

People matter, results count.

For internal use only

# Modulo SQL

Modificaciones de Datos con Insert Delete y Update



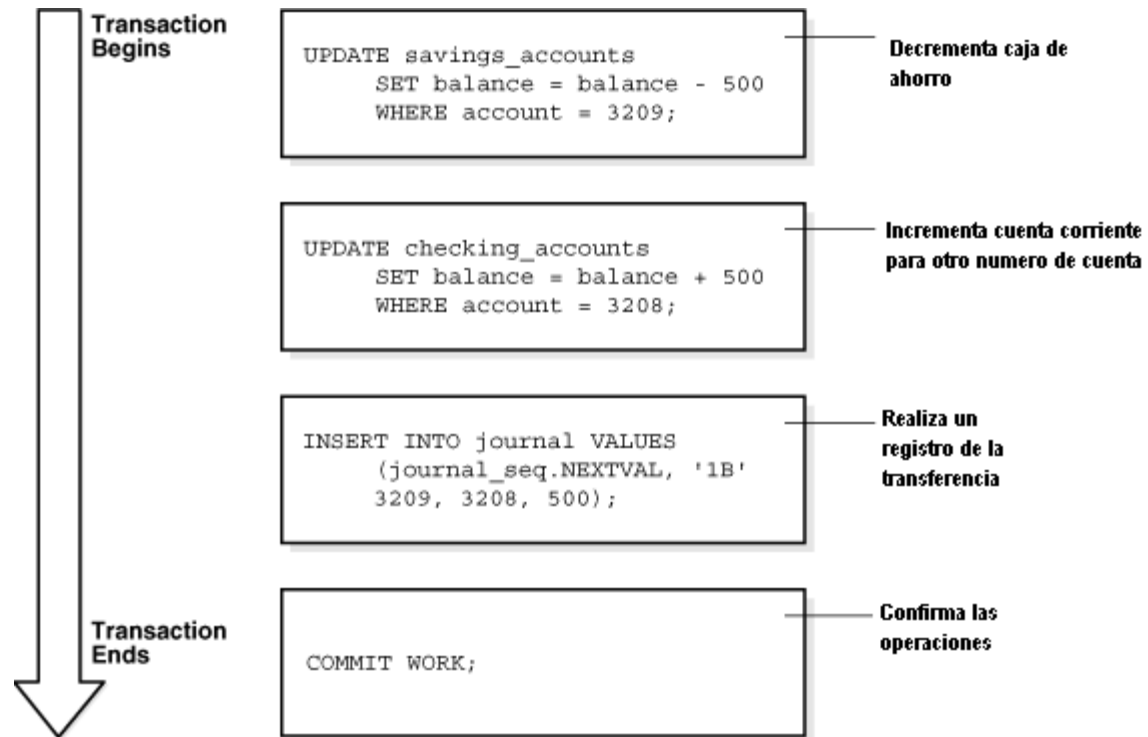


**Manipulando datos.**

# DML (Data Manipulation Language)

- Una sentencia de tipo DML se ejecuta cuando:
    - Insertamos filas nuevas en una tabla.
    - Modificamos filas existentes en una tabla.
    - Borrarnos filas de una tabla.
  - Llamamos transacción a un conjunto de sentencias de tipo DML's que conforman una unidad lógica de trabajo.
-

# Ejemplo de una transacción bancaria



# INSERT

## ■ Sintaxis general

```
INSERT INTO  table [(column [, column...])]  
VALUES      (value [, value...]);
```

```
INSERT INTO departments(department_id,  
                        department_name, manager_id, location_id)  
VALUES (70, 'Public Relations', 100, 1700);  
1 row created.
```

- Con esta sintaxis insertamos de a una fila por vez.
- Tener en cuenta que los valores de tipo carácter van entre comillas simples.
- Es opcional listar las columnas en la cláusula INSERT, pero para este caso es necesario respetar el orden de las columnas en la tabla.

## Insertando filas con valores nulos

- **Método Implícito:** se omite la columna de la lista de columnas.

```
INSERT INTO departments (department_id,  
                          department_name )  
VALUES (30, 'Purchasing');  
1 row created.
```

- **Método Explícito:** se especifica la palabra clave NULL en la cláusula VALUES.

```
INSERT INTO departments  
VALUES (100, 'Finance', NULL, NULL);  
1 row created.
```



## Insertando filas con valores especiales.

```
INSERT INTO employees (employee_id,  
                        first_name, last_name,  
                        email, phone_number,  
                        hire_date, job_id, salary,  
                        commission_pct, manager_id,  
                        department_id)  
VALUES  
    (113,  
     'Louis', 'Popp',  
     'LPOPP', '515.124.4567',  
     SYSDATE, 'AC_ACCOUNT', 6900,  
     NULL, 205, 100);  
  
1 row created.
```

- SYSDATE es una función que retorna la fecha y hora corriente del sistema operativo del servidor en el cual reside la base de datos. Retorna un valor de tipo DATE y la función no tiene argumentos.

# Copiando filas de otra tabla.

```
INSERT INTO sales_reps(id, name, salary, commission_pct)
SELECT employee_id, last_name, salary, commission_pct
FROM employees
WHERE job_id LIKE '%REP%';

4 rows created.
```

- Utilizamos una subconsulta que nos devuelve las filas a insertar.
- No se necesita la cláusula VALUES.
- Tienen que coincidir el número de columnas en la cláusula INSERT con los de la subconsulta.
- Ejemplo creación de tabla con as select.



# UPDATE

- Sintaxis general

```
UPDATE      table
SET         column = value [, column = value, ...]
[WHERE      condition];
```

```
UPDATE employees
SET    department id = 70
WHERE  employee_id = 113;
1 row updated.
```

```
UPDATE    copy_emp
SET       department_id = 110;
22 rows updated.
```

- Puede actualizar más de una fila a la vez de acuerdo con la cantidad de filas que cumplen la condición especificada en el WHERE.
- Si se omite la cláusula WHERE se modifican todas las filas de una tabla.

## Update de dos columnas con subconsultas.

```
UPDATE    employees
SET       job_id   = (SELECT  job_id
                      FROM     employees
                      WHERE     employee_id = 205),
          salary   = (SELECT  salary
                      FROM     employees
                      WHERE     employee_id = 205)
WHERE     employee_id   = 114;
1 row updated.
```

- Modificamos job y salary para el empleado con id 114 con los valores que tiene en esas columnas el empleado con id 205.
- Utilizamos para las subconsultas la misma tabla employees.

# Modificar filas con valores obtenidos de otra tabla.

```
UPDATE copy_emp
SET    department_id = (SELECT department_id
                        FROM employees
                        WHERE employee_id = 100)
WHERE  job_id        = (SELECT job_id
                        FROM employees
                        WHERE employee_id = 200);

1 row updated.
```

- Podemos utilizar subconsultas para actualizar filas de una tabla según los valores de otra tabla.

# DELETE

- **Sintaxis general**

```
DELETE [FROM]   table
[WHERE          condition];
```

```
DELETE FROM departments
WHERE department_name = 'Finance';
1 row deleted.
```

```
DELETE FROM copy_emp;
22 rows deleted.
```

- Puede borrar más de una fila a la vez de acuerdo con la cantidad de filas que cumplen la condición especificada en el WHERE.
- Si se omite la cláusula WHERE se borrar todas las filas de la tabla.

# Borrando filas con condiciones basadas en otra tabla.

```
DELETE FROM employees
WHERE department_id =
    (SELECT department_id
     FROM departments
     WHERE department_name
           LIKE '%Public%');

1 row deleted.
```

- Podemos utilizar subconsultas para eliminar filas a partir de valores de otra tabla.

# DML and Integrity Constraints.

## ■ Qué sucede si ?

- Hago un insert en la tabla EMPLOYEES haciendo referencia a un departamento que no existe en la tabla DEPARTMENTS ?
- Modifico un registro de la tabla EMPLOYEES y le asigno un job\_id que no existe en la tabla JOBS ?
- Intento borrar un registro de la tabla DEPARTMENTS que tiene empleados asignados en la tabla EMPLOYEES ?