Universidad César Vallejo

Pregrado

SESIÓN 11:
Diseño de base de datos: - Tablas y relaciones. - Sentencias SQL

Instrucciones básicas

A nivel de Base de datos se realizan instrucciones básicas de SQL estas son Select, Insert, Delete y Update.

La sentencia SELECT:

La selección total o parcial de una tabla se lleva a cabo mediante la instrucción **Select**. En dicha selección hay que especificar:

-Los campos que queremos seleccionar

-La tabla en la que hacemos la selección

En nuestra tabla modelo de clientes podríamos hacer por ejemplo una selección del nombre y razon social de los clientes con una instrucción de este tipo:

Select nombre, razonsocial From clientes

Si quisiésemos seleccionar todos los campos, es decir, **toda la tabla**, podríamos utilizar el comodín * del siguiente modo:

Select * From Clientes

Resulta también muy útil el filtrar los registros mediante condiciones que vienen expresadas después de la **cláusula Where**. Si quisiésemos mostrar los clientes con determinado apellido paterno usaríamos una expresión como esta:

```
Select * From clientes
Where Ap_Paterno='Gonzalez'
```

Además, podríamos **ordenar los resultados** en función de uno o varios de sus campos. Para este ultimo ejemplo los podríamos ordenar por nombre así:

```
Select * From clientes
Where Ape_Paterno='Gonzalez'
Order By nombres
```

Teniendo en cuenta que puede haber más de un cliente con el mismo nombre, podríamos dar un segundo criterio que podría ser el apellido:

```
Select * From clientes
Where Ape_Paterno='Gonzalez
Order By nombres, Ape_Materno
```

Es posible también **clasificar por orden inverso**. Si por ejemplo quisiésemos ver nuestros clientes por apellido paterno en orden descendente escribiríamos algo así:

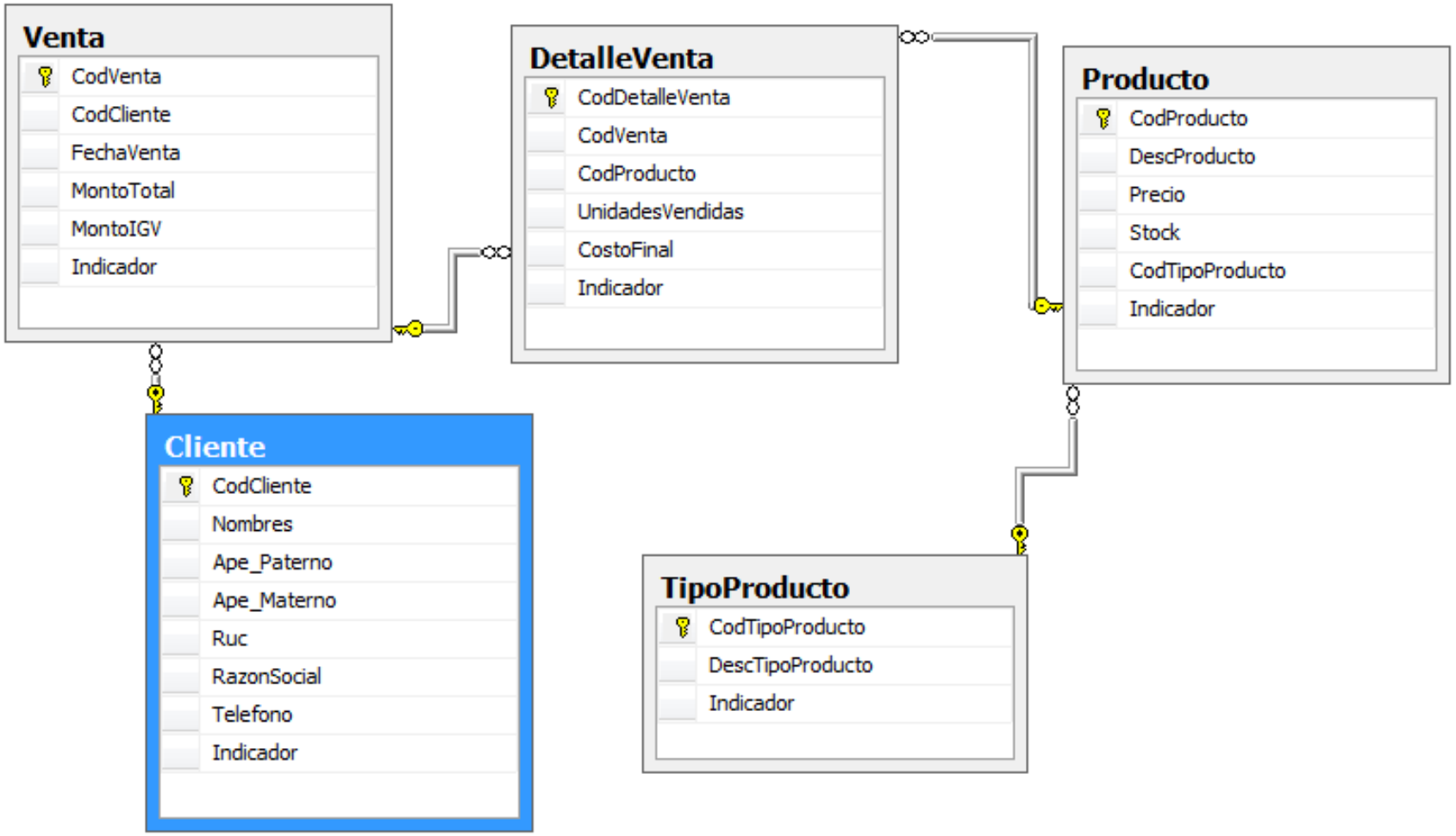
```
Select * From clientes
Order By pedidos Desc
```

Una opción interesante es la de efectuar **selecciones sin coincidencia**. Si por ejemplo buscásemos el saber en qué distritos se encuentran nuestros clientes sin necesidad de que para ello aparezca varias veces el mismo distrito usaríamos una sentencia de esta clase:

```
Select Distinct distrito
From clientes Order By distrito
```

Selección con Múltiples Tablas

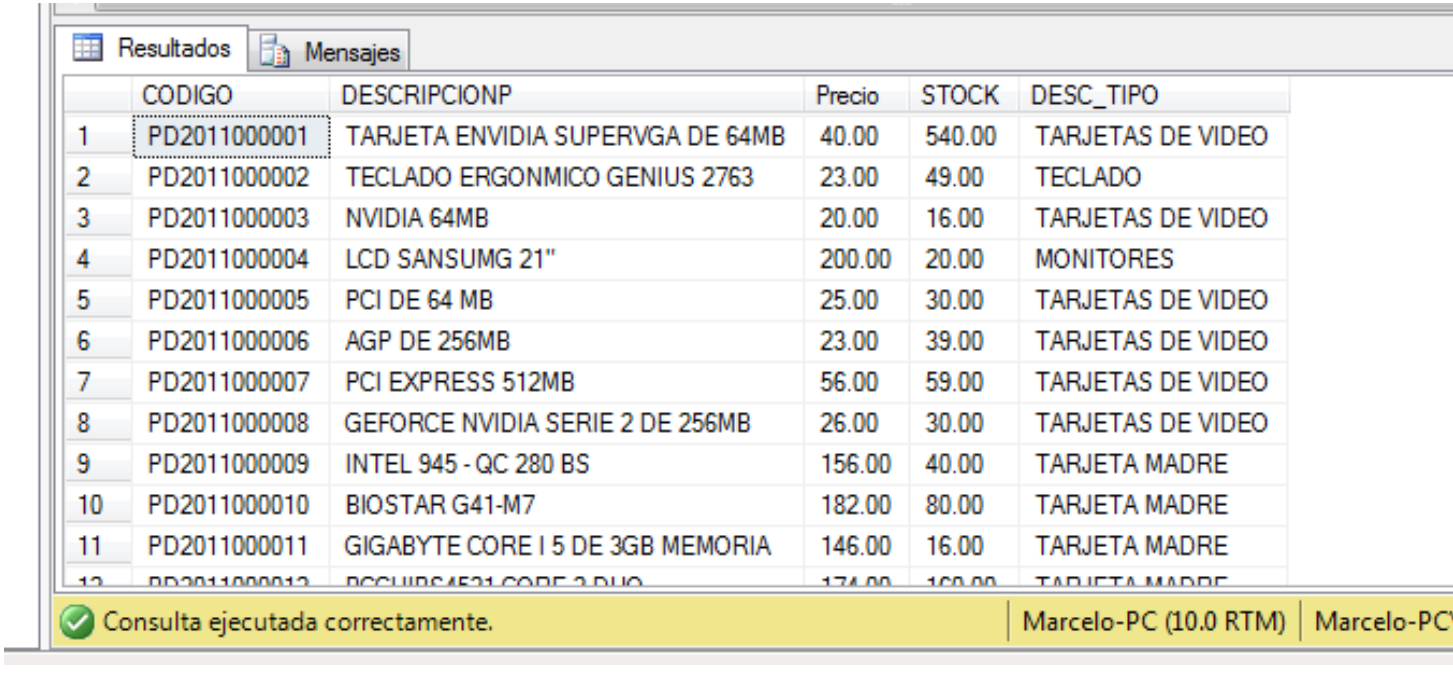
Asumiendo que tenemos la siguiente base de datos venta donde se encuentran tablas relacionas



Si quisiéramos traernos la información de los productos que se han registrado con su respetico tipo de producto daríamos la siguiente consulta

```
SELECT  PRODUCTO.CodProducto AS CODIGO,
        PRODUCTO.DescProducto AS DESCRIPCIONP,
        PRODUCTO.Precio AS Precio,
        PRODUCTO.Stock AS STOCK,
        TIPOPRODUCTO.DescTipoProducto AS DESC_TIPO
FROM PRODUCTO,TIPOPRODUCTO
WHERE (PRODUCTO.Indicador = 'S')
    AND PRODUCTO.CODTIPOPRODUCTO=TIPOPRODUCTO.CODTIPOPRODUCTO
```

Esta consulta traería la siguiente información de la Base de Datos



Resultados | Mensajes

	CODIGO	DESCRIPCIONP	Precio	STOCK	DESC_TIPO
1	PD2011000001	TARJETA ENVIDIA SUPERVGA DE 64MB	40.00	540.00	TARJETAS DE VIDEO
2	PD2011000002	TECLADO ERGONMICO GENIUS 2763	23.00	49.00	TECLADO
3	PD2011000003	NVIDIA 64MB	20.00	16.00	TARJETAS DE VIDEO
4	PD2011000004	LCD SANSUMG 21"	200.00	20.00	MONITORES
5	PD2011000005	PCI DE 64 MB	25.00	30.00	TARJETAS DE VIDEO
6	PD2011000006	AGP DE 256MB	23.00	39.00	TARJETAS DE VIDEO
7	PD2011000007	PCI EXPRESS 512MB	56.00	59.00	TARJETAS DE VIDEO
8	PD2011000008	GEFORCE NVIDIA SERIE 2 DE 256MB	26.00	30.00	TARJETAS DE VIDEO
9	PD2011000009	INTEL 945 - QC 280 BS	156.00	40.00	TARJETA MADRE
10	PD2011000010	BIOSTAR G41-M7	182.00	80.00	TARJETA MADRE
11	PD2011000011	GIGABYTE CORE I 5 DE 3GB MEMORIA	146.00	16.00	TARJETA MADRE

Consulta ejecutada correctamente. | Marcelo-PC (10.0 RTM) | Marcelo-PC

También si quisiéramos traer información de la tabla ventas y cliente que están unidas a través de campo CodCliente ha daríamos la siguiente consulta utilizando la clausula Inner Join como:

```
SELECT Venta.CodVenta, Venta.CodCliente,Venta.FechaVenta, Venta.MontoTotal,
       Venta.MontoIGV,Cliente.Nombres, Cliente.Ape_Paterno,
       Cliente.Ape_Materno, Cliente.Ruc,Cliente.RazonSocial,
       Cliente.Telefono
FROM Venta
INNER JOIN Cliente ON Venta.CodCliente = Cliente.CodCliente
WHERE Venta.Indicador='S'
```

Esta consulta traería la siguiente información de la Base de Datos

Resultados | Mensajes

	CodVenta	CodCliente	FechaVenta	MontoTotal	MontoIGV	Nombres	Ape_Paterno	Ape_Materno	Ruc	RazonSocial
1	VT2011000006	CL2011000023	2011-05-23 14:24:39.947	2000.00	380.00	DANIELA	PEREZ	SALAZAR	45217485963	CALLE LAS GARDE
2	VT2011000007	CL2011000003	2011-05-23 21:43:13.897	0.00	0.00	NAZLY	ESTRADA	GONZALEZ	85623254125	AV. UNO MZ V1 LO
3	VT2011000008	CL2011000016	2011-05-23 21:45:12.560	432.00	82.08	LILY	SALAZAR	CHAVEZ	65498765412	JESUS MARIA
4	VT2011000009	CL2011000004	2011-05-24 08:25:24.033	371.00	70.49	FERNANDA	MARTINEZ	ARELLANO	32165498750	CALLE SAN GENER
5	VT2011000010	CL2011000022	2011-05-24 08:33:37.410	3135.00	595.65	NELSON	DIAZ	LEIVA	98746321321	CALLAO
6	VT2011000011	CL2011000002	2011-05-24 08:48:06.980	2262.00	429.78	PEDRO	MOLINA	VELARDE	12369854258	AV. ROSA TORO 56

Consulta ejecutada correctamente. | Marcelo-PC (10.0 RTM) | Marcelo-PC\Marcelo (56) | BDVentas | 00:00:00 | 6 filas

Además de los criterios hasta ahora explicados para realizar las consultas en tablas, SQL permite también aplicar un conjunto de funciones predefinidas. Estas funciones, aunque básicas, pueden ayudarnos en algunos momentos a expresar nuestra selección de una manera más simple sin tener que recurrir a operaciones adicionales por parte del script del lenguaje de programación que estemos utilizados.

Aquí algunas funciones relevantes:

FUNCION	DESCRIPCION
Sum(campo)	Calcula la suma de los registros del campo especificado
Avg(Campo)	Calcula la media de los registros del campo especificado
Count(*)	Nos proporciona el valor del numero de registros que han sido seleccionados
Max(Campo)	Nos indica cual es el valor máximo del campo
Min(Campo)	Nos indica cual es el valor mínimo del campo

Si quisiera sumar todas las ventas que hay en la tabla ventas haria la siguiente consulta:

```
Select Sum(MontoTotal) As Suma_Ventas
From Venta
```

Resultados | Mensajes

	Suma_Ventas
1	8200.00

Una cláusula interesante en el uso de funciones es Group By. Esta cláusula nos permite agrupar registros a los cuales vamos a aplicar la función. Podemos por ejemplo calcular el **dinero gastado por cada cliente**:

```
Select CodCliente, Sum(MontoTotal) as suma_ventas
From Venta
Group By CodCliente
```

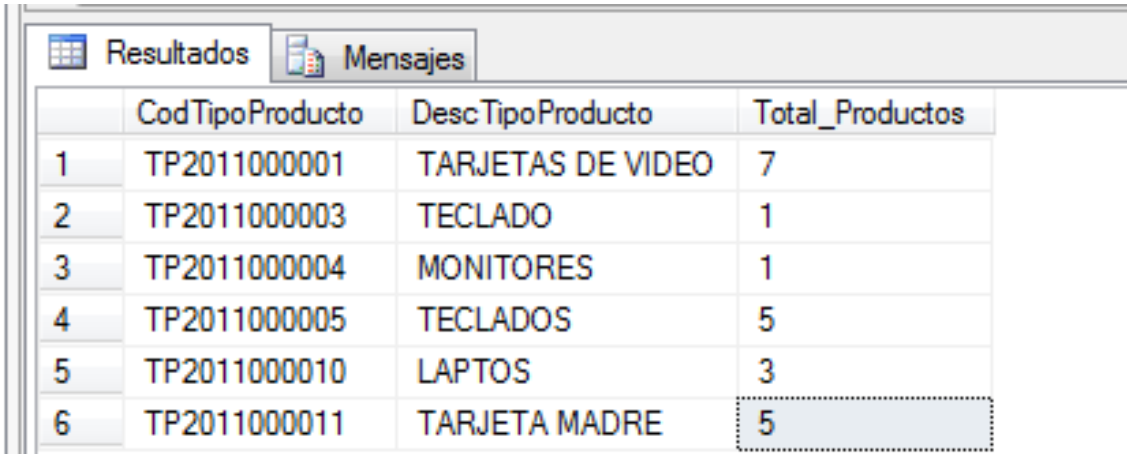
Resultados | Mensajes

	CodCliente	suma_ventas
1	CL2011000002	2262.00
2	CL2011000003	0.00
3	CL2011000004	371.00
4	CL2011000016	432.00
5	CL2011000022	3135.00
6	CL2011000023	2000.00

Si se quiere saber cuantos productos se han registrado por cada tipo de producto se haría la siguiente instrucción:

```
Select TipoProducto.CodTipoProducto,TipoProducto.DescTipoProducto,
       Count(*) as Total_Productos
From TipoProducto,Producto
Where TipoProducto.CodTipoProducto=Producto.CodTipoProducto
Group By TipoProducto.CodTipoProducto,TipoProducto.DescTipoProducto
```

Esta consulta traería la siguiente información de la Base de Datos



Resultados | Mensajes

	CodTipoProducto	DescTipoProducto	Total_Productos
1	TP2011000001	TARJETAS DE VIDEO	7
2	TP2011000003	TECLADO	1
3	TP2011000004	MONITORES	1
4	TP2011000005	TECLADOS	5
5	TP2011000010	LAPTOS	3
6	TP2011000011	TARJETA MADRE	5

Operadores en las Consultas

Los operadores lógicos soportados por SQL son: AND, OR, XOR, Eqv, Imp, Is y Not. A excepción de los dos últimos todos poseen la siguiente sintaxis:

<expresión1> **operador** <expresión2>

Por ejemplo si quisiéramos traer la información de los detalles de ventas cuyo montoparcial este entre 100 y 300 soles haríamos la siguiente consulta:

```
SELECT *
FROM DetalleVenta
WHERE CostoParcial> 100 AND CostoParcial<300
```

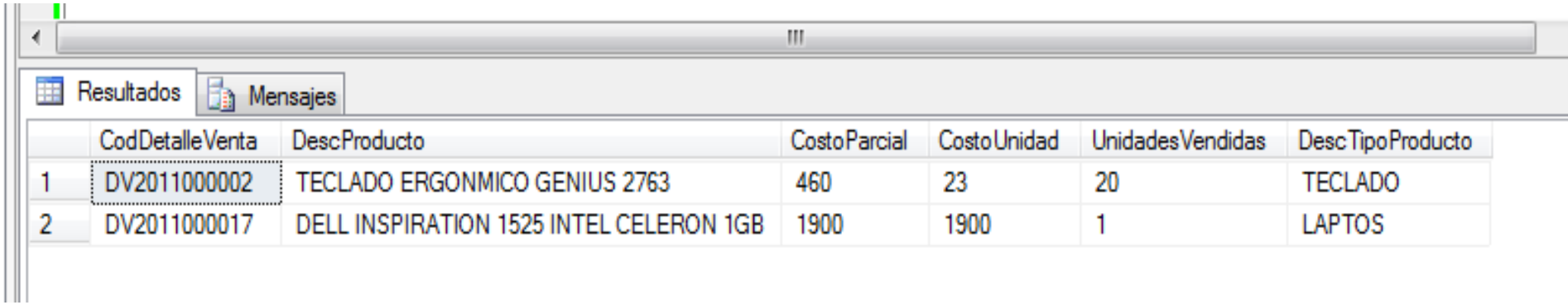
Resultados | Mensajes

	CodDetalleVenta	CodVenta	CodProducto	UnidadesVendidas	CostoUnidad	CostoParcial	Indicador
1	DV2011000003	VT2011000006	PD2011000003	12	20	240	N
2	DV2011000010	VT2011000009	PD2011000011	2	146	292	S
3	DV2011000019	VT2011000011	PD2011000011	2	146	292	S

Si quisiéramos saber que saber que costo parcial del detalle de venta supera los 200 y cuyo tipo de productos es teclado o Laptos haríamos la siguiente consulta:

```
SELECT DetalleVenta.CodDetalleVenta,Producto.DescProducto
       ,DetalleVenta.CostoParcial,DetalleVenta.CostoUnidad,
       DetalleVenta.UnidadesVendidas,TipoProducto.DescTipoProducto
FROM   DetalleVenta,Producto,TipoProducto
Where DetalleVenta.CodProducto=Producto.CodProducto and
      Producto.CodTipoProducto=TipoProducto.CodTipoProducto
      and DetalleVenta.CostoParcial>200
      and (TipoProducto.DescTipoProducto='Teclado' or
           TipoProducto.DescTipoProducto='Laptos')
```

Esta consulta traería la siguiente información de la Base de Datos os



Resultados | Mensajes

	CodDetalleVenta	DescProducto	CostoParcial	CostoUnidad	UnidadesVendidas	DescTipoProducto
1	DV2011000002	TECLADO ERGONMICO GENIUS 2763	460	23	20	TECLADO
2	DV2011000017	DELL INSPIRATION 1525 INTEL CELERON 1GB	1900	1900	1	LAPTOS

Operador like

Se utiliza para comparar una expresión de cadena con un modelo en una expresión SQL. Su sintaxis es:

expresión **Like** modelo

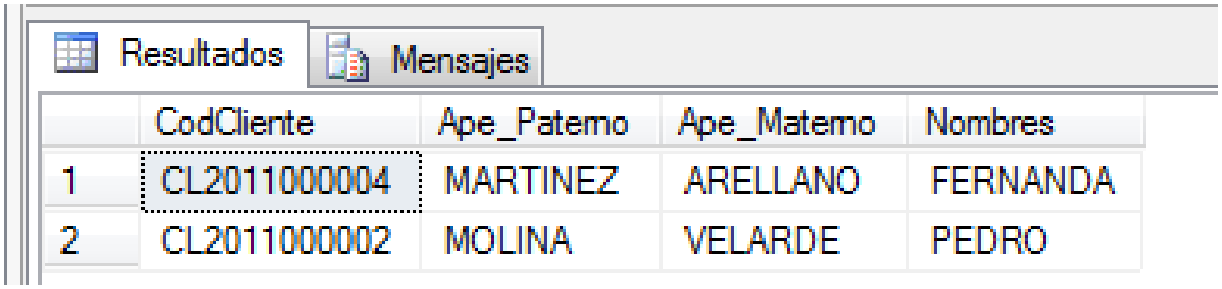
En donde expresión es una cadena modelo o campo contra el que se compara la expresión. Se puede utilizar el operador Like para encontrar valores en los campos que coincidan con el modelo especificado. Algunos ejemplos de su uso:

EJEMPLO	DESCRIPCION
LIKE 'A%'	Todo lo que comience por A
LIKE '_NG'	Todo lo que comience por cualquier carácter y luego siga NG
LIKE '[AF]%'	Todo lo que comience por A ó F
LIKE '[A-F]%'	Todo lo que comience por cualquier letra comprendida entre la A y la F
LIKE '[A^B]%'	Todo lo que comience por A y la segunda letra no sea una B

Si quiera saber cuales son las personas cuyo apellido paterno empieza con las letras S haríamos la siguiente consulta:

```
select CodCliente,Ape_Paterno,Ape_Materno,Nombres
from cliente
where Indicador='S'  and Ape_Paterno like 'M%'
order by Ape_Paterno
```

Esta consulta traería la siguiente información de la Base de Datos



	CodCliente	Ape_Paterno	Ape_Materno	Nombres
1	CL2011000004	MARTINEZ	ARELLANO	FERNANDA
2	CL2011000002	MOLINA	VELARDE	PEDRO

Utilizando el Insert

La sentencia insert permite insertar datos a una tabla se pueden insertar a todas las tablas o simplemente se puede insertan por medio de unos campos especificos veamos el formato y el ejemplo:

INSERT INTO nombre de la tabla(Columnas de la Tabla)
VALUES(Datos para columnas separados por comas)

Ejemplos:

Insertando todos los datos en la tabla

INSERT INTO Cliente **VALUES**('CL2011000020','Sandra', 'Andrade', 'Campos','3215425125 2', 'villa Sol 458','5374214', 'S')

Insertando solo en dos campos de la tabla

INSERT INTO Cliente (Codcliente, Ape_Paterno)
VALUES('CL2011000050','Gonzalez')

Utilizando el Delete

La sentencia delete permite eliminar los registro de una tabla y su formato es el siguiente:

DELETE FROM nombre de la tabla **WHERE** (Condicion)

Ejemplos:

Borrando todos los clientes cuyo apellido paterno empieza con la letra A

DELETE FROM Cliente **WHERE** (Ape_Paterno like ‘A%’)

Utilizando el Update

La sentencia Update permite actualizar los registros de una tabla a a traves de una condición veamos el formato y el ejemplo:

```
UPDATE NombreTabla SET Columna1=Valor, Columna2=valor ……
WHERE (Condicion)
```

Ejemplos:

Actualizando registros de la tabla Clientes y colocando en N el indicador

```
UPDATE Cliente SET Indicador='N'
WHERE (Ape_Paterno like 'Es%')
```

Actualizando un registro de la tabla producto

```
UPDATE producto SET DescProducto='Hp 5215 CoreI5 ' ,Precio=1600,
                    Stock=50, CodTipoProducto='TP2011000008'
WHERE CodProducto = 'PD2011000020'
```

Universidad César Vallejo

Licenciada por Sunedu
para que puedas salir adelante