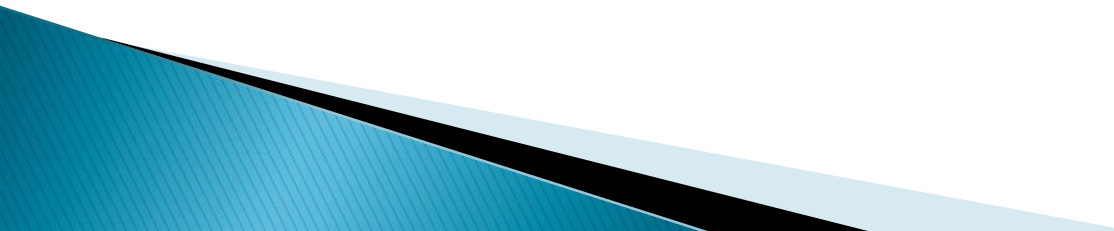


UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE COMPUTACIÓN

Tema V.  
Lenguaje Procedural  
Cursores

Ing. Lucila Patricia  
Arellano Mendoza  
2018

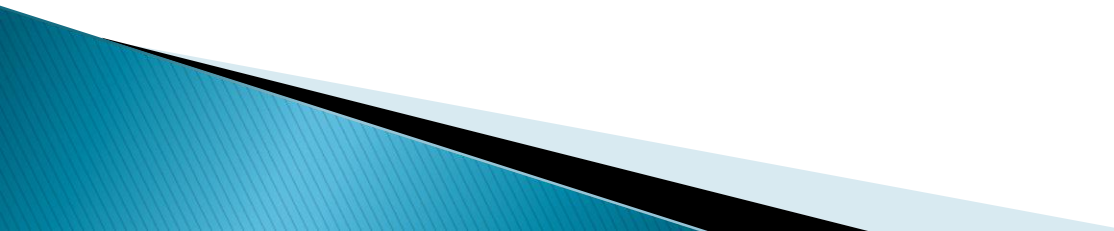


Un cursor es un área de memoria para realizar operaciones con los registros devueltos en una sentencia SELECT

Existen dos tipos:

- Implícitos
- Explícitos

Si deseamos obtener más de un valor utilizamos un cursor explícito para extraer individualmente cada fila.



Su sintaxis es la siguiente:

```
CURSOR <nombreCursor>  
IS  
  <ordenSelect>
```

Debe declararse en la zona de declaraciones

La apertura del cursor debe colocarse en la zona de instrucciones, con el siguiente formato:

```
OPEN <nombreCursor>;
```

La extracción de los datos y su almacenamiento en variables PL/SQL se realiza utilizando la siguiente sintaxis:

```
FETCH <nombreCursor> INTO {<variable> | <listaVariables>};
```

```
FETCH <nombreCursor> INTO <registroPL/SQL>;
```

Para cerrarlo utilizamos

```
CLOSE <nombreCursor>;
```

# Atributos de cursores utilizados

- ▶ Para conocer detalles de la situación del cursor tenemos 4 atributos:
- ▶ %FOUND y %NOTFOUND para controlar si la ultima orden FETCH devolvió o no una fila
- ▶ %ISOPEN para ver si el cursor esta abierto o no
- ▶ %ROWCOUNT devuelve el número de filas extraídas por el cursor.

Abra un archivo spool y realice lo siguiente

## Ejemplo 12. Uso de cursores

Genere la siguiente tabla

```
CREATE TABLE estudiante(  
  id NUMBER(5,0) PRIMARY KEY,  
  nombre VARCHAR2(20),  
  apellido VARCHAR2(20),  
  especialidad VARCHAR2(30),  
  avCreditos NUMBER(2,0)  
);
```

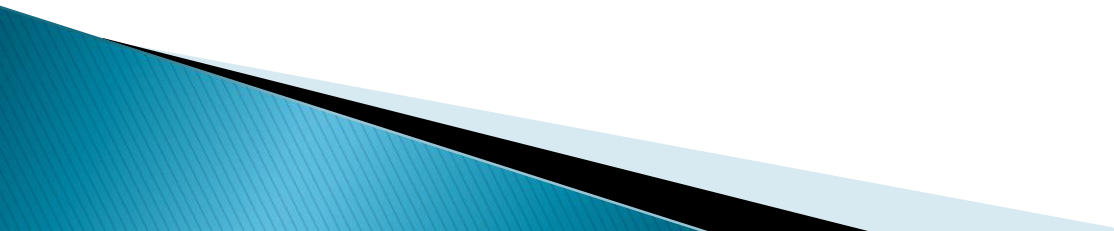
Cree un procedimiento para insertar valores a la tabla ESTUDIANTE y agregue la siguiente información

ID	NOMBRE	APELLIDO	ESPECIALIDAD	AVCREDITOS
1	Luis	Ramírez	Computación	10
2	Margarita	Mendoza	Historia	8
3	Patricia	Muñoz	Computación	20
4	Tomas	Tapia	Música	15
5	Alejandro	Pérez	Nutrición	25
6	Ester	Estrada	Nutrición	18
7	Rita	Rodríguez	Música	25
8	Rosa	Ramírez	Historia	9
9	David	Ortega	Música	22
10	Juan	Álvarez	Computación	19

El siguiente es un cursor que obtiene los nombres y apellidos de los alumnos ordenados por su especialidad.

```
DECLARE
  CURSOR estudReg          --declaramos el cursor
  IS
    SELECT *
    FROM estudiante
    ORDER BY especialidad;

  rEstud estudiante%ROWTYPE; /*variable de tipo
                              registro*/
```





BEGIN

OPEN estudReg; --abrimos el cursor

FETCH estudReg INTO rEstud;

DBMS\_OUTPUT.PUT\_LINE('Resultado : ');

/\*usamos un ciclo para capturar los registros\*/

WHILE estudReg%FOUND LOOP

DBMS\_OUTPUT.PUT\_LINE(rEstud.especialidad||' '||  
rEstud.nombre||' '||rEstud.apellido);

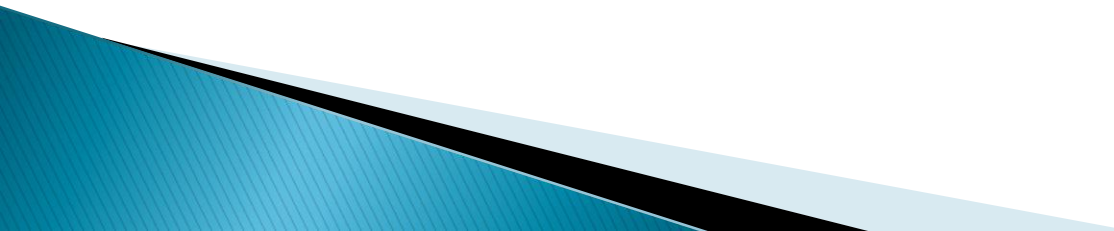
FETCH estudReg INTO rEstud;

END LOOP;

CLOSE estudReg; -- cerramos el cursor

END;

El trabajo de un cursor consiste en :

- ▶ Declarar un cursor
  - ▶ Declarar una variable que recogerá los datos del cursor
  - ▶ Abrir el cursor
  - ▶ Recuperar con FETCH, una a una, las filas extraídas introduciendo los datos en las variables, procesándolos y comprobando si se han recuperado datos o no.
- 

# Cursor FOR ...LOOP

- ▶ Para resumir todas estas tareas, tenemos la estructura cursor FOR...LOOP que hace todas estas cosas de forma implícita, todas excepto la declaración del cursor.

El formato y el uso de esta estructura es:

- ▶ Declarar la información cursor en la sección correspondiente
- ▶ Presentar el cursor utilizando el siguiente formato:

```
FOR <nombreVarReg> IN <nombreCursor> LOOP  
    . . .  
END LOOP;
```

Se declara implícitamente la variable nombreVarReg de tipo nombreCursor%ROWTYPE

Se ejecuta el primer FETCH cuyo resultado quedarán en nombreVarReg.



## Ejemplo: Ordena a los estudiantes por su nombre utilizando un cursor for.

```
DECLARE
  CURSOR estudReg
  IS
    SELECT * FROM estudiante
    ORDER BY nombre;
BEGIN
  DBMS_OUTPUT.PUT_LINE('Resultado : ');
  FOR rEstud IN estudReg LOOP
    DBMS_OUTPUT.PUT_LINE(rEstud.id||' '||
      rEstud.nombre||' '||restud.apellido);
  END LOOP;
END;
/
```

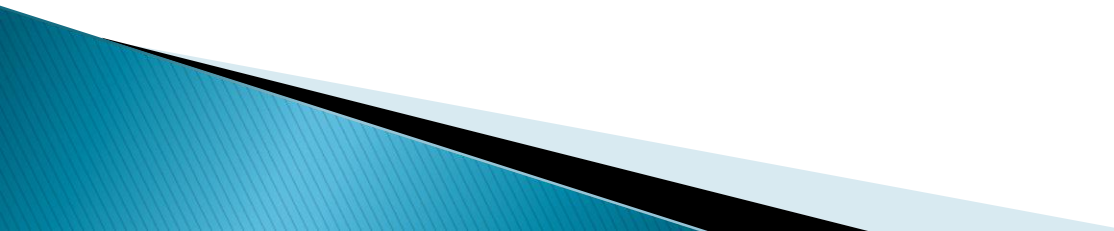
# Cursores con parámetros

Un cursor puede tener parámetros; su sintaxis es la siguiente:

```
CURSOR <nombreCursor> [(parámetros)]
```

```
IS
```

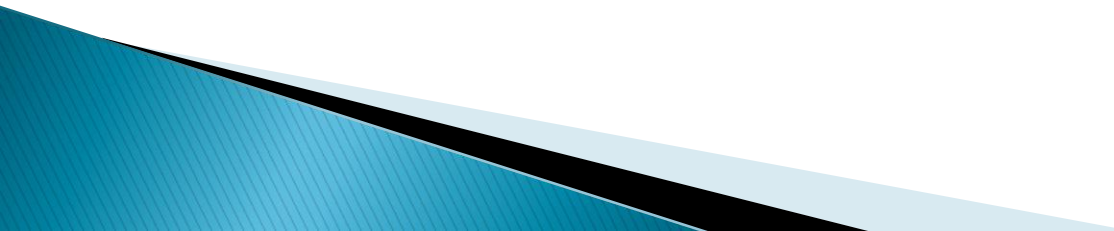
```
SELECT <sentencia select en la que  
intervendrán los parámetros>;
```



Los parámetros indicados después del nombre del cursor tienen la siguiente sintaxis:

`<nombreVarCursor> [IN] tipoDato [{:=|DEFAULT}  
valor]`

Todos los parámetros de un cursor son parámetros de entrada y su ámbito es local al cursor por eso sólo pueden ser referenciados dentro de la consulta.



# Ejemplo: Cursor con parámetros para definir un criterio de selección.

DECLARE

CURSOR espeEstud (vCreditos NUMBER:= &vCreditos)

IS

SELECT id, nombre, especialidad, avCreditos

FROM estudiante

WHERE avCreditos >= vCreditos;

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Estudiantes seleccionados ');

FOR estReg IN espeEstud LOOP

DBMS\_OUTPUT.PUT\_LINE(estReg.id||' '||

estReg.especialidad||' ' ||estReg.nombre);

END LOOP;

END;

/



Ahora vamos a crear un cursor para seleccionar los datos de los estudiantes que tienen la especialidad Historia y guardarlos en una tabla llamada temporal.

Primero creamos la tabla temporal donde se guardará la información

```
CREATE TABLE temporal(  
columnaNumerica NUMBER,  
columnaCaracter VARCHAR2(60)  
);
```

Después creamos el cursor



DECLARE

vClaveEst estudiante.id%TYPE;  
vNombre estudiante.nombre%TYPE;  
vApellido estudiante.apellido%TYPE;

--Creamos el cursor para almacenar a los estudiantes cuya  
especialidad es Historia

CURSOR cursorEstudiantesHistoria  
IS

SELECT id,nombre,apellido  
FROM estudiante  
WHERE especialidad='Historia';

BEGIN

-- Abrimos el cursor

```
OPEN cursorEstudiantesHistoria;
LOOP
    -- Recuperamos la información de los estudiantes
    FETCH cursorEstudiantesHistoria INTO vClaveEst, vNombre,
    vApellido;
    -- Salimos del bloque al terminar
    EXIT WHEN cursorEstudiantesHistoria%NOTFOUND;
    --Guardamos en una tabla temporal
    INSERT INTO temporal (columnanumerica,columnacaracter)
    VALUES(vClaveEst, vNombre || ' ' || vApellido);
END LOOP;
CLOSE cursorEstudiantesHistoria; -- Cerramos el cursor
COMMIT;
END;
/
```

Revisamos la tabla llamada temporal

```
SELECT * FROM temporal;
```

Cierre su archivo spool

