

3- UTILITZAR SENTÈNCIES SQL EN PL/SQL

Contenido

1. Quines sentències SQL es poden utilitzar en PL/SQL.....	2
2. SELECT: clàusula INTO	2
3. DML en PL/SQL: INSERT, UPDATE I DELETE	5
4. Cursors implícits	7
5. Utilitzar cursors implícits en sentències SQL	9

1. Quines sentències SQL es poden utilitzar en PL/SQL

Es poden utilitzar sentències:

- **DML:** INSERT, UPDATE, DELETE.
- SELECT
- Control de **TRANSACCIONS** (No en APEX online pq no permet transaccions). APEX ONLINE AUTOCCOMMIT ON → cada sentència AUTOCOMMIT. No es pot desahabilitar. => no TRANSACCIONS

No es poden utilitzar sentències:

- **DDL:** CREATE TABLE, ALTER TABLE, DROP TABLE
- **DCL:** GRANT, REVOKE

2. SELECT: clàusula INTO

Les dades que s'obtenen d'una consulta es poden guardar en variables PL/SQL amb la sentència **INTO**.

- ✓ Les consultes només poden tornar una fila
- ✓ S'ha d'indicar una variable per cada columna del SELECT, i s'assignen en ordre: primer camp a la primera variable, segon camp a la segona variable, ...

EXEMPLE

Bloc de codi anònim que busca el nom de l'empleat amb id 100 i el mostra.

```
DECLARE
    v_last_name employees.last_name%TYPE;
BEGIN
    SELECT LAST_NAME
    INTO v_last_name
    FROM EMPLOYEES
    WHERE employee_id = 100;

    DBMS_OUTPUT.PUT_LINE(v_last_name);
END;
```

King

Statement processed.

EXEMPLE

Bloc de codi anònim que obté la data de contractació i el salari de l'empleat amb id 100 i els mostra.

```
DECLARE
    v_emp_hiredate employees.hire_date%TYPE;
    v_emp_salary    employees.salary%TYPE;
BEGIN
    SELECT hire_date, salary
    INTO v_emp_hiredate, v_emp_salary
    FROM employees
    WHERE employee_id = 100;

    DBMS_OUTPUT.PUT_LINE('Hire date: ' || v_emp_hiredate);
    DBMS_OUTPUT.PUT_LINE('Salary: '   || v_emp_salary);
END;
```

```
Hire date: 06/17/1987
Salary: 24000

Statement processed.
```

Una consulta que no retorna cap fila o en retorna més d'una genera un error !

EXEMPLE

Bloc de codi anònim que consulta la data de contractació i salari d'un l'empleat que té l'id 99 (no existeix)

```
DECLARE
    v_emp_hiredate employees.hire_date%TYPE;
    v_emp_salary    employees.salary%TYPE;
BEGIN
    SELECT hire_date, salary
    INTO v_emp_hiredate, v_emp_salary
    FROM employees
    WHERE employee_id = 99;

    DBMS_OUTPUT.PUT_LINE('Hire date: ' || v_emp_hiredate);
    DBMS_OUTPUT.PUT_LINE('Salary: '   || v_emp_salary);
END;
```

```
ORA-01403: no data found
```

EXEMPLE

Bloc de codi anònim que consulta i mostra la data de contractació i salari de tots els empleats.

```
DECLARE
    v_emp_hiredate employees.hire_date%TYPE;
    v_emp_salary    employees.salary%TYPE;
BEGIN
    SELECT hire_date, salary
    INTO v_emp_hiredate, v_emp_salary
    FROM employees;

    DBMS_OUTPUT.PUT_LINE('Hire date: ' || v_emp_hiredate);
    DBMS_OUTPUT.PUT_LINE('Salary: '    || v_emp_salary);
END;
```

ORA-01422: exact fetch returns more than requested number of rows

- Es pot utilitzar una variable en una sentència SQL.
- La clàusula **WHERE** pot està formada per variables, constants, literals o expressions PL/SQL.

EXEMPLE

Bloc de codi anònim que consulta i mostra la suma dels salaris del departament que indica la variable **v_numDepartament**.

```
DECLARE
    v_salari NUMBER(10,2);
    v_numDepartament NUMBER NOT NULL := 60;
BEGIN
    SELECT sum(salary) -- Funció de grup
    INTO v_salari
    FROM employees
    WHERE department_id = v_numDepartament;

    DBMS_OUTPUT.PUT_LINE('Suma salaris del departament ' ||
                          v_numDepartament || ' : ' || v_salari);
END;
```

```
Suma salaris del departament 60 : 19200

Statement processed.
```

3.DML en PL/SQL: INSERT, UPDATE I DELETE

Es poden utilitzar les sentències DML **INSERT**, **UPDATE** i **DELETE** en PL/SQL sense restriccions.

Crearem una còpia de la taula EMPLOYEES per fer les proves, sense modificar els employees originals:

```
CREATE TABLE copy_emp AS  
SELECT *  
FROM employees;
```

EXEMPLE INSERT

Bloc de codi anònim que afegeix un nou empleat:

```
BEGIN  
  INSERT INTO copy_emp  
    (employee_id, first_name, last_name, email, hire_date, job_id, salary)  
  VALUES (99, 'Ruth', 'Cores', 'RCORES', SYSDATE, 'ADD_ASST', 4000);  
END;
```

```
1 row(s) inserted.
```

EXEMPLE UPDATE

Bloc de codi que augmenta en 800 el salari de tots els empleats que tenen com a *job_id* **ST_CLERK**:

```
DECLARE  
  v_sal_increase copy_emp.salary%TYPE := 800;  
BEGIN  
  UPDATE copy_emp  
  SET salary = salary + v_sal_increase  
  WHERE job_id = 'SA_REP';  
END;
```

```
1 row(s) updated.
```

Salaris abans de modificar		Salaris modificats	
LAST_NAME	SALARY	LAST_NAME	SALARY
Abel	11000	Abel	11800
Taylor	8600	Taylor	9400
Grant	7000	Grant	7800

EXEMPLE DELETE

Bloc de codi que elimina els empleats del departament 110

```
DECLARE
    v_deptno copy_emp.department_id%TYPE := 110;
BEGIN
    DELETE FROM copy_emp
    WHERE department_id = v_deptno;
END;
```

```
1 row(s) deleted.
```

Encara que mostri el missatge que elimina 1 fila, en realitat n'ha eliminat dues.

Inicialment té 2 treballadors. Després d'executar el DELETE:

```
no data found
```

4.Cursors implícits

Quan Oracle executa una sentència crea un **cursor implícit** (*implicit cursor*) per guardar informació sobre el resultat de l'execució de la sentència.

El cursor implícit té **3 propietats** que es poden consultar mitjançant variables predefinides de PL/SQL, anomenades ***implicit cursor attributes***.

Attribute	Description
SQL%FOUND	Boolean attribute that evaluates to TRUE if the most recent SQL statement returned at least one row.
SQL%NOTFOUND	Boolean attribute that evaluates to TRUE if the most recent SQL statement did not return even one row.
SQL%ROWCOUNT	An integer value that represents the number of rows affected by the most recent SQL statement.

EXEMPLE

Bloc de codi que elimina els empleats que tenen un id de departament determinat i mostra quants empleats han estat eliminats.

```
DECLARE
    v_deptno copy_emp.department_id%TYPE := 90;
BEGIN
    DELETE FROM copy_emp
    WHERE department_id = v_deptno;
    DBMS_OUTPUT.PUT_LINE(SQL%ROWCOUNT || ' files eliminades.');
```

END;

3 files eliminades.

1 row(s) deleted.

Elimina 3 files però mostra el missatge com si només n'eliminés una !!

EXAMPLE

Modifica el salari dels empleats que tenen job_id IT_PROG i mostra el nombre d'empleats modificats.

```
DECLARE
    v_salary_increase copy_emp.salary%TYPE := 800;
BEGIN
    UPDATE copy_emp
    SET     salary = salary + v_salary_increase
    WHERE  job_id = 'IT_PROG';
    DBMS_OUTPUT.PUT_LINE(SQL%ROWCOUNT || ' rows updated.');
```

END;

```
3 rows updated.

1 row(s) updated.
```


5.Utilitzar cursors implícits en sentències SQL

EXEMPLE

Bloc de codi que actualitza el salari dels empleats i guarda en la taula **RESULTATS** el número d'empleats actualitzats i la data d'actualització.

CREA LA TAULA RESULTATS

Crea la taula per guardar el nombre de files actualitzades i la data d'actualització.

```
CREATE TABLE RESULTATS (  
    num_files NUMBER(4),  
    data_actualitzacio DATE  
);
```

PRIMERA VERSIÓ: GENERA UN ERROR

```
-- Bloc de codi anònim que actualitza els empleats i  
-- afegeix un registre a la taula RESULTAT  
BEGIN  
    -- Actualitza el salari dels empleats  
    UPDATE employees  
    SET salary = salary + 100  
    WHERE job_id = 'SA_REP';  
  
    -- Afegeix el nombre d'empleats actualitzats i la data a la taula results  
    INSERT INTO RESULTATS (NUM_FILES, DATA_ACTUALITZACIO)  
    VALUES (SQL%ROWCOUNT, SYSDATE);  
END;
```

Genera un **error**. No es pot utilitzar directament el valor d'SQL%ROWCOUNT en una sentència SQL. S'ha de guardar en una variable i utilitzar la variable:

```
-- Bloc de codi anònim que actualitza els empleats i
-- afegeix un registre a la taula RESULTAT
DECLARE
    v_num_files NUMBER;
BEGIN
    -- Actualitza el salari dels empleats
    UPDATE employees
    SET salary = salary + 100
    WHERE job_id = 'SA_REP';

    -- Guarda la variable el valor de ROW%COUNT
    v_num_files := SQL%ROWCOUNT;

    -- Afegeix el nombre d'empleats actualitzats i la data la taula results
    INSERT INTO RESULTATS (NUM_FILES, DATA_ACTUALITZACIO)
    VALUES (v_num_files, SYSDATE);
END;
```

NUM_FILES	DATA_ACTUALITZACIO
3	03/14/2022