Ejemplos de CASE

```
SELECT   manu_code, stock_num, unit_price,
        'Rango de Precios' =
      CASE
         WHEN unit_price =  0 THEN '0 - Item no negociable'
         WHEN unit_price < 100 THEN '1 - Precio Menor $100'
         WHEN unit_price >= 100 and unit_price< 250 THEN '2 - Precio Menor a $250'
         WHEN unit_price >= 250 and unit_price < 500 THEN '3 - Precio Menor a $500'
         ELSE '4 - Precio mayor a $500'
      END
FROM products
ORDER BY "Rango de Precios",manu_code,stock_num
```

```
UPDATE products
SET unit_price =
    ( CASE
         WHEN (unit_price <= 250) THEN unit_price * 1.05
         ELSE (unit_price * 1.10)
       END
    )
WHERE manu_code = 'ANZ';
```

```
SELECT manu_code, stock_num, order_num, item_num
FROM items
WHERE manu_code IN ('ANZ','HRO')
ORDER BY CASE WHEN manu_code='HRO' THEN order_num END,
         CASE manu_code WHEN 'ANZ' THEN stock_num END;
```

```
SELECT manu_code, MAX(unit_price)AS precio,
       COUNT(manu_code) cant_reg
FROM products
GROUP BY manu_code
HAVING (COUNT(CASE WHEN manu_code = 'ANZ'
                   THEN manu_code
               ELSE NULL END) > 10.00
    OR COUNT(CASE WHEN manu_code <> 'ANZ'
                 THEN manu_code
              ELSE NULL END) > 15.00)
ORDER BY manu_code DESC
```

Ejemplo de COALESCE

```
SELECT order_date, paid_date
FROM orders
ORDER BY paid_date
```

```
SELECT order_date, COALESCE(paid_date,'2017-01-01')
FROM orders
ORDER BY paid_date
```

```
---------

SELECT c1.customer_num,c1.lname+', '+c1.fname Referido,
       'Referido por: '+c2.lname+', '+c2.fname Referente
FROM CUSTOMER c1 LEFT JOIN CUSTOMER c2
        ON (c1.customer_num_referedBy=c2.customer_num)

SELECT c1.customer_num,c1.lname+', '+c1.fname Referido,
       COALESCE('Referido por: '+c2.lname+', '+c2.fname,'Cliente Directo')
Referente
FROM CUSTOMER c1 LEFT JOIN CUSTOMER c2
        ON (c1.customer_num_referedBy=c2.customer_num)


---------
-- OTRO EJEMPLO DE COALESCE
--creo un produucto con unit_price en nulo
INSERT INTO products VALUES (1,'ANZ',NULL,NULL)

-- cuenta los unit_price que no son nulos, no toma en cuenta el recién insertado.
SELECT COUNT(unit_price) FROM products WHERE manu_code='ANZ'

-- ahora el count, si lo tiene en cuenta. Igual para esto uso COUNT(*)
SELECT COUNT(COALESCE(unit_price,0))
FROM products WHERE manu_code='ANZ'

SERIA LO MISMO QUE HACER
SELECT count(*)--COUNT(COALESCE(unit_price,0))
FROM products WHERE manu_code='ANZ'

-- Borro la fila dummy creada
DELETE FROM products WHERE stock_num=1 AND manu_code='ANZ'
```

Ejemplos de COALESCE y CASTEO por Tipo de Dato

```
-- Opcion 1 con COALESCE
-- falla por problemas de Casteo

SELECT c.customer_num,
       COALESCE(MAX(order_date),'No posee Productos') ultima_compra
FROM customer c LEFT JOIN orders o ON c.customer_num=o.customer_num
GROUP BY c.customer_num



-- Opcion 2 con COALESCE Y CAST

SELECT c.customer_num,
       COALESCE(CAST(MAX(order_date) AS CHAR),'No posee Productos') ultima_compra
FROM customer c LEFT JOIN orders o ON c.customer_num=o.customer_num
GROUP BY c.customer_num
```

Ejemplos de Condicion en el WHERE o en el ON para LEFT JOINS

```sql
--Condicion en JOIN o en WHERE
--Aca da lo mismo
SELECT c.customer_num,c.lname, order_num
FROM customer c JOIN orders o
ON (c.customer_num=o.customer_num AND lname='Higgins')

SELECT c.customer_num,c.lname, order_num
FROM customer c INNER JOIN orders o
ON c.customer_num=o.customer_num
WHERE lname='Higgins'

--Condicion en JOIN o en WHERE (OUTER JOIN)
--Aca da diferente
SELECT c.customer_num,c.lname, order_num
FROM customer c LEFT JOIN orders o
ON (c.customer_num=o.customer_num AND lname='Higgins')

SELECT c.customer_num,c.lname, order_num
FROM customer c LEFT JOIN orders o
ON c.customer_num=o.customer_num
WHERE lname='Higgins'
```