Antes de comenzar la clase, una Reflexión: LAS METAS.

Nadie alcanza la meta con un solo intento, ni perfecciona la vida con una sola rectificación, ni alcanza altura con un solo vuelo. Nadie camina la vida sin haber pisado en falso muchas veces.

Nadie recoge cosecha sin probar muchos sabores, enterrar muchas semillas y abonar mucha tierra. Nadie mira la vida sin acobardarse en muchas ocasiones, ni se mete en el barco sin temerle a la tempestad, ni llega a puerto sin remar muchas veces.

Nadie siente el amor sin probar sus lágrimas, ni recoge rosas sin sentir sus espinas. Nadie hace obras sin martillar sobre su edificio, ni cultiva amistad sin renunciar a sí mismo. Nadie llega a la otra orilla sin haber hecho puentes para pasar. Nadie deja el alma lustrosa sin el pulimento diario de la vida.

Nadie puede juzgar sin conocer primero su propia debilidad. Nadie consigue su ideal sin haber pensado muchas veces que perseguía un imposible. Nadie conoce la oportunidad hasta que esta pasa por su lado y la deja ir. Nadie encuentra el pozo del placer hasta caminar por la sed del desierto.

Pero nadie deja de llegar, cuando se tienen la claridad de un don, el crecimiento de la voluntad, la abundancia de la vida, el poder para realizarse y el impulso de si

mismo. Nadie deja de arder con fuego dentro de si antes de saber lo que es el calor de la amistad. Nadie deja de llegar cuando de verdad se lo propone.

Si sacas todo lo que tienes y confías en ti, esfuérzate, ¡Porque lo vas a lograr¡¡¡¡.

Mi Comentario: En mi opinión esta lectura se parece mucho a un poema, que habla mucho de una situación que todos hemos vivido, porque nuestra vida ha sido una lucha constante. Yo creo que la buena suerte existe, pero uno tiene que estar preparado para recibirla. De que vale que te ofrezcan el trabajo de tu vida, en la ciudad y el país que deseas, con el sueldo y los beneficios que siempre has soñado sino tienes la preparación y el conocimiento necesario para asumirlo.

A veces vemos a un individuo exitoso y decimos: "Que suerte tiene¡¡¡¡¡¡¡". Es verdad tiene buena suerte y además se encuentra preparado para recibirla.....

USO DEL SELECT.

En el Capítulo I , se explicó la sintaxis básica para utilizar la instrucción SELECT.

En lo sucesivo se ampliará esta información.

OMITIENDO LAS FILAS DUPLICADAS EN UNA CONSULTA

La cláusula **DISTINCT** elimina todas aquellas filas que se encuentran duplicadas en una consulta. (información retornada u obtenida por un **SELECT**).

Sintaxis

SELECT DISTINCT columna1, columna2 . . .
FROM tabla

Ejemplo

Muestra de manera unívoca los Clientes que tienen al menos una Factura.

```
SQL> select distinct codcli
  2  from facturas
  3  where estatus='A';

CODC
----
0002
0003
0004
0006
0007
0008
0009

7 rows selected.
```

USO DE ALIAS

El alias representa un seudónimo o nombre temporal que se le da a una columna o a una tabla, para facilitar el uso de la misma en una instrucción.

Sintaxis

SELECT columna1 as “nombre_alias”, columna2 as “nombre_alias2”
FROM nombre_tabla nombre_alias

Ejemplo

Mostrar el Nombre de los Clientes con Alias “Nombre del Cliente”, desde la Tabla Clientes con Alias cl. Cuando se utiliza Alias en los nombres de las Tablas, se puede lograr que los QUERYS sean mas cortos porque a veces los nombres de las tablas son largos, por eso se dice que de esta manera se pueden resumir.

Alias en las columnas

```
SQL> select nombre as "Nombre del Cliente"
  2  from clientes cl
  3  where estatus = 'A';

Nombre del Cliente
-------------------------
MAKRO
UCLA
EXITO
LOCATEL
BECO
RENE DESSES
SIDETUR
BANCOR
UPEL
EPA

10 rows selected.
```

CONCATENACION DE VALORES EN LAS COLUMNAS

Cuando en una consulta se necesita adicionar algún literal o concatenar dos o más columnas para mejorar la salida se utiliza el símbolo "PIPE" (| |). El literal debe estar encerrado entre comillas.

Sintaxis

SELECT campo1 | | ´Literal´, campo2 | | campo3, campo4,.., campoN
FROM tabla

Ejemplo

Mostrar la Fecha de Emisión y vencimiento de las facturas en términos amigables para los usuarios.

```
SQL> select 'La factura '||numero||' se emitió el: '||fecha||' y se vence el: ',fecha+diasvigencia
  2  from facturas
  3  where estatus = 'A';

'LAFACTURA'||NUMERO||'SEEMITIÓEL:'||FECHA||'YSEVENCEEL FECHA+DI
---------------------------------------------------- --------
La factura 1851 se emitió el: 05/01/04 y se vence el:  04/02/04
La factura 1852 se emitió el: 05/01/04 y se vence el:  05/03/04
La factura 1853 se emitió el: 06/01/04 y se vence el:  20/02/04
La factura 1854 se emitió el: 06/01/04 y se vence el:  05/02/04
La factura 1855 se emitió el: 07/01/04 y se vence el:  22/01/04
La factura 1856 se emitió el: 07/01/04 y se vence el:  06/02/04
La factura 1857 se emitió el: 08/01/04 y se vence el:  23/01/04
La factura 1858 se emitió el: 09/01/04 y se vence el:  09/03/04
La factura 1859 se emitió el: 09/01/04 y se vence el:  24/01/04
La factura 1860 se emitió el: 10/01/04 y se vence el:  30/01/04
La factura 1861 se emitió el: 11/01/04 y se vence el:  26/01/04
La factura 1862 se emitió el: 12/01/04 y se vence el:  22/01/04

12 rows selected.
```

Ciertamente el título de la Salida emitida no parece encajar con el resto de la Información, sin embargo, el estudiante no debe preocuparse por ese título ahora, mas adelante aprenderá que ese detalle tiene alternativas de solución. Aquí lo importante es que se visualice la capacidad de concatenar datos en procesos de emisión de información.

Aunque no hemos llegado a ese punto, se puede notar también que al sumarle a una fecha un número determinado de días, automáticamente se calcula una nueva fecha.

USO DE OPERADORES MATEMÁTICOS

Las expresiones aritméticas en una instrucción SQL, ejecutan varios cálculos numéricos usando los operadores aritméticos.

Mas de un operador aritmético puede ser utilizado en una instrucción SQL. Las expresiones aritméticas en una instrucción SELECT son mostradas como una columna más.

Ejemplo

Se desea realizar una proyección estimada de Precios. Para ello se solicita sumarle el 40% al costo de los productos para visualizar como quedarían los precios de

venta. En ese caso, es necesario realizar una Operación aritmética sobre la Tabla de Artículos como sigue:

```
SQL> select codigo,descripcion, costo, ((costo*0.40)+costo) as "Precio"
  2  from articulos
  3  where estatus = 'A';

CODI DESCRIPCION                       COSTO    Precio
---- ------------------------------ --------- ---------
0001 Nevera                             500       700
0002 Enfriador                         1630      2282
0003 Dvd                                420       588
0004 Cava Cuarto Pequeña               1300      1820
0005 Ventilador Mediano                  53      74,2
0006 Ventilador Pequeño                  29      40,6
0007 Licuadora                          459     642,6
0008 Quemador de Cds                    391     547,4

8 rows selected.
```

El núcleo de ORACLE evalúa cada expresión aritmética. El resultado de la expresión está determinado por el orden de precedencia de los operadores.

Orden de Evaluación	
*	MULTIPLICACION
/	DIVISIÓN
+	SUMA
-	RESTA

Cuando una expresión o una función hace referencia a una columna que contiene un valor nulo, el resultado también es nulo.

Ejemplo

Se desea emitir una salida, que sume 500 Unidades al Límite de Crédito de cada Cliente, con el objetivo de visualizar como quedaría un eventual Límite de Crédito nuevo. Ahora bien, se debe tener en cuenta que al Cliente MAKRO no le aparece Límite de Crédito, ni siquiera se encuentra en cero, por eso se concluye que esa columna (o campo) tiene valor NULO.

Al emitir la salida, los resultados son los siguientes:

```
SQL> select codigo,nombre,limcre as "Límite Crédito", limcre+500 as "Límite Propuesto"
  2  from clientes
  3  where estatus = 'A';

CODI NOMBRE                    Límite Crédito Límite Propuesto
---- ------------------------ -------------- ----------------
0001 MAKRO
0004 UCLA                               1100             1600
0005 EXITO                              7800             8300
0006 LOCATEL                            7000             7500
0002 BECO                               2500             3000
0008 RENE DESSES                        9500            10000
0007 SIDETUR                            7500             8000
0009 BANCOR                             9900            10400
0010 UPEL                               1100             1600
0003 EPA                                2800             3300

10 rows selected.
```

En lo sucesivo, nunca se debe olvidar que la existencia de un Valor NULO puede afectar cualquier cálculo, ya que al aplicarle cualquier operación, el Resultado será NULO.

USO DE FUNCIONES EN LAS COLUMNAS

TRUNC y ROUND

Ejemplo

Mostrar el Código del Cliente, el Límite de Crédito, el 12.5% del Límite de Crédito, el 12.5% del límite de crédito sin tomar en cuenta los Decimales, y el 12,5% del Límite de crédito redondeado a su valor superior.

```
SQL> select codigo,limcre,limcre*0.125,trunc(limcre*0.125),round(limcre*0.125)
  2  from clientes
  3  where estatus = 'A';

CODI    LIMCRE LIMCRE*0.125 TRUNC(LIMCRE*0.125) ROUND(LIMCRE*0.125)
---- --------- ------------ ------------------- -------------------
0001
0004      1100        137,5                 137                 138
0005      7800          975                 975                 975
0006      7000          875                 875                 875
0002      2500        312,5                 312                 313
0008      9500       1187,5                1187                1188
0007      7500        937,5                 937                 938
0009      9900       1237,5                1237                1238
0010      1100        137,5                 137                 138
0003      2800          350                 350                 350

10 rows selected.
```

UPPER y SUBSTR

Ejemplo

Mostrar todas las Descripciones en Mayúsculas, así como los 5 primeros caracteres de la Descripción de cada Artículo.

```
SQL> select descripcion, upper(descripcion),substr(descripcion,1,5)
  2  from articulos
  3  where estatus = 'A';

DESCRIPCION                    UPPER(DESCRIPCION)             SUBST
------------------------------ ------------------------------ -----
Nevera                         NEVERA                         Never
Enfriador                      ENFRIADOR                      Enfri
Dvd                            DVD                            Dvd
Cava Cuarto Pequeña            CAVA CUARTO PEQUEÑA            Cava
Ventilador Mediano             VENTILADOR MEDIANO             Venti
Ventilador Pequeño             VENTILADOR PEQUEÑO             Venti
Licuadora                      LICUADORA                      Licua
Quemador de Cds                QUEMADOR DE CDS                Quema

8 rows selected.
```

AVG,MAX,SUM

Ejemplo

Mostrar el promedio de los Costos, el Costo Mayor y la Suma de todos los Costos de la Tabla de Artículos.

```
SQL> select avg(costo),max(costo),sum(costo)
  2  from articulos
  3  where estatus = 'A';

AVG(COSTO) MAX(COSTO) SUM(COSTO)
---------- ---------- ----------
    597,75       1630       4782
```

FECHAS

Las Operaciones con Fechas se consideran importantes. Una Fecha se puede descomponer en Día, Mes y Año. Para ello se usa la Función TO_CHAR.

TO_CHAR

Los Parámetros de la esta Función se encuentran en la siguiente TABLA:

Elemento	Descripción
DD	Día del mes.
D	Día de la semana.
DAY	Nombre del día en mayúscula.
MM	Mes (01 – 12).
MONTH	Nombre del mes en mayúscula.
MON	Las tres primeras letras del mes en mayúscula.
HH: MI: SS	Horas minutos y segundo.
Fm	Suprime los blancos.

Ejemplo

Descomponer las Fechas de las facturas en Día, Mes y año.

```
SQL> select numero,fecha, to_char(fecha,'dd') as "Día",to_char(fecha,'Month') as "Mes",
  2  to_Char(fecha,'yyyy') as "Año"
  3  from facturas
  4  where estatus = 'A';

NUME FECHA    Dí Mes        Año
---- -------- -- ---------- ----
1851 05/01/04 05 Enero      2004
1852 05/01/04 05 Enero      2004
1853 06/01/04 06 Enero      2004
1854 06/01/04 06 Enero      2004
1855 07/01/04 07 Enero      2004
1856 07/01/04 07 Enero      2004
1857 08/01/04 08 Enero      2004
1858 09/01/04 09 Enero      2004
1859 09/01/04 09 Enero      2004
1860 10/01/04 10 Enero      2004
1861 11/01/04 11 Enero      2004
1862 12/01/04 12 Enero      2004

12 rows selected.
```

Mostrar la misma información en el siguiente formato: Número de la Factura, Fecha de la Factura con mes en letras – día en número – año en número.

```
SQL> select numero,substr(to_char(fecha,'fm month dd,yyyy'),1,20) as "Fecha Factura"
  2  from facturas
  3  where estatus = 'A';

NUME Fecha Factura
---- --------------------
1851  enero 5,2004
1852  enero 5,2004
1853  enero 6,2004
1854  enero 6,2004
1855  enero 7,2004
1856  enero 7,2004
1857  enero 8,2004
1858  enero 9,2004
1859  enero 9,2004
1860  enero 10,2004
1861  enero 11,2004
1862  enero 12,2004

12 rows selected.
```

OPERACIONES CON FECHA	
FECHA + número	Adiciona un número de días a una fecha, generando una nueva fecha.
FECHA - número	Sustrae un número de días a una fecha, generando una nueva fecha.
FECHA - FECHA2	Resta dos fechas, produciendo un número de días.

Ejemplo

Mostrar el Número de la Factura , Fecha de Emisión, Días de Vigencia, Fecha de Vencimiento la cual se calculará al sumarle los Días de Vigencia.

```
SQL> select numero, fecha as "Emisión",diasvigencia as "Días",
  2  fecha+diasvigencia as "Vencimiento"
  3  from facturas
  4  where estatus='A';

NUME Emisión       Días Vencimie
---- -------- --------- --------
1851 05/01/04        30 04/02/04
1852 05/01/04        60 05/03/04
1853 06/01/04        45 20/02/04
1854 06/01/04        30 05/02/04
1855 07/01/04        15 22/01/04
1856 07/01/04        30 06/02/04
1857 08/01/04        15 23/01/04
1858 09/01/04        60 09/03/04
1859 09/01/04        15 24/01/04
1860 10/01/04        20 30/01/04
1861 11/01/04        15 26/01/04
1862 12/01/04        10 22/01/04

12 rows selected.
```

Obtener la Fecha del Sistema.

Para obtener la Fecha del Sistema se utiliza el sysdate de la siguiente manera:

```
SQL> select to_char(sysdate,'dd/mm/yyyy') as "Hoy es:"
  2  from dual;

Hoy es:
----------
21/04/2005
```

Donde DUAL representa una Tabla inexistente utilizada para completar este tipo de sintaxis. También puede realizar cualquier operación aritmética utilizando el DUAL, como sigue:

```
SQL> select 5+5 from dual;

       5+5
----------
        10
```

Obtener la Hora del Sistema.

Para obtener la Fecha del Sistema se utiliza el sysdate de la siguiente manera:

```
SQL> select to_char(sysdate,'HH24:MI:SS') as "La hora es:"
  2  from dual;

La hora
--------
18:04:30
```

NVL

La función NVL convierte un valor nulo en un valor no nulo, para los efectos de evaluación en una expresión. Contribuye a minimizar el Problema de los NULOS en las operaciones aritméticas.

Sintaxsis

NVL (Exp1, Exp2)

Cuando la función NVL evalúa una expresión retorna el siguiente valor:

Exp1: Si el valor no es nulo.

Exp2: Si el valor de Exp1 es nulo.

Ejemplo

Mostrar el código, Límite de Crédito, Límite de Crédito mas 500, luego el mismo Límite de Crédito mas 500 pero aplicando la función NVL.

```
SQL> select codigo,limcre,limcre+500,nvl(limcre,0)+500
  2  from clientes
  3  where estatus='A';

CODI    LIMCRE LIMCRE+500 NVL(LIMCRE,0)+500
---- --------- ---------- -----------------
0001                                    500
0004      1100       1600              1600
0005      7800       8300              8300
0006      7000       7500              7500
0002      2500       3000              3000
0008      9500      10000             10000
0007      7500       8000              8000
0009      9900      10400             10400
0010      1100       1600              1600
0003      2800       3300              3300

10 rows selected.
```

DECODE

Ejemplo

Mostrar el Código de cada Artículo, Descripción, Costo, luego examinar las siguientes alternativas:

A. Si el Costo del Producto es 500 Sumarle 100 para calcular el Precio.

B. Si el Costo es 1630 sumarle 50 para calcular el Precio.

C. Cualquier otro costo sumarle 10 para calcular el Precio.

```
SQL> select codigo,descripcion, costo,
  2  decode(costo, 500, costo+100,
  3                 1630, costo+50,
  4                       costo+10) as "Precio"
  5  from articulos
  6  where estatus = 'A';

CODI DESCRIPCION                       COSTO    Precio
---- ------------------------------ --------- ---------
0001 Nevera                             500       600
0002 Enfriador                         1630      1680
0003 Dvd                                420       430
0004 Cava Cuarto Pequeña               1300      1310
0005 Ventilador Mediano                  53        63
0006 Ventilador Pequeño                  29        39
0007 Licuadora                          459       469
0008 Quemador de Cds                    391       401

8 rows selected.
```

COUNT

Ejemplo

Mostrar la cantidad de Artículos que existen

```
SQL> select count(*) as "Cant Artículos"
  2  from articulos
  3  where estatus = 'A';

Cant Artículos
--------------
             8
```

OTROS USOS DE LA CLAUSULA WHERE

La cláusula WHERE se utiliza para seleccionar un conjunto de filas específicas.

Su sintaxis es:

SELECT lista de columnas

FROM Nombre de la Tabla

WHERE condición;

UTILIZANDO LOS OPERADORES DE COMPARACION

BETWEEN

Ejemplo

Mostrar aquellas facturas que fueron elaboradas entre el 5 de Enero del 2004 y el 10 de Enero del 2004.

```
SQL> select * from facturas
  2  where fecha between '05/01/04' and '10/01/04'
  3  and estatus = 'A';

NUME CODC FECHA    DIASVIGENCIA  MONTOTAL E
---- ---- -------- ------------ --------- -
1851 0004 05/01/04           30      4500 A
1852 0003 05/01/04           60      6000 A
1853 0007 06/01/04           45      7500 A
1854 0004 06/01/04           30      1000 A
1855 0002 07/01/04           15      4000 A
1856 0007 07/01/04           30      2000 A
1857 0006 08/01/04           15      4500 A
1858 0009 09/01/04           60      7500 A
1859 0008 09/01/04           15     10000 A
1860 0009 10/01/04           20      2500 A

10 rows selected.
```

IN

IN representa la Operación PERTENECE de la Teoría de Conjuntos. Sirve para preguntar si un Valor PERTENECE a un conjunto determinado.

Ejemplo

Mostrar código, descripción y costo de aquellos artículos con costo igual a 500,420 y 391.

```
SQL> select codigo,descripcion,costo
  2  from articulos
  3  where costo IN (500,420,391)
  4  and estatus = 'A';

CODI DESCRIPCION                       COSTO
---- ------------------------------ ---------
0001 Nevera                             500
0003 Dvd                                420
0008 Quemador de Cds                    391
```

LIKE

Comodines que pueden ser utilizados:

%	Sustitución completa
_	Sustitución posicional

Ejemplos

Mostrar el Código, Nombre y Límite de Crédito de aquellos Clientes cuyo Nombre comienza con la Letra B.

```
SQL> select codigo, nombre, limcre
  2  from clientes
  3  where nombre like 'B%'
  4  and estatus = 'A';

CODI NOMBRE                         LIMCRE
---- ------------------------- ---------
0002 BECO                           2500
0009 BANCOR                         9900
```

Mostrar los Datos de los Clientes cuya segunda Letra del Nombre sea una A. Sin importar con letra empiece y sin revisar lo que sigue después de la segunda Letra.

```
SQL> select * from clientes
  2  where nombre like '_A%'
  3  and estatus = 'A';

CODI NOMBRE                         LIMCRE E
---- ------------------------- --------- -
0001 MAKRO                              A
0009 BANCOR                          9900 A
```

UTILIZANDO LOS OPERADORES LOGICOS

NOT, AND Y OR

Cuando una condición está compuesta por dos o mas expresiones el RDBMS evalúa cada expresión. El resultado de todas las condiciones está determinado por el orden de precedencia de los operadores.

Operadores de igual precedencia

=, !=, >, >=, IN, LIKE, IS NULL, BETWEEN . . . AND . . .

Los operadores lógicos son evaluados en este orden:

1	NOT
2	AND
3	OR

Para alterar las reglas de precedencia, se debe colocar parte de la expresión entre paréntesis; lo cual hará que dicha expresión se evalúe primero.

Ejemplo

Mostrar código, descripción y costo de aquellos artículos con costo Diferente a 500,420 y 391.

```
SQL> select codigo,descripcion,costo
  2  from articulos
  3  where costo NOT IN (500,420,391)
  4  and estatus = 'A';

CODI DESCRIPCION                       COSTO
---- ------------------------------ ---------
0002 Enfriador                           1630
0004 Cava Cuarto Pequeña                 1300
0005 Ventilador Mediano                    53
0006 Ventilador Pequeño                    29
0007 Licuadora                            459
```

UTILIZANDO LA CLAUSULA ORDER BY.

La cláusula ORDER BY permite ordenar la información obtenida de una consulta de diferentes maneras; ya sea por una o por varias columnas de las especificadas en la instrucción SELECT, estas columnas deben ser referenciadas en la cláusula ORDER BY ya sea a través del nombre o del número correspondiente a la posición de la columna.

Sintaxis

```
SELECT columna1, column2, columna3
FROM nombre_tabla
WHERE condición
ORDER BY columna1 [ DESC  o  ASC ]
```

Ejemplo

Usando la cláusula ORDER BY, para ordenar ascendentemente una salida. Se pide mostrar los Clientes Ordenados por Nombre.

```
SQL> select codigo,nombre,limcre
  2  from clientes
  3  where estatus = 'A'
  4  order by nombre;

CODI NOMBRE                         LIMCRE
---- ------------------------- ---------
0009 BANCOR                          9900
0002 BECO                            2500
0003 EPA                             2800
0005 EXITO                           7800
0006 LOCATEL                         7000
0001 MAKRO
0008 RENE DESSES                     9500
0007 SIDETUR                         7500
0004 UCLA                            1100
0010 UPEL                            1100

10 rows selected.
```

Usando la cláusula ORDER BY, para ordenar por múltiples columnas.

Ejemplo

Mostrar el Código, Nombre, Límite de Crédito de cada Cliente ordenado por Límite de Crédito y dentro de ese Límite por Nombre.

```
SQL> select codigo,nombre,limcre
  2  from clientes
  3  where estatus = 'A'
  4  order by limcre,nombre;

CODI NOMBRE                         LIMCRE
---- ------------------------- ---------
0004 UCLA                            1100
0010 UPEL                            1100
0002 BECO                            2500
0003 EPA                             2800
0006 LOCATEL                         7000
0007 SIDETUR                         7500
0005 EXITO                           7800
0008 RENE DESSES                     9500
0009 BANCOR                          9900
0001 MAKRO

10 rows selected.
```

UTILIZANDO LA CLAUSULA GROUP BY

La cláusula GROUP BY permite realizar operaciones de grupo sobre las columnas que conforman una tabla. Debe existir al menos una función de grupo en el SELECT y todas las columnas que no son funciones de grupo deben estar referenciadas en la cláusula GROUP BY.

Ejemplo

Mostrar Código del Clientes, el Número de facturas que posee y el Total de esas facturas por cada Cliente.

```
SQL> select codcli,count(*) "Num Facturas",sum(montotal) "Total"
  2  from facturas
  3  where estatus = 'A'
  4  group by codcli;

CODC Num Facturas     Total
---- ------------ ---------
0002            1      4000
0003            1      6000
0004            2      5500
0006            1      4500
0007            2      9500
0008            3     22500
0009            2     10000

7 rows selected.
```

UTILIZANDO LA CLAUSULA HAVING

La cláusula **HAVING** es utilizada para condicionar a la cláusula **GROUP BY.**

Ejemplo

Mostrar la misma información anterior, teniendo como restricción que se mostrarán solo los Totales inferiores a 10,000, ordenados por los Totales.

```
SQL> select codcli,count(*) "Num Facturas",sum(montotal) "Total"
  2  from facturas
  3  where estatus = 'A'
  4  group by codcli
  5  having sum(montotal)< 10000
  6  order by sum(montotal);

CODC Num Facturas     Total
---- ------------ ---------
0002            1      4000
0006            1      4500
0004            2      5500
0003            1      6000
0007            2      9500
```

PRACTICA II

Utilice la tabla ARTICULOS para realizar los siguientes ejercicios:

1. Para cada Artículo muestre el Código, Descripción y Existencia.
2. Muestre los Artículos cuyo Costo sea mayor de 500 Bs. Y coloque el encabezado "Articulo" a la columna Descripción.
3. Muestre todos los códigos y Descripciones de todos los Artículos, esta información debe ser ordenada por código y Descripción.
4. Muestre la Descripción de todos aquellos Artículos donde se cumpla que la letra "V" este contenida en la primera posición de dicha Descripción, o la letra "v" en la tercera posición.
5. Muestre todos aquellos Artículos cuya Existencia este en el rango de 200 a 615.
6. Muestre todos los Artículos con su Descripción en Mayúscula, el listado debe estar ordenado por Descripción, por Existencia y por Costo.
7. Muestre la fecha de cada Factura, en los siguientes formatos:

a	31 Diciembre, 1993
b	Dic 31 93
c	12/31/93

DESPUÉS DE ESTA CLASE TANNNNNNN!!!!!! LARGA: Algo de RELAX.

La Viejita estaba trabajando en el Jardín de su casa, cuando su pequeño nieto se acercó y le preguntó:

¿Abuela, Como se llama cuando una persona duerme encima de la otra?.

La Viejita lo pensó bastante y decidió decirle la verdad: "Se llama Relaciones Sexuales".

Inmediatamente, el nieto se retiró a seguir jugando.

Pasados 15 Minutos, el niño se apareció nuevamente y le dijo en tono muy enfadado:

Abuela no seas mentirosa!!!!!. Eso que te pregunté se llama LITERA y la Mamá de Carlitos quiere hablar contigo!!!!!!!!!.