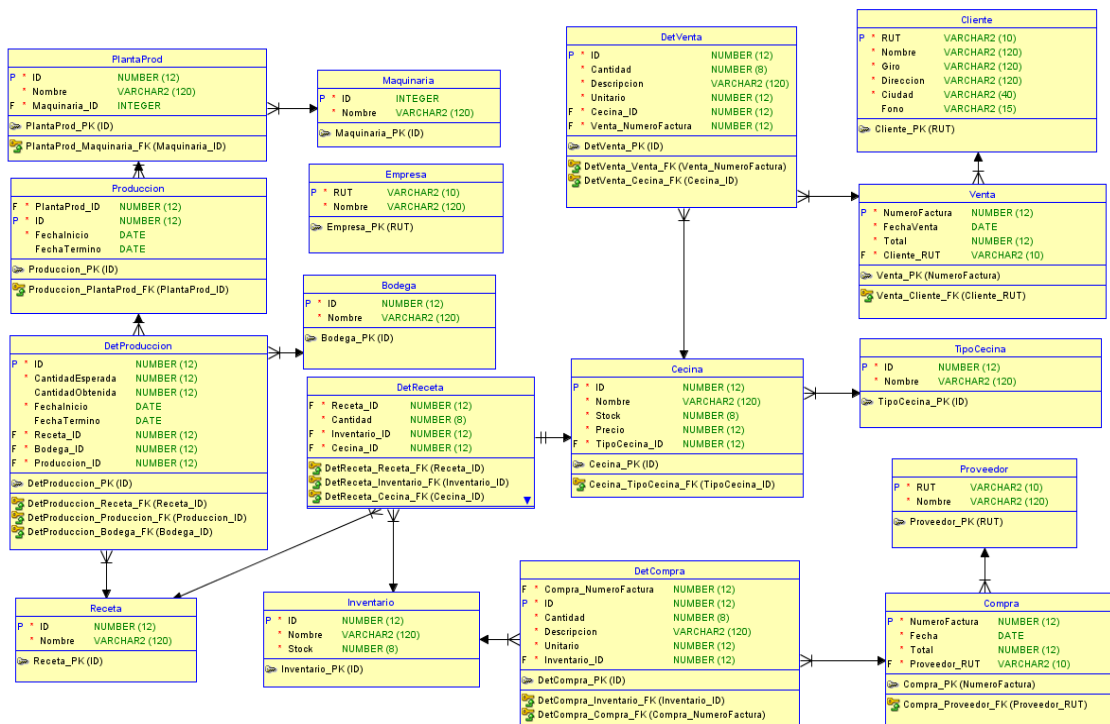
El resultado sería:

Nombre	NombreComuna
Carlos	Concepción
Carlos	Santiago
Juan	Concepción
Juan	Santiago
Pedro	Concepción
Pedro	Santiago

Esto representa como si los trabajadores ejercen en todas las comunas, y lo que se realizaron fueron todas las combinaciones posibles entre ambas tablas.

EXERCISE 2: REALIZANDO CONSULTAS MULTITABLAS.

Realizaremos consultas multitablas, basándonos en el siguiente Modelo Entidad - Relación:



Para realizar esta ejemplificación, trabajaremos en SQL Developer.

Lo primero que haremos será listar las compras del año 2020, con los atributos: Fecha, Total, NombreProveedor, RutProveedor, y Cantidad de ítems comprados.

Para esto, debemos relacionar las tablas "Compra" y "DetCompra". Éstas se relacionan mediante el atributo NumeroFactura. Por este motivo, escribiremos el comando:

```
1 SELECT * FROM COMPRA INNER JOIN DETCOMPRA ON NUMEROFACTURA =
2 COMPRA_NUMEROFACTURA
```

Obteniendo como resultado las columnas de ambas tablas:

Resultado de la Consulta x									
SQL Todas las Filas Recuperadas: 0 en 0,002 segundos									
NUMER...	FECHA	TOTAL	PROVEE...	ID	CANTIDAD	DESCRL...	UNITARIO	COMPR...	INVENT...

Filtraremos los atributos correspondientes. Para eso, debemos crear la relación de las tablas con la entidad proveedora.

Agregamos en la siguiente línea: **INNER JOIN PROVEEDOR ON PROVEEDOR_RUT = RUT**, quedando el comando total así:

```
1 SELECT * FROM COMPRA INNER JOIN DETCOMPRA ON NUMEROFACTURA =
2 COMPRA_NUMEROFACTURA
3 INNER JOIN PROVEEDOR ON PROVEEDOR_RUT = RUT;
```

Para filtrar la lista de las compras del año 2020, usaremos el comando:

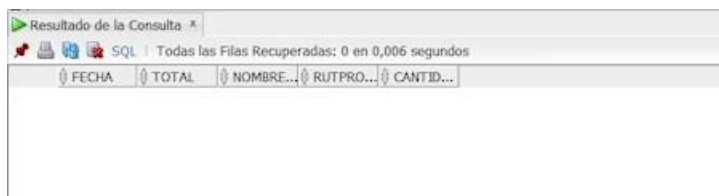
```
1 WHERE TO_CHAR (FECHA, 'YYYY') = '2020';
```

Utilizamos **TO_CHAR** para transformar el formato Fecha de **DATE**, a **CHARACTER**.

Continuamos reemplazando la primera parte del comando, por: **SELECT FECHA, TOTAL, NOMBRE AS NOMBREPROVEEDOR, RUT AS RUTPROVEEDOR, 0 AS CANTIDADITEM FROM COMPRA**, quedando finalmente:

```
1 SELECT FECHA,
2 TOTAL,
3 NOMBRE AS NOMBREPROVEEDOR,
4 RUT AS RUTPROVEEDOR,
5 0 AS CANTIDADITEM
6 FROM COMPRA
7 INNER JOIN DETCOMPRA ON NUMEROFACTURA = COMPRA_NUMEROFACTURA
8 INNER JOIN PROVEEDOR ON PROVEEDOR_RUT = RUT
9 WHERE TO_CHAR (FECHA, 'YYYY') = '2020';
```

Obteniendo así el siguiente resultado:



Tal como está definido, el atributo “CantidadItem” nos deja un problema. Al efectuar una consulta, se duplicarían las compras por cada uno de los detalles de éstas. Para evitarlo, agregaremos una línea al final, con la instrucción: **GROUP BY FECHA, TOTAL, NOMBRE, RUT**. Usamos la función “Count”, que devuelve el número de filas de la consulta.

Quedará finalmente como:

```
1 SELECT FECHA,
2 TOTAL,
3 NOMBRE AS NOMBREPROVEEDOR,
4 RUT AS RUTPROVEEDOR,
5 COUNT (*) AS CANTIDADITEM
6 FROM COMPRA
7 INNER JOIN DETCOMPRA ON NUMEROFACTURA = COMPRA_NUMEROFACTURA
8 INNER JOIN PROVEEDOR ON PROVEEDOR_RUT = RUT
9 WHERE TO_CHAR (FECHA, 'YYYY') = '2020'
10 GROUP BY FECHA, TOTAL, NOMBRE, RUT
```

Continuaremos realizando un listado de todas las ventas del año 2020, pero con las columnas: Fecha, Total, NombreCliente, el monto más alto del artículo vendido, y el promedio de los subtotales de los vendidos.

Para realizar esta consulta, escribiremos:

```
1 SELECT FECHAVENTA,
2 TOTAL,
3 NOMBRE AS NOMBRECLIENTE,
4 0 AS MONTOALTO,
5 0 AS PROMEDIO
6 FROM VENTA
```

```

7 INNER JOIN DETVENTA ON NUMEROFACTURA = VENTA_NUMEROFACTURA
8 INNER JOIN CLIENTE ON CLIENTE_RUT = RUT
9 WHERE TO_CHAR (FECHAVENTA, 'YYYY') = '2020'
10 GROUP BY FECHAVENTA, TOTAL, NOMBRE

```

De esta forma, obtenemos el siguiente resultado:



Para mostrar el monto más alto de una venta, vamos a desplegar la información. En la tabla DetVenta, podemos multiplicar el atributo Cantidad por Unitario, y se los entregamos como parámetros a una función de agregación llamada “max”. Ésta sirve para obtener el mayor valor de una columna determinada.

La consulta quedará de la siguiente forma:

```

1 SELECT FECHAVENTA,
2 TOTAL,
3 NOMBRE AS NOMBRECLIENTE,
4 MAX(CANTIDAD * UNITARIO) AS MONTOALTO,
5 0 AS PROMEDIO
6 FROM VENTA
7 INNER JOIN DETVENTA ON NUMEROFACTURA = VENTA_NUMEROFACTURA
8 INNER JOIN CLIENTE ON CLIENTE_RUT = RUT
9 WHERE TO_CHAR (FECHAVENTA, 'YYYY') = '2020'
10 GROUP BY FECHAVENTA, TOTAL, NOMBRE

```

Finalmente, para terminar de obtener los datos de esta consulta, nos centraremos en el promedio de los subtotales.

Vamos a utilizar la función AVG, y le entregaremos como parámetro los atributos Cantidad y Unitario. La consulta quedará de la siguiente forma:

```

1 SELECT FECHAVENTA,
2 TOTAL,
3 NOMBRE AS NOMBRECLIENTE,

```

```
4 MAX(CANTIDAD * UNITARIO) AS MONTOALTO,  
5 AVG (CANTIDAD * UNITARIO) AS PROMEDIO  
6 FROM VENTA  
7 INNER JOIN DETVENTA ON NUMEROFACTURA = VENTA_NUMEROFACTURA  
8 INNER JOIN CLIENTE ON CLIENTE_RUT = RUT  
9 WHERE TO_CHAR (FECHAVENTA, 'YYYY') = '2020'  
10 GROUP BY FECHAVENTA, TOTAL, NOMBRE
```

Al ejecutarla, no encontramos errores, y continuamos obteniendo la impresión de las columnas solicitadas. Sin embargo, no se despliegan datos, ya que no guardamos ninguno en las tablas.

Continuaremos mostrando las compras y ventas realizadas en el mes de enero del año 2020, con las columnas: Nombre, Total y Tipo.

Para poder obtener el resultado esperado, escribiremos la siguiente consulta para la tabla Venta:

```
1 SELECT NOMBRE,  
2 TOTAL,  
3 'VENTA' AS TIPO  
4 FROM VENTA V  
5 INNER JOIN CLIENTE C ON C.RUT = V.CLIENTE_RUT  
6 WHERE TO_CHAR (FECHAVENTA, 'YYYY-MM') = '2020-01'
```

Y para la tabla Compra escribiremos:

```
1 SELECT NOMBRE,  
2 TOTAL,  
3 'COMPRA' AS TIPO  
4 FROM COMPRA  
5 INNER JOIN PROVEEDOR C ON RUT = PROVEEDOR_RUT  
6 WHERE TO_CHAR (FECHA, 'YYYY-MM') = '2020-01'
```

Ambas consultas devuelven las columnas solicitadas.



Luego, para unir ambos resultados, agregaremos la palabra “unión”, que realizará la relación entre las tablas, quedando la consulta de la siguiente manera:

```

1 SELECT NOMBRE,
2 TOTAL,
3 'VENTA' AS TIPO
4 FROM VENTA V
5 INNER JOIN CLIENTE C ON C.RUT = V.CLIENTE_RUT
6 WHERE TO_CHAR (FECHAVENTA, 'YYYY-MM') = '2020-01'
7 UNION
8 SELECT NOMBRE,
9 TOTAL,
10 'COMPRA' AS TIPO
11 FROM COMPRA
12 INNER JOIN PROVEEDOR C ON RUT = PROVEEDOR_RUT
13 WHERE TO_CHAR (FECHA, 'YYYY-MM') = '2020-01'
  
```

Finalmente, listaremos las ventas por cada mes del año 2020, con los atributos: Mes, TotalMes y Promedio.

Nuestra consulta quedará como:

```

1 SELECT TO_CHAR (FECHAVENTA, 'YYYY-MM') MES,
2 SUM (CANTIDAD* UNITARIO) TOTALMES,
3 AVG (CANTIDAD*UNITARIO) PROMEDIO
4 FROM VENTA
5 INNER JOIN DETVENTA ON NUMEROFACTURA = VENTA_NUMEROFACTURA
6 WHERE TO_CHAR (FECHAVENTA, 'YYYY') = '2020'
7 GROUP BY TO_CHAR (FECHAVENTA, 'YYYY-MM')
  
```

Al ejecutar, obtenemos las columnas deseadas:



Resultado de la Consulta		
SQL Todas las Filas Recuperadas: 0 en 0,005 segundos		
MES	TOTALM...	PROMED...