**Section9**

CREATE [OR REPLACE] FUNCTION function\_name

[(parameter1 [mode1] datatype1, ...)]

RETURN datatype IS|AS

[local\_variable\_declarations; …]

BEGIN

-- actions;

RETURN expression;

END [function\_name];

CREATE OR REPLACE FUNCTION get\_sal

(p\_id IN employees.employee\_id%TYPE)

RETURN NUMBER IS

v\_sal employees.salary%TYPE := 0;

BEGIN

SELECT salary

INTO v\_sal

FROM employees

WHERE employee\_id = p\_id;

RETURN v\_sal;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN NULL; -- or you could raise a custom exception or handle it in another way

WHEN TOO\_MANY\_ROWS THEN

RAISE\_APPLICATION\_ERROR(-20001, 'More than one row found for the given employee\_id');

END get\_sal;

A screenshot of a computer

Description automatically generated

CREATE OR REPLACE FUNCTION get\_sal

(p\_id IN employees.employee\_id%TYPE) RETURN NUMBER IS

v\_sal employees.salary%TYPE := 0;

BEGIN

SELECT salary INTO v\_sal

FROM employees WHERE employee\_id = p\_id;

RETURN v\_sal;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN RETURN NULL;

END get\_sal;

A screenshot of a computer

Description automatically generated

DECLARE v\_sal employees.salary%type;

BEGIN

v\_sal := get\_sal(100); ...

END;

CREATE OR REPLACE FUNCTION valid\_dept

(p\_dept\_no departments.department\_id%TYPE)

RETURN BOOLEAN IS v\_valid VARCHAR2(1);

BEGIN

SELECT 'x‘ INTO v\_valid

FROM departments

WHERE department\_id = p\_dept\_no;

RETURN(true);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN RETURN(false);

WHEN OTHERS THEN NULL;

END;

BEGIN …

IF valid\_dept(v\_departmentid) THEN

-- this was a valid department, so we’ll do this

part of the code, e.g. an insert into employees

ELSE

-- valid\_dept returned a false, so we are not

doing the insert

END IF;

…

END;

DECLARE v\_today DATE;

BEGIN

v\_today := SYSDATE; ...

END;

CREATE [OR REPLACE] PROCEDURE name [parameters] IS|AS

(Mandatory)

Variables, cursors, etc. (Optional)

BEGIN (Mandatory)

SQL and PL/SQL statements;

EXCEPTION (Optional)

WHEN exception-handling actions;

END [name]; (Mandatory)

CREATE [OR REPLACE] FUNCTION name [parameters] (Mandatory)

RETURN datatype IS|AS (Mandatory)

Variables, cursors, etc. (Optional)

BEGIN (Mandatory)

SQL and PL/SQL statements;

RETURN ...; (One Mandatory, more optional)

EXCEPTION (Optional)

WHEN exception-handling actions;

END [name]; (Mandatory)

CREATE OR REPLACE FUNCTION tax(p\_value IN NUMBER)

RETURN NUMBER IS

BEGIN

RETURN (p\_value \* 0.08);

END tax;

SELECT employee\_id, last\_name, salary, tax(salary)

FROM employees

WHERE department\_id = 50;

SELECT employee\_id, tax(salary)

FROM employees

WHERE tax(salary) > (SELECT MAX(tax(salary))

FROM employees

WHERE department\_id = 20)

ORDER BY tax(salary) DESC;

DESCRIBE ALL\_TABLES

A screenshot of a computer

Description automatically generated

SELECT table\_name, owner FROM ALL\_TABLES;

A screenshot of a computer

Description automatically generated

SELECT object\_type, object\_name FROM USER\_OBJECTS;

A screenshot of a computer

Description automatically generated

SELECT object\_type, COUNT(\*) FROM USER\_OBJECTS GROUP BY object\_type;

A screenshot of a computer

Description automatically generated

SELECT COUNT(\*) FROM DICT WHERE table\_name LIKE 'USER%';

A screenshot of a computer

Description automatically generated

SELECT \* FROM DICT WHERE table\_name LIKE 'USER%IND%';

A screenshot of a computer

Description automatically generated

DROP {PROCEDURE procedure\_name | FUNCTION function\_name}

SELECT text

FROM USER\_SOURCE

WHERE name = 'TAX'

ORDER BY line;

A black and white screen

Description automatically generated

CREATE OR REPLACE PACKAGE check\_emp\_pkg

IS

g\_max\_length\_of\_service CONSTANT NUMBER := 100;

PROCEDURE chk\_hiredate

(p\_date IN employees.hire\_date%TYPE);

PROCEDURE chk\_dept\_mgr

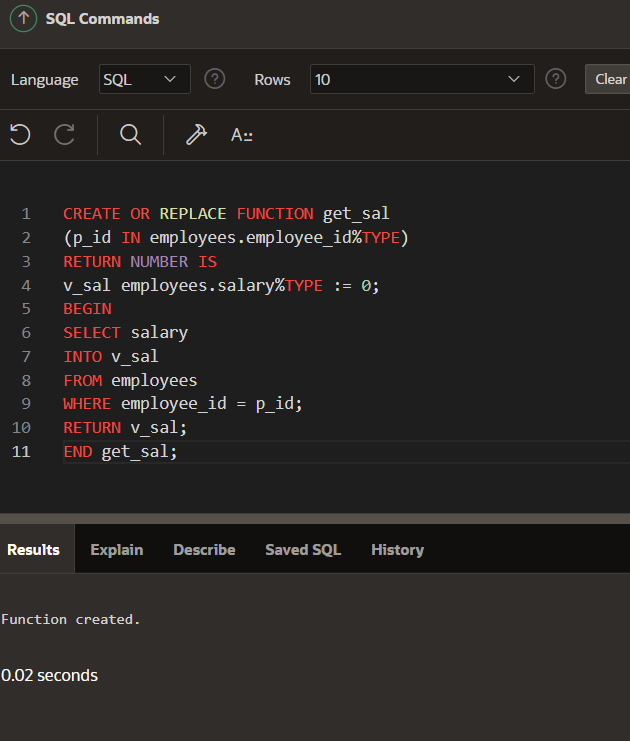
(p\_empid IN employees.employee\_id%TYPE,

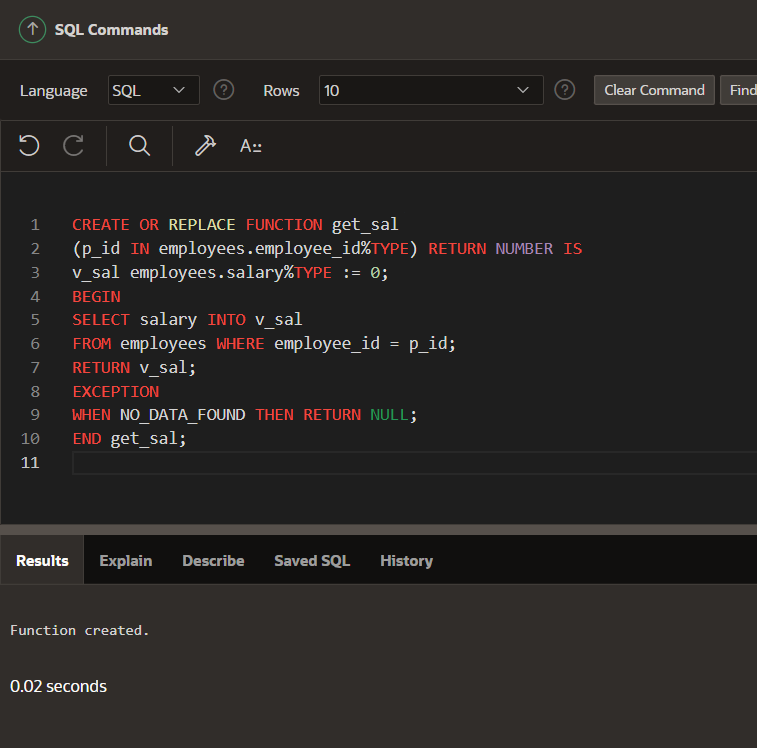
p\_mgr IN employees.manager\_id%TYPE);

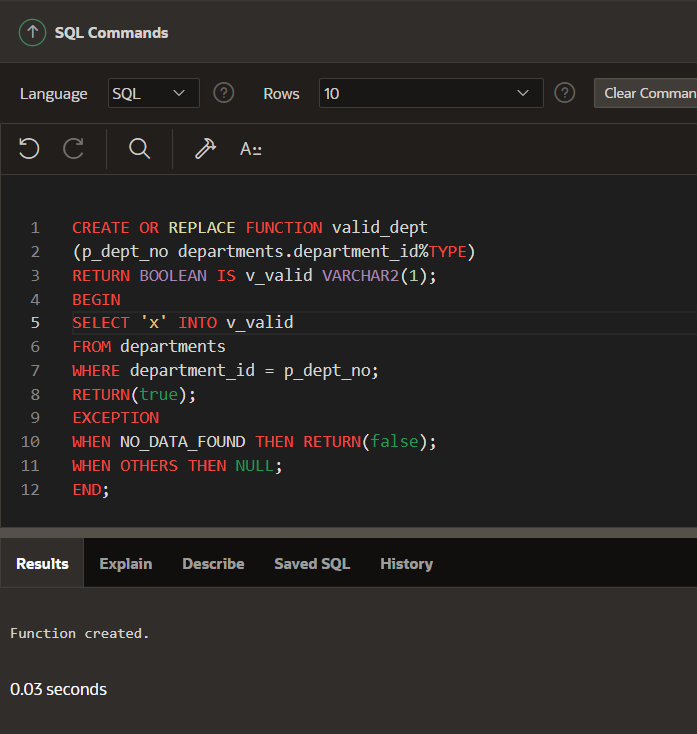
END check\_emp\_pkg;

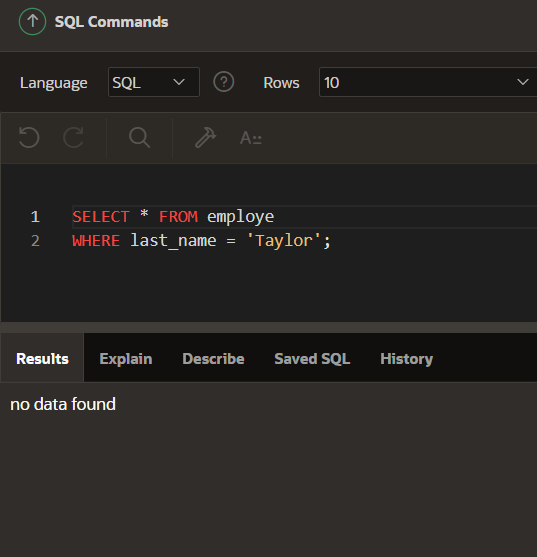
A screenshot of a computer

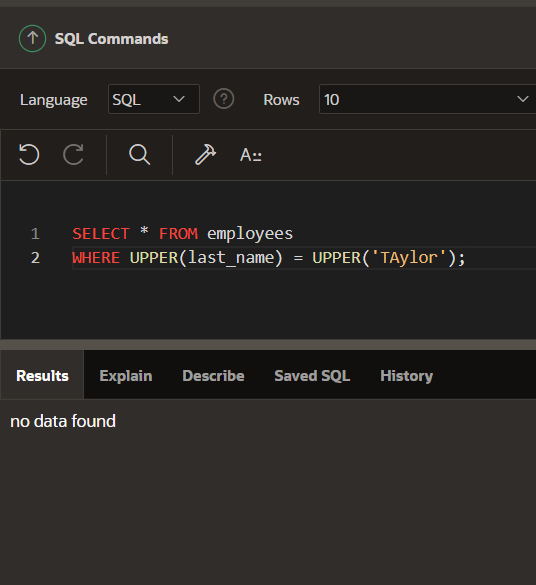
Description automatically generated

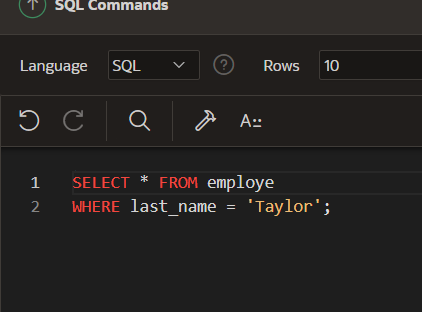
****

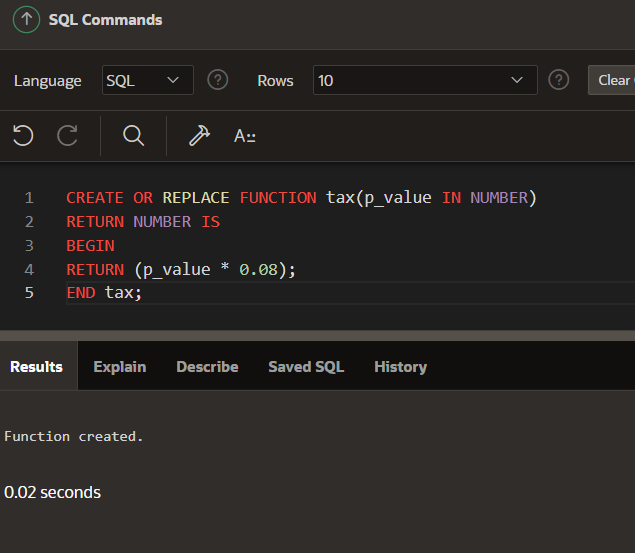
****

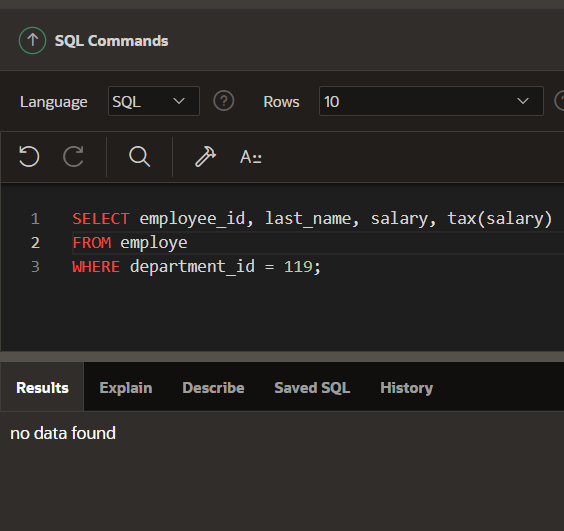
****

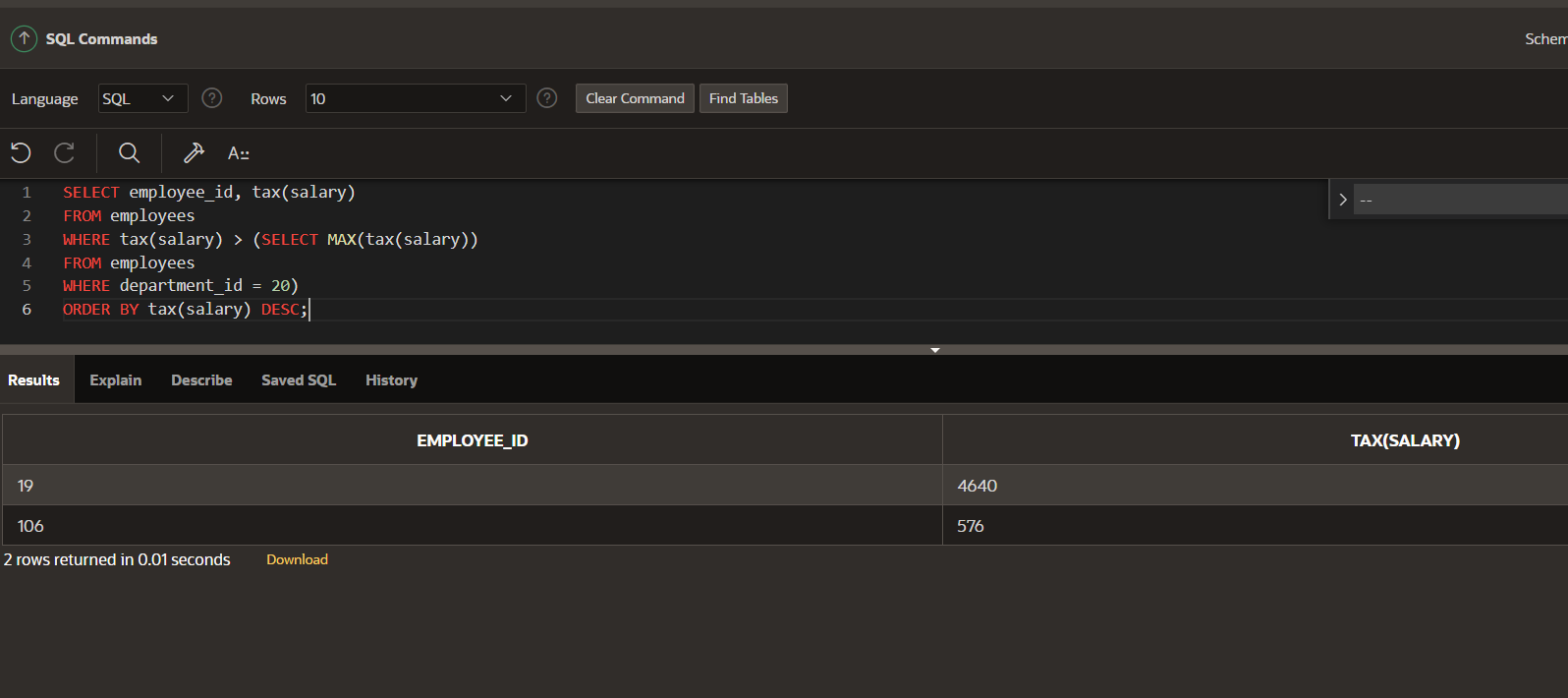
****

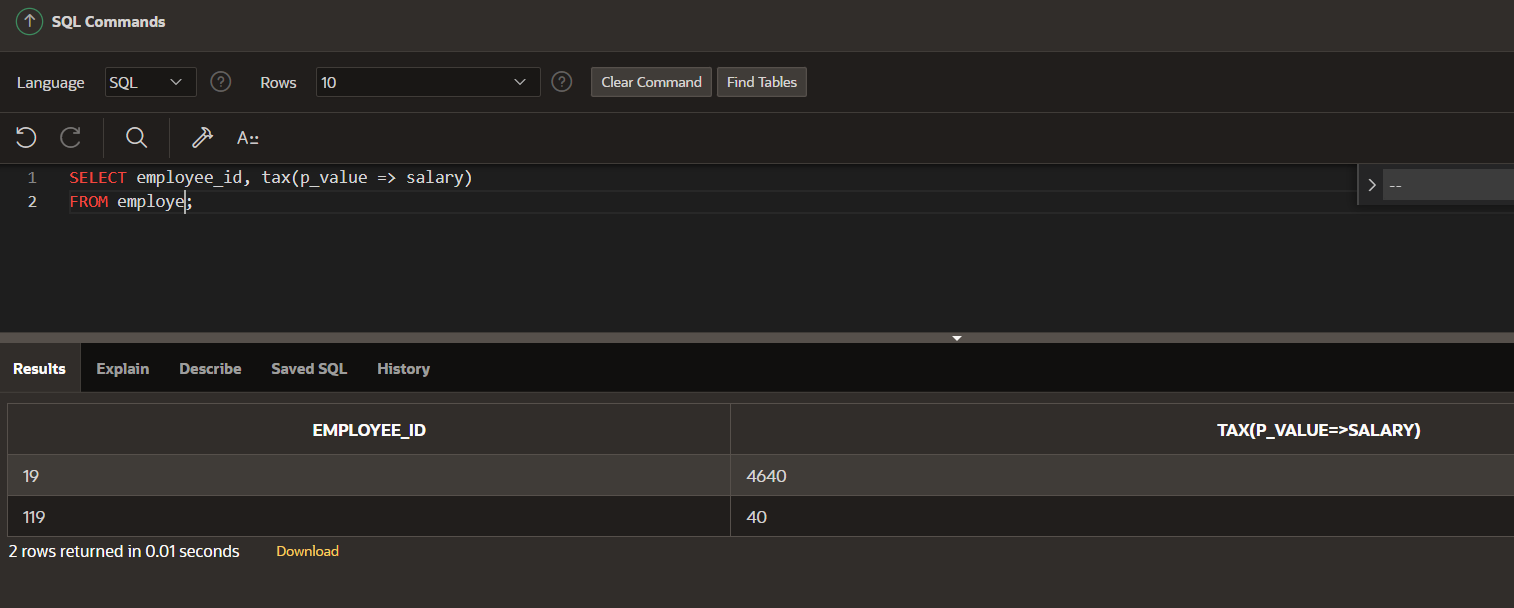
****

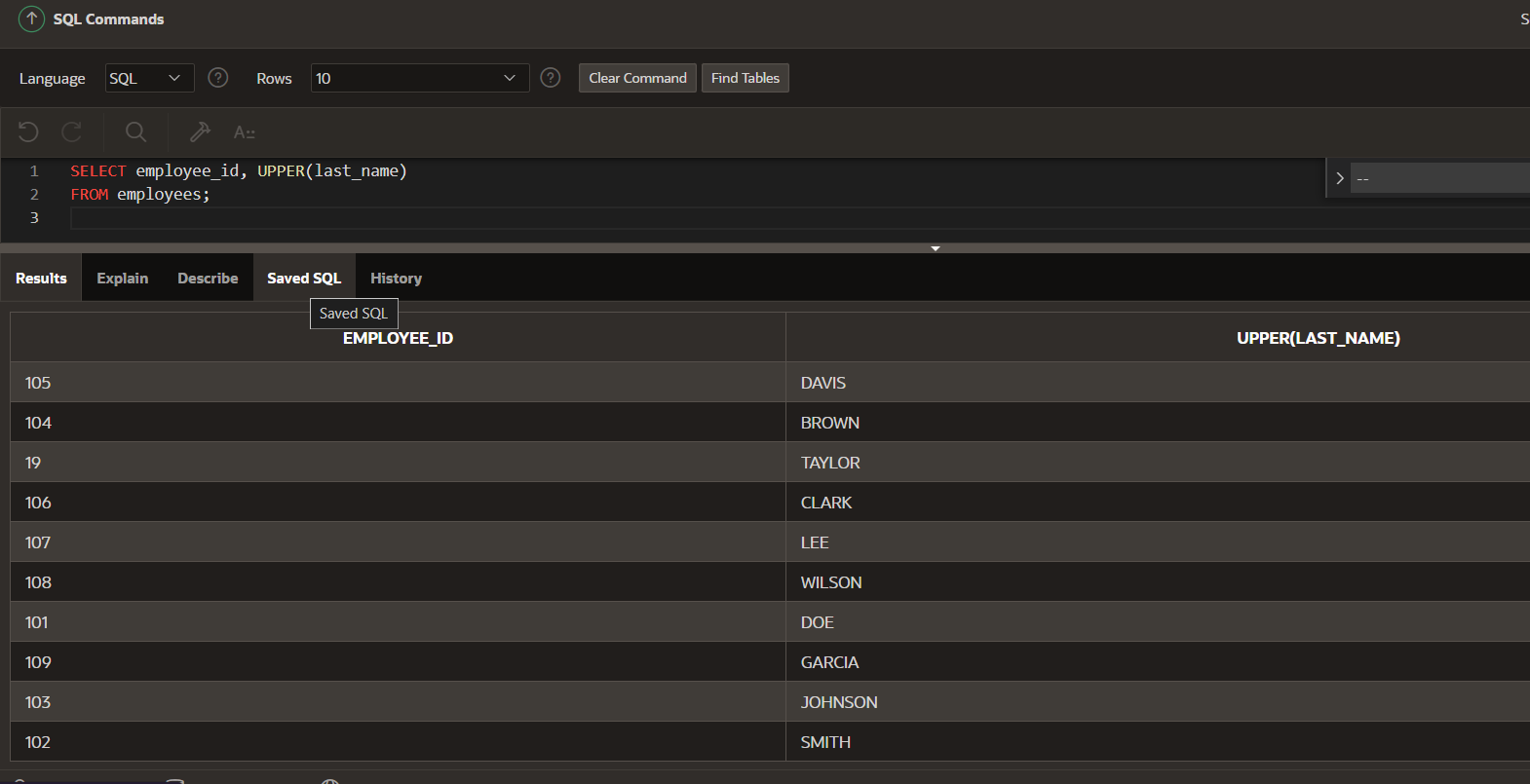
****

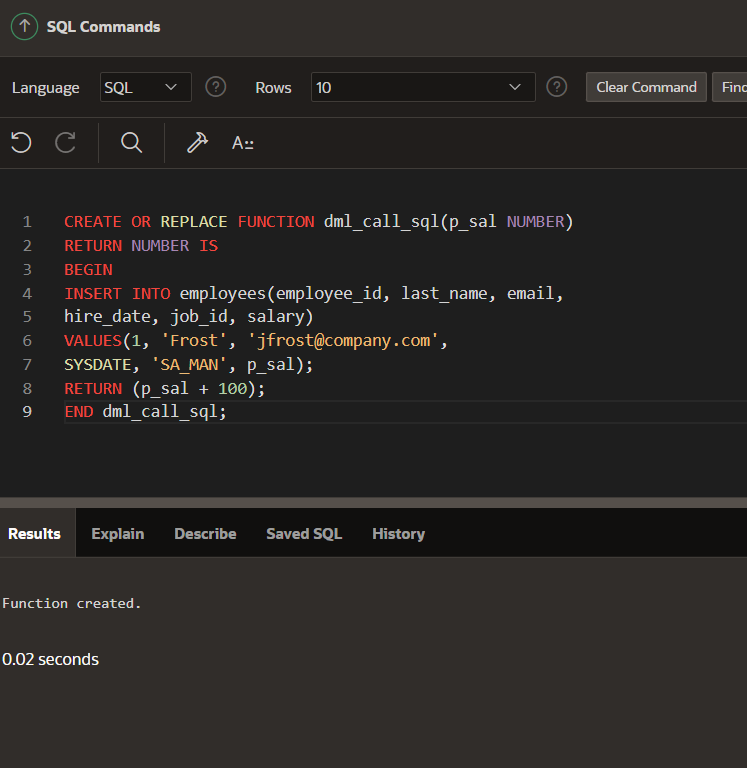
****

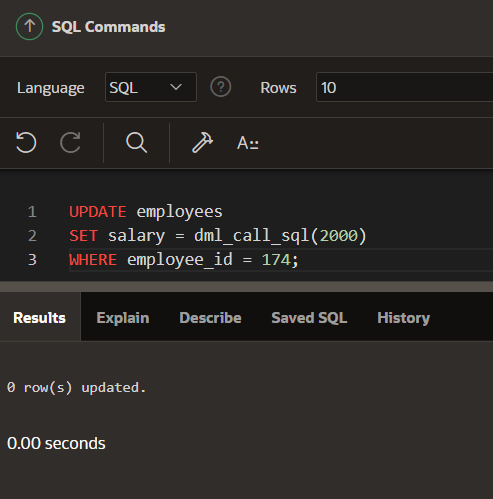
****

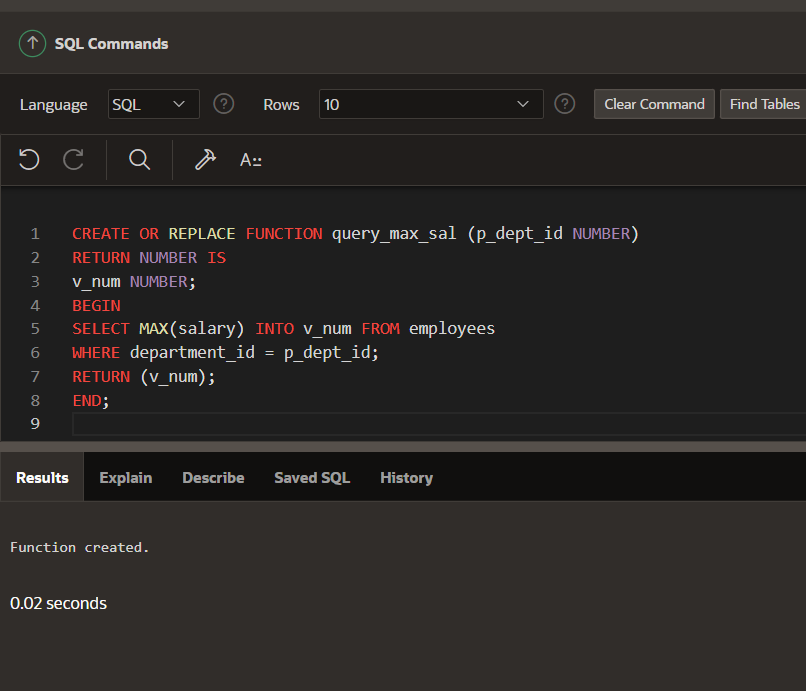
****

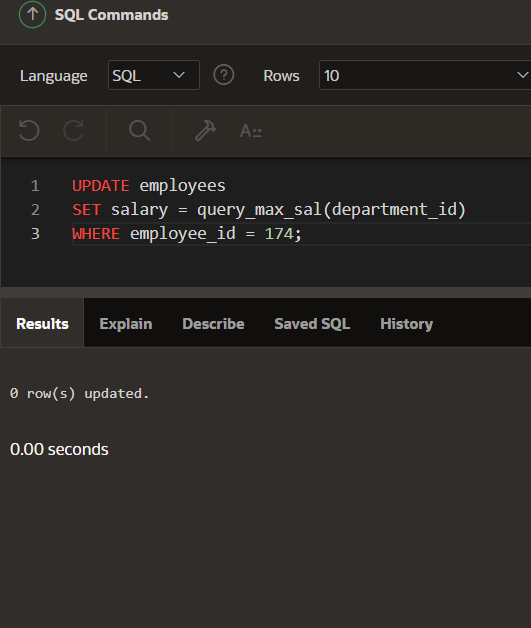
****

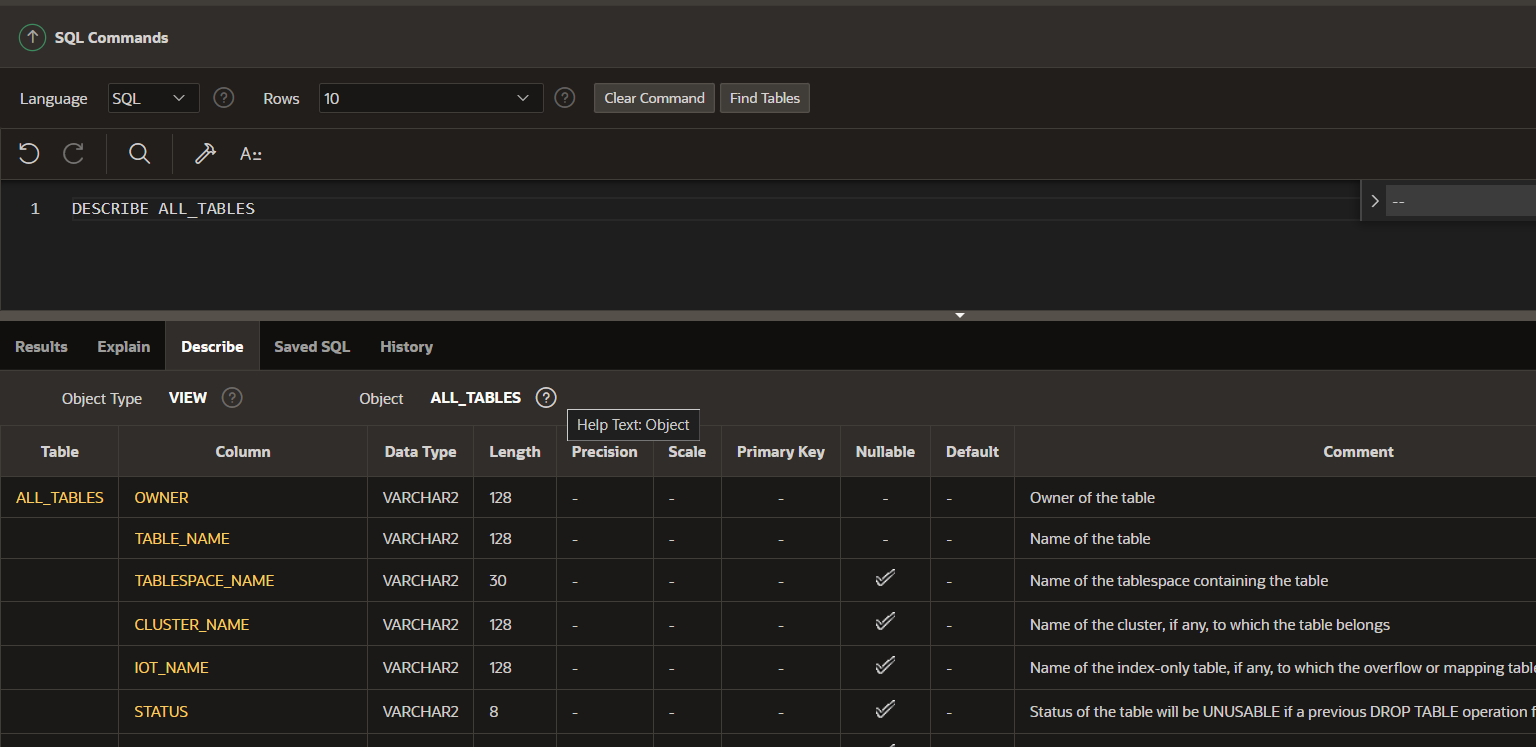
****

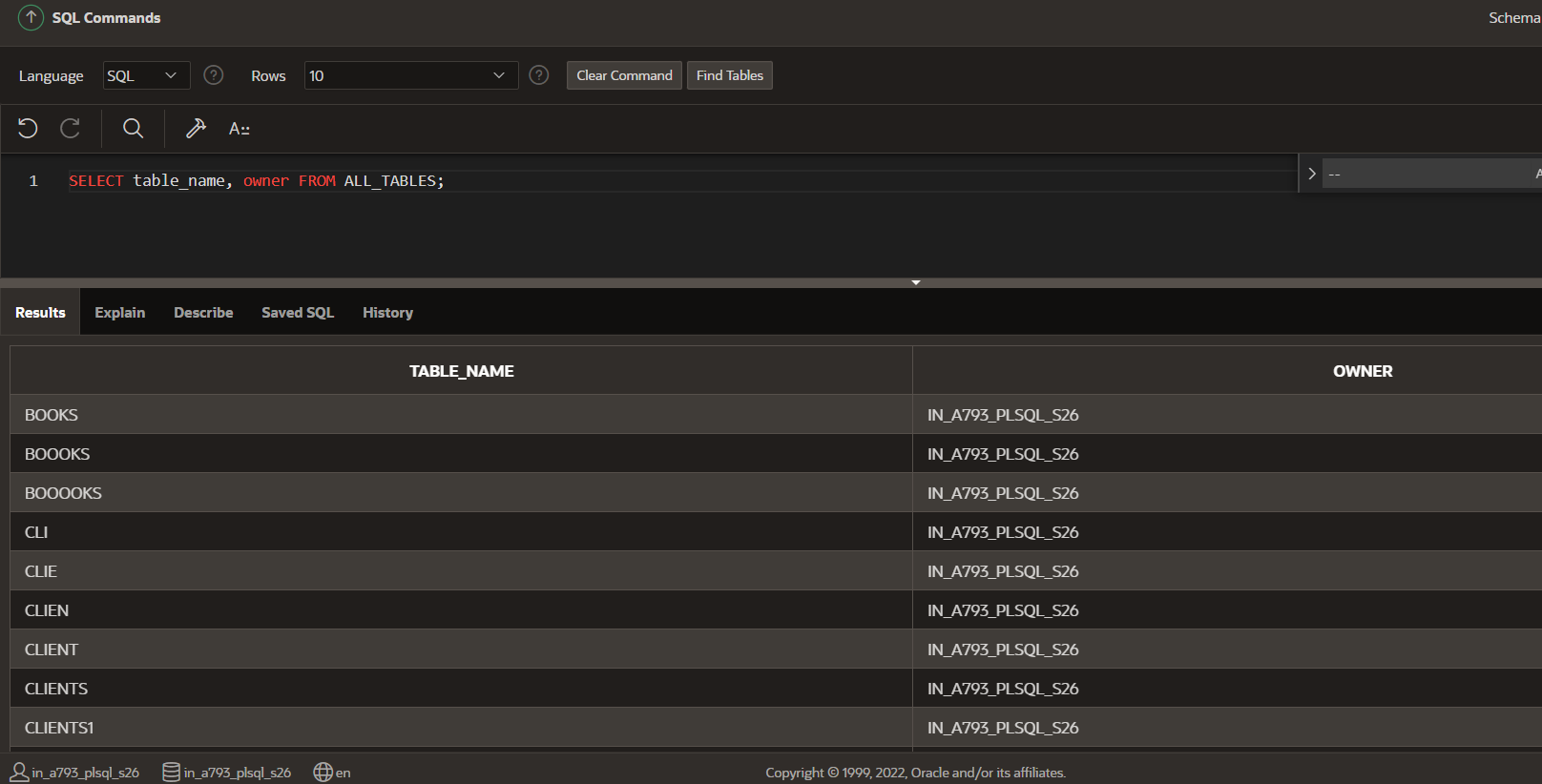
****

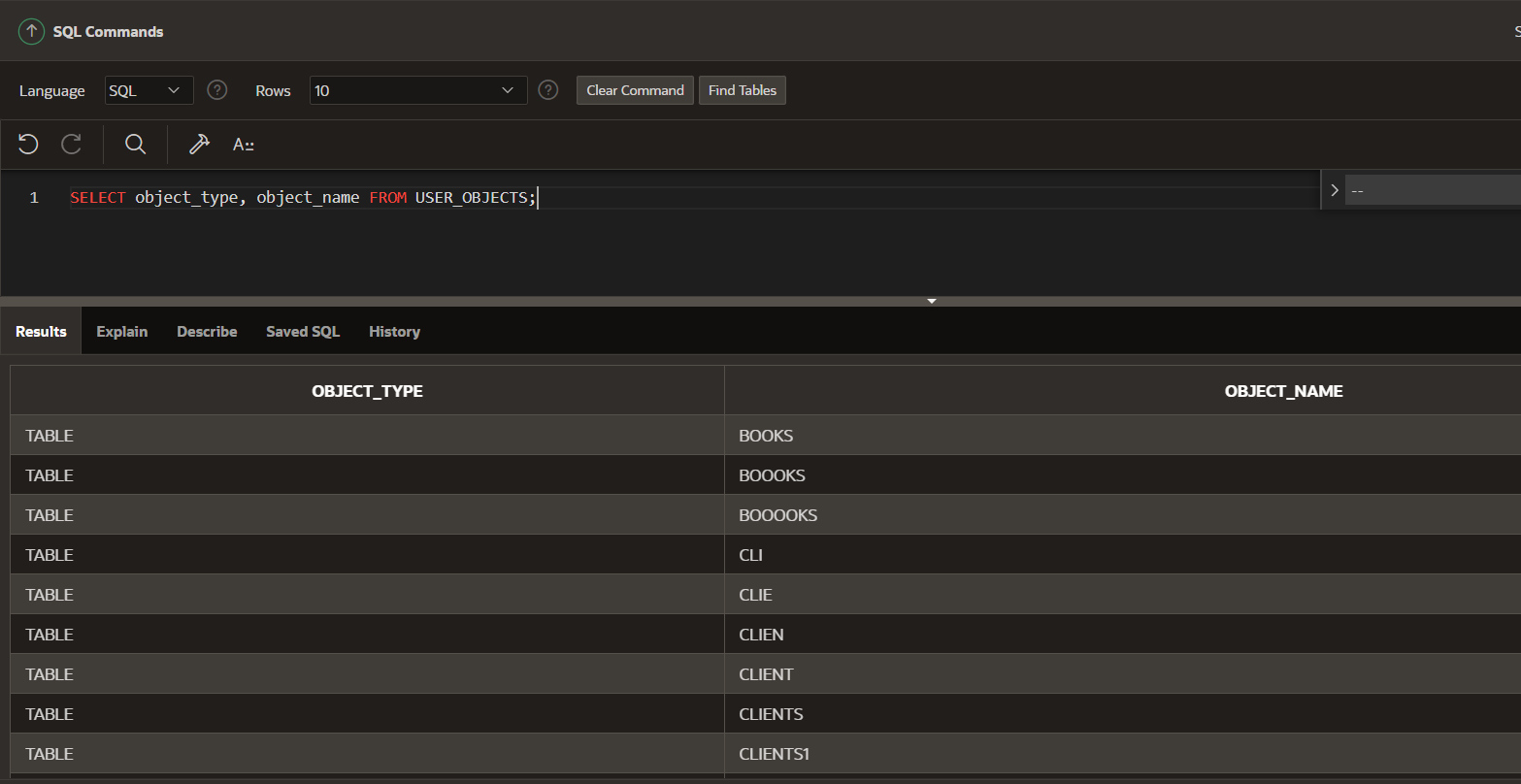
****

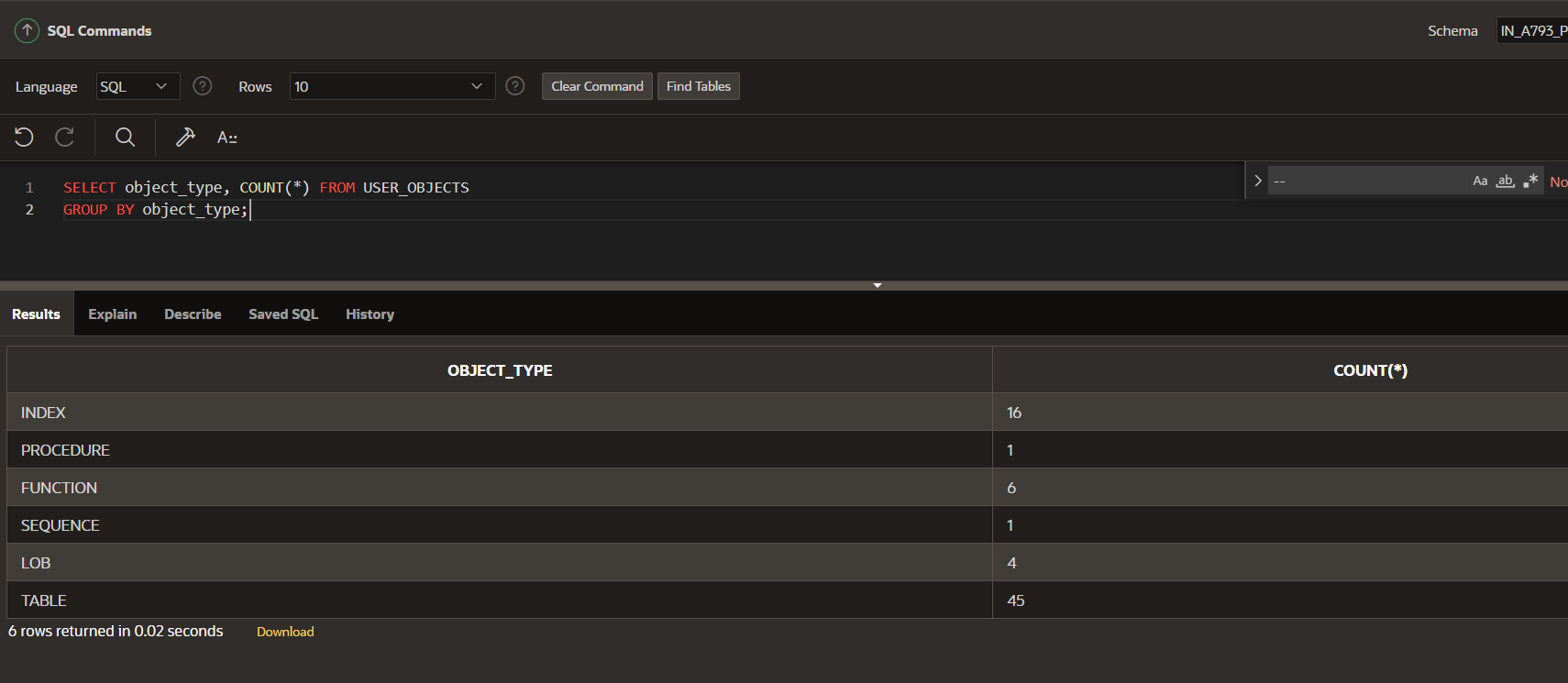
****

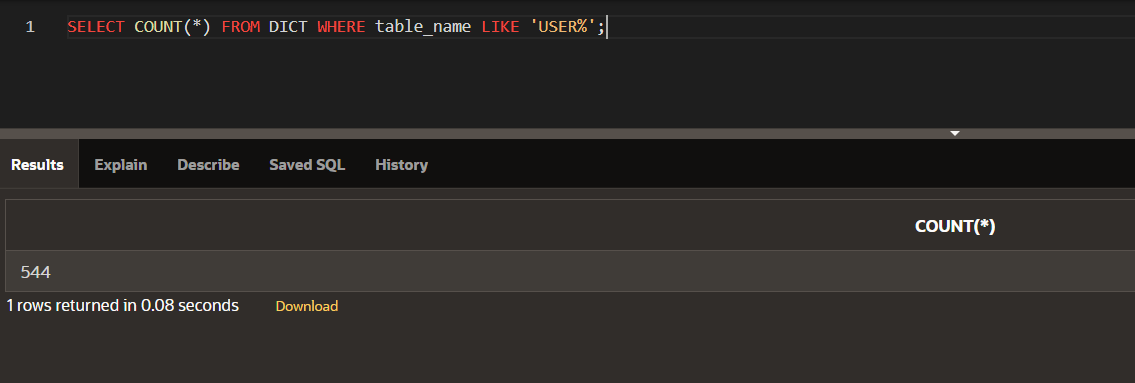
****

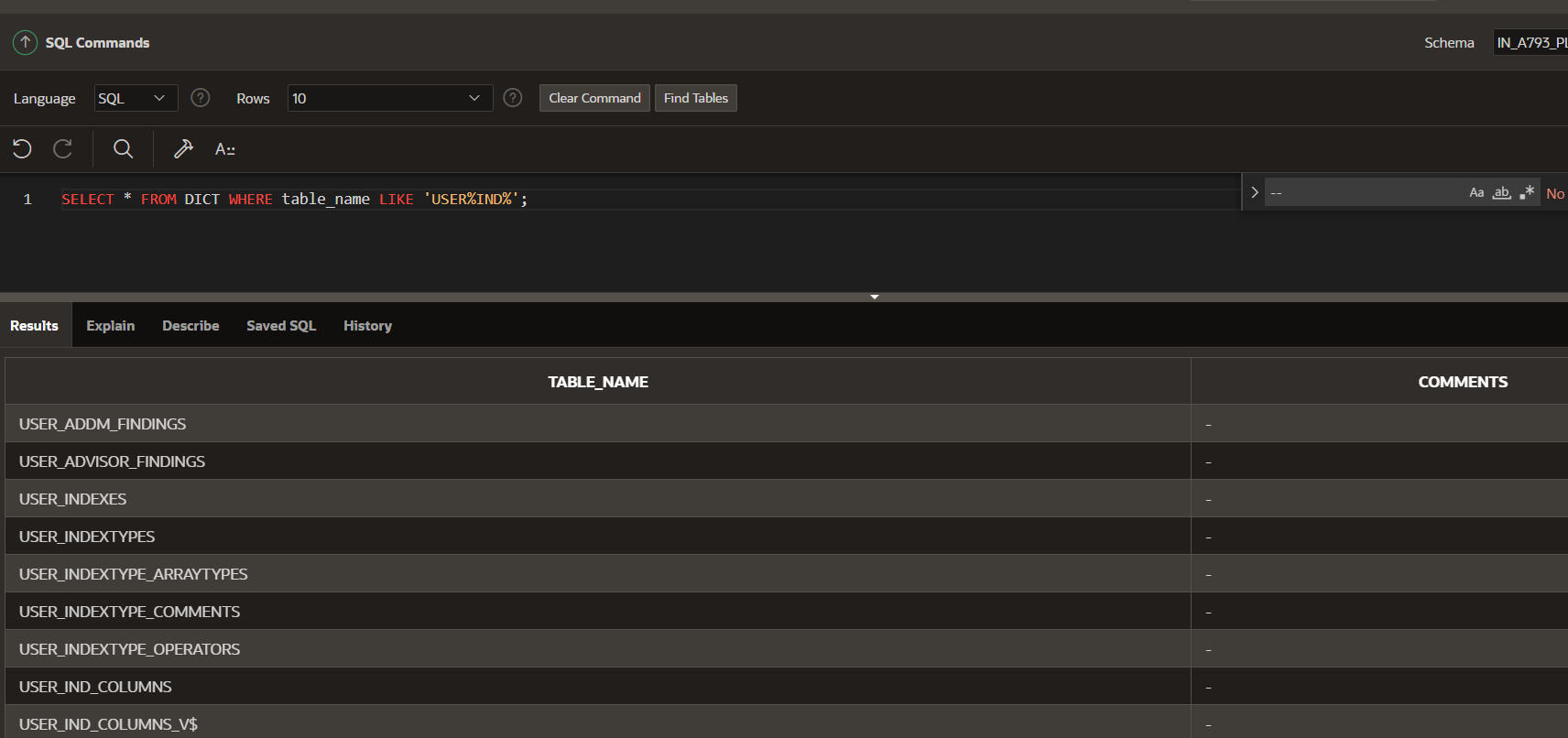
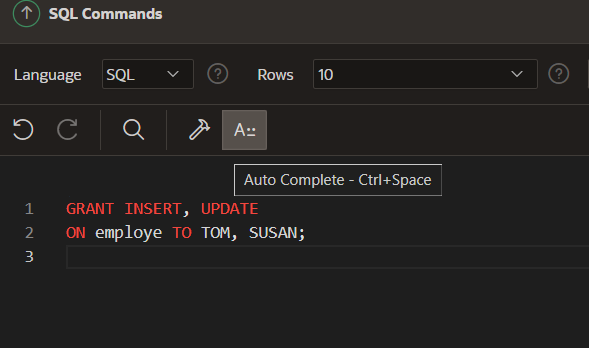
****

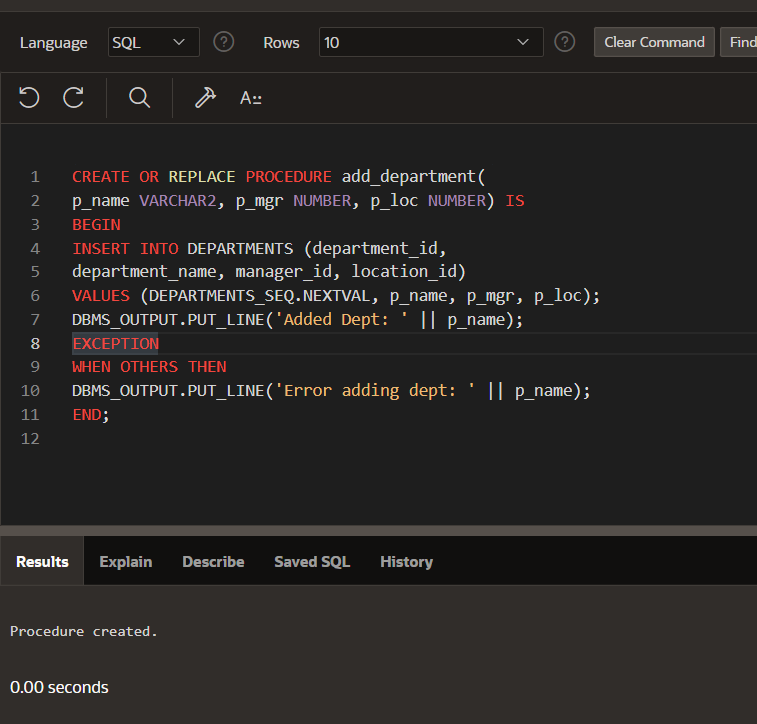
****

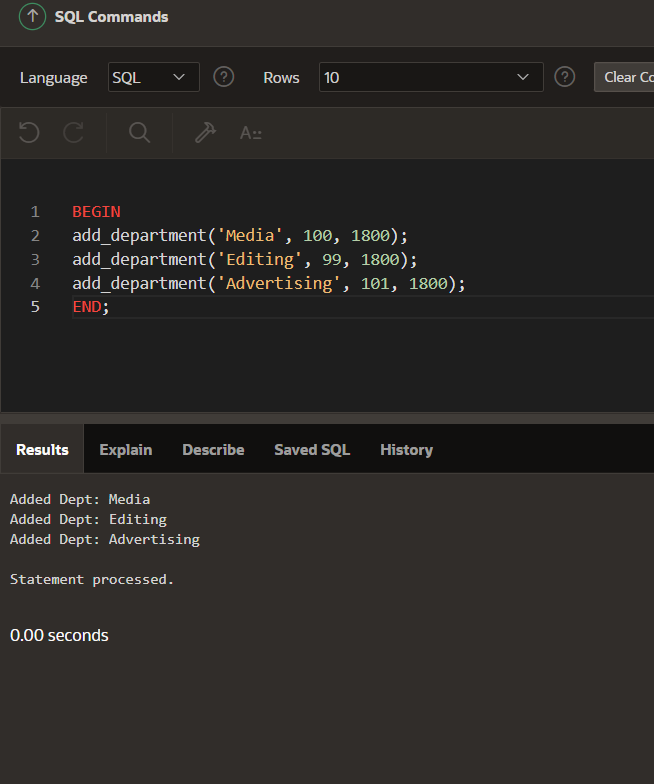
****

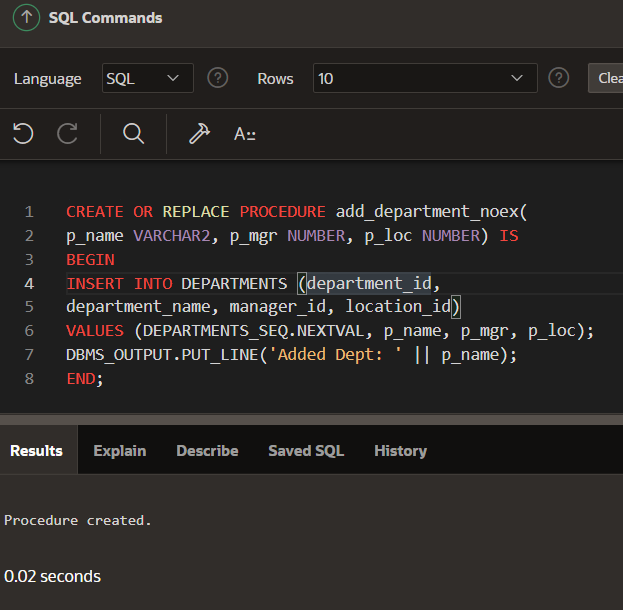
****

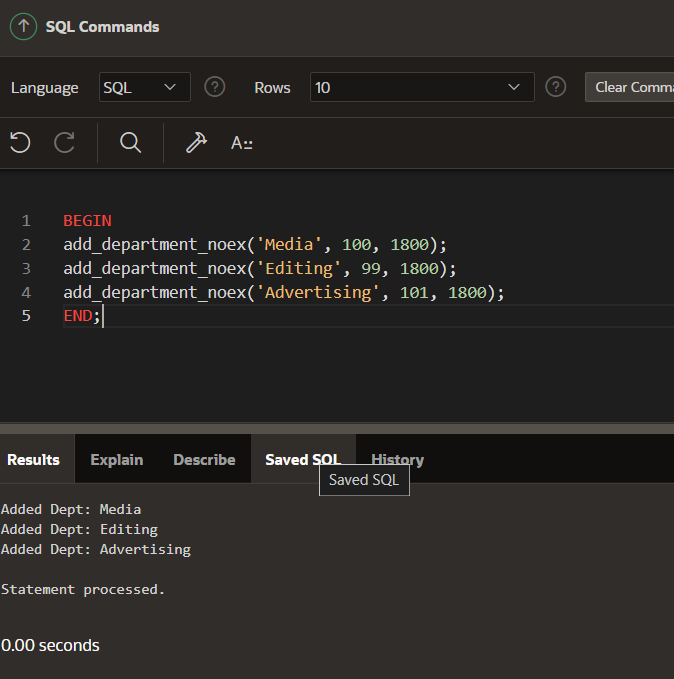
****

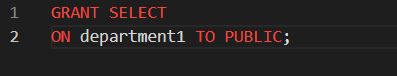
****

****

****

****

****

****