



# Uso de Cursores Explícitos Simples

**MDY3131**

# Experiencia de Aprendizaje y Competencia Asociada



ESCUELA DE  
INFORMÁTICA Y  
TELECOMUNICACIONES

Experiencia	Nombre	Unidad de Competencia Especialidad – Nivel de la Competencia de Empleabilidad
Nº 2	Construyendo Bloques Anónimos PL/SQL Complejos	Desarrolla operaciones sobre la base de datos para efectuar procesamiento de datos utilizando el lenguaje asociado de acuerdo a los requerimientos de información.
		Resolución de Problemas (N1)

# Objetivos de la Clase

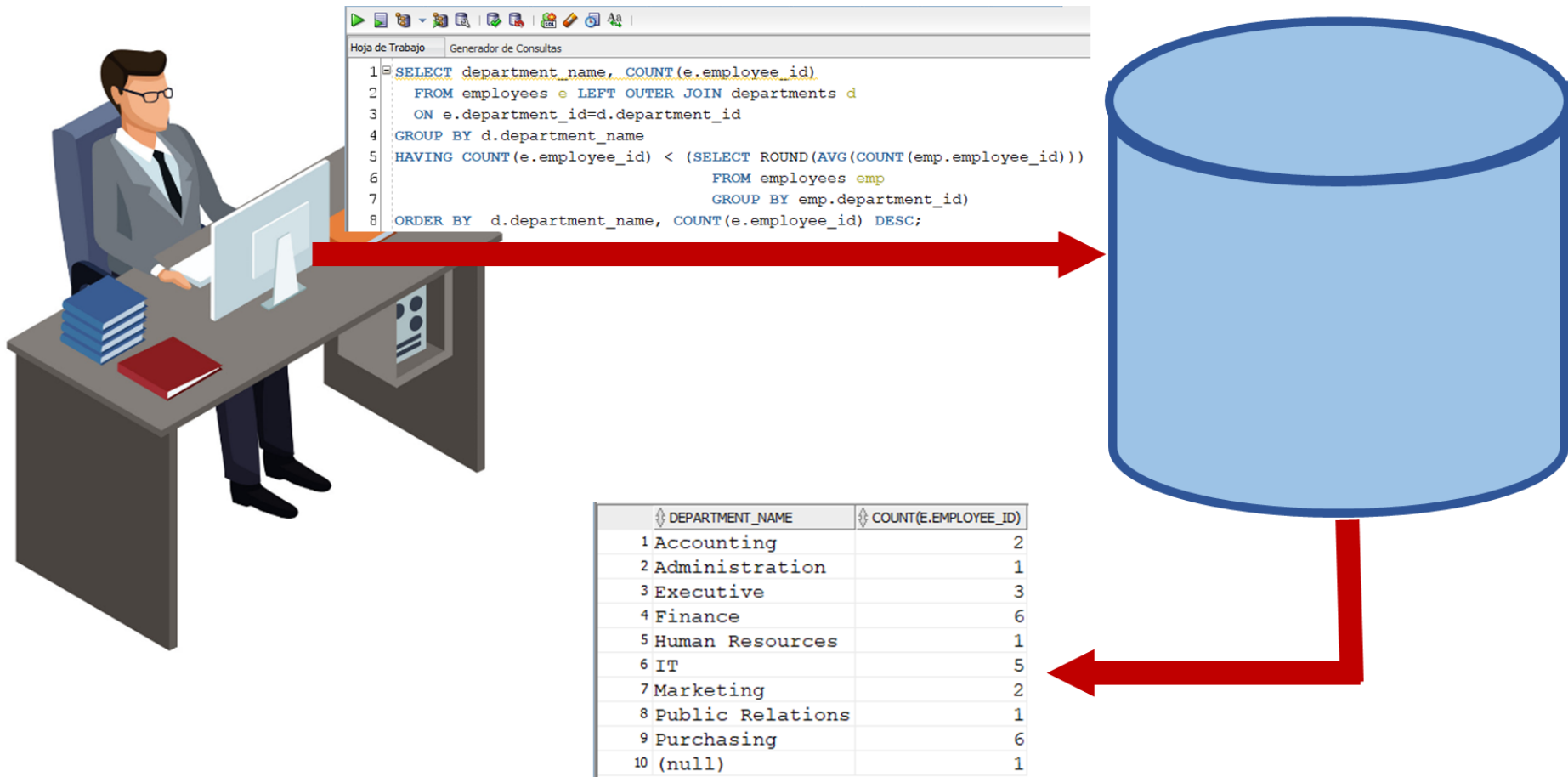
- Características de los Cursores Explícitos.
- Cómo se manejan los Cursores Explícitos.
- Cómo se Declara un Cursor Explícito y las acciones que se realizan en el Servidor de Base de Datos.
- Cómo se Abre un Cursor Explícito y las acciones que se realizan en el Servidor de Base de Datos.
- Cómo se Leen las filas del Cursor Explícito y las acciones que se realizan en el Servidor de Base de Datos.
- Cómo se Cierra un Cursor Explícito y las acciones que se realizan en el Servidor de Base de Datos.



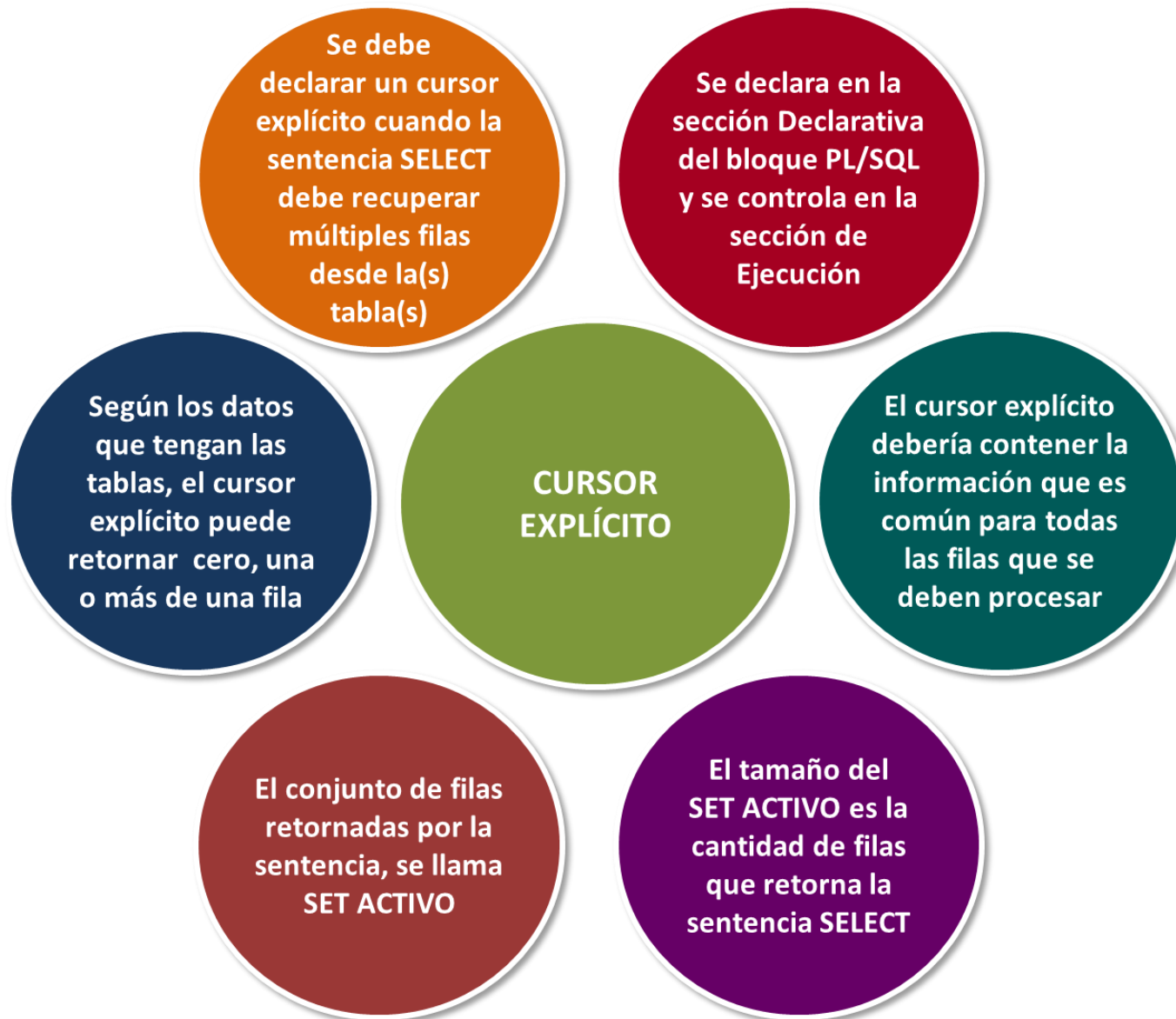
# Cursores Explícitos en Bloques PL/SQL

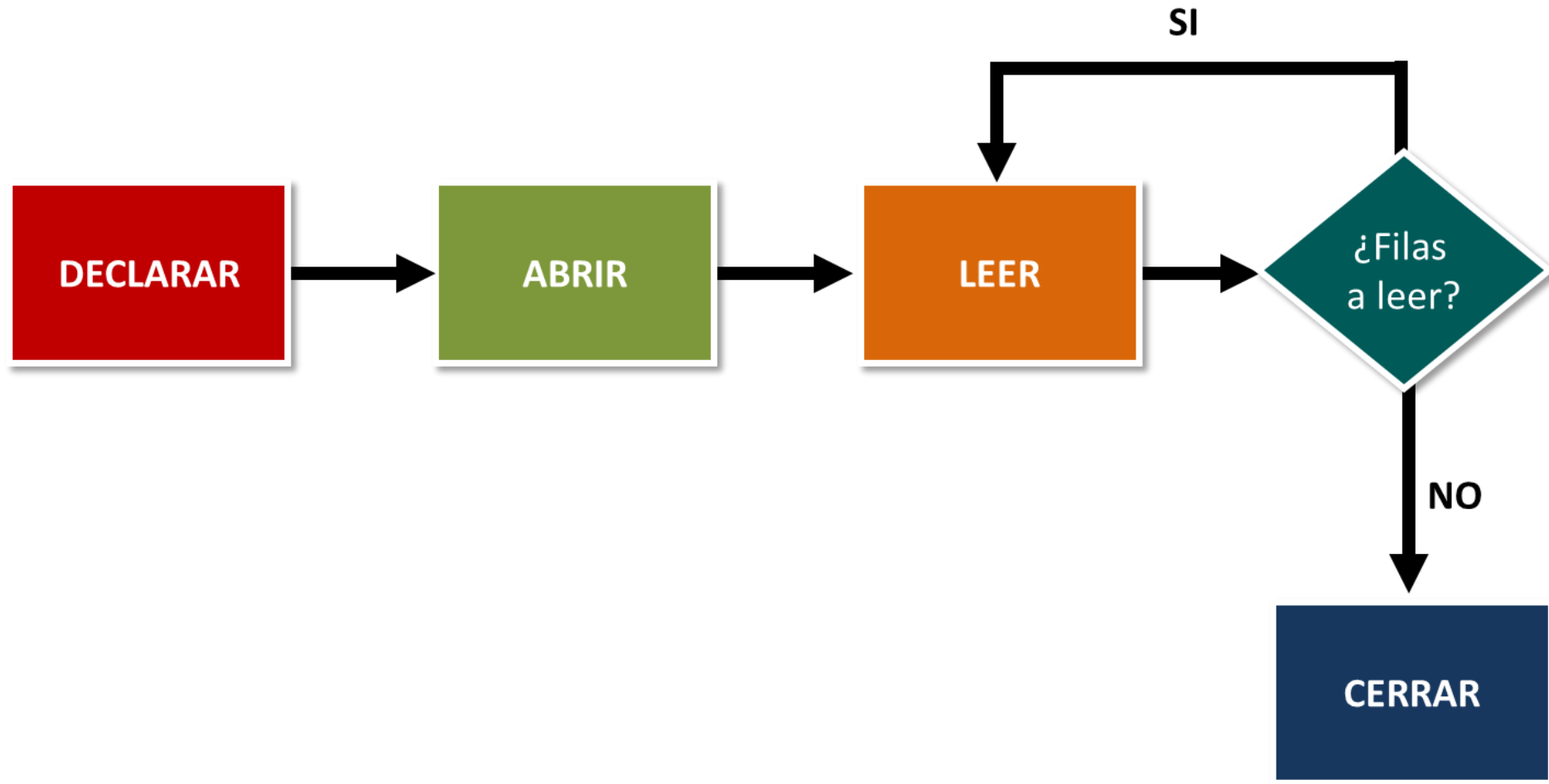
\* Los bloques PL/SQL de los ejemplos usan tablas del esquema HR de la Base de Datos Oracle.

# Características Generales







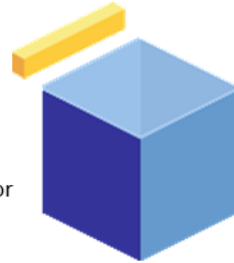


# Control de Cursores Explícitos

1

## CURSOR

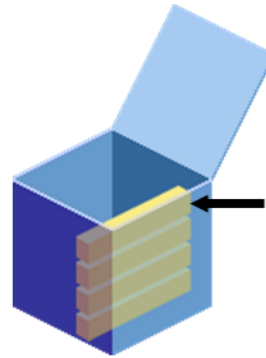
- Asigna nombre al cursor
- Define la sentencia SELECT del cursor



2

## OPEN

- Ejecuta la consulta del cursor
- Identifica el set activo
- Posiciona el cursor en la primera fila

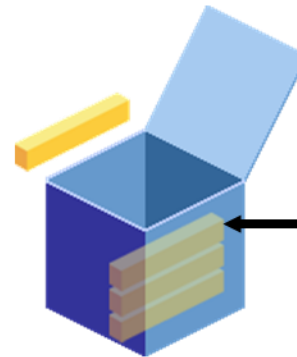


Puntero del Cursor

3

## FETCH

- Recupera la fila actual
- Avanza el cursor a la siguiente fila



Puntero del cursor

4

## CLOSE

- Libera el cursor



Puntero del  
cursor



Se declara en la sección Declarativa del bloque PL/SQL. Como buena práctica, el nombre debería comenzar con CUR\_ o C\_

La sentencia SELECT del cursor no contiene la cláusula INTO porque sólo se está definiendo el cursor

El cursor puede contener todos los elementos de un sentencia SELECT, además de variables PL/SQL y BIND

El SET ACTIVO lo determina la sentencia SELECT del cursor

- Sintaxis:

```
CURSOR nombre_cursor IS  
    sentencia_select;
```

- Ejemplo:

```
DECLARE  
CURSOR cur_dato_empleado IS  
    SELECT employee_id, e.first_name || ' ' || e.last_name, e.salary  
    FROM employees e;
```

- Ejemplo:

```
DECLARE  
CURSOR cur_empleados IS  
  SELECT e.employee_id,  
         e.first_name || ' ' ||  
         e.last_name, j.job_title, l.city, l.street_address  
FROM employees e JOIN jobs j  
      ON(e.job_id = j.job_id)  
LEFT OUTER JOIN departments d  
      ON(e.department_id = d.department_id)  
LEFT OUTER JOIN locations l  
      ON(d.location_id = l.location_id)  
ORDER BY e.employee_id;
```

- Ejemplo:

```
VAR b_salario NUMBER
EXEC :b_salario:=10000
DECLARE
CURSOR cur_empleados IS
    SELECT e.employee_id,e.first_name || ' ' || e.last_name,
           j.job_title, l.city, l.street_address
    FROM employees e JOIN jobs j
        ON(e.job_id = j.job_id)
    LEFT OUTER JOIN departments d
        ON(e.department_id = d.department_id)
    LEFT OUTER JOIN locations l
        ON(d.location_id = l.location_id)
    WHERE e.job_id IN (SELECT job_id
                       FROM employees
                       GROUP BY job_id
                       HAVING ROUND(AVG(salary)) > :b_salario)
    ORDER BY e.employee_id;
```

El cursor se abre en la sección de Ejecución del Bloque PL/QL

Dinámicamente se asigna memoria para el cursor

Parseo de la sentencia SELECT, se examinan los valores de las variables y se identifica el Set Activo

Posiciona el puntero en la primera fila del Set Activo

- Sintaxis:

```
OPEN nombre_cursor;
```

- Ejemplo:

```
DECLARE
CURSOR cur_dato_empleado IS
  SELECT employee_id, e.first_name || ' ' || e.last_name, e.salary
  FROM employees e;
BEGIN
  OPEN cur_dato_empleado;
```

## SET ACTIVO

	EMPLOYEE_ID	E.FIRST_NAME  ' ' E.LAST_NAME	SALARY
1	100	Steven King	24000
2	101	Neena Kochhar	17000
3	102	Lex De Haan	17000
4	103	Alexander Hunold	9000
5	104	Bruce Ernst	6000
.....			
	104	203 Susan Mavris	6500
	105	204 Hermann Baer	10000
	106	205 Shelley Higgins	12008
	107	206 William Gietz	8300

- Ejemplo:

```
DECLARE  
CURSOR cur_empleados IS  
    SELECT e.employee_id,  
           e.first_name || ' ' ||  
           e.last_name, j.job_title, l.city, l.street_address  
FROM employees e JOIN jobs j  
    ON(e.job_id = j.job_id)  
    LEFT OUTER JOIN departments d  
    ON(e.department_id = d.department_id)  
    LEFT OUTER JOIN locations l  
    ON(d.location_id = l.location_id)  
    ORDER BY e.employee_id;  
BEGIN  
OPEN cur_empleados;
```

**SET ACTIVO**

	EMPLOYEE_ID	E.FIRST_NAME  ' ' E.LAST_NAME	JOB_TITLE	CITY	STREET_ADDRESS
1	100	Steven King	President	Seattle	2004 Charade Rd
2	101	Neena Kochhar	Administration Vice President	Seattle	2004 Charade Rd
3	102	Lex De Haan	Administration Vice President	Seattle	2004 Charade Rd
4	103	Alexander Hunold	Programmer	Southlake	2014 Jabberwocky Rd
5	104	Bruce Ernst	Programmer	Southlake	2014 Jabberwocky Rd
.....					
104	203	Susan Mavris	Human Resources Representa...	London	8204 Arthur St
105	204	Hermann Baer	Public Relations Represent...	Munich	Schwanthalerstr. 7031
106	205	Shelley Higgins	Accounting Manager	Seattle	2004 Charade Rd
107	206	William Gietz	Public Accountant	Seattle	2004 Charade Rd

- Ejemplo:

```
VAR b_salario NUMBER
EXEC :b_salario:=10000
DECLARE
CURSOR cur_empleados IS
    SELECT e.employee_id,e.first_name || ' ' || e.last_name,
           j.job_title, l.city, l.street_address
    FROM employees e JOIN jobs j
        ON(e.job_id = j.job_id)
    LEFT OUTER JOIN departments d
        ON(e.department_id = d.department_id)
    LEFT OUTER JOIN locations l
        ON(d.location_id = l.location_id)
    WHERE e.job_id IN (SELECT job_id
                      FROM employees
                      GROUP BY job_id
                      HAVING ROUND(AVG(salary)) > :b_salario)
    ORDER BY e.employee_id;
BEGIN
OPEN cur_empleados;
```

EMPLOYEE_ID	E.FIRST_NAME  E.LAST_NAME	JOB_TITLE	CITY	STREET_ADDRESS
1	100 Steven King	President	Seattle	2004 Charade Rd
2	101 Neena Kochhar	Administration Vice President	Seattle	2004 Charade Rd
3	102 Lex De Haan	Administration Vice President	Seattle	2004 Charade Rd
4	108 Nancy Greenberg	Finance Manager	Seattle	2004 Charade Rd
5	114 Den Raphaely	Purchasing Manager	Seattle	2004 Charade Rd
6	145 John Russell	Sales Manager	Oxford	Magdalen Centre, The Oxford Science Park
7	146 Karen Partners	Sales Manager	Oxford	Magdalen Centre, The Oxford Science Park
8	147 Alberto Errazuriz	Sales Manager	Oxford	Magdalen Centre, The Oxford Science Park
9	148 Gerald Cambrault	Sales Manager	Oxford	Magdalen Centre, The Oxford Science Park
10	149 Eleni Zlotkey	Sales Manager	Oxford	Magdalen Centre, The Oxford Science Park
11	201 Michael Hartstein	Marketing Manager	Toronto	147 Spadina Ave
12	205 Shelley Higgins	Accounting Manager	Seattle	2004 Charade Rd

SET ACTIVO



Se lee la fila actual del Set Activo del cursor y los datos se almacenan en las variables de la cláusula INTO

Además, avanza el puntero a la siguiente fila en el Set Activo

El atributo %NOTFOUND permite validar si se han leído todas las filas del Set Activo del cursor

Para poder leer todas las filas recuperadas en el cursor se debe generar una iteración

- Sintaxis:

```
FETCH nombre_cursor INTO lista_de_variables | registro_PL/SQL;
```

- Ejemplo:

```
DECLARE
CURSOR cur_dato_empleado IS
  SELECT employee_id, e.first_name || ' ' || e.last_name, e.salary
  FROM employees e;
TYPE tipo_reg_empleado RECORD IS
  id NUMBER(3),
  nombre VARCHAR2(30),
  salario NUMBER(8));
reg_empleado tipo_reg_empleado;
OPEN cur_dato_empleado;
FETCH cur_dato_empleado INTO reg_empleado;
END;
```

- Ejemplo:

```
DECLARE
CURSOR cur_empleados IS
    SELECT e.employee_id, e.first_name || ' ' || e.last_name,
           j.job_title, l.city, l.street_address
    FROM employees e JOIN jobs j
        ON(e.job_id = j.job_id)
    LEFT OUTER JOIN departments d
        ON(e.department_id = d.department_id)
    LEFT OUTER JOIN locations l
        ON(d.location_id = l.location_id)
    ORDER BY e.employee_id;
v_id_emp NUMBER(3);
v_nombre VARCHAR2(30);
v_trabajo VARCHAR2(30);
v_ciudad VARCHAR2(30);
v_direcc VARCHAR2(30);
BEGIN
    OPEN cur_empleados;
    FETCH cur_empleados INTO v_id_emp, v_nombre, v_trabajo, v_ciudad, v_direcc;
END;
```

- Ejemplo:

```
VAR b_salario NUMBER
EXEC :b_salario:=10000
DECLARE
CURSOR cur_empleados IS
    SELECT e.employee_id,e.first_name || ' ' || e.last_name, j.job_title, l.city, l.street_address
    FROM employees e JOIN jobs j
        ON(e.job_id = j.job_id)
    LEFT OUTER JOIN departments d
        ON(e.department_id = d.department_id)
    LEFT OUTER JOIN locations l
        ON(d.location_id = l.location_id)
    WHERE e.job_id IN (SELECT job_id
        FROM employees
        GROUP BY job_id
        HAVING ROUND(AVG(salary)) > :b_salario)
    ORDER BY e.employee_id;
v_id_emp NUMBER(3);
v_nombre VARCHAR2(30);
v_trabajo VARCHAR2(30);
v_ciudad VARCHAR2(30);
v_direcc VARCHAR2(30);
BEGIN
    OPEN cur_empleados;
    FETCH cur_empleados INTO v_id_emp, v_nombre,v_trabajo,v_ciudad,v_direcc;
END;
```

# Cerrar el Cursor

Se cierra el cursor se, libera el área de contexto que se le asignó en memoria y el set activo se elimina

Un cursor puede ser reabierto sólo si éste se ha cerrado anteriormente

- Sintaxis:

**CLOSE *nombre\_cursor*;**

- Ejemplo:

```
DECLARE
CURSOR cur_dato_empleado IS
  SELECT e.employee_id, e.first_name || ' ' || e.last_name, e.salary
  FROM employees e;
TYPE tipo_reg_empleado RECORD IS
  id NUMBER(3),
  nombre VARCHAR2(30),
  salario NUMBER(8));
reg_empleado tipo_reg_empleado;
BEGIN
  OPEN cur_dato_empleado;
  FETCH cur_dato_empleado INTO reg_empleado;
  CLOSE cur_dato_empleado;
END;
```

- Ejemplo:

```
DECLARE
CURSOR cur_empleados IS
    SELECT e.employee_id, e.first_name || ' ' || e.last_name,
           j.job_title, l.city, l.street_address
    FROM employees e JOIN jobs j
        ON(e.job_id = j.job_id)
    LEFT OUTER JOIN departments d
        ON(e.department_id = d.department_id)
    LEFT OUTER JOIN locations l
        ON(d.location_id = l.location_id)
    ORDER BY e.employee_id;
v_id_emp NUMBER(3);
v_nombre VARCHAR2(30);
v_trabajo VARCHAR2(30);
v_ciudad VARCHAR2(30);
v_direcc VARCHAR2(30);
BEGIN
    OPEN cur_empleados;
    FETCH cur_empleados INTO v_id_emp, v_nombre, v_trabajo, v_ciudad, v_direcc;
    CLOSE cur_empleados;
END;
```

- Ejemplo:

```
VAR b_salario NUMBER
EXEC :b_salario:=10000
DECLARE
CURSOR cur_empleados IS
    SELECT e.employee_id,e.first_name || ' ' || e.last_name, j.job_title, l.city, l.street_address
    FROM employees e JOIN jobs j
        ON(e.job_id = j.job_id)
    LEFT OUTER JOIN departments d
        ON(e.department_id = d.department_id)
    LEFT OUTER JOIN locations l
        ON(d.location_id = l.location_id)
    WHERE e.job_id IN (SELECT job_id
        FROM employees
        GROUP BY job_id
        HAVING ROUND(AVG(salary)) > :b_salario)
    ORDER BY e.employee_id;
v_id_emp NUMBER(3);
v_nombre VARCHAR2(30);
v_trabajo VARCHAR2(30);
v_ciudad VARCHAR2(30);
v_direcc VARCHAR2(30);
BEGIN
    OPEN cur_empleados;
    FETCH cur_empleados INTO v_id_emp, v_nombre,v_trabajo,v_ciudad,v_direcc;
    CLOSE cur_empleados;
END;
```



- Se describieron las características de los Cursores Explícitos.
- Se explicó cómo se manejan los Cursores Explícitos.
- Se explicó cómo se Declara un Cursor Explícito y las acciones que se realizan en el Servidor de Base de Datos.
- Se explicó cómo se Abre un Cursor Explícito y las acciones que se realizan en el Servidor de Base de Datos.
- Se explicó cómo se Leen las filas del Cursor Explícito y las acciones que se realizan en el Servidor de Base de Datos.
- Se explicó cómo se Cierra un Cursor Explícito y las acciones que se realizan en el Servidor de Base de Datos.