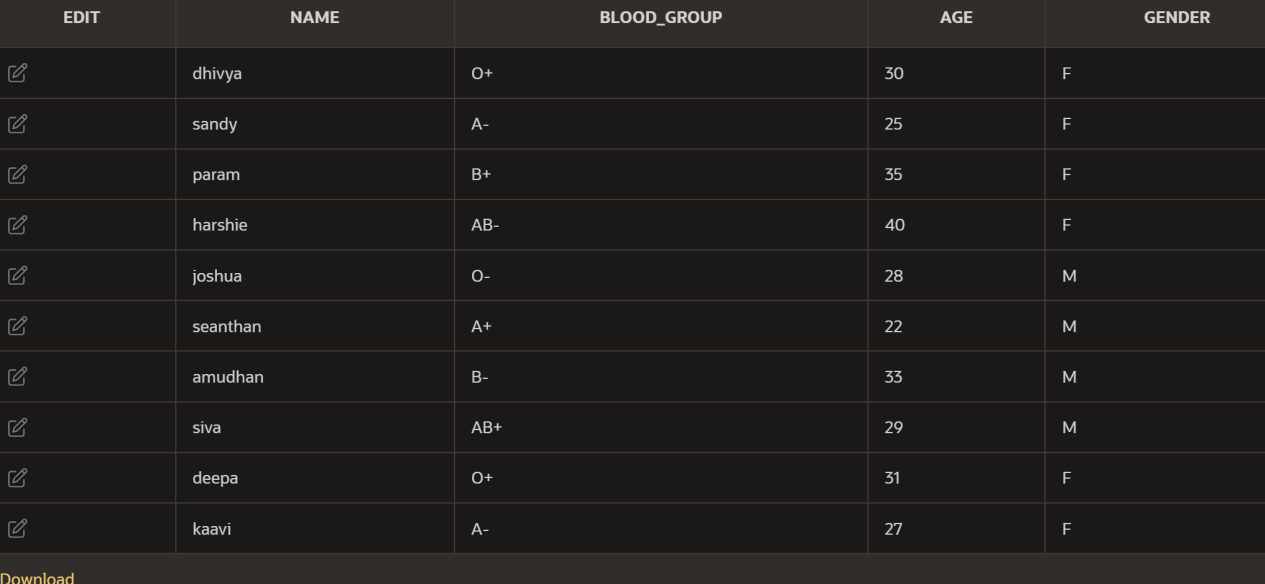
# SQL FROM 4.1

**CREATE TABLE PERSONAL DETAILS AND ARRANGE THEM IN ASCENDING ORDER:**

THE TABLE AFTER CREATING AND INSERTING DATA:



DECLARE

CURSOR my\_cursor IS

SELECT name,blood\_group,age,gender

FROM personal\_details

ORDER BY blood\_group ASC;

my\_record my\_cursor%ROWTYPE;

BEGIN

OPEN my\_cursor;

LOOP

FETCH my\_cursor INTO my\_record;

EXIT WHEN my\_cursor%NOTFOUND;

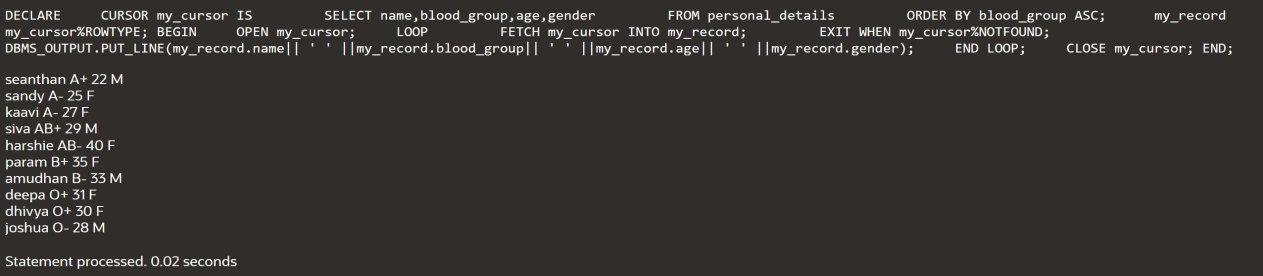
DBMS\_OUTPUT.PUT\_LINE(my\_record.name|| ' ' ||my\_record.blood\_group|| ' ' ||my\_record.age|| ' ' ||my\_record.gender);

END LOOP;

CLOSE my\_cursor;

END;

**OUTPUT:**



**Using the FORALL statement with SQL%BULK\_ROWCOUNT:**

* **Create table:**

CREATE TABLE num\_table (

n NUMBER

);

DECLARE

TYPE num\_list\_type IS TABLE OF NUMBER

INDEX BY BINARY\_INTEGER;

v\_nums num\_list\_type;

BEGIN

v\_nums (1) := 1;

v\_nums (2) := 3;

v\_nums (3) := 5;

v\_nums (4) := 7;

v\_nums (5) := 11;

FORALL i IN v\_nums.FIRST .. v\_nums.LAST

INSERT INTO num\_table (n) VALUES (v\_nums (i));

FOR i IN v\_nums.FIRST .. v\_nums.LAST

LOOP

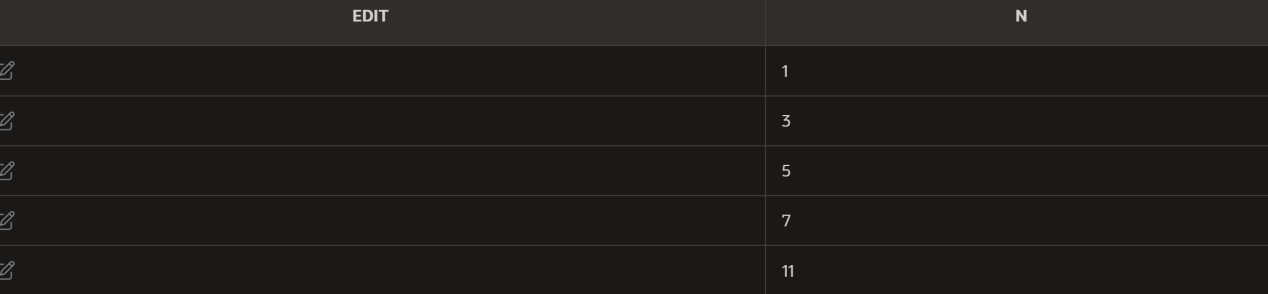
DBMS\_OUTPUT.PUT\_LINE ('Inserted '||

SQL%BULK\_ROWCOUNT (i) || ' row (s)' ||

' on iteration ' || i );

END LOOP;

END;



* **CREATE TABLE FOR N AND N1 AND ADD THEM:**

DECLARE

TYPE num\_list\_type IS TABLE OF NUMBER

INDEX BY BINARY\_INTEGER;

v\_n num\_list\_type;

v\_n1 num\_list\_type;

BEGIN

v\_n(1) := 1;

v\_n(2) := 3;

v\_n(3) := 5;

v\_n(4) := 7;

v\_n(5) := 11;

v\_n1(1) := 10;

v\_n1(2) := 30;

v\_n1(3) := 50;

v\_n1(4) := 70;

v\_n1(5) := 110;

FORALL i IN v\_n.FIRST .. v\_n.LAST

INSERT INTO num\_table1 (n, n1, n2) VALUES (v\_n(i), v\_n1(i), v\_n(i) + v\_n1(i));

FOR i IN v\_n.FIRST .. v\_n.LAST

LOOP

DBMS\_OUTPUT.PUT\_LINE('Inserted values: ' ||

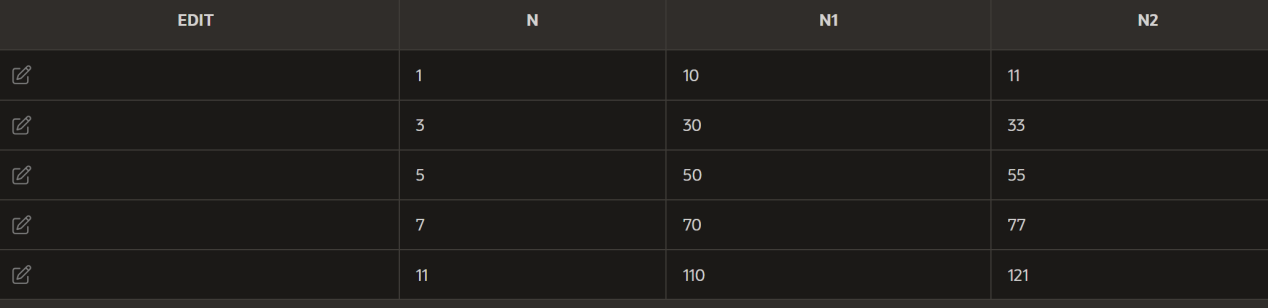
'n=' || v\_n(i) || ', ' ||

'n1=' || v\_n1(i) || ', ' ||

'n2=' || (v\_n(i) + v\_n1(i)));

END LOOP;

END;



**GOTO:**

DECLARE

a number(2) := 30;

BEGIN

<<loopstart>>

WHILE a < 50 LOOP

dbms\_output.put\_line ('value of a: ' || a);

a := a + 1;

IF a = 35 THEN

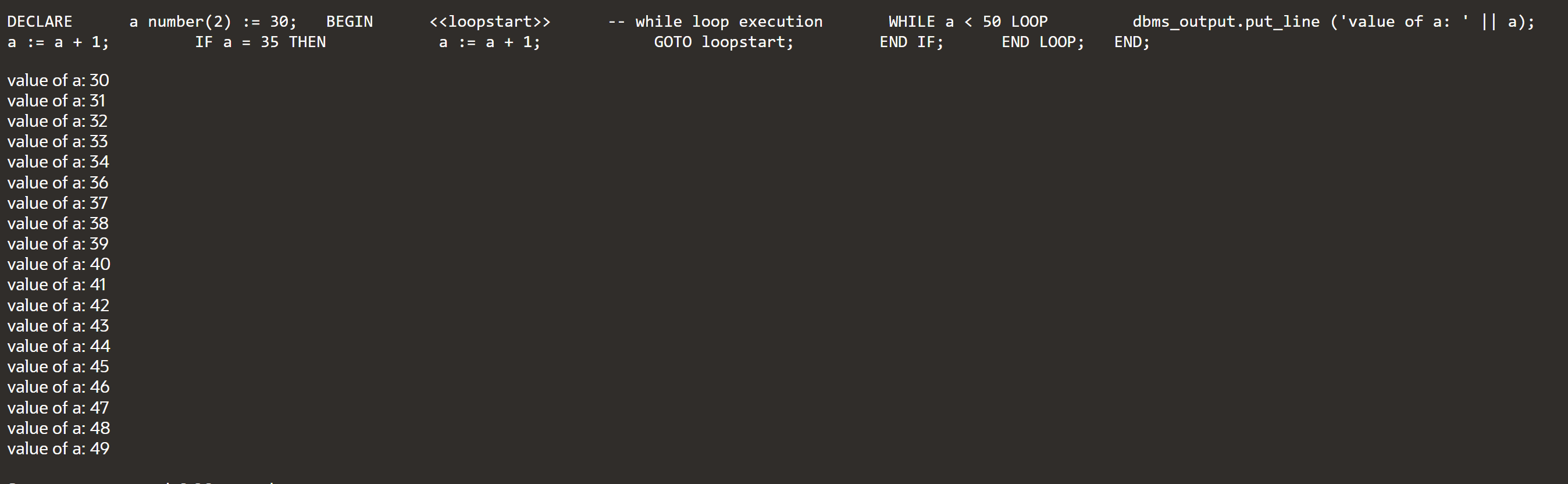
a := a + 1;

GOTO loopstart;

END IF;

END LOOP;

END;



**HANDLING EXCEPTION WHEN A DIVISION BY ZERO OCCURS:**

DECLARE

dividend NUMBER := 10;

divisor NUMBER := 0;

result NUMBER;

BEGIN

BEGIN

result := dividend / divisor;

DBMS\_OUTPUT.PUT\_LINE('Result: ' || result);

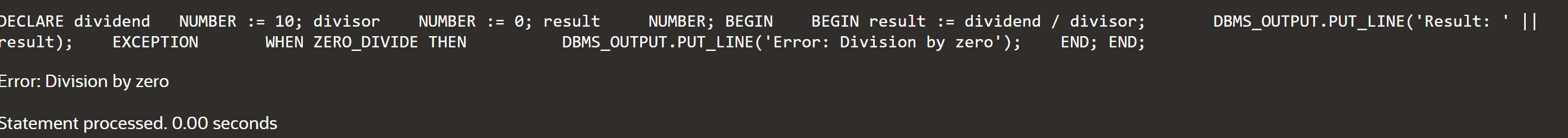
EXCEPTION

WHEN ZERO\_DIVIDE THEN

DBMS\_OUTPUT.PUT\_LINE('Error: Division by zero');

END;

END;



**NO DATA FOUND:**

* **Handle the NO\_DATA\_FOUND exception when retrieving a row from a table and no matching record is found:**

DECLARE

emp\_name VARCHAR2(100);

emp\_id NUMBER := 225;

BEGIN

BEGIN

SELECT first\_name INTO emp\_name FROM employee2 WHERE employee\_id = emp\_id;

DBMS\_OUTPUT.PUT\_LINE('Employee Name: ' || emp\_name);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Error: No matching record found');

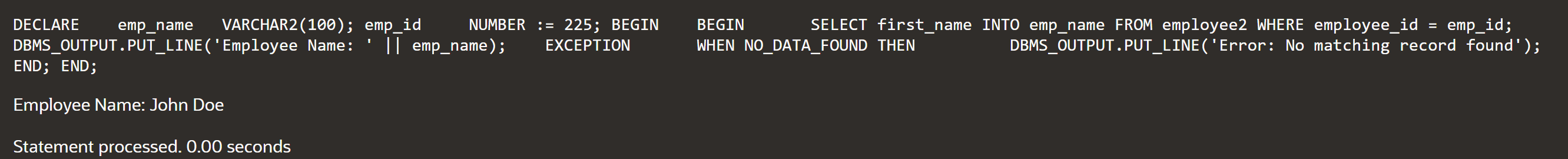
END;

END;

**TABLE:**



**OUTPUT:**

****

* **Handle the TOO\_MANY\_ROWS exception when retrieving multiple rows instead of a single row from a table.**

CREATE TABLE employee3 (

employee\_id INTEGER PRIMARY KEY,

first\_name VARCHAR2(25),

last\_name VARCHAR2(25),

email VARCHAR2(25),

phone\_number VARCHAR2(15),

hire\_date DATE,

job\_id VARCHAR2(25),

salary INTEGER,

commission\_pct NUMBER(5,2),

manager\_id INTEGER,

department\_id INTEGER

);

INSERT INTO employee3 (employee\_id, first\_name, last\_name, email, phone\_number, hire\_date, job\_id, salary, commission\_pct, manager\_id, department\_id)

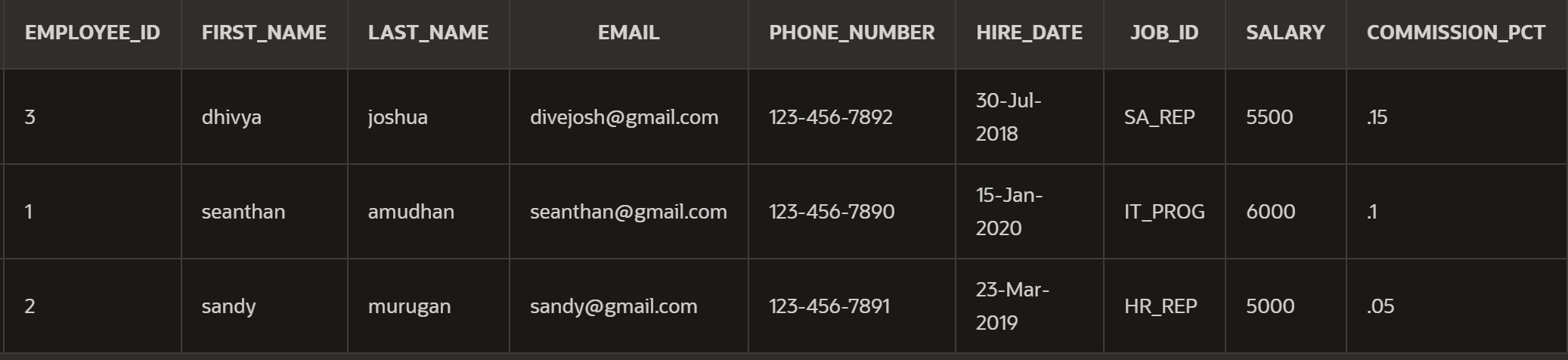
VALUES (1, 'seanthan', 'amudhan', 'seanthan@gmail.com', '123-456-7890', TO\_DATE('2020-01-15', 'YYYY-MM-DD'), 'IT\_PROG', 6000, 0.10, 0, 90);

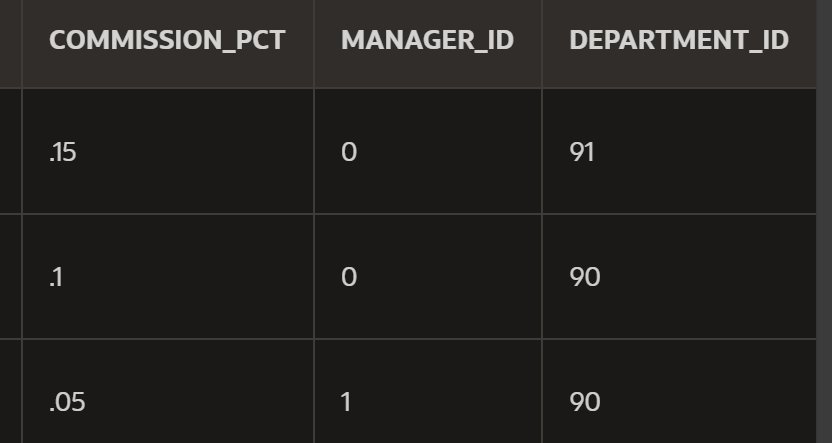
INSERT INTO employee3 (employee\_id, first\_name, last\_name, email, phone\_number, hire\_date, job\_id, salary, commission\_pct, manager\_id, department\_id)

VALUES (2, 'sandy', 'murugan', 'sandy@gmail.com', '123-456-7891', TO\_DATE('2019-03-23', 'YYYY-MM-DD'), 'HR\_REP', 5000, 0.05, 1, 90);

INSERT INTO employee3 (employee\_id, first\_name, last\_name, email, phone\_number, hire\_date, job\_id, salary, commission\_pct, manager\_id, department\_id)

VALUES (3, 'dhivya', 'joshua', 'divejosh@gmail.com', '123-456-7892', TO\_DATE('2018-07-30', 'YYYY-MM-DD'), 'SA\_REP', 5500, 0.15, 0, 91);





DECLARE

v\_dep\_id employees.department\_id%TYPE := 90;

v\_emp\_id employees.employee\_id%TYPE;

v\_emp\_name employees.first\_name%TYPE;

v\_emp\_salary employees.salary%TYPE;

v\_exception\_msg VARCHAR2(200);

BEGIN

SELECT first\_name, department\_id, salary

INTO v\_emp\_name,v\_dep\_id, v\_emp\_salary

FROM employees

WHERE department\_id = v\_dep\_id;

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN

v\_exception\_msg := 'Multiple rows found for the given department ID: ' || v\_dep\_id;

DBMS\_OUTPUT.PUT\_LINE(v\_exception\_msg);

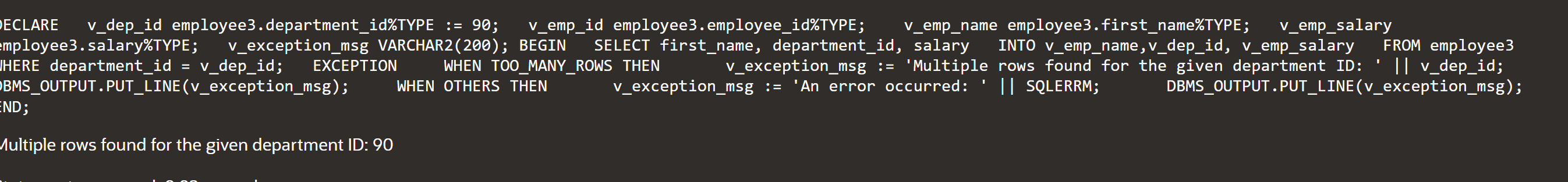
WHEN OTHERS THEN

v\_exception\_msg := 'An error occurred: ' || SQLERRM;

DBMS\_OUTPUT.PUT\_LINE(v\_exception\_msg);

END;

**OUTPUT:**



**Handle the INVALID\_NUMBER exception when converting a non-numeric value to a number:**

DECLARE

v\_input VARCHAR2(10) := 'abc';

v\_number NUMBER;

BEGIN

BEGIN

v\_number := TO\_NUMBER(v\_input);

DBMS\_OUTPUT.PUT\_LINE('Conversion successful. Number: ' || v\_number);

EXCEPTION

WHEN INVALID\_NUMBER THEN

DBMS\_OUTPUT.PUT\_LINE('Error: Invalid number');

END;

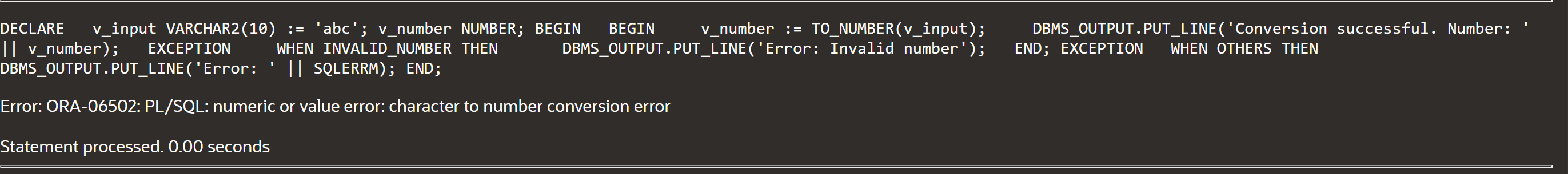
EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

END;

**OUTPUT:**



**Handle the DUP\_VAL\_ON\_INDEX exception when inserting a duplicate value into a table with a unique constraint:**

DECLARE

v\_duplicate\_value NUMBER := 125;

BEGIN

BEGIN

INSERT INTO employees (employee\_id) VALUES (v\_duplicate\_value);

DBMS\_OUTPUT.PUT\_LINE('Insertion successful.');

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

-- Handle the exception

DBMS\_OUTPUT.PUT\_LINE('Error: Duplicate value already exists.');

-- You can perform additional error handling or logging here

END;

END;

**BEFORE:**

**AFTER:**

