grupoesfera | INNOVACION Y DESARROLLO

Introducción al lenguaje de consulta estructurado (SQL)

Structured Query Language

Agenda

Sentencias avanzadas en SQL.

- Múltiples tablas

Consultas en más de una tabla

- Hay dos formas de utilizar más de una tabla en una consulta
  - UNION: Requiere dos (o *N*) tablas con las mismas columnas. Se asemeja a la unión de conjuntos
  - COMPOSICIÓN: Toma dos (o *N*) tablas y produce una tercera de resultado con campos de ambas tablas.

Consultas multitabla - UNION

- Junta los resultados de todas las tablas participantes en la Unión dando por resultado una nueva tabla.
- Estas condiciones tienen que cumplirse para poder incluir consultas en un UNION:
  - La cantidad y el orden de las columnas debe ser el mismo en todas las consultas.
  - Los tipos de datos de las columnas deben ser compatibles entre los distintas consultas.

Consultas multitabla – UNION (cont.)

- Sintaxis:

```
SELECT campo1, campo2, campo3
FROM Tabla1

UNION

SELECT campo4, campo5, campo6
FROM Tabla2
```

Consultas multitabla – UNION (cont.)

Los resultados de la unión de las tablas no repiten valores.

- Se utiliza la palabra ALL para que los valores duplicados formen parte del resultado de una consulta.

```
SELECT campo1, campo2, campo3
FROM Tabla1
UNION ALL
SELECT campo4, campo5, campo6
FROM Tabla2
```

Ejemplo de UNION

- Supongamos que existen dos tablas, una de Clientes y otra de Usuarios donde ambas tiene un campo llamado nombre.
- ¿Qué sentencia SQL habría que ejecutar para poder obtener todos los nombres del sistema, ya sean usuarios o clientes?

```
SELECT nombre
  FROM CLIENTE
  UNION
  SELECT nombre
  FROM USUARIO
```

Consultas multitabla - COMPOSICIÓN

- Las consultas pueden involucrar más de una tabla.
- En estos casos las tablas se indican a la derecha de la cláusula FROM.
- Se pueden incluir todas las tablas que se deseen.
- En la cláusula SELECT se pueden poner campos de cualquier tabla.
- Si más de una tabla tiene un campo con el mismo nombre:

```
Se debe diferenciar el campo
referenciandolo de la forma:
nombreTabla.nombreCampo
```

Consultas en más de una tabla

- Los tipos de composición de tablas son en consultas que involucran más de una tabla son:
  - `El producto cartesiano`
  - `El INNER JOIN`
  - `El LEFT / RIGHT JOIN`

Consultas en más de una tabla (cont.)

Producto Cartesiano

- Se indica colocando en la cláusula FROM las tablas que queremos componer separadas por comas.
- Se pueden utilizar desde 2 a *N* tablas.
- Se puede componer una tabla consigo misma.
- El criterio para combinar las tablas se debe especificar en la cláusula WHERE.

```
Si no se especifica ningún criterio, el
resultado es el producto cartesiano de
los todos los registros de todas las
tablas.
```

Consultas en más de una tabla (cont.)

IMPORTANTE: El resultado de un producto cartesiano incluye todas las combinaciones de los registros de todas las tablas involucradas.

- Si se hace el producto cartesiano de dos tablas que tienen 100 registros cada una, se obtienen 10.000 registros.
- Si se hace el producto cartesiano de tres tablas que tienen 100 registros cada una, se obtienen 1.000.000 registros.
- Si se hace el producto cartesiano de tres tablas que tienen 100 registros cada una con una cuarta que tiene 1000 registros, se obtienen 1.000.000.000 registros.

Consultas en más de una tabla (cont.)

- ALIAS: Es un nombre alternativo que puede dársele a una tabla o columna para referenciarlas dentro de una consulta.
- Sintaxis

```
SELECT [nombreTabla].campo1 as nombreCampo1,
       [nombreTabla].campo2 as nombreCampo2, … ,
       [nombreTabla].campoN as nombreCampoN
FROM tabla1 as nombreTabla1,
     tabla2 as nombreTabla2, … ,
     tablaN as nombreTablaN
WHERE condicion1, condicion2, … , condicionN
```

Ejemplo de COMPOSICIÓN

- Supongamos que existen dos tablas, una de Clientes y otra de Facturas donde la tabla de Facturas tiene un campo llamado idCliente y otro llamado montoTotal. La tabla de Clientes tiene un campo llamado id y otro campo llamado nombre.
- ¿Qué sentencia SQL habría que ejecutar para poder obtener todos los clientes junto con el total de cada una de sus facturas?

```
SELECT c.nombre, f.montoTotal
  FROM CLIENTE as c, FACTURA as f
  WHERE f.idCliente = c.id
```

Consultas en más de una tabla (cont.)

INNER JOINS

- Este tipo de consultas permite realizar consultas en más de una tabla especificando las condiciones en que las tablas se relacionan.
- A diferencia del producto cartesiano visto previamente, donde la cláusula WHERE para asociar los registros de las distintas tablas era opcional, en el INNER JOIN, esta cláusula es explícita.
- El resultado contendrá a todas las filas que satisfacen la condición en ambas tablas.

Consultas en más de una tabla (cont.)

- Sintaxis

```
SELECT tabla1.campo1, tabla2.campo2, … , campoN
FROM tabla1
     INNER JOIN tabla2
     ON tabla1.campo1 = tabla2.campo2
WHERE condicion1, condicion2, … , condicionN
```

- *Consideraciones:*
  - Las columnas sobre las que se especificarán las condiciones deben ser del mismo tipo.
  - Los operadores para comparar pueden ser =, <, >, <=, >=, o <>.

Consultas en más de una tabla (cont.)

LEFT / RIGHT JOINS

- Las composiciones vistas hasta ahora (el producto cartesiano y el INNER JOIN) son composiciones internas ya que todos los valores de los registros del resultado son valores que están en las tablas que se combinan.
- Con una composición interna sólo se obtienen los registros que tienen registros en todas las que se están consultando.
- LEFT JOIN y RIGHT JOIN se denominan OUTER JOINS puesto que los resultados de la composición no necesariamente tienen que estar en todas las tablas que se están consultando.

Consultas en más de una tabla (cont.)

- En un LEFT JOIN el resultado contiene a todos los registros de la tabla de la izquierda, aún si no hay ningún registro que coincida con el criterio establecido en la tabla de la derecha.
- Análogamente, en un RIGHT JOIN el resultado contiene a todos los registros de la tabla de la derecha, aún si no hay ningún registro que coincida con el criterio establecido en la tabla de la izquierda.

Consultas en más de una tabla (cont.)

- Sintaxis

```
SELECT tabla1.campo1, tabla2.campo2, … , campoN
FROM tabla1
     LEFT JOIN tabla2
     ON tabla1.campo1 = tabla2.campo2
WHERE condicion1, condicion2, … , condicionN

SELECT tabla1.campo1, tabla2.campo2, … , campoN
FROM tabla1
     RIGHT JOIN tabla2
     ON tabla1.campo1 = tabla2.campo2
WHERE condicion1, condicion2, … , condicionN
```

Ejemplo de JOINS

- Supongamos que existen dos tablas, una de Clientes y otra de Facturas donde la tabla de Facturas tiene un campo llamado idCliente y otro llamado montoTotal. La tabla de Clientes tiene un campo llamado id y otro campo llamado nombre.
- ¿Qué sentencia SQL habría que ejecutar para poder obtener todos los clientes junto con el total de cada una de sus facturas?

```
SELECT c.nombre, f.montoTotal
  FROM CLIENTE as c
       INNER JOIN FACTURA as f
       ON (f.idCliente = c.id)
```

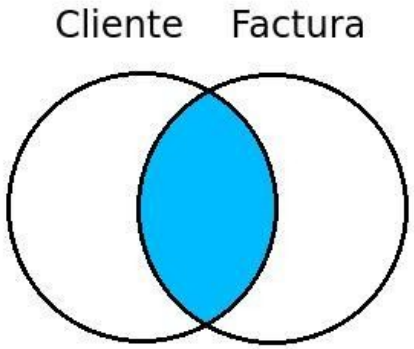
Ejemplo de JOINS

- Supongamos que existen dos tablas, una de Clientes y otra de Facturas donde la tabla de Facturas tiene un campo llamado idCliente y otro llamado montoTotal. La tabla de Clientes tiene un campo llamado id y otro campo llamado nombre.
- ¿Qué sentencia SQL habría que ejecutar para poder obtener todos los clientes junto con el total de cada una de sus facturas, aún si no tienen ninguna factura?

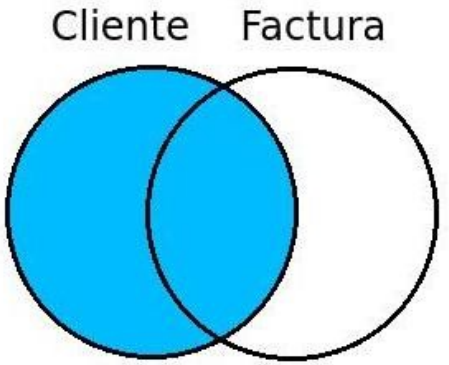
```
SELECT c.nombre, f.montoTotal
  FROM CLIENTE as c
       LEFT JOIN FACTURA as f
       ON (f.idCliente = c.id)
```

Comparando los JOINS

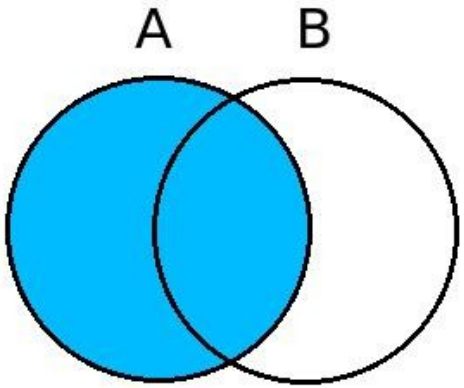
```
SELECT c.nombre, f.montoTotal
 FROM CLIENTE as c
      INNER JOIN FACTURA as f
      ON (f.idCliente = c.id)
```




```
SELECT c.nombre, f.montoTotal
 FROM CLIENTE as c
      LEFT JOIN FACTURA as f
      ON (f.idCliente = c.id)
```



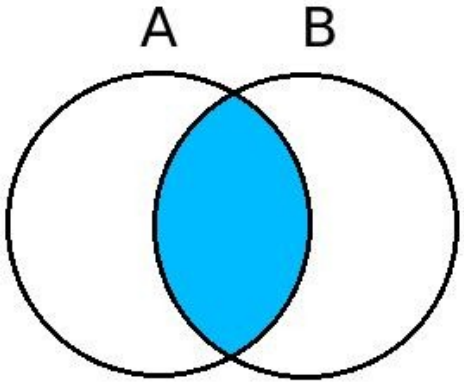
Comparando los JOINS



A LEFT OUTER JOIN B

EQUIVALENTE A:

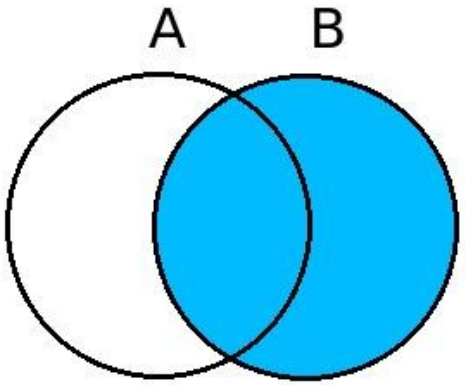
B RIGHT OUTER JOIN A



A INNER JOIN B

EQUIVALENTE A:

B INNER JOIN A



A RIGHT OUTER JOIN B

EQUIVALENTE A:

B LEFT OUTER JOIN A

Entendiendo el OUTER JOIN

- Las siguientes sentencias son equivalentes:

```
(1) SELECT c.nombre, f.montoTotal FROM CLIENTE as c
          LEFT JOIN FACTURA as f ON (f.idCliente = c.id)


(2) SELECT c.nombre, f.montoTotal
    FROM Cliente as c INNER JOIN Factura as f
         ON (f.idCliente = c.id)
    UNION ALL
    SELECT clienteSinRelacion.nombre, NULL
    FROM Cliente as clienteSinRelacion
    WHERE clienteSinRelacion.id NOT IN
                  (SELECT f.idCliente from Factura as f)
```

Entendiendo el OUTER JOIN

```
(2) SELECT c.nombre, f.montoTotal
    FROM Cliente as c INNER JOIN Factura as f
         ON (f.idCliente = c.id)
    UNION ALL
    SELECT clienteSinRelacion.nombre, NULL
    FROM Cliente as clienteSinRelacion
    WHERE clienteSinRelacion.id NOT IN
                  (SELECT f.idCliente from Factura as f)
```

Observaciones:

(a) Se usó la palabra 'clienteSinRelacion' en el alias para clarificar el ejemplo ya que los Clientes que se están obteniendo en el segundo SELECT no tienen relación con ninguna Factura (no están asociados) esto es así debido a la condición del WHERE. Se puede usar 'c' como alias, aún así no afectaría al primer SELECT dado que el intérprete del motor entiende que son 2 SELECT diferentes a pesar de que se trata de una sola consulta

Ejercicios

- E11A: Obtener todos los nombres de los clientes y los empleados en un listado (Todos los nombres de las personas involucradas en el Negocio)
- E11B: Obtener todas las fechas de los eventos tanto para las altas de una Factura como para los ingresos de Empleados
- E12: Obtener todos los clientes de categoría ALTA que posean facturas cuyo monto total supere los $5000, para ellos se desea obtener el nombre, apellido y el número y monto de sus facturas.
- E13: Obtener a todos los clientes que hayan comprado en un producto llamado “recuperador” en el último año (2012).
- E14: Obtener para la sucursal número 3, el nombre de cada uno de sus clientes.

Ejercicios (cont.)

- E15: Obtener las facturas mostrando su número, monto total, nombre y apellido del cliente, y fecha de confección de la misma.
- E16: Obtener el nombre, apellido y fecha de contratación de cada uno de los empleados y el nombre de la sucursal donde trabaja.
- E17: Obtener las sucursales con objetivo superior a $300.000 indicando para cada una de ellas el nombre de su jefe (si es que tiene).
- E18: Obtener las facturas cuyo monto total sea superior a $12.000, incluyendo el nombre del cliente que la solicitó, la sucursal donde se realizó dicha venta y el nombre del empleado que la realizó.

Ejercicios (cont.)

- E19: Obtener un listado de los clientes que no tienen facturas.
- E20: Obtener los empleados que realizaron un pedido el mismo día en que fueron contratados.
- E21: Obtener los empleados con una cuota superior a la de su jefe; para cada empleado obtener sus datos y el nombre y cuota de su jefe.
- E22: Obtener los nombres de los empleados que tienen un cuota superior a $10.000 o que su jefe tenga una cuota inferior a $15.000.

FIN