Manejo de errores y excepciones



Salgado Miranda Jorge Sanchez de Santiago Julián

Nosotros



"CTRL"

Jorge Salgado













Errores







Ejemplos



INTRODUCCIÓN



MANEJO DE EXCEPCIONES





ZPARA QUÉ?

Estos elementos permitirán identificar y dar manejo a los problemas los cuales ocurren en la ejecución de un programa y los cuales no se apegan al funcionamiento del mismo.

Diferencias entre errores y excepciones



Errores

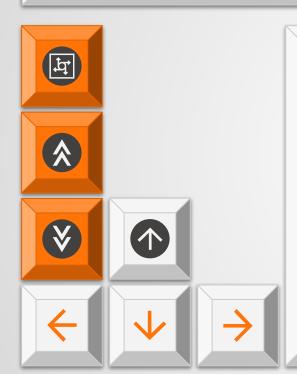
Un error es un problema que ocurre durante la ejecución de una sentencia SQL y que impide que la sentencia se complete correctamente.



Excepciones

Una excepción es un evento que ocurre durante la ejecución de una función o bloque de código que interrumpe el flujo normal de ejecución.

Mensajes de error



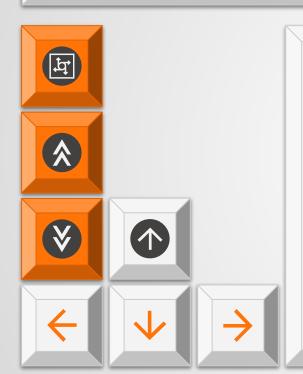
Los mensajes de error son más detallados y específicos que los mensajes de excepción. Los mensajes de error incluyen información sobre la causa del problema y el contexto en el que ocurrió. Los mensajes de excepción son más genéricos y pueden no proporcionar tanta información sobre la causa del problema.

Manejo de errores y excepciones



Los errores se manejan de manera diferente a las excepciones. Los errores se producen en el momento en que se ejecuta una sentencia y pueden ser detectados por el cliente o la aplicación que está realizando la consulta. Las excepciones se producen durante la ejecución de un bloque de código y deben ser manejadas explícitamente mediante la declaración de un bloque de gestión de excepciones.

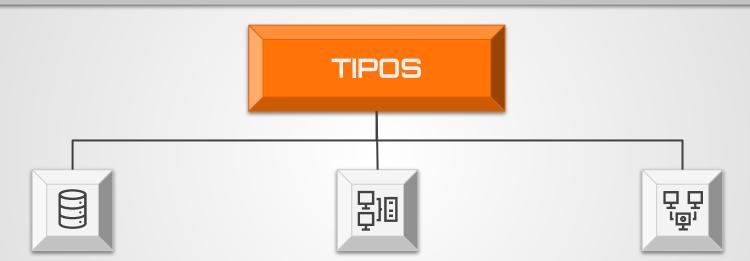
Tipos de errores y excepciones



Existen diferentes tipos de errores y excepciones en PostgreSQL. Los errores incluyen errores de sintaxis, errores de autorización, errores de integridad de datos y errores de ejecución. Las excepciones incluyen excepciones de pl/pgSQL, excepciones de SQL y excepciones de otros lenguajes de programación.

Cómo se producen: Los errores se producen cuando se ejecuta una sentencia y no se puede completar debido a un problema. Las excepciones se producen cuando se ejecuta un bloque de código y se produce un evento inesperado que interrumpe el flujo normal de ejecución.

Tipos de excepciones



PREDEFINIDOS

No es necesario definir estas expresiones para errores frecuentes en PSQL

NO PREDEFINIDOS

Se definen en la parte declarativa, servidor levanta el error y se pueden capturar en el manejador de excepciones

DEFINIDOS POR USUARIO

Se declaran y se levantan de manera explícita por el usuario, para condiciones anormales

Códigos de errores



https://www.post gresql.org/docs/c urrent/errcodes-a ppendix.html

Error Code	Condition Name
Class 00 —	Successful Completion
00000	successful_completion
Class 01 —	Warning
01000	warning





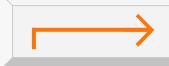


Class 02 — No Data (this is also a warning class per the SQL standard)	
02000	no_data
02001	no_additional_dynamic_result_sets_returned
Class 03 — SQL Statement Not Yet Complete	
03000	sql_statement_not_yet_complete
Class 08 — Connection Exception	
08000	connection_exception

Class 09 — Triggered Action Exception		
09000	triggered_action_exception	
Class 0A — Feature Not Supported		
0A000	feature_not_supported	
Class 0B —	Invalid Transaction Initiation	
0B000	invalid_transaction_initiation	
Class 0F —	Locator Exception	
0F000	locator_exception	
0F001	invalid_locator_specification	
Class 0L — Invalid Grantor		
0L000	invalid_grantor	
0LP01	invalid_grant_operation	
Class 0P — Invalid Role Specification		
0P000	invalid_role_specification	
Class 0Z —	Diagnostics Exception	
0Z000	diagnostics_exception	

Ejemplo Excepciones



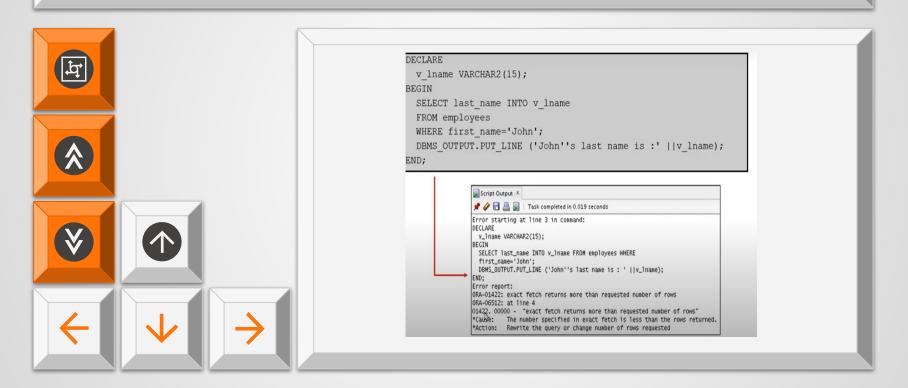




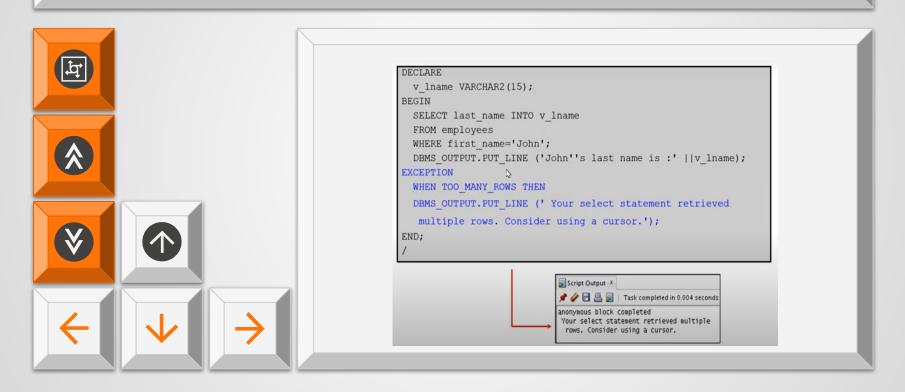




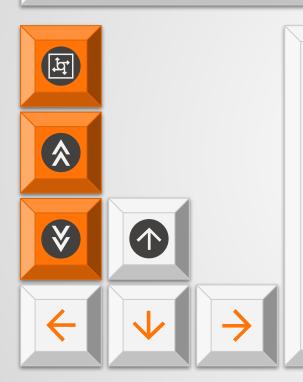
Ejemplo Manejo de Excepción



Ejemplo Manejo de Excepción



Ejemplo manejo de excepciones



```
CREATE [OR REPLACE] PROCEDURE procedure name
   [ (parameter [,parameter]) ]
   [declaration section]
BEGIN
  executable section
EXCEPTION
  WHEN exception name1 THEN
      [statements]
  WHEN exception_name2 THEN
      [statements]
  WHEN exception name n THEN
      [statements]
  WHEN OTHERS THEN
      [statements]
END [procedure name];
```

Funciones para manejo de errores



- SQLCODE: Devuelve el valor numérico para el código de error
- SQLERRM: Devuelve el mensaje asociado con el número de error

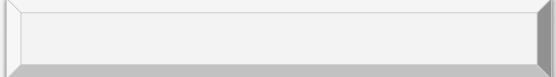
```
DECLARE
error_code NUMBER;
error_message VARCHAR2(255);

BEGIN
...
EXCEPTION
...
WHEN OTHERS THEN
ROLLBACK;
error_code := SQLCODE;
error_message := SQLERRM;
INSERT INTO errors (e_user, e_date, error_code,
error_message) VALUES(USER, SYSDATE, error_code,
error_message);
END;
/
```





Conclusiones



Conclusiones



Para concluir, es importante tener en cuenta que las excepciones y errores en PostgreSQL son una parte integral del manejo de bases de datos. Es esencial saber cómo gestionar y controlar estos eventos para garantizar la integridad y consistencia de la base de datos. Algunas de las principales conclusiones a tener en cuenta son las siguientes:



"Un error es simplemente una oportunidad para hacer las cosas mejor en el futuro"

-Anonymous

