```
********* QUERY **********
CREATE OR REPLACE FUNCTION my_sum(x integer, y integer)
RETURNS integer AS
$$
        SELECT x + y;
$$
LANGUAGE SQL;
**************************

CREATE FUNCTION
********* QUERY **********
SELECT my_sum(4, 6);
**************************

 my_sum
--------
     10
(1 row)

********* QUERY **********
DROP FUNCTION my_sum;
**************************

DROP FUNCTION
********* QUERY **********
DROP TABLE IF EXISTS temp_posts;
**************************

DROP TABLE
********* QUERY **********
CREATE TEMP TABLE IF NOT EXISTS temp_posts AS
SELECT * FROM posts;
**************************

SELECT 5
********* QUERY **********
CREATE OR REPLACE FUNCTION delete_posts(p_title text)
RETURNS SETOF integer AS
$$
        DELETE FROM temp_posts WHERE title = p_title RETURNING pk;
$$
LANGUAGE SQL;
**************************

CREATE FUNCTION
********* QUERY **********
SELECT pk, title FROM temp_posts ORDER BY pk;
**************************

 pk |     title
----+---------------
  5 | my orange
  6 | my new apple
  7 | Re:my orange
  9 | my new orange
 11 | my tomato
(5 rows)

********* QUERY **********
SELECT delete_posts('my tomato');
```

```
**************************

 delete_posts
--------------
           11
(1 row)

********* QUERY **********
SELECT pk, title FROM temp_posts ORDER BY pk;
**************************

 pk |     title
----+---------------
  5 | my orange
  6 | my new apple
  7 | Re:my orange
  9 | my new orange
(4 rows)

********* QUERY **********
DROP TABLE IF EXISTS temp_posts;
**************************

DROP TABLE
********* QUERY **********
CREATE TEMP TABLE IF NOT EXISTS temp_posts AS
SELECT * FROM posts;
**************************

SELECT 5
********* QUERY **********
DROP FUNCTION delete_posts;
**************************

DROP FUNCTION
********* QUERY **********
CREATE OR REPLACE FUNCTION delete_posts(p_title text)
RETURNS TABLE (ret_key integer, ret_title text) AS
$$
        DELETE FROM temp_posts WHERE title = p_title RETURNING pk, title;
$$
LANGUAGE SQL;
**************************

CREATE FUNCTION
********* QUERY **********
SELECT pk, title FROM temp_posts ORDER BY pk;
**************************

 pk |     title
----+---------------
  5 | my orange
  6 | my new apple
  7 | Re:my orange
  9 | my new orange
 11 | my tomato
(5 rows)

********* QUERY **********
SELECT * FROM delete_posts('my tomato');
```

```
**************************

 ret_key | ret_title
---------+-----------
      11 | my tomato
(1 row)

********* QUERY **********
SELECT pk, title FROM temp_posts ORDER BY pk;
**************************

 pk |     title
----+---------------
  5 | my orange
  6 | my new apple
  7 | Re:my orange
  9 | my new orange
(4 rows)

********* QUERY **********
DROP TABLE IF EXISTS temp_posts;
**************************

DROP TABLE
********* QUERY **********
DROP FUNCTION delete_posts;
**************************

DROP FUNCTION
********* QUERY **********
CREATE OR REPLACE FUNCTION nvl(anyelement, anyelement)
RETURNS anyelement AS
$$
        SELECT coalesce($1, $2);
$$
LANGUAGE SQL;
**************************

CREATE FUNCTION
********* QUERY **********
SELECT nvl(null::integer, 1);
**************************

 nvl
-----
   1
(1 row)

********* QUERY **********
SELECT nvl(''::text, 'n'::text);
**************************

 nvl
-----

(1 row)

********* QUERY **********
SELECT nvl('a'::text, 'n'::text);
**************************
```

```
 nvl
-----
 a
(1 row)

********* QUERY **********
DROP FUNCTION nvl;
**************************

DROP FUNCTION
********* QUERY **********
CREATE OR REPLACE FUNCTION my_sum(x integer, y integer)
RETURNS integer AS
$$
        DECLARE
                ret integer;
        BEGIN
                ret := x + y;
                RETURN ret;
        END;
$$
LANGUAGE 'plpgsql';
**************************

CREATE FUNCTION
********* QUERY **********
SELECT my_sum(4, 6);
**************************

 my_sum
--------
     10
(1 row)

********* QUERY **********
DROP FUNCTION IF EXISTS my_sum;
**************************

DROP FUNCTION
********* QUERY **********
CREATE OR REPLACE FUNCTION my_sum(integer, integer)
RETURNS integer AS
$$
        DECLARE
                x alias for $1;
                y alias for $2;
                ret integer;
        BEGIN
                ret := x + y;
                RETURN ret;
        END;
$$
LANGUAGE 'plpgsql';
**************************

CREATE FUNCTION
********* QUERY **********
SELECT my_sum(7, 11);
**************************
```

```
 my_sum
--------
     18
(1 row)

********* QUERY **********
DROP FUNCTION IF EXISTS my_sum;
**************************

DROP FUNCTION
********* QUERY **********
CREATE OR REPLACE FUNCTION my_sum(integer, integer)
RETURNS integer AS
$$
        DECLARE
                ret integer;
        BEGIN
                ret := $1 + $2;
                RETURN ret;
        END;
$$
LANGUAGE 'plpgsql';
**************************

CREATE FUNCTION
********* QUERY **********
SELECT my_sum(15, 16);
**************************

 my_sum
--------
     31
(1 row)

********* QUERY **********
DROP FUNCTION IF EXISTS my_sum;
**************************

DROP FUNCTION
********* QUERY **********
CREATE OR REPLACE FUNCTION my_sum_3_params(
        IN x integer,
        IN y integer,
        OUT z integer) AS
$$
        BEGIN
                z := x + y;
        END;
$$
LANGUAGE 'plpgsql';
**************************

CREATE FUNCTION
********* QUERY **********
SELECT my_sum_3_params(121, 121);
**************************

 my_sum_3_params
-----------------
```

```
             242
(1 row)

********* QUERY **********
DROP FUNCTION IF EXISTS my_sum_3_params;
**************************

DROP FUNCTION
********* QUERY **********
CREATE OR REPLACE FUNCTION my_sum_mul(
        IN x integer,
        IN y integer,
        OUT z integer,
        OUT w integer) AS
$$
        BEGIN
                z := x + y;
                w := x * y;
        END;
$$
LANGUAGE 'plpgsql';
**************************

CREATE FUNCTION
********* QUERY **********
SELECT my_sum_mul(6, 6);
**************************

 my_sum_mul
------------
 (12,36)
(1 row)

********* QUERY **********
SELECT * FROM my_sum_mul(6, 6);
**************************

 z  | w
----+----
 12 | 36
(1 row)

********* QUERY **********
SELECT * FROM my_sum_mul(6, 6) WHERE z = 12;
**************************

 z  | w
----+----
 12 | 36
(1 row)

********* QUERY **********
DROP FUNCTION IF EXISTS my_sum_mul;
**************************

DROP FUNCTION
********* QUERY **********
CREATE OR REPLACE FUNCTION my_check(x integer, y integer)
RETURNS text AS
$$
```

```
        BEGIN
                IF x > y THEN
                        RETURN 'the first parameter is greater than the sec
ond one';
                ELSIF y > x THEN
                        RETURN 'the second parameter is greater than the fi
rst one';
                ELSE
                        RETURN 'the two parameters are equal';
                END IF;
        END;
$$
LANGUAGE 'plpgsql';
**************************

CREATE FUNCTION
********* QUERY **********
SELECT my_check(1, 2);
**************************

                      my_check
-----------------------------------------------------
 the second parameter is greater than the first one
(1 row)

********* QUERY **********
SELECT my_check(2, 1);
**************************

                      my_check
-----------------------------------------------------
 the first parameter is greater than the second one
(1 row)

********* QUERY **********
SELECT my_check(1, 1);
**************************

           my_check
-------------------------------
 the two parameters are equal
(1 row)

********* QUERY **********
DROP FUNCTION IF EXISTS my_check;
**************************

DROP FUNCTION
********* QUERY **********
CREATE OR REPLACE FUNCTION my_check_value(x integer DEFAULT 0)
RETURNS text AS
$$
        BEGIN
                CASE x
                        WHEN 1 THEN RETURN 'value = 1';
                        WHEN 2 THEN RETURN 'value = 2';
                        ELSE RETURN 'value >= 3';
                END CASE;
        END;
$$
```

```
LANGUAGE 'plpgsql';
**************************

CREATE FUNCTION
********* QUERY **********
SELECT my_check_value(1);
**************************

 my_check_value
----------------
 value = 1
(1 row)

********* QUERY **********
SELECT my_check_value(2);
**************************

 my_check_value
----------------
 value = 2
(1 row)

********* QUERY **********
SELECT my_check_value(5);
**************************

 my_check_value
----------------
 value >= 3
(1 row)

********* QUERY **********
DROP FUNCTION IF EXISTS my_check_value;
**************************

DROP FUNCTION
********* QUERY **********
CREATE OR REPLACE FUNCTION my_check_case(x integer DEFAULT 0, y integer DEF
AULT 0)
RETURNS text AS
$$
        BEGIN
                CASE
                        WHEN x > y THEN RETURN 'the first parameter is grea
ter than the second one';
                        WHEN y > x THEN RETURN 'the second parameter is gre
ater than the first one';
                        ELSE RETURN 'the two parameters are equal';
                END CASE;
        END;
$$
LANGUAGE 'plpgsql';
**************************

CREATE FUNCTION
********* QUERY **********
SELECT my_check_case(2, 1);
**************************

                    my_check_case
```

```
-------------------------------------------------
 the first parameter is greater than the second one
(1 row)

********* QUERY **********
SELECT my_check_case(1, 2);
**************************

                  my_check_case
-------------------------------------------------
 the second parameter is greater than the first one
(1 row)

********* QUERY **********
SELECT my_check_case(1, 1);
**************************

        my_check_case
------------------------------
 the two parameters are equal
(1 row)

********* QUERY **********
SELECT my_check_case();
**************************

        my_check_case
------------------------------
 the two parameters are equal
(1 row)

********* QUERY **********
DROP FUNCTION IF EXISTS my_check_case;
**************************

DROP FUNCTION
********* QUERY **********
CREATE TYPE my_type AS (
        id integer,
        title text,
        record_data hstore);
**************************

CREATE TYPE
********* QUERY **********
CREATE OR REPLACE FUNCTION my_first_func(p_id integer)
RETURNS SETOF my_type AS
$$
        DECLARE
                rw posts%ROWTYPE;
                ret my_type;
        BEGIN
                FOR rw IN SELECT * FROM posts WHERE pk = p_id LOOP
                        ret.id = rw.pk;
                        ret.title = rw.title;
                        ret.record_data = hstore(ARRAY['title', rw.title, '
title and content', FORMAT('%s %s', rw.title, rw.content)]);
                        RETURN NEXT ret;
                END LOOP;
        END;
```

```
$$
LANGUAGE 'plpgsql';
**************************

CREATE FUNCTION
********* QUERY **********
CREATE OR REPLACE FUNCTION my_second_func(p_id integer)
RETURNS SETOF my_type AS
$$
        DECLARE
                rw record;
                ret my_type;
        BEGIN
                FOR rw IN SELECT * FROM posts WHERE pk = p_id LOOP
                        ret.id = rw.pk;
                        ret.title = rw.title;
                        ret.record_data = hstore(ARRAY['title', rw.title, '
title and content', FORMAT('%s %s', rw.title, rw.content)]);
                        RETURN NEXT ret;
                END LOOP;
        END;
$$
LANGUAGE 'plpgsql';
**************************

CREATE FUNCTION
********* QUERY **********
SELECT * FROM my_first_func(11);
**************************

 id |   title   |                                                record_data

----+-----------+-----------------------------------------------------------
-----------------------------------------
 11 | my tomato | "title"=>"my tomato", "title and content"=>"my tomato my
tomato is the best orange in the world"
(1 row)

********* QUERY **********
SELECT * FROM my_second_func(11);
**************************

 id |   title   |                                                record_data

----+-----------+-----------------------------------------------------------
-----------------------------------------
 11 | my tomato | "title"=>"my tomato", "title and content"=>"my tomato my
tomato is the best orange in the world"
(1 row)

********* QUERY **********
DROP FUNCTION IF EXISTS my_first_func;
**************************

DROP FUNCTION
********* QUERY **********
DROP FUNCTION IF EXISTS my_second_func;
**************************

DROP FUNCTION
```

```
********* QUERY **********
DROP TYPE IF EXISTS my_type;
**************************

DROP TYPE
********* QUERY **********
CREATE OR REPLACE FUNCTION my_first_except(x numeric, y numeric)
RETURNS real AS
$$
        DECLARE
                ret real;
        BEGIN
                ret := x / y;
                RETURN ret;
        EXCEPTION
                WHEN division_by_zero THEN
                        RAISE INFO 'DIVISION BY ZERO';
                        RAISE INFO 'Error % %', SQLSTATE, SQLERRM;
                        RETURN 0;
        END;
$$
LANGUAGE 'plpgsql';
**************************

CREATE FUNCTION
********* QUERY **********
SELECT my_first_except(4, 2);
**************************

 my_first_except
-----------------
               2
(1 row)

********* QUERY **********
SELECT my_first_except(4, 0);
**************************

 my_first_except
-----------------
               0
(1 row)

********* QUERY **********
DROP FUNCTION IF EXISTS my_first_except;
**************************

DROP FUNCTION
```