DBMS PRACTICE:

1.

CREATE FUNCTION sum\_of\_two\_numbers(m integer, n integer)

RETURNS integer AS $$

BEGIN

RETURN m + n;

END;

$$ LANGUAGE plpgsql;

2.

CREATE FUNCTION FUN\_TO\_TEST () RETURNS double precision

AS $TEST$

BEGIN

RETURN 4.295806896E-29;

END;

$TEST$ LANGUAGE PLPGSQL

3.

CREATE FUNCTION TODAY\_IS () RETURNS CHAR(22) AS '

BEGIN

RETURN ''Today''''is '' || CAST(CURRENT\_DATE AS CHAR(10));

END;

'

LANGUAGE PLPGSQL

4.

CREATE FUNCTION FUN\_TO\_TEST(dt DATE, ing INTEGER)

RETURNS DATE AS $test$

DECLARE ss ALIAS FOR dt;

ff ALIAS FOR ing;

BEGIN

RETURN ss + ff \* INTERVAL '2 DAY';

END;

$test$

LANGUAGE PLPGSQL

5.

CREATE FUNCTION get\_employee(text) RETURNS text AS '

DECLARE

frst\_name ALIAS FOR $1;

lst\_name employees.last\_name%TYPE;

BEGIN

SELECT INTO lst\_name last\_name FROM employees

WHERE first\_name = frst\_name;

return frst\_name || '' '' || lst\_name;

END;

' LANGUAGE 'plpgsql';

SELECT get\_employee('John');

get\_first\_name

----------------

John Chen

(1 row)

6.

CREATE FUNCTION get\_employee (integer) RETURNS text AS '

DECLARE

emp\_id ALIAS FOR $1;

found\_employee employees%ROWTYPE;

BEGIN

SELECT INTO found\_employee \* FROM employees WHERE employee\_id = emp\_id;

RETURN found\_employee.first\_name || '' '' || found\_employee.last\_name;

END;

' LANGUAGE 'plpgsql';

SELECT get\_employee(108);

get\_employee

-----------------

Nancy Greenberg

(1 row)

7.

Table: “JOBS”

job\_id | job\_title | min\_salary | max\_salary

------------+---------------------------------+------------+------------

AD\_PRES | President | 20000 | 40000

AD\_VP | Administration Vice President | 15000 | 30000

AD\_ASST | Administration Assistant | 3000 | 6000

FI\_MGR | Finance Manager | 8200 | 16000

FI\_ACCOUNT | Accountant | 4200 | 9000

AC\_MGR | Accounting Manager | 8200 | 16000

Create table Jobs( job\_id varchar(20), job\_title varchar(20), min\_sal int, max\_sal int)

CREATE FUNCTION not\_equal(maxa decimal, minb decimal)

RETURNS boolean AS $$

BEGIN

RETURN maxa <> minb;

END;

$$ LANGUAGE plpgsql;

SELECT not\_equal(max\_salary, min\_salary) FROM jobs;

not\_equal

-----------

t

t

8.

CREATE FUNCTION not\_equal(fst\_number integer, snd\_number integer)

RETURNS boolean AS $$

DECLARE

fstnum integer := fst\_number;

sndname integer:= snd\_number;

BEGIN

RETURN fstnum<> sndname;

END;

$$ LANGUAGE plpgsql;

SELECT not\_equal(15, 20);

……………………

9.

CREATE OR REPLACE FUNCTION demo ()

RETURNS text AS $$

BEGIN

PERFORM (SELECT \* FROM mytable);

RETURN ’OK’;

END;

$$ LANGUAGE plpgsql;

10.

CREATE OR REPLACE FUNCTION ifstat (date)

RETURNS text

AS

$$

BEGIN

IF EXTRACT(DAY FROM current\_date) = 1

THEN

RETURN '1st day of the Month';

ELSE

RETURN 'Other day';

END IF;

END;

$$

LANGUAGE plpgsql;

11.

SELECT salary,

CASE WHEN department\_id =90 THEN 'High Salary'

WHEN department\_id =100 THEN '2nd grade salary'

ELSE 'Low Salary'

END

AS salary\_status

FROM employees

LIMIT 15;

salary | salary\_status

----------+------------------

24000.00 | High Salary

17000.00 | High Salary

17000.00 | High Salary

BUILTIN FUNCTION:

1.

SELECT translate ('translate', 'rnlt', '123');

translate

-----------

1a2s3ae

(1 row)

2.

SELECT replace ('test string', 'st', '\*\*');

replace

-------------

te\*\* \*\*ring

(1 row)