

Insertar Datos en Tablas

La instrucción básica para insertar valores a los atributos (columnas) de una tabla es la instrucción **INSERT INTO**

Insertar una sola tupla

Para insertar una tupla en la tabla, se utiliza la siguiente sintaxis

```
INSERT INTO nombre_tabla VALUES (valor_1, 'valor_2', 'aaaa-mm-dd')
```

En este caso se insertará en una tabla de tres columnas cuyo primer valor podría ser un entero, el segundo una cadena de caracteres y el tercero un tipo fecha. Para esto hay que recordar el orden de los datos, es decir, se debe saber que el segundo es una cadena de caracteres y el tercero una fecha.

Insertar según nombre de columna

Para evitar esto, es conveniente usar una sintaxis que permita insertar no en el orden de las columnas, sino de acuerdo al nombre de la columna.

```
INSERT INTO nombre_tabla (columna_1, columna_2, columna_3) VALUES (valor_1,  
'valor_2', 'aaaa-mm-dd')
```

De esta forma a columna_1 se le asigna el valor_1, a columna_2 se le asigna 'valor_2' y a columna_3 se le asigna 'aaaa-mm-dd', se deben tener tantos valores como columnas en la expresión.

Insertar varias tuplas

Es posible insertar varias tuplas a la vez en una tabla, se deben encerrar entre paréntesis y separar por comas, esto aplica para cualquiera de las dos opciones anteriores

```
INSERT INTO nombre_tabla (columna_1, columna_2) VALUES  
(valor_1, 'valor_1_c'),  
(valor_2, 'valor_2_c'),  
(valor_3, 'valor_3_c')
```

Esto insertará tres tuplas en la tabla

Seleccionar Datos en Tablas

La instrucción básica para la selección de datos es

```
SELECT * FROM nombre_tabla
```

Esto mostrará todas las columnas y todas las tuplas de una tabla .

Se pueden limitar la cantidad de columnas a desplegar, esto se realiza con la instrucción .

```
SELECT columna_1,columna_2, columna_8 FROM nombre_tabla
```

Esto solamente mostrará los valores de las columnas columna_1, columna_2 y columna_8 de todas las tuplas .

Mostrar solamente los valores únicos

Si solo se quieren desplegar los resultados únicos de una tabla, sin desplegar los repetidos , se puede usar la sentencia:

```
SELECT DISTINCT columna_1 FROM nombre_tabla
```

Se pueden seleccionar las tuplas que cumplan con una cierta propiedad, esto es con la instrucción .

```
SELECT * FROM nombre_tabla WHERE condición
```

Que mostrará solamente las tuplas que cumplan con la condición especificada.

Seleccionar valores máximos o mínimos

```
SELECT MAX (columna) FROM nombre_tabla  
SELECT MIN (columna) FROM nombre_tabla
```

Desplegará el valor Máximo y Mínimo de una tabla de acuerdo a los valores de una cierta columna .

Renombrando columnas

Para cambiar el nombre de una columna cuando se muestran los datos en una consulta, se utiliza la siguiente sentencia:

```
SELECT columna_1 AS 'Nombre a desplegar'
```

En este caso en lugar de mostrar el nombre de *columna_1* se desplegará el valor colocado en 'Nombre a desplegar', si el nombre a desplegar no lleva espacios, no es necesario usar apóstrofes . Esta instrucción no cambia el nombre de la columna en la tabla.

Límite en las consultas

Es posible especificar que solo se quieren desplegar una cierta cantidad de resultados .

```
SELECT * FROM nombre_tabla LIMIT valor
```

En donde valor es un número entero, esto desplegará solamente las valor tuplas de la tabla .

Es posible especificar un rango de tuplas a mostrar .

```
SELECT * FROM usuarios LIMIT A,B
```

En donde A y B son números enteros, esto despliega de la tupla A hasta la tupla B, considerar que si se quiere el primer elemento el valor de A debe ser 1 .

Ordenando resultados

Cuando se selecciona de una tabla, los resultados se ordenan de acuerdo a cómo fueron insertados, se puede especificar un cierto orden ascendente o descendente dependiendo de una o varias columnas, si no se especifica nada se toma por defecto **ASC**

```
SELECT * FROM nombre_tabla ORDER BY columna_1 DESC;  
SELECT * FROM nombre_tabla ORDER BY columna_1 ASC;
```

Es posible especificar varios criterios de ordenamiento .

```
SELECT * FROM nombre_tabla ORDER BY columna_1, columna_2 DESC
```

En este caso, primero se ordenan los resultados por columna_1 de manera ascendente, posteriormente aquellos cuyo valor sea el mismo en columna_1, se ordenan por su valor en columna_2 de manera descendente .

Total de elementos

Para obtener el total de resultados se utiliza la sentencia :

```
SELECT COUNT(*) FROM nombre_tabla
```

Actualizar Datos en Tablas

Una vez insertado un dato en una Tabla, es posible actualizar sus valores .

Actualizar información de tuplas

Para actualizar la información de tuplas, se debe usar la instrucción UPDATE

```
UPDATE nombre_tabla SET columna = nuevo_valor
```

Cambiaría el valor del atributo a nuevo_valor para todas las tuplas .

Se pueden modificar más de una columna a la vez.

```
UPDATE nombre_tabla SET columna = nuevo_valor, columna_2 = nuevo_valor
```

Si se desea especificar una o un grupo particular de tuplas, se debe usar la sentencia **WHERE**

```
UPDATE nombre_tabla SET columna = nuevo_valor WHERE condición
```

En donde la condición puede involucrar la columna a modificar, otra o una combinación de columnas

Agregando y Concatenando

Es posible actualizar un valor numérico o uno tipo VARCHAR tomando el valor existente a través de una sentencia UPDATE

Ésta sentencia agregará la “CADENA EXTRA” al valor de un determinado atributo

```
UPDATE nombre_tabla SET atributo = CONCAT (atributo, 'CADENA EXTRA')
```

Ésta sentencia sumará un valor al contenido de un atributo (se pueden realizar otras operaciones aritméticas)

```
UPDATE nombre_tabla SET atributo = atributo + N
```

En donde N es la cantidad a agregar.

Eliminar Datos en Tablas

Para eliminar tuplas de una base de datos, se utiliza la instrucción:

```
DELETE FROM nombre_tabla
```

Esto eliminará todos los registros de la tabla indicada, no es posible especificar que se quiere eliminar el valor solamente de algunas columnas.

Para especificar que se quieren borrar solamente algunas tuplas en particular se utiliza la siguiente instrucción :

```
DELETE FROM nombre_tabla WHERE condicion
```

En donde condición es un predicado que permite elegir ciertas tuplas que cumplan con una determinada propiedad.

Condicionales en las Sentencias

Condiciones en sentencias SQL

Acompañando a las instrucciones UPDATE, SELECT o DELETE se puede usar la instrucción WHERE seguido de un predicado, aquí se explican varias de estas condiciones y su uso en las sentencias.

La sentencia WHERE tiene la siguiente estructura:

WHERE condicion_1 **AND/OR** condicion_2 **AND/OR** condicion_n

Las condiciones se basan en comparaciones, para las comparaciones se tienen los operandos básicos:

Símbolo	Significado
=	Igualdad
!=	Distinto de
>	Mayor
<	Menor
>=	Mayor que
<=	Menor que

Con esto se puede comparar si un cierto valor cumple con lo especificado

Ejemplo:

```
SELECT * FROM nombre_tabla WHERE columna_1 = 'valor_comparable'
```

```
UPDATE nombre_tabla SET columna_1 = valor WHERE columna_2 = valor_2 AND  
columna_3 = 'valor_3'
```

```
DELETE FROM nombre_tabla WHERE columna_1 > valor
```

El operador LIKE

Cuando se comparan cadenas, los operadores anteriores funcionan bien cuando se conoce el valor a comparar completo, si se quieren seleccionar aquellos que cumplan con un cierto patrón se tiene el operador **LIKE**, para especificar al patrón se dispone del carácter %

Ejemplo:

```
SELECT * FROM nombre_tabla WHERE columna_1 LIKE '%abc' AND ...
```

Aquí se seleccionarán todos los valores de las tuplas que finalicen con el patrón 'abc', el operador % funciona como el operador * en un sistema Unix, de esta forma:

```
SELECT * FROM nombre_tabla WHERE columna_1 LIKE 'abc%'
SELECT * FROM nombre_tabla WHERE columna_1 LIKE '%abc%'
```

La primera sentencia selecciona aquellos que comienzan con el patrón 'abc'.

La segunda sentencia selecciona aquellos que contienen el patrón 'abc'.

El operador NOT IN

Una opción para sustituir el operador de diferencia es el operador NOT IN

```
SELECT * FROM nombre_tabla WHERE columna_1 NOT IN (valor_1) AND ...
SELECT * FROM nombre_tabla WHERE columna_1 NOT IN (valor_1, valor_2,
valor_n) OR ...
SELECT * FROM nombre_tabla WHERE columna_1 NOT IN ("valor_1", "valor_n")
AND ...
```