

Excepciones en SQL

Integrantes

1. **Diego Armenta Tecuzcano**
2. **Daniel Alberto Zarco Manzanares**

Tipod de excepciones

<i>Excepción</i>	<i>Descripción</i>	<i>Instrucciones de Manejo</i>
Errores Predefinidos de Oracle .	Errores que ocurren con frecuencia en PL/SQL .	No hay necesidad de declarar esta excepciones. Son predefinidas por Oracle y se levantan de forma implícita.
Errores No-Predefinidos de Oracle .	Cualquier otro error estándar del servidor de Oracle.	Es necesario declararlos dentro de la parte declarativa; el servidor Oracle levanta el error de forma implícita, y luego puedes capturarlo en el controlador de excepciones.
Errores definidos por e usuario.	X condición que el programador entiende como anormal.	Deben ser declarados y levantados de forma explícita.

Ejemplos

<i>Excepción</i>	<i>Número de Error en Oracle</i>	<i>Descripción</i>
ACCESS_INTO_NULL	ORA-06530	Intento de asignar valores a los atributos de un objeto no inicializado.
CASE_NOT_FOUND	ORA-06592	No se selecciona ninguna de las opciones en las cláusulas WHEN de una sentencia CASE , y no hay una cláusula ELSE .

Ejemplos

TOO_MANY_ROWS	ORA-01422	Sentencia SELECT retorna mas de una fila.
VALUE_ERROR	ORA-06502	Se produjo un error: aritmético, de conversión, de truncamiento o de restricción de tamaño.
ZERO_DIVIDE	ORA-01476	Se ha intentado dividir por cero.

Excepciones definidas

```
WHEN NO_DATA_FOUND THEN  
    statement1;
```

...

```
WHEN TOO_MANY_ROWS THEN  
    statement1;
```

...

```
WHEN OTHERS THEN  
    statement1;
```

No definidas

DECLARE

e_insert_excep **EXCEPTION**;

PRAGMA EXCEPTION_INIT(*e_insert_excep*, -01400);

BEGIN

INSERT INTO *departments* (*department_id*, *department_name*)

VALUES (280, **NULL**);

EXCEPTION

WHEN *e_insert_excep* **THEN**

DBMS_OUTPUT.PUT_LINE('INSERCIÓN FALLIDA.');

DBMS_OUTPUT.PUT_LINE(**SQLERRM**);

END;

Definidas por el usuario

```
SET SERVEROUTPUT ON
DECLARE
  v_deptno NUMBER := 500;
  v_name VARCHAR2(20) := 'Testing';
  e_invalid_department EXCEPTION;
BEGIN
  UPDATE departments
  SET department_name = v_name
  WHERE department_id = v_deptno;
  IF SQL%NOTFOUND THEN
    RAISE e_invalid_department;
  END IF;
  COMMIT;
EXCEPTION
  WHEN e_invalid_department THEN
    DBMS_OUTPUT.PUT_LINE(v_deptno||' No es un número de departamento Valido.');
```

END;

Ejemplo 1

```
CREATE OR REPLACE FUNCTION montoactual(id_monto integer, monto_abonar integer)
RETURNS FLOAT AS $$
DECLARE
    monto_actual FLOAT;
    monto_nuevo FLOAT;
BEGIN
    SELECT credmonto INTO monto_actual from credito WHERE credid=id_monto;

    monto_nuevo= monto_actual-monto_abonar;

    IF(monto_abonar IS NULL) THEN

        RAISE EXCEPTION USING MESSAGE = 'Se necesita el valor del monto a abonar.';
    ELSIF(id_monto IS NULL) THEN

        RAISE EXCEPTION USING MESSAGE = 'Se necesita el valor del id.';
    ELSE

        UPDATE credito

        SET credmonto=credmonto-monto_abonar

        WHERE

            credid=id_monto;

    END IF;

    RETURN monto_nuevo;
END; $$
LANGUAGE 'plpgsql';
```

```
Command Prompt - psql postgres postgres

postgres=# select * from credito;
 credid | credmonto | credinteres
-----+-----+-----
      82 |    60000 |           85
      12 |    14800 |           40
(2 rows)

postgres=# select montoactual(12,200);
 montoactual
-----
        14600
(1 row)

postgres=# select * from credito;
 credid | credmonto | credinteres
-----+-----+-----
      82 |    60000 |           85
      12 |    14600 |           40
(2 rows)
```

Ejemplo 1, capturando las Exception

Cuando el monto abonar es nulo.

```
postgres=# select * from credito;
 credid | credmonto | credinteres
-----+-----+-----
      82 |    60000 |           85
      12 |    14600 |           40
(2 rows)
```

```
postgres=# select montoactual(12,null);
ERROR:  Se necesita el valor del monto a abonar.
CONTEXT:  PL/pgSQL function montoactual(integer,integer) line 9 at RAISE
postgres=#
```

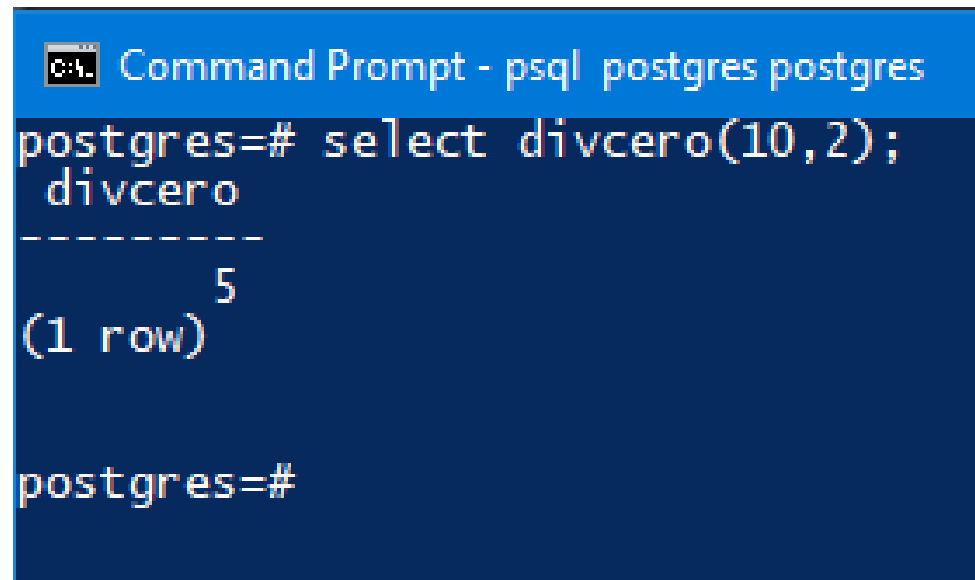
Ejemplo 1, capturando las Exception

Cuando el id del monto es nulo

```
postgres=# select montoactual(null,200);  
ERROR:  Se necesita el valor del id.  
CONTEXT:  PL/pgSQL function montoactual(integer,integer) line 11 at RAISE  
postgres=#
```

Ejemplo 2, división entre dos números

```
CREATE OR REPLACE FUNCTION divcero(num1 integer, num2 integer)
RETURNS FLOAT AS $$
DECLARE
    result FLOAT;
BEGIN
    IF(num2=0)THEN
        RAISE EXCEPTION USING MESSAGE='Error,
división entre cero.';
    ELSE
        result=num1/num2;
    END IF;
    RETURN result;
END; $$
LANGUAGE 'plpgsql';
```



The screenshot shows a Windows Command Prompt window with a blue title bar that reads "Command Prompt - psql postgres postgres". The window has a dark blue background with white text. The text inside shows a PostgreSQL session where the query `select divcero(10,2);` was executed. The output shows a single row with the value 5. The prompt `postgres=#` is visible at the end of the line.

```
postgres=# select divcero(10,2);
divcero
-----
          5
(1 row)

postgres=#
```

Ejemplo 2, capturando el error

Cuando el dividendo es cero

```
postgres=# select divcero(10,0);  
ERROR:  Error, divisi n entre cero.  
CONTEXT:  PL/pgSQL function divcero(integer,integer) line 6 at RAISE  
postgres=#
```

Gracias por su atención