

# Cursores

# Cursores Implícitos - Atributos

- %FOUND

```
DELETE FROM emp WHERE empno = mi_empno;  
IF SQL%FOUND THEN / *si la eliminación fue  
                    exitosa */  
INSERT INTO nuevos_empleados  
VALUES (mi_empno, mi_enombre, ...);
```

- %ISOPEN

- %NOTFOUND

# Cursores Implícitos - Atributos

- %ROWCOUNT

```
PROCEDURE borrar_filas (p_orden_id NUMBER) IS
v_filas_borradas NUMBER;
BEGIN
DELETE FROM item WHERE orden_id =p_orden_id;
v_filas_borradas := SQL%ROWCOUNT;
DBMS_OUTPUT.PUT_LINE(TO_CHAR(v_filas_borradas)||
‘ filas borradas.’);
END borrar_filas;
```

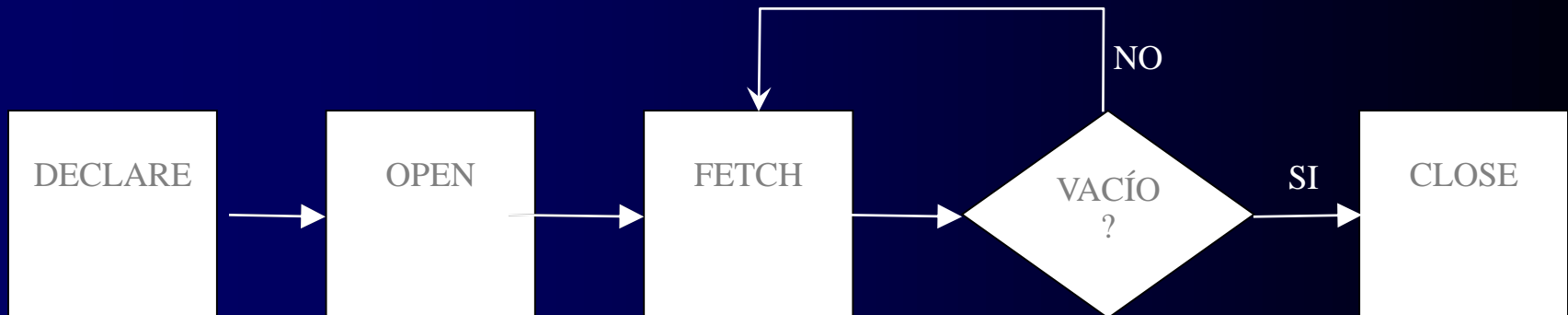
# Cursores Explícitos

Permiten:

- procesar, fila por fila, más allá de la primera fila retornada por una consulta
- mantener el control sobre la fila que está siendo procesada
- controlar los cursores manualmente dentro de un bloque PL/SQL.

# Control de Cursores Explícitos

- OPEN
- FETCH
- CLOSE



# Declaración de un Cursor

- Sintaxis:

```
CURSOR nombre_cursor [(parámetro[, parámetro]...)]  
IS sentencia_select;
```

```
nombre_parametro_cursor [IN] tipo_dato [{:=  
DEFAULT} expresión]
```

# Declaración de un Cursor

- Ejemplos:

```
DECLARE  
CURSOR c1 IS SELECT empno, enombre, tarea,  
                    salario  
FROM emp  
WHERE salario > 2000;
```

```
CURSOR c2 (bajo INTEGER DEFAULT 0,  
          alto INTEGER DEFAULT 99)  
IS SELECT ...;
```

# Acciones de la sentencia OPEN

- Asigna memoria dinámicamente.
- Interpreta la sentencia SELECT.
- Inicializa los parámetros de entrada obteniendo sus direcciones de memoria.
- Identifica el conjunto de filas que satisfacen el criterio de búsqueda.
- Posiciona el cursor en la primera fila del conjunto activo.



# Apertura de un Cursor

- Sintaxis:

```
OPEN nombre_cursor;
```

- Ejemplo:

```
DECLARE
```

```
    CURSOR c1 IS SELECT enombre, tarea  
                  FROM emp  WHERE salario < 3000;
```

```
BEGIN
```

```
    ...  
    OPEN c1;
```

```
    ...
```

```
END;
```

# Acciones de la sentencia FETCH

- Avanza el puntero a la próxima fila del conjunto activo.
- Lee la información de la fila activa en las variables de salida.
- Abandona el ciclo FOR de cursores si el puntero se posiciona al final del conjunto activo.

# Recuperación de Datos desde el Cursor

- Sintaxis:

```
FETCH c1 INTO v_empno, v_enombre;
```

- Ejemplo:

```
LOOP
```

```
    FETCH c1 INTO mi_registro;
```

```
    EXIT WHEN c1%NOTFOUND;
```

```
    -- procesa el registro de datos
```

```
END LOOP;
```

# Cierre de un Cursor

- Sintaxis:

`CLOSE nombre_cursor;`

- La excepción `INVALID_CURSOR` se dispara cuando se trata realizar una operación sobre un cursor cerrado
- Se deben cerrar los cursores explícitos para liberar recursos
- El parámetro `OPEN_CURSORS` define la cantidad de cursores abiertos por sesión de usuario

# Ejemplo

```
DECLARE
v_salario emp.salario%TYPE;
v_factor  NUMBER := 2;
CURSOR c1(p_factor number) IS
    SELECT p_factor*salario
    FROM emp WHERE tarea = mi_tarea;
BEGIN
OPEN c1(v_factor); -- aquí v_factor es igual a 2
LOOP
    ...
    FETCH c1 INTO v_salario;
    EXIT WHEN c1%NOTFOUND;
    v_factor := v_factor + 1; -- no afecta al FETCH
END LOOP;
CLOSE c1;
END;
```

# Cursores Explícitos - Atributos

- %FOUND

```
LOOP
```

```
    FETCH c1 INTO v_enombre, v_salario,  
                  v_fecha_empleo;
```

```
    IF c1%FOUND THEN -- FETCH exitoso
```

```
        ...
```

```
    ELSE -- el FETCH falló, sale del ciclo
```

```
        EXIT;
```

```
    END IF;
```

```
END LOOP;
```

# Cursores Explícitos - Atributos

- %ISOPEN

```
IF c1%ISOPEN THEN -- el cursor está abierto
```

```
...
```

```
    FETCH c1 INTO v_enombre, v_salario
```

```
...
```

```
ELSE -- el cursor está cerrado, entonces lo abre
```

```
    OPEN c1;
```

```
END IF;
```

# Cursores Explícitos - Atributos

- %NOTFOUND

```
LOOP
```

```
    FETCH c1 INTO v_enombre, v_salario;
```

```
    EXIT WHEN c1%NOTFOUND;
```

```
    ...
```

```
END LOOP;
```

```
EXIT WHEN c1%NOTFOUND OR  
         c1%NOTFOUND IS NULL;
```



# Cursores Explícitos - Atributos

- %ROWCOUNT

```
LOOP
    FETCH c1 INTO mi_enombre, mi_deptno;
    IF c1%ROWCOUNT > 10 THEN
        ...
    END IF;
    ...
END LOOP;
```

# Ciclos FOR para Cursores

```
DECLARE
v_resultado temp.col1%TYPE;
CURSOR c1 IS SELECT n1, n2, n3
                FROM tabla_datos WHERE experimento_nro =1;
BEGIN
    FOR c1_reg IN c1 LOOP
        /* calcula y almacena los resultados */
        v_resultado := c1_reg.n2 / (c1_reg.n1 + c1_reg.n3);
        INSERT INTO temp
            VALUES (v_resultado, NULL, NULL);
    END LOOP;
    COMMIT;
END;
```

# Uso de Subconsultas

```
DECLARE
    v_gratificacion number;
BEGIN
    FOR emp_reg IN      (SELECT empno, salario, comision
                        FROM emp) LOOP
        v_gratificacion := (emp_reg.salario * 0.05) +
                            (emp_reg.comision * 0.25);
        INSERT INTO gratificaciones
            VALUES (emp_reg.empno, v_gratificacion);
    END LOOP;
    COMMIT;
END;
```

# Utilización de la Cláusula WHERE CURRENT OF

```
...  
CURSOR emp_cursor IS  SELECT ...  
    FROM ... WHERE ...  
        FOR UPDATE [OF ref_columna] [NOWAIT];  
BEGIN  
  
...  
FOR emp_registro IN emp_cursor LOOP  
    UPDATE ...  
        WHERE CURRENT OF emp_cursor;  
  
...  
END LOOP;  
COMMIT;  
END;
```