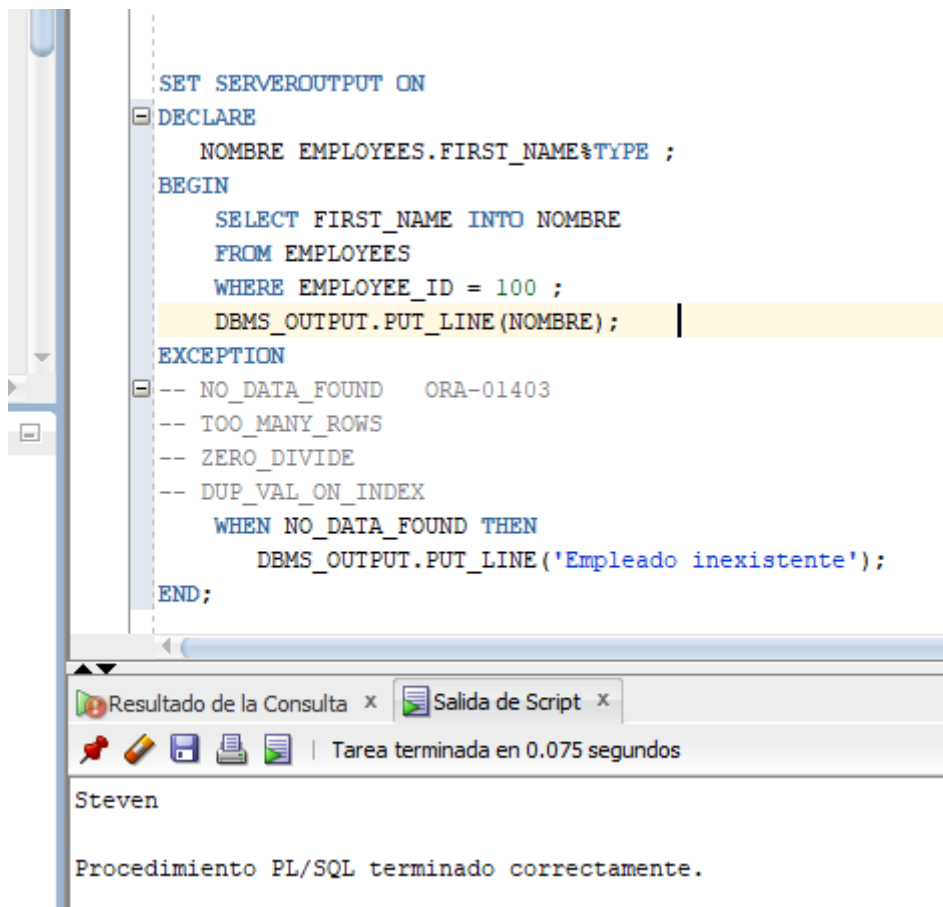


Prácticas ORACLE PL/SQL

Práctica con EXCEPCIONES

1- Crear bloque pl/Sql que devuelva el nombre de un empleado pasándole el EMPLOYEE_ID en el WHERE,

- Comprobar en primer lugar que funciona pasando un empleado existente
- Pasar un empleado inexistente y comprobar que genera un error
- Crear una zona de EXCEPTION controlando el NO_DATA_FOUND para que pinte un mensaje como "Empleado inexistente"



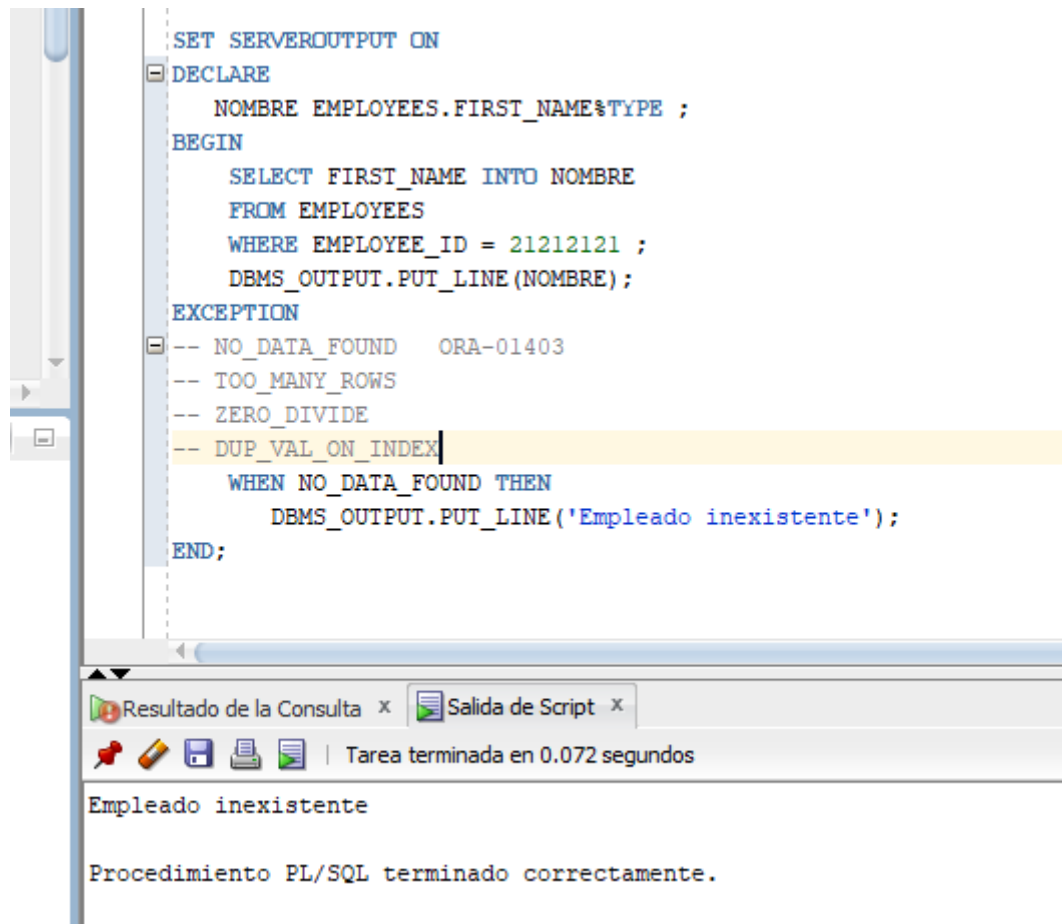
```
SET SERVEROUTPUT ON
DECLARE
    NOMBRE EMPLOYEES.FIRST_NAME%TYPE ;
BEGIN
    SELECT FIRST_NAME INTO NOMBRE
    FROM EMPLOYEES
    WHERE EMPLOYEE_ID = 100 ;
    DBMS_OUTPUT.PUT_LINE(NOMBRE);
EXCEPTION
    -- NO_DATA_FOUND      ORA-01403
    -- TOO_MANY_ROWS
    -- ZERO_DIVIDE
    -- DUP_VAL_ON_INDEX
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Empleado inexistente');
END;
```

Resultado de la Consulta x Salida de Script x

Tarea terminada en 0.075 segundos

Steven

Procedimiento PL/SQL terminado correctamente.



```
SET SERVEROUTPUT ON
DECLARE
  NOMBRE EMPLOYEES.FIRST_NAME%TYPE ;
BEGIN
  SELECT FIRST_NAME INTO NOMBRE
  FROM EMPLOYEES
  WHERE EMPLOYEE_ID = 21212121 ;
  DBMS_OUTPUT.PUT_LINE(NOMBRE);
EXCEPTION
  -- NO_DATA_FOUND    ORA-01403
  -- TOO_MANY_ROWS
  -- ZERO_DIVIDE
  -- DUP_VAL_ON_INDEX
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('Empleado inexistente');
END;
```

Resultado de la Consulta x Salida de Script x

Tarea terminada en 0.072 segundos

Empleado inexistente

Procedimiento PL/SQL terminado correctamente.

2- Modificar la SELECT para que devuelva más de un empleado, por ejemplo, poniendo EMPLOYEE_ID > 100. Debe generar un error. Controlar la excepción para que genere un mensaje como “Demasiados empleados en la consulta”

```
--"Demasiados empleados en la consulta"
SET SERVEROUTPUT ON
DECLARE
    NOMBRE EMPLOYEES.FIRST_NAME%TYPE ;
BEGIN
    SELECT FIRST_NAME INTO NOMBRE
    FROM EMPLOYEES
    WHERE EMPLOYEE_ID > 100 ;
    DBMS_OUTPUT.PUT_LINE(NOMBRE);
EXCEPTION
    -- NO_DATA_FOUND    ORA-01403
    -- TOO_MANY_ROWS
    -- ZERO_DIVIDE
    -- DUP_VAL_ON_INDEX
    WHEN TOO_MANY_ROWS THEN
        DBMS_OUTPUT.PUT_LINE('Demasiados empleados en la consulta');
END;
```

Resultado de la Consulta x Salida de Script x

Tarea terminada en 0.073 segundos

Demasiados empleados en la consulta

Procedimiento PL/SQL terminado correctamente.

3- Modificar la consulta para que devuelva un error de división por CERO, por ejemplo, vamos a devolver el salario en vez del nombre y lo dividimos por 0. En este caso, en vez de controlar la excepción directamente, creamos una sección WHEN OTHERS y pintamos el error con SQLCODE y SQLERR

```
DECLARE
    PSALARY EMPLOYEES.SALARY%TYPE ;
    CODE NUMBER;
    MESSAGE VARCHAR2(100);
BEGIN
    SELECT SALARY INTO PSALARY
    FROM EMPLOYEES
    WHERE EMPLOYEE_ID = 100 ;
    PSALARY := PSALARY / 0 ;

EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(SQLCODE);
        DBMS_OUTPUT.PUT_LINE(SQLERRM);
        CODE:=SQLCODE;
        MESSAGE:=SQLERRM;
        -- INSERT INTO ERRORS1 VALUES (CODE,MESSAGE,sysdate);
        COMMIT;
END;
```

Resultado de la Consulta x Salida de Script x

Tarea terminada en 0.067 segundos

ORA-01476: el divisor es igual a cero

Procedimiento PL/SQL terminado correctamente.

4- El error -00001 es clave primaria duplicada.

- Aunque ya existe una predefinida (DUP_VAL_ON_INDEX) vamos a crear una excepción no -predefinida que haga lo mismo.
- Vamos a usar la tabla REGIONS para hacerlo más fácil
- Usamos PRAGMA EXCEPTION_INIT y creamos una excepción denominada "duplicado".
- Cuando se genere ese error debemos pintar "Clave duplicada, intente otra".

```
SET SERVEROUTPUT ON
DECLARE
    duplicado EXCEPTION;
    PRAGMA EXCEPTION_INIT(duplicado,-1);

BEGIN
    INSERT INTO REGIONS VALUES (1, 'MARTE AREA 1');

EXCEPTION
    WHEN duplicado THEN
        DBMS_OUTPUT.PUT_LINE('Clave duplicada, intente otra');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('ERROR INDEFINIDO');
END;
```

Resultado de la Consulta x Salida de Script x

Tarea terminada en 0.089 segundos

Clave duplicada, intente otra

Procedimiento PL/SQL terminado correctamente.