**Profesor**
M. en I.A. Erick Orlando Matla Cruz
**Ayudantes**
L. en C.C. Efraín Hipólito Chamú
L. en C.C. Anahí Quiroz Jiménez
L. en C.C. Karen Monserrat Zavala Correa

# *Práctica 08*

## *Lenguaje de Manipulación de Datos (DML) Avanzado*

### 1. Objetivo General

Profundizar en el uso de palabras reservadas de SQL para construir consultas específicas.

### 2. Objetivos Secundarios

- Crear scripts (código en SQL) de mayor nivel para las consultas.
- Agrupar información que sirva de pauta para dar una lectura más fácil a los datos del negocio.
- Hacer uso de filtros para optimizar las consultas de una base de datos

### 3. Introducción

Siguiendo con la manipulación de registros de una base de datos, en muchas ocasiones se requerirá tener consultas que organicen información y que a su vez sea optima en tiempo de consulta.

En este aparatado se describe el uso de filtros y agrupamientos en cantidades grandes de información.

#### 3.1. Condiciones y Agrupamientos

El filtrar información es de mucha utilidad para generar consultas que solo contengan datos importantes para el negocio.

SQL cuenta con operadores que permiten construir consultas más complejas. En la Figura 8.1 se muestra ésta sintaxis.

```
SELECT nombre_columna1, nombre_columna2, ...
FROM nombre_tabla
WHERE condición
GROUP BY atributo_agrupador
HAVING condición_de_agrupación;
```

Figura 8.1 - Sintaxis avanzada de consultas en SQL.

En la sintaxis de la Figura 8.1 encontramos:

1. SELECT
   Palabra reservada para indicar que atributos se mostrarán como resultado de la consulta.
2. nombre_columna
   Es el nombre de la columna, o columnas separadas por comas, que queremos consultar.
3. FROM
   Palabra reservada para indicarle al SMBD la tabla de la cual obtener la columna que necesitamos.
4. nombre_tabla
   Es el nombre de la tabla que contiene la columna o columnas que solicitamos.
5. WHERE
   Palabra reservada para indicar bajo qué condición seleccionaremos los registros que queremos consultar.
6. condicion
   Es una característica o conjunto de características que tiene que cumplir los registros para ser seleccionados.
7. GROUP BY
   Palabras reservadas para agrupar los datos de una columna dependiendo de su valor.
8. atributo_agrupador
   Es el nombre de la columna, o columnas separadas por comas, cuyos valores dirigen la agrupación.
9. HAVING
   Palabra reservada para indicar que se requiere filtrar el o los grupos que se formaron en el punto 7 y 8. Su función es equivalente a WHERE con la diferencia de que únicamente es usado para grupos.
10. condición_de_agrupación
    Característica o conjunto de características que tiene que cumplir un grupo para ser seleccionado.
11. ;
    Indica la finalización de la instrucción.

Para explicar con claridad lo descrito anteriormente, se tomará como ejemplo el conjunto de las siguientes tablas mostradas en las Tablas 8.1 a 8.5 obtenidas del ejemplo de la Pizzería.

**Profesor**
M. en I.A. Erick Orlando Matla Cruz
**Ayudantes**
L. en C.C. Efraín Hipólito Chamú
L. en C.C. Anahí Quiroz Jiménez
L. en C.C. Karen Monserrat Zavala Correa

Tabla 8.1 - Tabla Cliente de la Pizzería

| ***CLIENTE*** | | | |
|---|---|---|---|
| ***Id_Cliente*** | ***Nombre*** | ***Ap_Materno*** | ***Ap_Paterno*** |
| 108 | Isidro | Juárez | Altamirano |
| 105 | Jorge | Arzola | Cervantes |
| 100 | Edna | Pérez | León |
| 109 | Estefanía | González | Leyva |
| 101 | David | Avila | Martínez |
| 106 | Carlos | Lozano | Morales |
| 104 | Norma | Ponce | Noriega |
| 110 | María | Meza | Segura |
| 107 | Ricardo | Navarro | Tapia |
| 103 | Sara | Monroy | Vega |
| 102 | Genaro | López | Villegas |

Tabla 8.2 - Tabla Orden de la Pizzería

| ***ORDEN*** | | | |
|---|---|---|---|
| ***Id_Orden*** | ***Fecha*** | ***Numero_Articulos*** | ***Total*** |
| 20507 | 19/04/2013 | 1 | 62.72 |
| 20511 | 21/04/2013 | 4 | 292.04 |
| 20501 | 15/04/2013 | 5 | 347.90 |
| 20504 | 16/04/2013 | 6 | 486.08 |
| 20517 | 24/04/2013 | 7 | 583.10 |
| 20505 | 16/04/2013 | 8 | 568.40 |
| 20513 | 22/04/2013 | 9 | 899.64 |
| 20514 | 22/04/2013 | 11 | 846.72 |
| 20502 | 15/04/2013 | 15 | 1014.30 |
| 20515 | 22/04/2013 | 15 | 911.40 |
| 20510 | 21/04/2013 | 17 | 1123.08 |
| 20508 | 19/04/2013 | 19 | 1229.90 |
| 20503 | 15/04/2013 | 23 | 2325.54 |
| 20516 | 23/04/2013 | 23 | 1865.92 |
| 20509 | 19/04/2013 | 33 | 3350.62 |
| 20512 | 22/04/2013 | 36 | 2380.42 |
| 20506 | 18/04/2013 | 42 | 2840.04 |

Tabla 8.3 - Tabla Detalle de Orden de la Pizzería

**Profesor**
M. en I.A. Erick Orlando Matla Cruz
**Ayudantes**
L. en C.C. Efraín Hipólito Chamú
L. en C.C. Anahí Quiroz Jiménez
L. en C.C. Karen Monserrat Zavala Correa

| DETALLE_DE_ORDEN | | | |
|---|---|---|---|
| Id_Orden | Id_Articulo | Cantidad | Total_Por_Art |
| 20511 | A-3 | 1 | 59.78 |
| 20501 | B-3 | 1 | 62.72 |
| 20507 | B-3 | 1 | 62.72 |
| 20511 | C-4 | 1 | 67.62 |
| 20501 | A-5 | 1 | 82.32 |
| 20505 | B-4 | 1 | 99.96 |
| 20504 | C-4 | 2 | 135.24 |
| 20503 | C-5 | 2 | 150.92 |
| 20504 | C-5 | 2 | 150.92 |
| 20505 | A-5 | 2 | 164.64 |
| 20511 | A-5 | 2 | 164.64 |
| 20504 | B-4 | 2 | 199.92 |
| 20501 | C-4 | 3 | 202.86 |
| 20514 | B-5 | 4 | 270.48 |
| 20516 | C-5 | 4 | 301.84 |
| 20505 | A-4 | 5 | 303.8 |
| 20502 | B-5 | 5 | 338.1 |
| 20508 | A-3 | 7 | 418.46 |
| 20510 | C-4 | 8 | 540.96 |
| 20514 | A-5 | 7 | 576.24 |
| 20510 | C-2 | 9 | 582.12 |
| 20517 | C-6 | 7 | 583.1 |
| 20502 | C-4 | 10 | 676.2 |
| 20503 | A-6 | 7 | 679.14 |
| 20512 | B-3 | 11 | 689.92 |
| 20508 | C-4 | 12 | 811.44 |
| 20513 | B-4 | 9 | 899.64 |
| 20515 | A-4 | 15 | 911.4 |
| 20509 | A-6 | 14 | 1358.28 |
| 20503 | A-2 | 14 | 1495.48 |
| 20516 | A-5 | 19 | 1564.08 |
| 20512 | B-5 | 25 | 1690.5 |
| 20509 | B-2 | 19 | 1992.34 |
| 20506 | C-4 | 42 | 2840.04 |

Tabla 8.4 - Tabla Costo de la Pizzería

| COSTO | |
|---|---|
| Id_Articulo | Costo_Unitario |
| A-3 | 59.78 |
| A-4 | 60.76 |
| B-3 | 62.72 |
| C-2 | 64.68 |
| B-5 | 67.62 |
| C-4 | 67.62 |
| C-5 | 75.46 |
| A-5 | 82.32 |
| C-6 | 83.3 |
| A-6 | 97.02 |
| B-4 | 99.96 |
| B-2 | 104.86 |
| A-2 | 106.82 |

Tabla 8.5 - Tabla Orden por Cliente de la Pizzería

| ORDEN_POR_CLIENTE | |
|---|---|
| Id_Orden | Id_Cliente |
| 20506 | 101 |
| 20517 | 102 |
| 20507 | 102 |
| 20511 | 103 |
| 20512 | 104 |
| 20516 | 105 |
| 20514 | 105 |
| 20502 | 106 |
| 20509 | 107 |
| 20510 | 107 |
| 20503 | 108 |
| 20504 | 108 |
| 20515 | 109 |
| 20501 | 109 |
| 20513 | 110 |
| 20508 | 110 |
| 20505 | 110 |

La Figura 8.2 muestra una consulta de varias columnas que pertenecen a una misma tabla con dos condiciones.

```
SELECT Id_Orden, Id_Articulo, Cantidad, Total_Por_Art
FROM Detalle_De_Orden
WHERE Cantidad > 4
AND Total_Por_Art > 500;
```

Figura 8.2. Consulta de la tabla Detalle_Orden.

Como resultado esta consulta arroja una tabla con todos los registros que cumplen que el valor de Cantidad sea mayor a 4 y que el Total por Art sea mayor a 500. Ver Tabla 8.6.

Tabla 8.6 - Tabla resultante de la consulta de la Figura 8.2.

| ***DETALLE_DE_ORDEN*** | | | |
|---|---|---|---|
| ***Id_Orden*** | ***Id_Articulo*** | ***Cantidad*** | ***Total_Por_Art*** |
| 20510 | C-4 | 8 | 540.96 |
| 20514 | A-5 | 7 | 576.24 |
| 20510 | C-2 | 9 | 582.12 |
| 20517 | C-6 | 7 | 583.1 |
| 20502 | C-4 | 10 | 676.2 |
| 20503 | A-6 | 7 | 679.14 |
| 20512 | B-3 | 11 | 689.92 |
| 20508 | C-4 | 12 | 811.44 |
| 20513 | B-4 | 9 | 899.64 |
| 20515 | A-4 | 15 | 911.4 |
| 20509 | A-6 | 14 | 1358.28 |
| 20503 | A-2 | 14 | 1495.48 |
| 20516 | A-5 | 19 | 1564.08 |
| 20512 | B-5 | 25 | 1690.5 |
| 20509 | B-2 | 19 | 1992.34 |
| 20506 | C-4 | 42 | 2840.04 |

En la consulta de la Figura 8.2, se muestra una consulta sencilla que utiliza dos condiciones mediante el uso del operador lógico AND (Y en inglés); este operador permite definir una nueva condición cada que vez que se incluye en el código, con la restricción de que todas las condiciones definidas seguidas de un AND tienen que resultar verdaderas simultáneamente para que el registro sea seleccionado y mostrado como resultado.

Existen otros operadores lógicos, el operador OR (O en inglés) permite definir una nueva condición cada vez que se incluye en el código, a diferencia de AND, en este caso al menos una de las condiciones definidas tiene que resultar verdadera para que el registro sea seleccionado y mostrado como resultado.

En otras palabras, si se usa AND todas las condiciones tienen que resultar verdaderas para seleccionar el registro; en el caso de OR, bastará con que una condición resulte verdadera para que el registro sea seleccionado.

El operador NOT (NO en inglés) permite cambiar el resultado de la condición, es decir, si la condición resultara verdadera, la palabra NOT la convertirá en falsa y viceversa.

Dentro de las condiciones se utilizan operadores matemáticos, que sirven como herramientas para comparar dos valores, es decir, es un operador lógico que regresa *Verdadero* o *Falso* dependiendo si se cumple o no la comparación matemática de ambos valores. En la Figura 8.3 se muestran los posibles operadores:

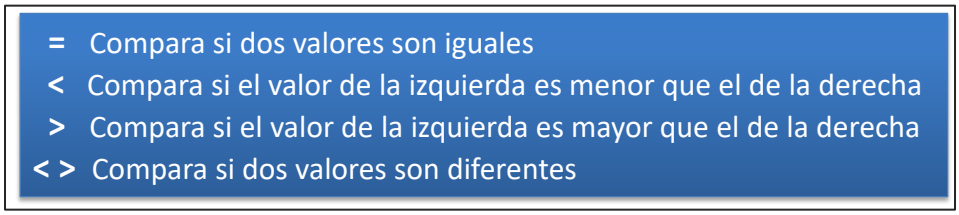


Figura 8.3 - Operadores Básicos para comparación.

Otros operadores funcionan también comparando valores de manera más explícita, lo que permite crear consultas más especializadas. Algunos ejemplos de estos se muestran en la Figura 8.4.

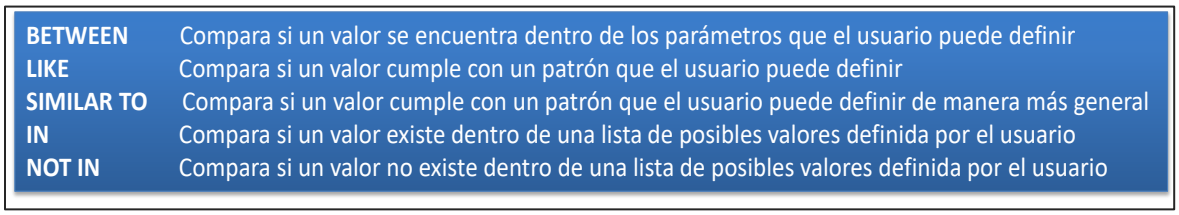


Figura 8.4 - Operadores Explícitos para comparación.

Para ejemplificar algunos de estos operadores, en la Figura 8.5, se muestra una consulta que regresa los registros cuyo nombre comienza con la letra S o el nombre es Edna o Carlos.

```
SELECT Nombre
FROM Cliente
WHERE Nombre LIKE 'S%'
OR Nombre IN ('Edna' , 'Carlos');
```

Figura 8.5 - Consulta de la tabla Nombre.

En esta sintaxis analizaremos únicamente la clausula WHERE y sus operadores:

1. WHERE
   Palabra reservada para indicar bajo qué condición seleccionaremos los registros que queremos consultar.
2. Nombre
   Es la columna que queremos comparar, es decir, los valores de cada registro de la columna Nombre de la tabla Cliente serán comparados.
3. LIKE
   Es el operador de comparación que utilizaremos.
4. '
   La comilla simple indica la apertura para escribir una cadena de caracteres.
5. S

La letra S mayúscula indica que solo se seleccionarán aquellos registros en los que el atributo nombre empiece con la letra S mayúscula.

6. %
   El porcentaje, es un comodín que deja libre el número y tipo de carácter que siga de la letra “S” del punto anterior. Es decir, discrimina el resto de la cadena. Existe el símbolo “_” (guión bajo) que indica solo la posición de un carácter dentro de una cadena, es decir, el lugar que ocupa un determinado carácter dentro de una cadena. Por ejemplo, LIKE ‘_a%’ regresaría todos los registros cuyos valores coincidan con que su segundo carácter sea la letra “a” minúscula, no importando cual sea el primer carácter ni cuales sean del tercero al último.
7. '
   La comilla simple indica el cierre de la escritura de la cadena de caracteres.
8. OR
   Indica que se definirá una nueva condición y que si un registro cumple al menos una de ellas, entonces será seleccionado.
9. Nombre
   Es la columna que queremos comparar, es decir, los valores de cada registro de la columna Nombre de la tabla Cliente serán comparados.
10. IN
    Indica que a continuación se definirá una lista de valores. Si un valor de la columna Nombre de la tabla Cliente se encuentra en ésta lista, será seleccionado por el SMBD.
11. (
    La apertura de paréntesis indica el comienzo de la lista de valores a buscar en la columna Nombre de la tabla Cliente.
12. ‘Edna’ , ‘Carlos’
    Son los valores que se deberán buscar en la columna Nombre de la tabla Cliente.
13. )
    El cierre de paréntesis indica el término de la lista de valores.
14. ;
    Indica la terminación de la instrucción.

Esta consulta seleccionará de los registros de la tabla Cliente que se muestran en la Tabla 8.1 El resultado final de la consulta se muestra en la Tabla 8.7, estos registros son mostrados ya que cumplen con alguna de las condiciones.

Tabla 8.7 - Tabla final de la consulta de la Figura 8.5.

| ***CLIENTE*** |
|---|
| ***Nombre*** |
| Sara |
| Edna |
| Carlos |

Notemos que solo nos regresa una columna ya que en la consulta solo se especificó la columna *Nombre* de la tabla *Cliente*.

El operador NOT IN es la negación del operador IN, es decir, el SMBD seleccionará todos los registros excepto los que se encuentren en la lista definida en la consulta.

A manera de repaso, la consulta mostrada en la Figura 8.6 utiliza diferentes operadores.

```
SELECT *
FROM Orden
WHERE Numero_Articulos BETWEEN 5 AND 11
AND Fecha NOT IN ('22/04/2013', '18/04/2013');
```

Figura 8.6 - Consulta a la tabla *Orden.*

En esta sintaxis únicamente se detallarán los elementos nuevos, tenemos entonces:

1. *
   El asterisco es el método abreviado para seleccionar todas las columnas de una tabla.
2. BETWEEN
   Es el operador de comparación que se utilizará y verificará que el valor del atributo se encuentre dentro de un rango especificado.
3. 5
   Primer parámetro del rango para comparar los valores de la columna en cuestión.
4. AND
   Palabra reservada para definir el segundo parámetro del rango del operador BETWEEN. No se debe confundir con el operador AND para agregar una nueva condición.
5. 11
   Segundo parámetro del rango para comparar los valores de la columna en cuestión.
6. AND

Indica que se agregará una nueva condición. La condición previa, en este caso BETWEEN, y la que se definirá a continuación deberán de resultar verdaderas para que el registro sea seleccionado.

7. Fecha
   Nombre de la columna cuyos valores se compararán.
8. NOT IN
   Indica que se definirá una lista de valores. Los registros, cuyos valores de la columna Fecha que se encuentren en ésta lista, no se seleccionarán.
9. ('22/04/2013' , '18/04/2013')
   Listado de valores a comparar, nótese que al ser una fecha ésta sigue un formato establecido.

La tabla resultante de la consulta que se encuentra en la Figura 8.6, se muestra en la Tabla 8.8.

Tabla 8.8 - Tabla resultante de la consulta de la Figura 8.7.

| ***ORDEN*** | | | |
|---|---|---|---|
| ***Id_Orden*** | ***Fecha*** | ***Numero_Articulos*** | ***Total*** |
| 20501 | 15/04/2013 | 5 | 347.90 |
| 20504 | 16/04/2013 | 6 | 486.08 |
| 20517 | 24/04/2013 | 7 | 583.10 |
| 20505 | 16/04/2013 | 8 | 568.40 |

Es posible observar que se incluyeron los resultados de las órdenes que tienen entre 5 y 11 artículos. Por otro lado los registros cuya fecha haya sido el 22/04/2013 o 18/04/2013 no se incluyeron en la lista debido a la cláusula AND de la consulta NOT IN.

### 3.2. Funciones de Agregación

Cuando el dato no se encuentra implícito en la base, es decir, que no está físicamente almacenado pero es posible derivarlo a partir del valor de otro atributo, se deberá calcular mediante funciones de agregación. Por ejemplo para contar los registros que se obtuvieron de la consulta anterior o para sumar los totales de las órdenes.

Las funciones de agregación se indican en el renglón de SELECT. En la Figura 8.7 se muestra una consulta que utiliza funciones de agregación y la cláusula GROUP BY.

```
SELECT Fecha, SUM(Total)
FROM Orden
GROUP BY Fecha;
```

Figura 8.7 - Consulta con GROUP BY a la tabla Orden.

En esta sintaxis encontramos:

1. SUM
   Palabra reservada, en este caso representa a la función suma. Es importante mencionar que la función SUM solo recibe valores de tipo numérico, en otro caso se provocará un error.
2. (Total)
   Indica la columna cuyos valores serán sumados.
3. GROUP BY
   Palabras reservadas para indicar qué columna tomará para realizar la agrupación. Este operador selecciona todos los valores diferentes que existen en una columna e indica a la función de agregación que realice su operación agrupando los resultados por cada valor de la columna que encontró el GROUP BY.
4. Fecha
   Nombre de la columna sobre la cual se realizará el agrupamiento.

La consulta de la Figura 8.8, que arrojará la suma de los totales agrupados por fecha de las Órdenes. El resultado de esta consulta se muestra en la Tabla 8.9.

Tabla 8.9 - Resultado de la consulta de la Figura 8.8.

| ***ORDEN*** | |
|---|---|
| ***Fecha*** | ***SUM (Total)*** |
| 15/04/2013 | 3687.74 |
| 16/04/2013 | 1054.48 |
| 18/04/2013 | 2840.04 |
| 19/04/2013 | 4643.24 |
| 21/04/2013 | 1415.12 |
| 22/04/2013 | 5038.18 |
| 23/04/2013 | 1865.92 |
| 24/04/2013 | 583.1 |

Otras funciones de agregación existentes son: COUNT, MIN, MAX y AVG. COUNT regresa el número de registros en la columna señalada. Si se utiliza COUNT (*) se cuentan todos los registros de la tabla. Por otra parte si se utiliza COUNT DISTINCT sólo se contarán los registros que son diferentes; es decir, que si un valor en una columna se encuentra repetido, esta función de agregación solo lo contará una vez.

MIN y MAX traen solo el registro cuyo valor sea el mínimo o máximo de una columna determinada o una agrupación. Esta función sólo puede recibir datos de tipo numérico. AVG es la función de agregación para determinar el promedio de los valores de una columna o agrupación.

**Profesor**
M. en I.A. Erick Orlando Matla Cruz
**Ayudantes**
L. en C.C. Efraín Hipólito Chamú
L. en C.C. Anahí Quiroz Jiménez
L. en C.C. Karen Monserrat Zavala Correa

## 4. Ejercicios

*(NOTA: Resuelve los siguientes ejercicios en relación al proyecto que realizarás durante el curso, en dado caso que no tengas un proyecto, utiliza la información en el apéndice Seguros parte 08 al final de esta práctica para realizarlos)*

1. Realiza 4 consultas a diferentes tablas donde utilices diferentes operadores de comparación y diferentes tipos de JOIN.
2. Realiza 4 consultas a diferentes tablas donde utilices funciones de agregación, agrupación y diferentes tipos de JOIN.
3. Realiza 2 consultas que utilicen funciones de operadores de comparación, agregación, agrupación y JOIN.

**Entregables requeridos para prácticas subsecuentes, un archivo en formato SQL:**

- 4 consultas con JOIN y operadores de comparación.
- 4 consultas con JOIN, funciones de agregación y agrupación.
- 2 consultas con JOIN, funciones de comparación, agregación y agrupación.