Databases

PL/pgSQL – Exceptions, Procedures and Functions

Marco Vieira

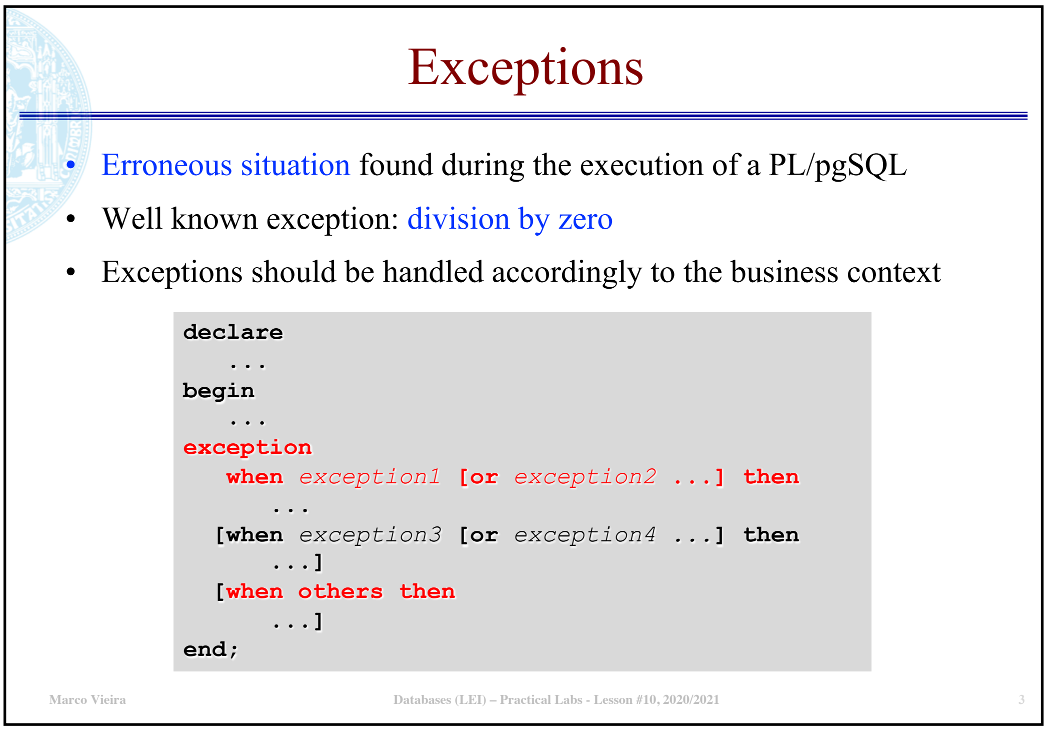
Bachelor in Informatics Engineering
Department of Informatics Engineering
University of Coimbra
2020/2021

1

Outline

- Exceptions
- Procedures
- Functions

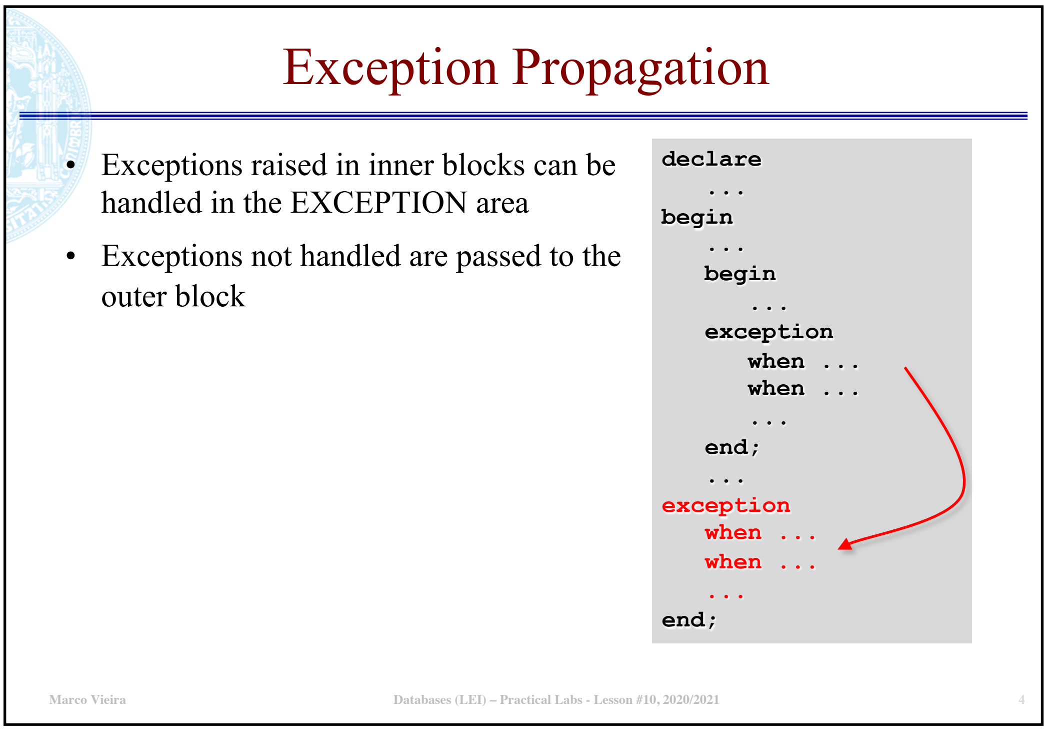
2



Exceptions

- Erroneous situation found during the execution of a PL/pgSQL
- Well known exception: division by zero
- Exceptions should be handled accordingly to the business context

```
declare
    ...
begin
    ...
exception
    when exception1 [or exception2 ...] then
        ...
   [when exception3 [or exception4 ...] then
        ...]
   [when others then
        ...]
end;
```



Exception Propagation

- Exceptions raised in inner blocks can be handled in the EXCEPTION area
- Exceptions not handled are passed to the outer block

```
declare
    ...
begin
    ...
    begin
        ...
    exception
        when ...
        when ...
        ...
    end;
    ...
exception
    when ...
    when ...
    ...
end;
```

Examples of Exceptions

- no_data_found
- too_many_rows
- division_by_zero
- invalid_cursor_state
- no_active_sql_transaction
- insufficient_privilege
- unique_violation
- foreign_key_violation
- …

Example of Exception Handling

```
do $$
declare
   v_dep dep%ROWTYPE;
begin
   ...
   select * into strict v_dep
   from dep
   where ndep = 100;
   ...
exception
   when no_data_found then
      insert into dep
      values(v_deptno, 'Sales', 'Coimbra');
   when too_many_rows then
       insert into errors(cod, mens, data)
       values('-1','Duplicate departments',current_date);
end;
$$;
```

SQLSTATE and SQLERRM

- Local variables blocks with EXCEPTION clause
  - SQLSTATE: error code
  - SQLERRM: error message

```
do $$
declare
   ...
begin
   ...
exception
   when others then
       insert into errors(cod, mens, data)
       values(sqlstate,sqlerrm,current_date);
end;
$$;
```

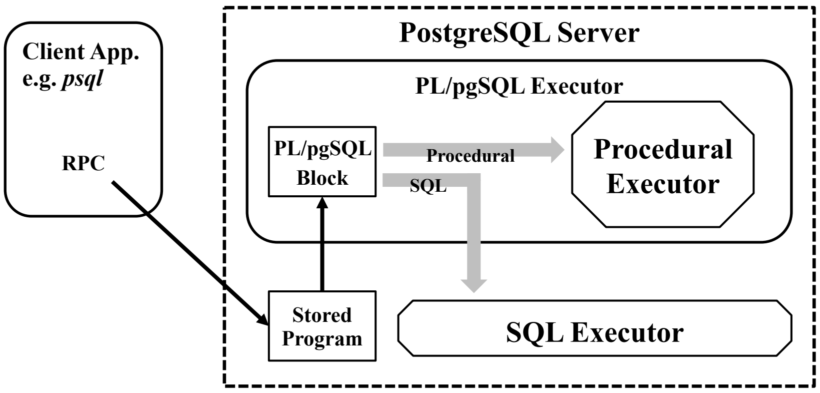
Raising Exceptions

```
raise [level] 'format' [, expression [, ... ]] [using option = expression [, ... ] ];
raise [level] condition_name [ using option = expression [, ... ] ];
raise [level] sqlstate 'sqlstate' [using option = expression [, ... ] ];
raise [level] using option = expression [, ... ];
raise ;
```

- Use the RAISE statement to report messages and raise errors
- *level*:
  - DEBUG, LOG, INFO, NOTICE, WARNING, EXCEPTION
- *option*:
  - MESSAGE, DETAIL, HINT, ERRCODE
  - COLUMN, CONSTRAINT, DATATYPE, TABLE, SCHEMA

```
raise exception 'Nonexistent ID: %', user_id
      using hint = 'Please check your user ID';
```

Procedures & Functions



call add_student('António Silva', 23212, ...);

Procedures

```
create [or replace] procedure procname ([arg1[, arg2,...]])
language plpgsql
as $$
declare
    -- declarations
begin
    -- actions
[exception
     -- exception handling]
end;
```

- Arguments:
  - [*argmode*] [*argname*] *argtype* [{ DEFAULT | = } *default_expr*]
- *argmode* can be:
  - IN: input arguments
  - INOUT: input/output arguments
  - VARIADIC: to accept a variable numbers of arguments (arrays only)

Procedures: Example

-
```
create or replace procedure proc1(a integer)
language plpgsql
as $$
<<proc1>>
begin
  insert into mytab values (a);
exception
  when others then
    raise exception 'error';
end;
$$;
```

```
call proc1(5);
```

```
drop procedure proc1;
```

Create and Execute & Functions

```
create [or replace] function funcname ([arg1[, arg2,...]])
returns datatype
language plpgsql
as $$
declare
    -- declarations
begin
    -- actions
    return value;
[exception
      -- exception handling]
end;
```

- Only input (IN) arguments
- Result is returned using the RETURN instruction

Functions: Example

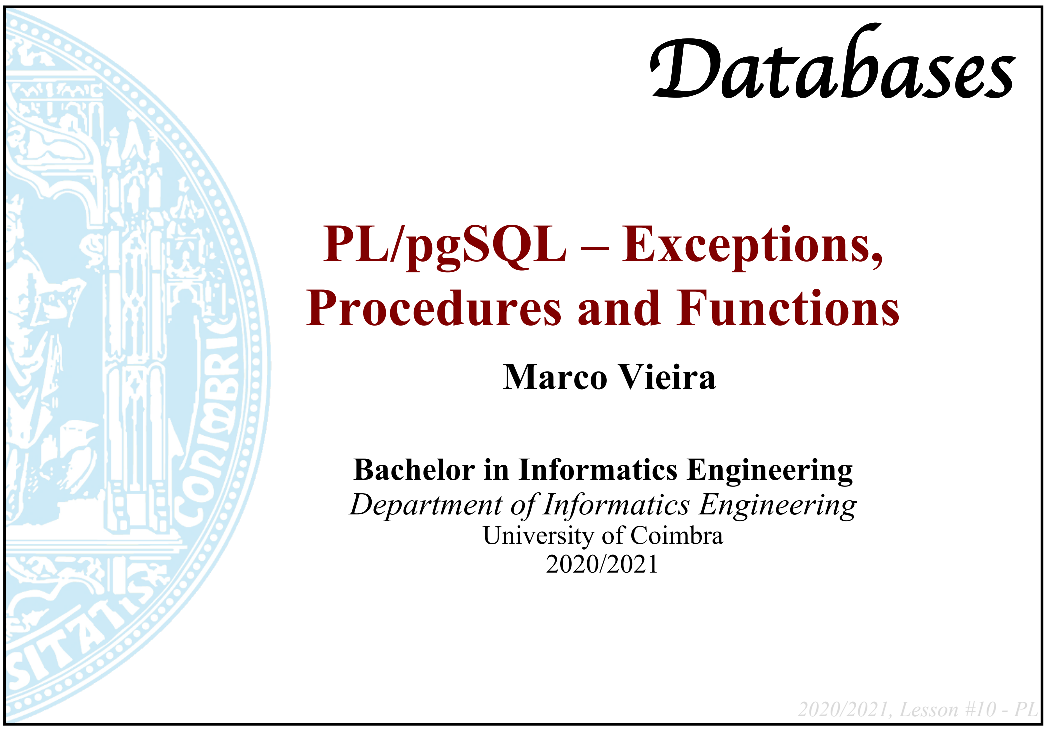
```
create or replace function func1() returns integer
language plpgsql
as $$
declare
  x integer;
begin
  select sum(col) into x from mytab;
  return(x);
exception
  when others then
    raise exception 'error';
end;
$$;
```

```
select func1();
```

```
var:=func1()
```

```
drop function func1();
```

Q&A



Databases

PL/pgSQL – Exceptions, Procedures and Functions

Marco Vieira

Bachelor in Informatics Engineering
Department of Informatics Engineering
University of Coimbra
2020/2021

15