

# 7

## Paquetes ORACLE

**Bases de datos II**  
**Universidad del Cauca**  
**Ing. Wilson Ortega**

# Introducción

- Un paquete es un objeto de un esquema que agrupa lógicamente tipos, variables y subprogramas.
- Usualmente los paquetes tienen dos partes:
  - Una especificación
  - Cuerpo; algunas veces el cuerpo no es necesario.

# Introduccion(2)

- La especificación es la interfaz del paquete. Declara tipos, variables, constantes, excepciones, cursores y subprogramas que pueden ser llamados desde fuera.
- El cuerpo define las consultas para el cursor y el código para los subprogramas.

# Contenido de un paquete

- Metodos GET y SET para variables en caso de que no se desee que desde fuera del paquete se lea y escriba en ellas directamente.
- Declaraciones de cursores para reusar.
- Declaración de excepciones
- Declaración de subprogramas
- Declaración de subprogramas sobrecargados (mismo nombre, distintos parámetros).
- Variables globales
- Declaraciones de tipos compuestos. Esto permitirá usarlos como parámetros de subprogramas.

# Ventajas de los paquetes

- **Modularidad**
  - División por funcionalidades comunes
- **Facilidad en el diseño de aplicaciones**
  - Definir primero la interfaz
- **Encapsulamiento**
  - Se ocultan detalles de implementación
- **Datos en sesión**
  - Variables y cursores
- **Mejor rendimiento**
  - Se cargan en memoria en la primera ejecución

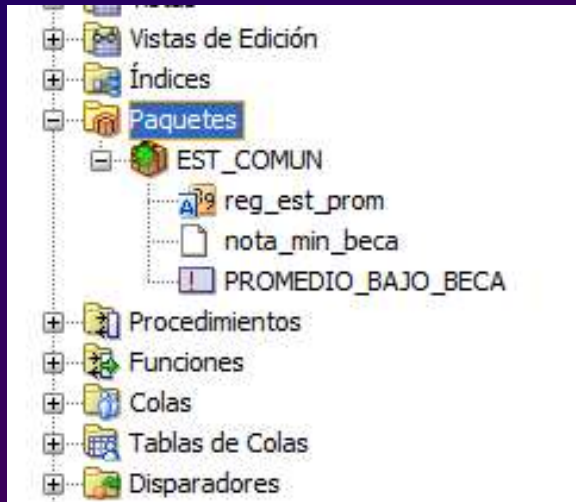
# Especificación del paquete

- La especificación del paquete contiene las declaraciones públicas
  - Accesibles desde el paquete o desde subprogramas del mismo esquema
- Si no se declaran cursores o subprogramas no es necesario crear el cuerpo del paquete.

```
CREATE [OR REPLACE] PACKAGE nombre_paquete AS  
  - variables  
  - tipos  
  - excepciones  
END;
```

# Especificación del paquete

## Ejemplo



```
CREATE OR REPLACE PACKAGE EST_COMUN AS
  TYPE reg_est_prom IS RECORD
  (
    nombre ESTUDIANTE.codigo%TYPE,
    promedio ESTUDIANTE.promedio%TYPE
  );
  nota_min_beca ESTUDIANTE.promedio%TYPE;
  PROMEDIO_BAJO_BECA exception;
END;
```

## Usar el paquete:

```
BEGIN
  EST_COMUN.nota_min_beca := 4;
  DBMS_OUTPUT.PUT_LINE('Nota mínima para beca: ' || EST_COMUN.nota_min_beca);
END ;
```



# Cuerpo del paquete

- Contiene la implementación de cada cursor y subprograma definido en la especificación
- Un subprograma definido en el cuerpo es accesible desde el exterior solamente si su definición aparece en la especificación del paquete
- El cuerpo se debe definir en el mismo esquema que su especificación
- La especificación y la implementación de los subprogramas debe coincidir palabra a palabra
- Se pueden definir variables en el cuerpo. En este caso sólo serán visibles internamente.

```
CREATE [OR REPLACE] PACKAGE BODY nombre_paquete AS  
  - variables internas  
  - implementación de subprogramas  
END;
```



# Cuerpo del paquete

## Ejemplo (Especificación del paquete)

```
CREATE OR REPLACE PACKAGE EST_COMUN AS
  TYPE reg_est_prom IS RECORD
  (
    nombre ESTUDIANTE.codigo%TYPE,
    promedio ESTUDIANTE.promedio%TYPE
  );
  nota_min_beca ESTUDIANTE.promedio%TYPE;
  PROMEDIO_BAJO_BECA exception;

  FUNCTION ObtenerPromedio(p_cod ESTUDIANTE.codigo%TYPE)
  RETURN ESTUDIANTE.promedio%TYPE;

  PROCEDURE ActualizarPromedio(p_cod ESTUDIANTE.codigo%TYPE,
    p_nuevo_prom ESTUDIANTE.promedio%TYPE);
END;
```

# Cuerpo del paquete

## Ejemplo (Implementación del cuerpo)

```
CREATE OR REPLACE PACKAGE BODY EST_COMUN AS

FUNCTION ObtenerPromedio(p_cod ESTUDIANTE.codigo%TYPE)
RETURN ESTUDIANTE.promedio%TYPE
IS
    v_prom ESTUDIANTE.promedio%TYPE := 0;
BEGIN
    SELECT promedio INTO v_prom FROM ESTUDIANTE
    WHERE codigo = p_cod;
    RETURN v_prom;

    EXCEPTION
    WHEN OTHERS THEN
        RETURN NULL;
END;

PROCEDURE ActualizarPromedio(p_cod ESTUDIANTE.codigo%TYPE,
    p_nuevo_prom ESTUDIANTE.promedio%TYPE)
IS BEGIN
    UPDATE ESTUDIANTE SET promedio = p_nuevo_prom WHERE codigo = p_cod;
END;
END;
```

# Uso del paquete

## Ejemplo

```
SET SERVEROUTPUT ON;  
DECLARE  
v_cod ESTUDIANTE.codigo%TYPE := 23;  
BEGIN  
    DBMS_OUTPUT.PUT_LINE('Promedio del estudiante ' || v_cod || ' : ' ||  
EST_COMUN.ObtenerPromedio(v_cod));  
END ;
```

```
Promedio del estudiante 23 : 2,1
```

# Usando cursores en paquetes

- En la especificación del paquete se define el nombre y tipo de retorno del cursor.
- En el cuerpo se define la sentencia SELECT y el tipo de retorno (debe coincidir con la especificación).
- El uso de cursores en paquetes incrementa la flexibilidad pues es posible cambiar el select y la definición se mantendrá igual.

# Usando cursores en paquetes

## Ejemplo (Especificación)

```
CREATE OR REPLACE PACKAGE EST_COMUN AS
  TYPE reg_est_prom IS RECORD
  (
    nombre ESTUDIANTE.codigo%TYPE,
    promedio ESTUDIANTE.promedio%TYPE
  );
  nota_min_beca ESTUDIANTE.promedio%TYPE;
  PROMEDIO_BAJO_BECA exception;

  FUNCTION ObtenerPromedio(p_cod ESTUDIANTE.codigo%TYPE)
  RETURN ESTUDIANTE.promedio%TYPE;

  PROCEDURE ActualizarPromedio(p_cod ESTUDIANTE.codigo%TYPE,
    p_nuevo_prom ESTUDIANTE.promedio%TYPE);

  CURSOR cur_est_pro RETURN ESTUDIANTE%ROWTYPE;
END;
```

# Usando cursores en paquetes

## Ejemplo (Cuerpo)

```
CREATE OR REPLACE PACKAGE BODY EST_COMUN AS

  CURSOR cur_est_pro RETURN ESTUDIANTE%ROWTYPE IS
  SELECT * FROM ESTUDIANTE
  WHERE programa_id = 1;

  -- Implementación de los subprogramas
  --
  --
  --
END;
```

# Uso del cursor

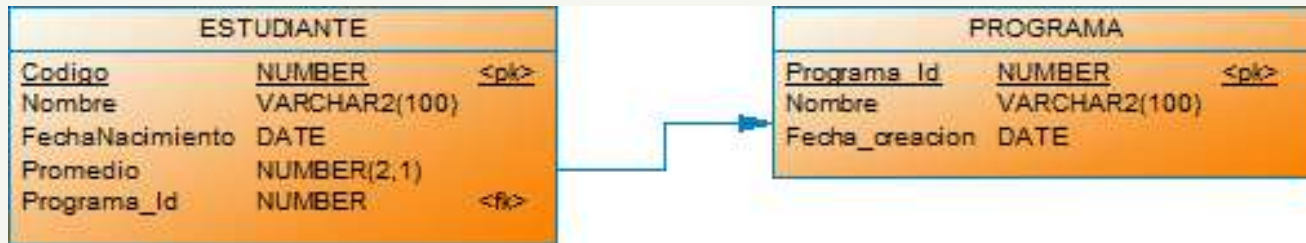
## Ejemplo

```
SET SERVEROUTPUT ON;
DECLARE
v_pro ESTUDIANTE.programa_id%TYPE := 1;
BEGIN
  FOR reg_est IN EST_COMUN.cur_est_pro LOOP
    DBMS_OUTPUT.PUT_LINE('Nombre: ' || reg_est.nombre || ' - Promedio: ' || reg_est.promedio);
  END LOOP;
END ;
```

```
Nombre: Juan - Promedio: 2,1
Nombre: Sara - Promedio: 4
Nombre: Pedro - Promedio: 3,4
```



# Ejercicio



- Cree un paquete para la gestión de programas académicos llamado PRO\_GESTION que implemente:
  - Un cursor que retorne los datos de todos los programas creados antes del año 2000.
  - Una función que retorne el número de estudiantes de un programa determinado. Se recibe como parámetro el programa\_Id.
  - Un procedimiento que guarde en una tabla anidad pública del paquete Nombre y Fecha de creación de todos los programas.
- Cree un bloque anónimo que muestre los nombres de los programas creados antes del año 2000 usando el cursor creado en el punto anterior.
- Cree un bloque anónimo que use la función del paquete creado en el primer punto.

# Bibliografía

- **Oracle® Database PL/SQL Language Reference**  
-11g Release 1 (11.1) - 2009