

Manipulación de Datos (INSERT, DELETE y UPDATE)

Inserción de datos. Orden INSERT

La orden INSERT sirve para añadir filas de datos en una tabla. El formato de esta orden es el siguiente:

```
INSERT INTO NombreTabla [(columna [, columna] ...)]  
VALUES (valor [, valor] ...);
```

- *NombreTabla* es la tabla en la que se van a insertar las filas
- *[columna [, columna] ...]* representa la columna o columnas donde se van a introducir valores. Si las columnas no se especifican en la cláusula INSERT, se consideran, por defecto, todas las columnas de la tabla.
- *(valor [, valor] ...)* representa los valores que se van a dar a las columnas. Éstos se deben corresponder con cada una de las columnas que aparecen; además, deben coincidir con el tipo de dato definido para cada columna. Cualquier columna que no se encuentre en la lista de columnas recibirá el valor NULL, siempre y cuando no esté definida como NOT NULL, en cuyo caso INSERT fallará. Si no se da la lista de columnas se han de introducir los valores en todas las columnas

Inserción con SELECT

Si ese añade a INSERT una consulta, es decir, una sentencia SELECT, se añaden tantas filas como devuelva la consulta. El formato de INSERT con SELECT es el siguiente:

```
INSERT INTO NombreTabla [(columna [, columna] ...)]  
  SELECT {columna [, columna] ... | *}  
  FROM NombreTabla2 [CLÁUSULA DE SELECT];
```

Modificación. Orden UPDATE

Para actualizar los valores de las columnas de una o varias filas de una tabla utilizamos la orden UPDATE, cuyo formato es el siguiente:

```
UPDATE NombreTabla  
SET columna1 = valor1, ..., columnan=valorn  
WHERE condición;
```

- *NombreTabla* es la tabla cuyas columnas se van a actualizar
- *SET* indica las columnas que se van a actualizar y sus valores
- *WHERE* selecciona las filas que se van a actualizar. Si se omite, la actualización afectará a todas las filas de la tabla.

UPDATE con SELECT

Podemos incluir una subconsulta en una sentencia UPDATE que puede estar contenida en la cláusula WHERE o puede formar parte de SET. Cuando la subconsulta (orden SELECT) forma parte de SET, debe seleccionar una única fila y el mismo número de columnas (con tipos de datos adecuados) que las que hay entre paréntesis al lado de SET. Los formatos son:

1er Formato:

```
UPDATE NombreTabla
SET columna1 = valor, columna2 = valor2, ...
WHERE columna3 = (SELECT ...);
```

2º Formato:

```
UPDATE NombreTabla
SET (columna1, columna2, ...) = (SELECT col1, col2, ...)
WHERE condición;
```

3er Formato:

```
UPDATE NombreTabla
SET columna1 = (SELECT col1 ...), columna2 = (SELECT col2 ...)
WHERE condición;
```

Borrado de filas. Orden DELETE

Para eliminar una fila o varias filas de una tabla se usa la orden DELETE. La cláusula WHERE es esencial para eliminar sólo aquellas filas deseadas. Sin la cláusula WHERE, DELETE borrará todas las filas de la tabla. El espacio usado por las filas que han sido borradas no se reutiliza, a menos que se realice un EXPORT o un IMPORT. La condición puede incluir una subconsulta. Éste es su formato

```
DELETE [FROM] NombreTabla
WHERE Condición;
```

ROLLBACK, COMMIT y AUTOCOMMIT

Si se comete algún error, como, por ejemplo, borrar sin querer todas las filas de una tabla, es posible solucionarlo dando marcha atrás al trabajo mediante la **orden ROLLBACK**, siempre y cuando no hayamos validado todos los cambios en la base de datos mediante la **orden COMMIT**.

Cuando hacemos transacciones sobre la base de datos (insertar, actualizar o eliminar datos), los cambios no se aplicarán hasta que no hagamos un COMMIT. Es decir que si se hemos estado realizando transacciones y no hemos hecho ningún COMMIT y de pronto se va la luz, todo el trabajo se habrá perdido y nuestras tablas estarán en la situación de partida.

ROLLBACK

La orden ROLLBACK aborta la transacción volviendo a la situación de las tablas de la base de datos desde el último COMMIT.

```
ROLLBACK;  
Rollback terminado.
```

COMMIT

La orden COMMIT sirve para validar los cambios que se hagan en la base de datos

```
COMMIT;  
Validación terminada.
```

Es posible validar automáticamente las transacciones sin tener que indicarlo de forma explícita con el parámetro **AUTOCOMMIT**. El valor de este parámetro se puede mostrar con la orden **SHOW**, de la siguiente manera:

```
SHOW AUTOCOMMIT;  
autocommit OFF
```

OFF es el valor por defecto, de manera que las transacciones (INSERT, UPDATE y DELETE) no son definitivas hasta que no hagamos COMMIT. Para activar el parámetro AUTOCOMMIT se utiliza la **orden SET**:

```
SET AUTOCOMMIT ON;  
SHOW AUTOCOMMIT;  
autocommit IMMEDIATE
```

Ahora, cualquier INSERT, UPDATE y DELETE se validará automáticamente.

COMMIT implícito

Hay varias órdenes SQL que fuerzan a que se ejecute un COMMIT sin necesidad de indicarlo

QUIT	DISCONNECT	CREATE VIEW	ALTER
EXIT	CREATE TABLE	DROP VIEW	REVOKE
CONNECT	DROP TABLE	GRANT	AUDIT
			NOAUDIT

Usar cualquiera de estas órdenes es como usar un COMMIT.

ROLLBACK automático

Si después de haber realizado cambios en nuestras tablas se produce un fallo de sistema (por ejemplo, se va la luz) y no hemos validado el trabajo. Oracle hace un ROLLBACK automático obre cualquier trabajo no validado. Esto significa que tendremos que repetir el trabajo cuando pongamos en marcha la base de datos