Reporting Messages

Reporting Messages

RAISE is used to raise errors and report messages, PostgreSQL provides various parameters to report an error, warning, and information at a detailed level. Below is the basic syntax for the RAISE command.

`raise level format;`

- `level`
  - `debug`
  - `log`
  - `notice`
  - `info`
  - `warning`
  - `exception(default)`
- `format`
  - `%`

FUSE Github

Note: raise statement uses the exception level by default.

Raising errors

Besides raising an error, you can add more information by using the following additional clause:

`raise exception ‘message’` *`using option = expression;`*

- `option`
  - `message:` set error message
  - `hint:` provide the hint message so that the root cause of the error is easier to be discovered.
  - `detail:` give detailed information about the error.
  - `errcode:` identify the error code, which can be either by condition name or directly five-character SQLSTATE code.

- `raise sqlstate ‘five-character’;`

sqlstate documentation

Trap Errors

By default, any error occurring in a PL/pgSQL function aborts execution of the function and the surrounding transaction. You **can trap errors** and recover from them by using a BEGIN block with an **EXCEPTION clause**.

```
[ <<label>> ]
[ DECLARE
    declarations ]
BEGIN
    statements
EXCEPTION
    WHEN condition [ OR condition ... ] THEN
        handler_statements
    [ WHEN condition [ OR condition ... ] THEN
          handler_statements
      ... ]
    WHEN OTHERS THEN
          handler_statements
END;
```

```
WHEN division_by_zero THEN ...
WHEN SQLSTATE '22012' THEN ...
```

A block containing an EXCEPTION clause is significantly more expensive to enter and exit than a block without one. Therefore, don't use EXCEPTION without need

Postgresql documentation

Obtaining Information About An Error

Within an exception handler, one may also retrieve information about the current exception by using the **GET STACKED DIAGNOSTICS** command.

```
GET STACKED DIAGNOSTICS variable { = | := } item [ , ... ];
```

[Postgresql documentation]

Assert Statement

The assert statement is a useful shorthand for inserting debugging checks into **PL/pgSQL** code(**Since postgreSQL 9.5**).

```
assert condition [, message];
show plpgsql.check_asserts;
set plpgsql.check_asserts = [on/off];
```

Note: you should use the assert statement solely for detecting bugs, not for reporting. To report a message or an error, you use the raise statement instead.

Interview Questions