



UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior
de Ingeniería Informática
Departamento de Lenguajes
y Sistemas Informáticos

Pruebas en PL/SQL

*Grupo de Ingeniería del Software y Bases de Datos
Departamento de Lenguajes y Sistemas Informáticos
Universidad de Sevilla
diciembre 2013*



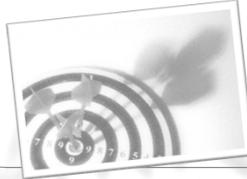
ETSII
UNIVERSARIO



UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior
de Ingeniería Informática
Departamento de Lenguajes
y Sistemas Informáticos

Pruebas en PL/SQL

- **Objetivos de este tema**
 - Entender la necesidad de realizar pruebas para detectar posibles errores en las operaciones de manipulación y consulta de datos.
 - Proporcionar un esquema sencillo para implementar pruebas en Oracle usando el lenguaje PL/SQL.



diciembre 2013 Introducción a la Ingeniería del Software y a los Sistemas de Información 1



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Pruebas en PL/SQL

- Las **pruebas software** tienen como objetivo detectar errores en el software.
- Una prueba está formada por:
 - { **valores de entrada**, **salida esperada** }
- Ejemplo:
 - { **Insertar("Manuel","28807980L")**, **Inserción correcta** }
 - { **Insertar("Antonio","288070L")**, **"Fallo, DNI no válido"** }

1. Pruebas software

2. Pruebas en PL/SQL

3. Ejercicios

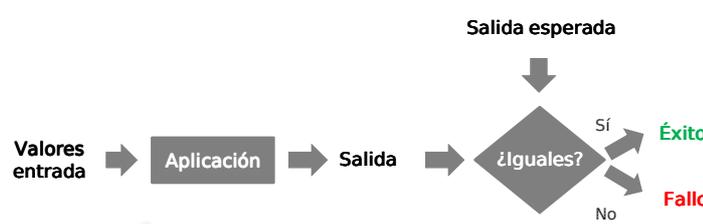
diciembre 2013 Introducción a la Ingeniería del Software y a los Sistemas de Información 2



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Pruebas en PL/SQL

- Proceso general de prueba:



```

graph LR
    A[Valores entrada] --> B[Aplicación]
    B --> C[Salida]
    C --> D{¿Iguales?}
    E[Salida esperada] --> D
    D -- Sí --> F[Éxito]
    D -- No --> G[Fallo]
            
```

1. Pruebas software

2. Pruebas en PL/SQL

3. Ejercicios

diciembre 2013 Introducción a la Ingeniería del Software y a los Sistemas de Información 3



UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Pruebas en PL/SQL

- Usaremos pruebas para detectar posibles errores en las operaciones de **inserción, actualización, eliminación y consulta** de nuestra base de datos.
- Un ejemplo

Nombre = 'Manuel'
Salario = 1000
Comisión = 0.7

Salida esperada =
Inserción realizada

```

    graph LR
      A[Insertar Empleado] --> B[Inserción realizada]
      B --> C{¿Iguales?}
      C -- Sí --> D[Éxito]
      C -- No --> E[Fallo]
      
```

diciembre 2013

Introducción a la Ingeniería del Software y a los Sistemas de Información

4



UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Pruebas en PL/SQL

- Usaremos pruebas para detectar posibles errores en las operaciones de **inserción, actualización, eliminación y consulta** de nuestra base de datos.
- Un ejemplo:

Nombre = 'Manuel'
Salario = 1000
Comisión = 1.7 (*)

Salida esperada =
Inserción fallida

```

    graph LR
      A[Insertar Empleado] --> B[Inserción realizada]
      B --> C{¿Iguales?}
      C -- No --> D[Fallo]
      C -- Sí --> E[Éxito]
      
```

(*) La comisión de un empleado debe ser un valor entre 0 y 1.

diciembre 2013

Introducción a la Ingeniería del Software y a los Sistemas de Información

5



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Pruebas en PL/SQL

- Para la implementación de las pruebas se recomienda el uso de paquetes PL/SQL.
- Un **paquete** es una estructura que agrupa objetos de PL/SQL como procedimientos o funciones.
- Los paquetes se definen en dos partes:
 - **Especificación.** Incluye, entre otros, el prototipo de las funciones y procedimientos que contiene.
 - **Cuerpo.** Implementación de las funciones y procedimientos del paquete.

1. Pruebas software
2. Pruebas en PL/SQL
3. Ejercicios

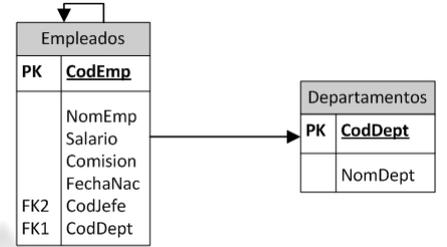
diciembre 2013 Introducción a la Ingeniería del Software y a los Sistemas de Información 6



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Pruebas en PL/SQL

- Un ejemplo:



```

    graph TD
      Empleados --> Departamentos
      subgraph Empleados
        direction TB
        E_PK[PK CodEmp]
        E_Nom[NomEmp]
        E_Sal[Salario]
        E_Com[Comision]
        E_Fecha[FechaNac]
        E_FK2[FK2 CodJefe]
        E_FK1[FK1 CodDept]
      end
      subgraph Departamentos
        direction TB
        D_PK[PK CodDept]
        D_Nom[NomDept]
      end
    
```

1. Pruebas software
2. Pruebas en PL/SQL
3. Ejercicios

diciembre 2013 Introducción a la Ingeniería del Software y a los Sistemas de Información 7



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

- 1. Pruebas software
- 2. Pruebas en PL/SQL
- 3. Ejercicios

Pruebas en PL/SQL

- Proceso de pruebas:
 1. Por cada tabla de la BD, crear un paquete PL/SQL con nombre "PRUEBAS_NOMBRETABLA"
 2. En cada paquete crear, al menos, cuatro procedimientos:
 1. Inicializar.
 2. Insertar.
 3. Actualizar.
 4. Eliminar.

diciembre 2013

Introducción a la Ingeniería del Software y a los Sistemas de Información

8



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

- 1. Pruebas software
- 2. Pruebas en PL/SQL
- 3. Ejercicios

Pruebas en PL/SQL

- Especificación del paquete PRUEBAS_DEPARTAMENTOS

```

create or replace
PACKAGE PRUEBAS_DEPARTAMENTOS AS

  PROCEDURE inicializar;
  PROCEDURE insertar
    (nombre_prueba VARCHAR2, w_nom_dept VARCHAR2, salidaEsperada BOOLEAN);
  PROCEDURE actualizar
    (nombre_prueba VARCHAR2, w_cod_dept INTEGER, w_nom_dept VARCHAR2, salidaEsperada BOOLEAN);
  PROCEDURE eliminar
    (nombre_prueba VARCHAR2, w_cod_dept INTEGER, salidaEsperada BOOLEAN);

END PRUEBAS_DEPARTAMENTOS;
        
```

NOMBRE_PROC (NOMBRE_PRUEBA, VALORES_ENTRADA, SALIDA_ESPERADA)

diciembre 2013

Introducción a la Ingeniería del Software y a los Sistemas de Información

9



UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Pruebas en PL/SQL

- Cuerpo del paquete PRUEBAS_DEPARTAMENTOS
- El procedimiento *inicializar* realiza las operaciones necesarias para dejar a la tabla un estado inicial controlado. Ej. Dejar la tabla vacía.

```

create or replace
PACKAGE BODY PRUEBAS_DEPARTAMENTOS AS

  /* INICIALIZACIÓN */
  PROCEDURE inicializar AS
  BEGIN
    /* Borrar contenido de la tabla */
    DELETE FROM departamentos;
  END inicializar;

```

1. Pruebas software
2. Pruebas en PL/SQL
3. Ejercicios



diciembre 2013
 Introducción a la Ingeniería del Software y a los Sistemas de Información
 10

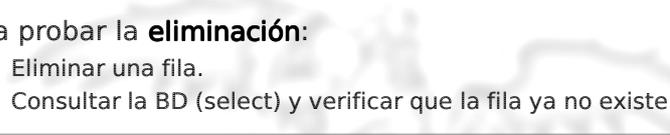


UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Pruebas en PL/SQL

- Para cada tabla de la BD se deben implementar procedimientos para probar la inserción, actualización y eliminación.
- Para probar la **inserción**:
 1. Insertar una fila con valores de prueba.
 2. Consultar la BD (select) y comprobar que los valores de la nueva fila coinciden con los valores de prueba.
- Para probar la **actualización**:
 1. Actualizar una fila con valores de prueba.
 2. Consultar la BD (select) y comprobar que los valores actualizados coinciden con los valores de prueba.
- Para probar la **eliminación**:
 1. Eliminar una fila.
 2. Consultar la BD (select) y verificar que la fila ya no existe.

1. Pruebas software
2. Pruebas en PL/SQL
3. Ejercicios



diciembre 2013
 Introducción a la Ingeniería del Software y a los Sistemas de Información
 11



UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Pruebas en PL/SQL

- Cuerpo del paquete PRUEBAS_DEPARTAMENTOS
 - Procedimiento *insertar*

```

/* PRUEBA PARA LA INSERCIÓN DE DEPARTAMENTOS */
PROCEDURE insertar (nombre_prueba VARCHAR2, w_nom_dept VARCHAR2, salidaEsperada BOOLEAN) AS
  salida BOOLEAN := true;
  departamento departamentos%ROWTYPE;
  w_cod_dept INTEGER;
BEGIN

  /* Insertar departamento */
  INSERT INTO departamentos VALUES (sec_dept.nextval, w_nom_dept);

  /* Seleccionar departamento y comprobar que los datos se insertaron correctamente */
  w_cod_dept := sec_dept.currval;
  SELECT * INTO departamento FROM departamentos WHERE cod_dept=w_cod_dept;
  IF (departamento.nom_dept<>w_nom_dept) THEN
    salida := false;
  END IF;
  COMMIT WORK;

  /* Mostrar resultado de la prueba */
  DBMS_OUTPUT.put_line(nombre_prueba || ':' || ASSERT_EQUALS(salida, salidaEsperada));

EXCEPTION
WHEN OTHERS THEN
  DBMS_OUTPUT.put_line(nombre_prueba || ':' || ASSERT_EQUALS(false, salidaEsperada));
  ROLLBACK;
END insertar;

```

1. Pruebas software
2. Pruebas en PL/SQL
3. Ejercicios

diciembre 2013 Introducción a la Ingeniería del Software y a los Sistemas de Información 12



UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Pruebas en PL/SQL

- Función auxiliar ASSERT_EQUALS

```

create or replace
FUNCTION ASSERT_EQUALS (salida BOOLEAN, salida_esperada BOOLEAN) RETURN VARCHAR2 AS
BEGIN
  IF (salida = salida_esperada) THEN
    RETURN 'EXITO';
  ELSE
    RETURN 'FALLO';
  END IF;
END ASSERT_EQUALS;

```

1. Pruebas software
2. Pruebas en PL/SQL
3. Ejercicios

diciembre 2013 Introducción a la Ingeniería del Software y a los Sistemas de Información 13



UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Pruebas en PL/SQL

- Cuerpo del paquete PRUEBAS_DEPARTAMENTOS
 - Procedimiento *actualizar*

```

/* PRUEBA PARA LA ACTUALIZACIÓN DE DEPARTAMENTOS */
PROCEDURE actualizar (nombre_prueba VARCHAR2,w_cod_dept INTEGER, w_nom_dept VARCHAR2,
                    salidaEsperada BOOLEAN) AS
    salida BOOLEAN := true;
    departamento departamentos%ROWTYPE;
BEGIN

    /* Actualizar empleado */
    UPDATE departamentos SET nom_dept=w_nom_dept WHERE cod_dept=w_cod_dept;

    /* Seleccionar departamento y comprobar que los campos se actualizaron correctamente */
    SELECT * INTO departamento FROM departamentos WHERE cod_dept=w_cod_dept;
    IF (departamento.nom_dept<>w_nom_dept) THEN
        salida := false;
    END IF;
    COMMIT WORK;

    /* Mostrar resultado de la prueba */
    DBMS_OUTPUT.put_line(nombre_prueba || ':' || ASSERT_EQUALS(salida,salidaEsperada));

EXCEPTION
WHEN OTHERS THEN
    DBMS_OUTPUT.put_line(nombre_prueba || ':' || ASSERT_EQUALS(false,salidaEsperada));
    ROLLBACK;
END actualizar;
            
```

1. Pruebas software
2. Pruebas en PL/SQL
3. Ejercicios

diciembre 2013 Introducción a la Ingeniería del Software y a los Sistemas de Información 14



UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

Pruebas en PL/SQL

- Cuerpo del paquete PRUEBAS_DEPARTAMENTOS
 - Procedimiento *eliminar*

```

/* PRUEBA PARA LA ELIMINACIÓN DE DEPARTAMENTOS */
PROCEDURE eliminar (nombre_prueba VARCHAR2,w_cod_dept INTEGER, salidaEsperada BOOLEAN) AS
    salida BOOLEAN := true;
    n_departamentos INTEGER;
BEGIN

    /* Eliminar empleado */
    DELETE FROM departamentos WHERE cod_dept=w_cod_dept;

    /* Verificar que el departamento no se encuentra en la BD */
    SELECT COUNT(*) INTO n_departamentos FROM departamentos WHERE cod_dept=w_cod_dept;
    IF (n_departamentos <> 0) THEN
        salida := false;
    END IF;
    COMMIT WORK;

    /* Mostrar resultado de la prueba */
    DBMS_OUTPUT.put_line(nombre_prueba || ':' || ASSERT_EQUALS(salida,salidaEsperada));

EXCEPTION
WHEN OTHERS THEN
    DBMS_OUTPUT.put_line(nombre_prueba || ':' || ASSERT_EQUALS(false,salidaEsperada));
    ROLLBACK;
END eliminar;
            
```

1. Pruebas software
2. Pruebas en PL/SQL
3. Ejercicios

diciembre 2013 Introducción a la Ingeniería del Software y a los Sistemas de Información 15



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

- 1. Pruebas software
- 2. Pruebas en PL/SQL
- 3. Ejercicios

Pruebas en PL/SQL

- **Proceso de pruebas:**
 1. Por cada tabla de la BD, crear un paquete PL/SQL con nombre "PRUEBAS_NOMBRETABLA"
 2. En cada paquete crear, al menos, cuatro procedimientos:
 1. Inicializar. Deja la BD en un estado inicial controlado.
 2. Insertar. Detección de errores en la inserción.
 3. Actualizar. Detección de errores en la actualización.
 4. Eliminar. Detección de errores en la eliminación.
 3. Crear un fichero PL/SQL de pruebas. Para cada tabla de la BD invocar a los procedimientos de pruebas con valores de entrada y salidas esperadas.

diciembre 2013
Introducción a la Ingeniería del Software y a los Sistemas de Información
16



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

- 1. Pruebas software
- 2. Pruebas en PL/SQL
- 3. Ejercicios

Pruebas en PL/SQL

- **Fichero principal de pruebas**

```

/* Activar salida de texto por pantalla */
SET SERVEROUTPUT ON;

DECLARE
    cod_dept INTEGER;
    cod_emp_paco INTEGER;
    cod_emp_manuel INTEGER;
BEGIN
    /*****
     PRUEBAS DE LAS OPERACIONES SOBRE LA TABLA DEPARTAMENTOS
     *****/
    PRUEBAS_DEPARTAMENTOS.INICIALIZAR;
    PRUEBAS_DEPARTAMENTOS.INSERTAR('Prueba 1 - Inserción depto','Marketing',true);
    cod_dept := sec_dept.currval;
    PRUEBAS_DEPARTAMENTOS.INSERTAR('Prueba 2 - Inserción depto con nombre null',null,false);
    PRUEBAS_DEPARTAMENTOS.ACTUALIZAR('Prueba 3 - Actualización nombre de depto',cod_dept,'Publicidad',true);
    PRUEBAS_DEPARTAMENTOS.ACTUALIZAR('Prueba 4 - Actualización nombre depto a null',cod_dept,null,false);
    PRUEBAS_DEPARTAMENTOS.ELIMINAR('Prueba 5 - Eliminar depto',cod_dept,true);

    /*****
     PRUEBAS DE LAS OPERACIONES SOBRE LA TABLA EMPLEADOS
     *****/
    ...
END;
                
```

diciembre 2013
Introducción a la Ingeniería del Software y a los Sistemas de Información
17

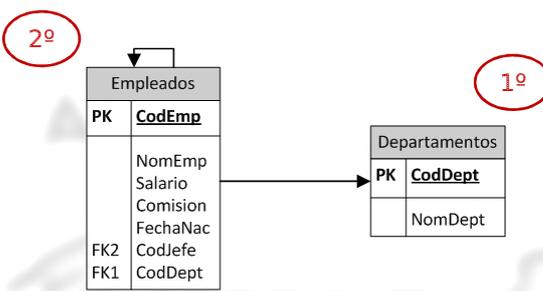
Pruebas en PL/SQL



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

1. Pruebas software
2. Pruebas en PL/SQL
3. Ejercicios

- Para facilitar el aislamiento de errores, se probarán primera las tablas sin dependencias con otras tablas (sin claves ajenas) y posteriormente las tablas que dependen de tablas ya probadas.



diciembre 2013
Introducción a la Ingeniería del Software y a los Sistemas de Información
18

Pruebas en PL/SQL



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

1. Pruebas software
2. Pruebas en PL/SQL
3. Ejercicios

- Deben realizarse tantas inserciones, actualizaciones y eliminaciones como sean necesarias para probar todos los triggers definidos.

diciembre 2013
Introducción a la Ingeniería del Software y a los Sistemas de Información
19

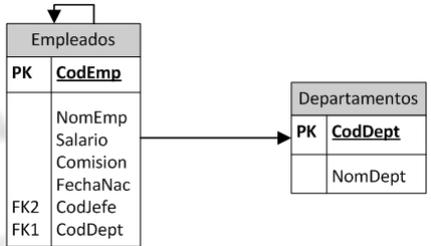
Pruebas en PL/SQL



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

1. Pruebas software
2. Pruebas en PL/SQL
3. Ejercicios

- **Ejercicio 1:** Implemente el siguiente modelo relacional y el código necesario para probar la inserción, actualización y eliminación de datos en la tabla Departamentos.



```

graph TD
    Empleados --> Departamentos
    subgraph Empleados
        direction TB
        E_PK[PK CodEmp]
        E_Nom[NomEmp]
        E_Sal[Salario]
        E_Com[Comision]
        E_Fecha[FechaNac]
        E_CodJefe[FK2 CodJefe]
        E_CodDept[FK1 CodDept]
    end
    subgraph Departamentos
        direction TB
        D_PK[PK CodDept]
        D_Nom[NomDept]
    end
        
```

diciembre 2013
Introducción a la Ingeniería del Software y a los Sistemas de Información
20

Pruebas en PL/SQL



Escuela Técnica Superior de Ingeniería Informática
Departamento de Lenguajes y Sistemas Informáticos

1. Pruebas software
2. Pruebas en PL/SQL
3. Ejercicios

- **Ejercicio 2:** Elimine la restricción “NOT NULL” de la columna nombre departamento para simular un error de programación. ¿Son sus pruebas capaces de detectar el error? ¿Y si introduce una comisión fuera de rango?

diciembre 2013
Introducción a la Ingeniería del Software y a los Sistemas de Información
21

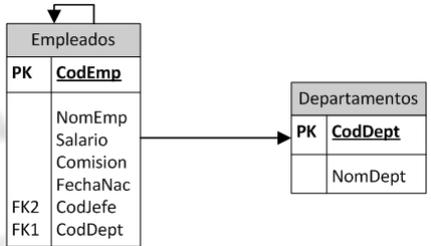
Pruebas en PL/SQL



Escuela Técnica Superior
de Ingeniería Informática
Departamento de Lenguajes
y Sistemas Informáticos

1. Pruebas software
2. Pruebas en PL/SQL
3. Ejercicios

- **Ejercicio 3:** Añada el código necesario para probar la inserción, actualización y eliminación de datos en la tabla Empleados.



```

    graph TD
      Empleados --> Departamentos
      subgraph Empleados
        direction TB
        PK1[PK CodEmp]
        NomEmp[NomEmp]
        Salario[Salario]
        Comision[Comision]
        FechaNac[FechaNac]
        CodJefe[CodJefe]
        FK1[FK1 CodDept]
      end
      subgraph Departamentos
        direction TB
        PK2[PK CodDept]
        NomDept[NomDept]
      end
      Empleados --- PK1
      Empleados --- NomEmp
      Empleados --- Salario
      Empleados --- Comision
      Empleados --- FechaNac
      Empleados --- CodJefe
      Empleados --- FK1
      Departamentos --- PK2
      Departamentos --- NomDept
      Empleados --> Departamentos
    
```

diciembre 2013
Introducción a la Ingeniería del Software y a los Sistemas de Información
22