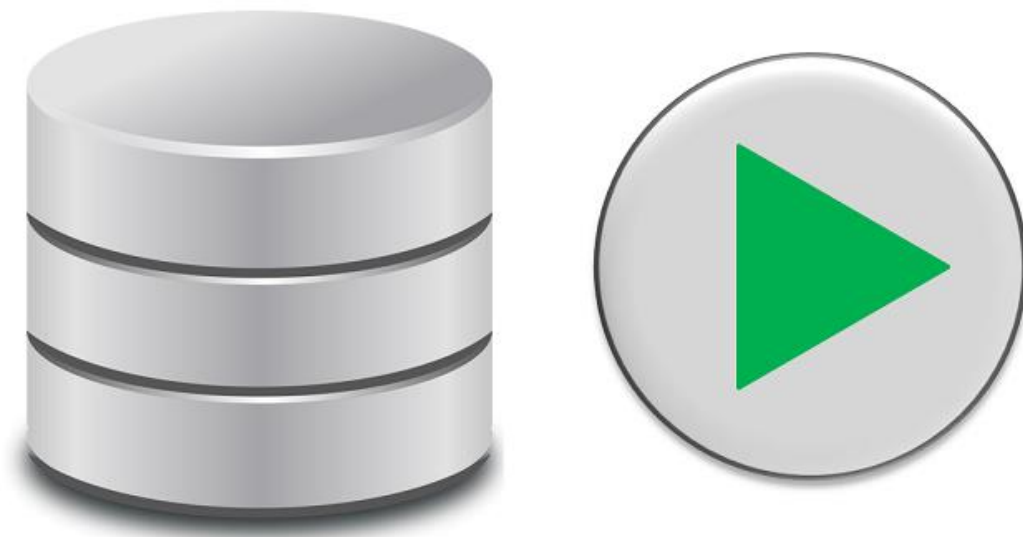




# ORACLE PL/SQL



MÓDULO 07

## TRATAMIENTO CON CURSORES

**GUSTAVO CORONEL**  
*desarrollasoftware.com*



# ORACLE PL/SQL



## Contenido

<b>PROCESAMIENTO DE CURSORES.....</b>	<b>3</b>
PASOS A SEGUIR .....	3
DECLARAR UN CURSOR.....	3
APERTURA DE UN CURSOR .....	3
EXTRACCIÓN DE DATOS DESDE UN CURSOR .....	4
CERRAR UN CURSOR .....	4
<b>ATRIBUTOS DE LOS CURSORES .....</b>	<b>5</b>
<b>BUCLES DE EXTRACCIÓN .....</b>	<b>5</b>
BUCLES SIMPLES.....	5
BUCLES WHILE .....	7
BUCLE FOR.....	10
<b>BUCLES FOR IMPLÍCITOS .....</b>	<b>12</b>
<b>CURSORES SELECT FOR UPDATE .....</b>	<b>13</b>
<b>CURSORES IMPLÍCITOS .....</b>	<b>14</b>
<b>CURSOS VIRTUALES .....</b>	<b>15</b>
JAVA ORIENTADO A OBJETOS.....	15
PROGRAMACIÓN DE BASE DE DATOS CON JAVA JDBC .....	15



## PROCESAMIENTO DE CURSORES

Los cursores permiten realizar recorridos a través de las filas de una consulta.

### Pasos a Seguir

1. Declarar el cursor
2. Apertura del cursor
3. Extracción de los resultados
4. Cerrar el cursor

### Declarar un Cursor

#### Sintaxis

```
CURSOR NOMBRE_CURSOR [ ( PARÁMETROS ) ]  
IS SENTENCIA_SELECT;
```

#### Script 1

```
DECLARE  
CURSOR C_DEMO IS SELECT * FROM DEPT;
```

### Apertura de un Cursor

#### Sintaxis

```
OPEN nombre_cursor [(argumentos)];
```

#### Script 2

```
OPEN C_DEMO;
```



# ORACLE PL/SQL



## Extracción de datos desde un cursor

### Sintaxis 1

```
Fetch nombre_cursor into lista_variables;
```

### Sintaxis 2

```
Fetch nombre_cursor into registro;
```

### Script 3

```
FETCH C_DEMO INTO COD, NOM, LOC;
```

## Cerrar un cursor

### Sintaxis

```
CLOSE NOMBRE_CURSOR;
```

### Script 4

```
create or replace procedure pr114
is
  cursor c_demo is select * from dept;
  r dept%rowtype;
begin
  open c_demo;
  fetch c_demo into r;
  close c_demo;
  dbms_output.put_line('deptno: ' || r.deptno);
  dbms_output.put_line('dname: ' || r.dname);
  dbms_output.put_line('loc: ' || r.loc);
end;
```



# ORACLE PL/SQL



Ejecución:

```
SQL> exec pr114;
deptno: 10
dname: ACCOUNTING
loc: NEW YORK

PL/SQL procedure successfully completed.
```

## ATRIBUTOS DE LOS CURSORES

ATRIBUTO	DESCRIPCIÓN
<b>%Found</b>	Devuelve TRUE si la última sentencia fetch tuvo éxito.
<b>%NotFound</b>	Devuelve TRU si la última sentencia fetch no tuvo éxito.
<b>%IsOpen</b>	Este atributo se utiliza para averiguar si un cursor esta abierto ó no.
<b>%RowCount</b>	Este atributo se utiliza para averiguar la cantidad de filas que se van extrayendo del cursor.

## BUCLES DE EXTRACCIÓN

### Bucles Simples

Formato

```
open cursor_name ;
loop
  fetch cursor_name into ... ;
  exit when cursor_name%notfound;
  -----
  -----
  -----
end loop;
close cursor_name;
```



# ORACLE PL/SQL



## Script 5

Procedimiento para listar los códigos y nombres de los empleados.

```
create or replace procedure pr115
is
  cursor c_emp is select * from emp;
  r emp%rowtype;
begin
  open c_emp;
  loop
    fetch c_emp into r;
    exit when c_emp%notfound;
    dbms_output.put_line(r.empno || ' - ' || r.ename);
  end loop;
  close c_emp;
end;
```

Ejecución:

```
SQL> exec pr115;
7369 - SMITH
7499 - ALLEN
7521 - WARD
7566 - JONES
7654 - MARTIN
7698 - BLAKE
7782 - CLARK
7788 - SCOTT
7839 - KING
7844 - TURNER
7876 - ADAMS
7900 - JAMES
7902 - FORD
7934 - MILLER

PL/SQL procedure successfully completed.
```



# ORACLE PL/SQL



## Bucles While

### Formato

```
OPEN CURSOR_NAME;  
FETCH CURSOR_NAME INTO ... ;  
WHILE CURSOR_NAME%FOUND LOOP  
    -----  
    -----  
    -----  
    FETCH CURSOR_NAME INTO ... ;  
END LOOP;  
CLOSE CURSOR_NAME;
```

### Script 6

Tenemos una tabla de nombre PLANILLAMES para guardar la planilla por mes, por departamento, la estructura es la siguiente:

```
CREATE TABLE PLANILLAMES(  
    ANIO NUMBER(4),  
    MES NUMBER(2),  
    DEPTNO NUMBER(2),  
    EMPS NUMBER(2) NOT NULL,  
    PLANILLA NUMBER(10,2) NOT NULL,  
    CONSTRAINT PK_PLANILLAMES PRIMARY KEY(ANIO,MES, DEPTNO)  
);
```



# ORACLE PL/SQL



Ahora desarrollaremos un procedimiento para generar la planilla de un determinado mes. Este procedimiento debe verificar si la planilla ya fue generada.

```
create or replace procedure pr116(p_anio number, p_mes number)
is
  cursor c_dept is select deptno from dept;
  v_deptno dept.deptno%type;
  cont number;
  v_emps number;
  v_planilla number;
begin
  select count(*) into cont
    from planillames
   where anio = p_anio and mes = p_mes;
  if (cont > 0) then
    dbms_output.put_line('Ya esta procesado');
    return;
  end if;
  open c_dept;
  fetch c_dept into v_deptno;
  while c_dept%found loop
    select count(*), sum(sal) into v_emps, v_planilla
      from emp
     where deptno = v_deptno;
    insert into planillames
      values(p_anio, p_mes, v_deptno, v_emps, nvl(v_planilla,0));
    fetch c_dept into v_deptno;
  end loop;
  close c_dept;
  commit;
  dbms_output.put_line('Proceso ok.');
```

end;

Ejecución:

```
SQL> exec pr116(2005,2);
Proceso ok.

PL/SQL procedure successfully completed.
```





# ORACLE PL/SQL



Consultemos el resultado:

```
SQL> select * from planillames;
```

ANIO	MES	DEPTNO	EMPS	PLANILLA
2005	2	10	3	8750
2005	2	20	5	10875
2005	2	30	6	9400
2005	2	40	0	0
2005	2	50	0	0



# ORACLE PL/SQL



## Bucle For

### Formato

```
for variable in cursor_name loop
    -----
    -----
end loop
```

### Script 7

Procedimiento para determinar el número de empleados y el importe de la planilla, por departamento.

```
create or replace procedure pr117
is
    cursor c_dept is select * from dept;
    emps number;
    planilla number;
    cad varchar2(100);
begin
    for r in c_dept loop
        select count(*), sum(nvl(sal,0)) into emps, planilla
            from emp where deptno = r.deptno;
        cad := r.deptno || ' - ' || emps || ' - ' || nvl(planilla,0);
        dbms_output.put_line(cad);
    end loop;
end;
```

### Ejecución:

```
SQL> exec pr117;
10 - 3 - 8750
20 - 5 - 10875
30 - 6 - 9400
40 - 0 - 0
50 - 0 - 0

PL/SQL procedure successfully completed.
```

### Ejercicio 1

Desarrollar un procedimiento que determine el empleado con mayor sueldo por departamento.



# ORACLE PL/SQL





# ORACLE PL/SQL



## BUCLAS FOR IMPLÍCITOS

### Formato

```
FOR VARIABLE IN (SENTENCIA_SELECT) LOOP
    -----
    -----
END LOOP
```

### Script 8

Determinar el sueldo promedio por departamento.

```
create or replace procedure pr118
is
    prom number;
begin
    for r in (select deptno from dept) loop
        select avg(nvl(sal,0)) into prom
            from emp where deptno = r.deptno;
        dbms_output.put_line(r.deptno || '-' || to_char(nvl(prom,0),'999,990.00'));
    end loop;
end;
```

### Ejecución:

```
SQL> exec pr118;
10-    2,916.67
20-    2,175.00
30-    1,566.67
40-         0.00
50-         0.00

PL/SQL procedure successfully completed.
```



## CURSORES SELECT FOR UPDATE

### Sintaxis

```
FOR UPDATE [OF LISTA_COLUMNAS] [NOWAIT | WAIT N]
```

### Script 9

Listado de empleados.

```
create or replace procedure pr119
is
  cursor c_demo is select * from emp for update wait 2;
begin
  for r in c_demo loop
    dbms_output.put_line(r.empno || '-' || r.ename);
  end loop;
end;
```

Antes de ejecutar el procedimiento, en otra ventana inicie una transacción sobre la tabla **emp**.

```
SQL> exec pr119;
BEGIN pr119; END;

*
ERROR at line 1:
ORA-30006: resource busy; acquire with WAIT timeout expired
ORA-06512: at "SCOTT.PR119", line 3
ORA-06512: at "SCOTT.PR119", line 5
ORA-06512: at line 1;
```



# ORACLE PL/SQL



## CURSORES IMPLÍCITOS

Se puede utilizar **SQL%Atributo** para verificar la ejecución de una sentencia SQL.

### Script 10

Actualizar el salario de un empleado.

```
create or replace procedure pr120(cod number, delta number)
is
begin
    update emp
        set sal = sal + delta
        where empno = cod;
    if sql%notfound then
        dbms_output.put_line('no existe');
    else
        commit;
        dbms_output.put_line('proceso ok');
    end if;
end;
```

Ejecución:

```
SQL> exec pr120(7369,200);
proceso ok

PL/SQL procedure successfully completed.
```



# ORACLE PL/SQL



## CURSOS VIRTUALES

En esta URL tienes cupones de descuento:

<http://gcoronelc.blogspot.com/p/cursos-virtuales.html>

## Java Orientado a Objetos



### **JAVA ORIENTADO A OBJETOS**

---

Eric Gustavo Coronel Castillo  
[www.desarrollasoftware.com](http://www.desarrollasoftware.com)  
INSTRUCTOR

<https://www.udemy.com/java-orientado-a-objetos/>

## Programación de Base de Datos con Java JDBC



### **PROGRAMACIÓN DE BASE DE DATOS ORACLE CON JAVA JDBC**

---

Eric Gustavo Coronel Castillo  
[www.desarrollasoftware.com](http://www.desarrollasoftware.com)  
INSTRUCTOR

[https://www.udemy.com/java-jdbc-oracle /](https://www.udemy.com/java-jdbc-oracle/)