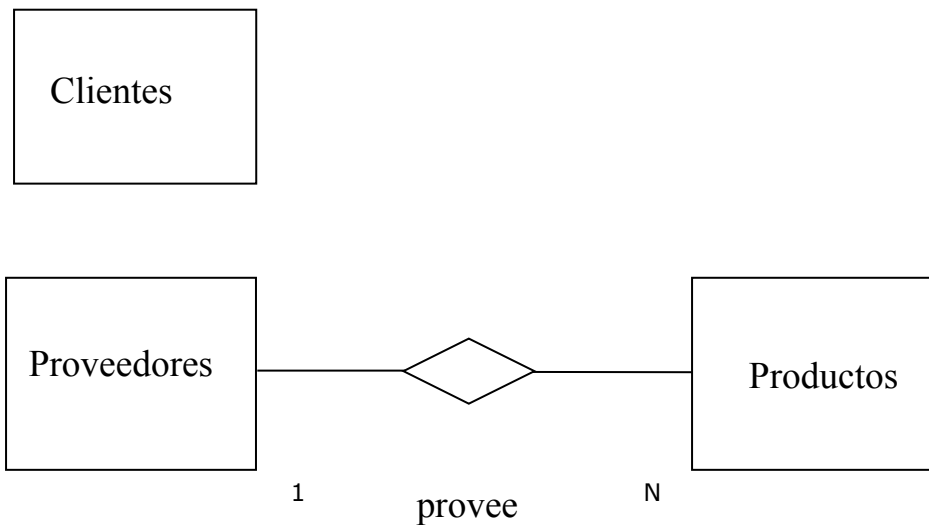




SQL Consultas – Material de Apoyo de Clases – Base de Ejemplo

Para la clase de apoyo de JOIN y Subconsultas (Sub Query) se utilizó el siguiente modelo lógico:



El Modelo Físico de datos es:

```
CREATE TABLE PROVEEDORES
( CUIT VARCHAR(11) PRIMARY KEY,
  NOMBRE VARCHAR(50),
  CODPOS INTEGER,
  VENTASULTIMOANIO DECIMAL(12,2)
);

CREATE TABLE PRODUCTOS
( CODIGO INTEGER,
  NOMBRE VARCHAR(50),
  PRECIO DECIMAL(5,2),
  PROVEEDOR_CUIT VARCHAR(11),          /* FK Proveedores(CUIT) */
  PRIMARY KEY (CODIGO)
);

CREATE TABLE CLIENTES
( CUIT VARCHAR(11) PRIMARY KEY,
  NOMBRE VARCHAR(50),
  CODPOSCLI INTEGER
);
```

Nota la base de datos física resultante se encuentra en esta página con el nombre:

- BdConsulta.fdb

Se ingresaron los siguientes clientes, proveedores y productos:

```
INSERT INTO PROVEEDORES VALUES ( '10','JUAN',3000, 30000000);
INSERT INTO PROVEEDORES VALUES ( '12','PEDRO',3100, 10000);
INSERT INTO PROVEEDORES VALUES ( '15','MARIA',NULL, 30000);
INSERT INTO PROVEEDORES VALUES ( '18','MARTA',3000, NULL);

INSERT INTO PRODUCTOS  VALUES (50, 'P1' , 10, 10);
INSERT INTO PRODUCTOS  VALUES (60, 'P2' , 5, 10);
INSERT INTO PRODUCTOS  VALUES (70, 'P3' , 100, 15);
INSERT INTO PRODUCTOS  VALUES (80, 'P4' , NULL, NULL);

INSERT INTO CLIENTES VALUES ( '85','ADRIANA',3100);
INSERT INTO CLIENTES VALUES ( '10','JUAN',3000);
```



El Valor nulo (NULL)

Tratamiento de valores nulos en funciones de agregación

--Filas consultadas

```
SELECT * FROM PROVEEDORES
```

CUIT	NOMBRE	CODPOS	VENTASULTIMOANIO
10	JUAN	3000	30000000
12	PEDRO	3100	10000
15	MARIA	null	30000
18	MARTA	3000	null

--Consulta global count(*) y count(codpos)

```
SELECT COUNT(*) AS COUNTASTERISCO, COUNT(CODPOS) AS COUNTCODPOS  
FROM PROVEEDORES
```

```
COUNTASTERISCO COUNTCODPOS
```

```
-----  
4                3
```

Nota: count(codpos) no cuenta los nulos.

--Restricción con NULL INCORRECTA

```
SELECT * FROM PROVEEDORES WHERE CODPOS = NULL
```

--Restricción con NULL CORRECTA

```
SELECT * FROM PROVEEDORES WHERE CODPOS IS NULL
```

El Agrupamiento

--Idem Sum

```
SELECT COUNT(*) AS CANTIDADFILAS,  
       COUNT(CODPOS) AS CANTCODPOS,  
       SUM(VENTASULTIMOANIO) AS VENTULTANIO  
FROM PROVEEDORES
```

--Ejemplo de Agrupamiento por código postal

```
SELECT CODPOS, COUNT(*)  
FROM PROVEEDORES  
GROUP BY CODPOS
```

CODPOS	COUNT
null	1
3000	2
3100	1

--Ejemplo de Agrupamiento por código postal filtrando filas con HAVING

```
SELECT CODPOS, COUNT(*)  
FROM PROVEEDORES  
GROUP BY CODPOS  
HAVING COUNT(*) > 1
```

CODPOS	COUNT
3000	2



El operador LIKE

```
--Ejemplos de LIKE usando expresiones regulares  
SELECT * FROM PROVEEDORES WHERE NOMBRE LIKE 'MA%'
```

Busca todos los proveedores cuyo nombre comience con MA inclusive nombre = MA

```
--Ejemplos de LIKE usando expresiones regulares  
--Busca todos los proveedores cuyo nombre contiene 4 y solo 4 letras  
SELECT * FROM PROVEEDORES WHERE NOMBRE LIKE '____'
```

Reuniones (JOIN)

```
--Reunion con Restricción - SQL 86  
SELECT * FROM PROVEEDORES, PRODUCTOS  
WHERE CUIT = PROVEEDOR_CUIT
```

```
--Reunion con operador JOIN                INNER (Interna)  
SELECT * FROM PROVEEDORES  
JOIN PRODUCTOS  
ON CUIT = PROVEEDOR_CUIT
```

```
SELECT * FROM PROVEEDORES  
INNER JOIN PRODUCTOS                -- INNER opcional  
ON CUIT = PROVEEDOR_CUIT
```

```
--Reunion con operador JOIN                LEFT (Externa Izquierda)  
SELECT * FROM PROVEEDORES  
LEFT JOIN PRODUCTOS  
ON CUIT = PROVEEDOR_CUIT
```

```
--Reunion con operador JOIN                LEFT (Externa Derecha)  
SELECT * FROM PROVEEDORES  
RIGHT JOIN PRODUCTOS  
ON CUIT = PROVEEDOR_CUIT
```

```
SELECT * FROM PROVEEDORES  
RIGHT OUTER JOIN PRODUCTOS          -- OUTER Opcional  
ON CUIT = PROVEEDOR_CUIT
```

```
--Reunion con operador JOIN                FULL (Todo)  
SELECT * FROM PROVEEDORES  
FULL JOIN PRODUCTOS  
ON CUIT = PROVEEDOR_CUIT
```

```
--Idem anterior manejando el orden del conjunto resultado de salida  
SQL Statement  
SELECT * FROM PROVEEDORES  
FULL JOIN PRODUCTOS  
ON CUIT = PROVEEDOR_CUIT  
ORDER BY PROVEEDOR_CUIT, CUIT
```



Sub Select - Sub Query

```
-- SubSelect en proyección
SELECT CODIGO, NOMBRE, PROVEEDOR_CUIT,
      ( SELECT NOMBRE FROM PROVEEDORES WHERE CUIT = PROVEEDOR_CUIT      )
FROM PRODUCTOS
```

```
-- Idem con Join
SELECT CODIGO, PRODUCTOS.NOMBRE, PROVEEDOR_CUIT, PROVEEDORES.NOMBRE
FROM PRODUCTOS
LEFT JOIN PROVEEDORES
      ON CUIT = PROVEEDOR_CUIT
```

```
-- SubSelect en restricción
SELECT * FROM PROVEEDORES
WHERE CUIT IN ( SELECT DISTINCT PROVEEDOR_CUIT FROM PRODUCTOS )
```

```
Primero se ejecuta el query                                SELECT DISTINCT PROVEEDOR_CUIT FROM PRODUCTOS
Esto da como resultado el conjunto                        ( null, 10, 15)
Luego se ejecuta
      SELECT * FROM PROVEEDORES
      WHERE CUIT IN ( null, 10, 15)
```

```
-- Idem con Join
SELECT DISTINCT PV.*
FROM PROVEEDORES PV
JOIN PRODUCTOS PD
      ON CUIT = PROVEEDOR_CUIT
```

```
-- Aclaración el distinct no es necesario
SELECT * FROM PROVEEDORES
WHERE CUIT IN ( SELECT PROVEEDOR_CUIT FROM PRODUCTOS )
```

```
--SubSelect en lista de relaciones - Tablas derivadas
SELECT *
FROM PROVEEDORES,
      (SELECT CODPOS, SUM(VENTASULTIMOANIO) AS VENTAS
      FROM PROVEEDORES
      GROUP BY CODPOS ) TOTALVENTAS
WHERE TOTALVENTAS.CODPOS = PROVEEDORES.CODPOS
```

```
--Idem anterior pero con JOIN - Tablas derivadas
SELECT *
FROM PROVEEDORES
JOIN (SELECT CODPOS, SUM(VENTASULTIMOANIO) AS VENTAS
      FROM PROVEEDORES
      GROUP BY CODPOS ) TOTALVENTAS
ON TOTALVENTAS.CODPOS = PROVEEDORES.CODPOS
```



UNION - Operador de Conjunto

```
--Filas usadas para todas las operaciones de conjunto
SELECT * FROM PROVEEDORES
```

CUIT	NOMBRE	CODPOS	VENTASULTIMOANIO
10	JUAN	3000	30000000
12	PEDRO	3100	10000
15	MARIA	null	30000
18	MARTA	3000	null

```
SELECT * FROM CLIENTES
```

CUIT	NOMBRE	CODPOS
10	JUAN	3000
85	ADRIANA	3100

```
-- Union USO ERRONEO - Conjuntos distintos
```

```
SELECT * FROM CLIENTES
```

```
UNION
```

```
SELECT * FROM PROVEEDORES
```

```
--Da un error por que describe que tiene cantidad de conlumnas distintas
```

```
-- Union USO CORRECTO
```

```
SELECT * FROM CLIENTES
```

```
UNION
```

```
SELECT CUIT, NOMBRE, CODPOS FROM PROVEEDORES
```

```
--Salida
```

CUIT	NOMBRE	CODPOS
10	JUAN	3000
12	PEDRO	3100
15	MARIA	NULL
18	MARTA	3000
85	ADRIANA	3100

```
-- Union ALL
```

```
SELECT * FROM CLIENTES
```

```
UNION ALL
```

```
SELECT CUIT, NOMBRE, CODPOS FROM PROVEEDORES
```

```
--Salida
```

CUIT	NOMBRE	CODPOS
10	JUAN	3000
10	JUAN	3000
12	PEDRO	3100
15	MARIA	NULL
18	MARTA	3000
85	ADRIANA	3100

INTERSECCION - Operador de Conjunto

```
SELECT * FROM CLIENTES
```

```
WHERE CUIT IN
```

```
(SELECT CUIT FROM PROVEEDORES)
```

```
--Salida
```

CUIT	NOMBRE	CODPOS
10	JUAN	3000



DIFERENCIA – Operador de Conjunto

```
SELECT * FROM CLIENTES
WHERE CUIT NOT IN
(SELECT CUIT FROM PROVEEDORES)
```

```
--Salida
CUIT      NOMBRE      CODPOS
-----
85        ADRIANA      3100
```

EXISTS – Operador ampliados

```
-- exists
SELECT * FROM CLIENTES C
WHERE EXISTS (SELECT CUIT FROM PROVEEDORES P where P.cuit = C.CUIT)
```

IN

```
-- Ya tratado en - SubSelect en restricción
SELECT * FROM PROVEEDORES
WHERE CUIT IN ( SELECT DISTINCT PROVEEDOR_CUIT FROM PRODUCTOS )

-- SubSelect en restricción con duplas, triplas, etc
-- Problema con el operador IN en firebird - NO FUNCIONA
SELECT CUIT, NOMBRE FROM PROVEEDORES
WHERE (CUIT, NOMBRE) IN (SELECT CUIT, NOMBRE FROM CLIENTES)
```

Tablas

```
CREATE TABLE VENTAS
( FECHA DATE,
  CLIENTE_CUIT VARCHAR(11),
  IMPORTE DECIMAL(5,2),
  SUCURSAL INTEGER
);

CREATE TABLE SUCURSAL
( CODIGO INTEGER,
  NOMBRE VARCHAR(30),
  VENTA_MINIMA DECIMAL(5,2),
  VENTA_OBJETIVO DECIMAL(8,2)
);
```

Insert

```
INSERT INTO VENTAS VALUES ( '2010/01/12', 10, 700, 3)
INSERT INTO VENTAS VALUES ( '2010/03/18', 10, 30, 3)
INSERT INTO VENTAS VALUES ( '2010/03/18',85, 304,13)
INSERT INTO VENTAS VALUES ( '2010/05/28',85, 1000, 23)
INSERT INTO VENTAS VALUES ( '2010/03/18', 10, 304,13)

INSERT INTO SUCURSAL VALUES ( 3, 'PARANA', 200, 500);
INSERT INTO SUCURSAL VALUES ( 13, 'CONCORDIA', 800, 1000);
INSERT INTO SUCURSAL VALUES ( 23, 'CORDOBA',350, 600);
```



```
-- SubSelect en restricción comparacion >, <, etc
SELECT * FROM VENTAS
WHERE IMPORTE > SOME
(SELECT VENTA_MINIMA FROM SUCURSAL)

-- SubSelect en restricción comparacion >, <, etc
SELECT * FROM VENTAS
WHERE IMPORTE > ALL
(SELECT VENTA_MINIMA FROM SUCURSAL)
```
