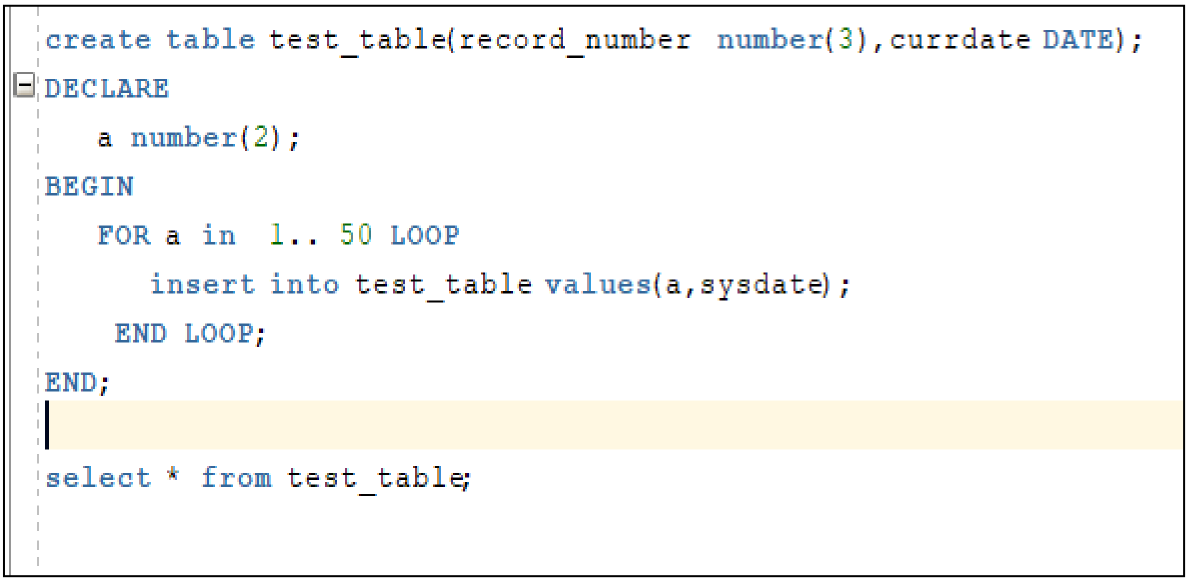
Name:- Harshvardhan Arvind Patil

PRN No:- 2020BTECS00050

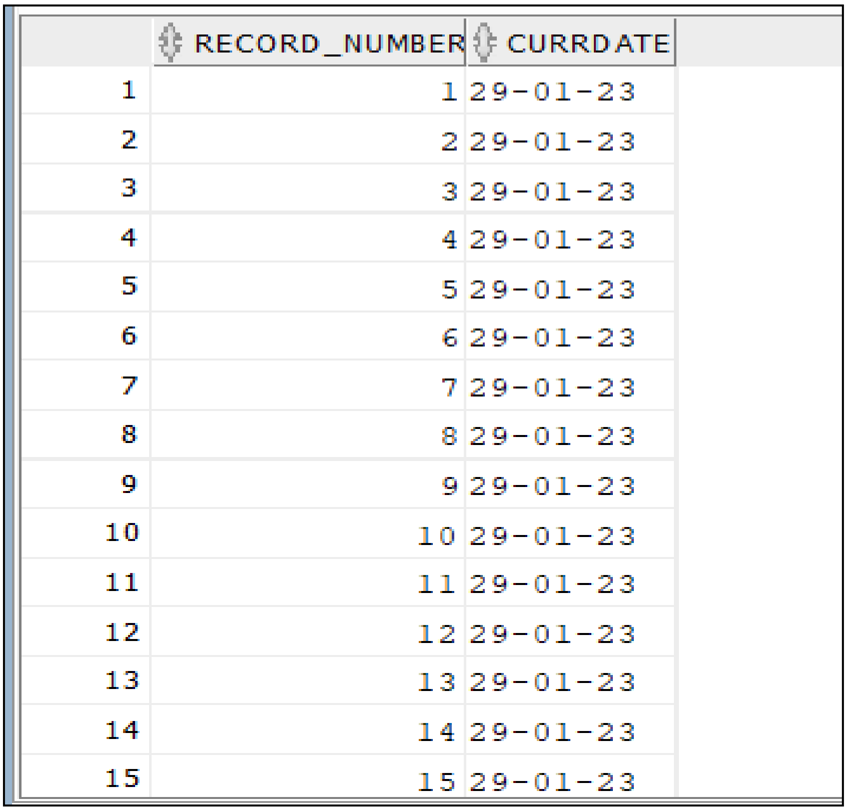
- ## PL/SQL Code blocks :

1 .Create a table called test_table with 2 columns RecordNumber (type : Number(3)) and CurrentDate (type : Date)). Write PL/SQL block which will insert 50 records into test_table. Insert the current date value into the table.



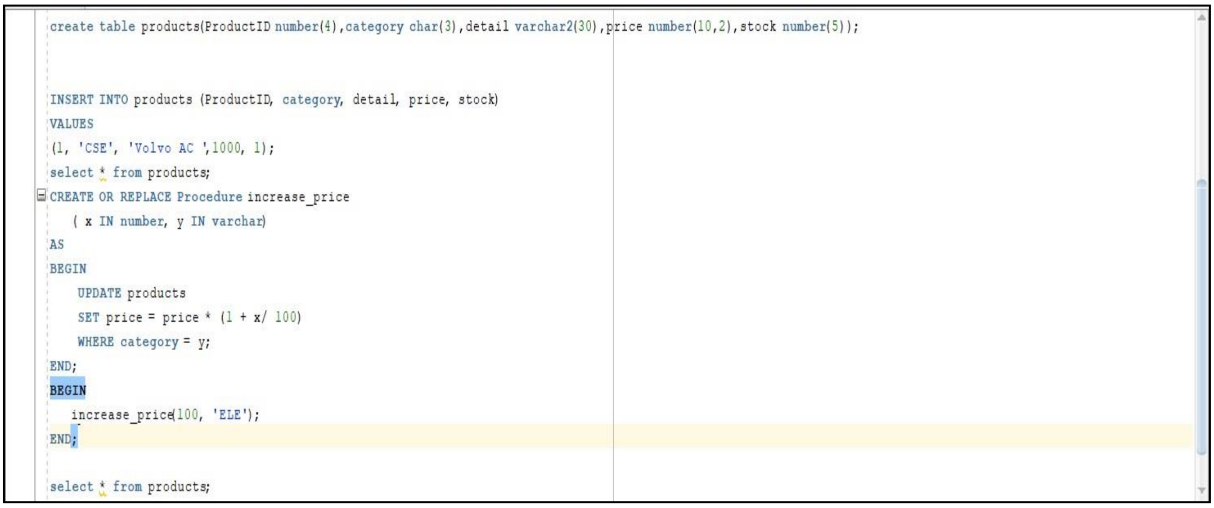
```
create table test_table(record_number  number(3),currdate DATE);
DECLARE
   a number(2);
BEGIN
   FOR a in  1.. 50 LOOP
      insert into test_table values(a,sysdate);
    END LOOP;
END;

select * from test_table;
```

Result :



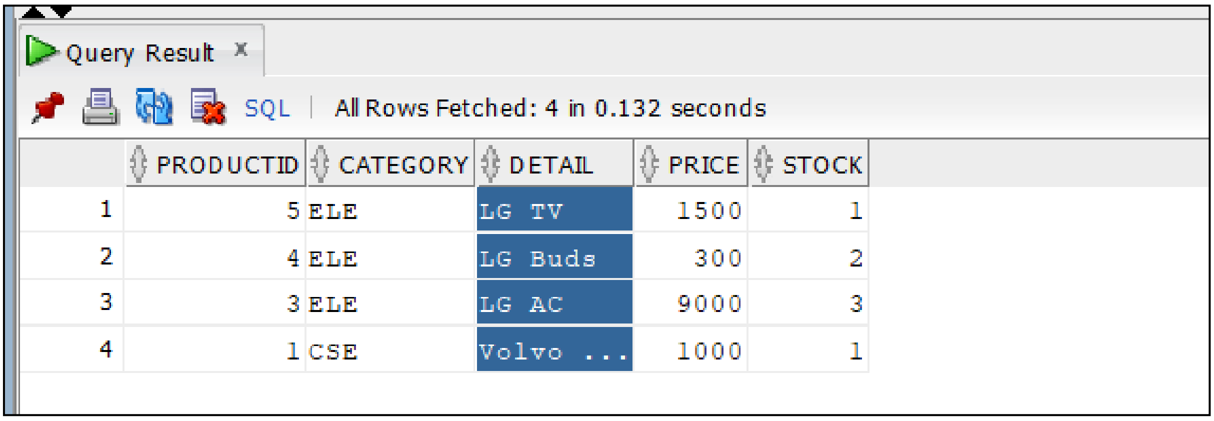
	RECORD_NUMBER	CURRDATE
1	1	29-01-23
2	2	29-01-23
3	3	29-01-23
4	4	29-01-23
5	5	29-01-23
6	6	29-01-23
7	7	29-01-23
8	8	29-01-23
9	9	29-01-23
10	10	29-01-23
11	11	29-01-23
12	12	29-01-23
13	13	29-01-23
14	14	29-01-23
15	15	29-01-23

2 . Create a products table products(ProductID number(4),category char(3),detail varchar2(30),price number(10,2),stock number(5)). Insert the sample data. Write PL/SQL procedure with two arguments X & Y which will increase price by X% for all products in category Y. X and Y will be given by user.



```
create table products(ProductID number(4),category char(3),detail varchar2(30),price number(10,2),stock number(5));


INSERT INTO products (ProductID, category, detail, price, stock)
VALUES
(1, 'CSE', 'Volvo AC ',1000, 1);
select * from products;
CREATE OR REPLACE Procedure increase_price
   ( x IN number, y IN varchar)
AS
BEGIN
    UPDATE products
    SET price = price * (1 + x/ 100)
    WHERE category = y;
END;
BEGIN
   increase_price(100, 'ELE');
END;

select * from products;
```

Result :

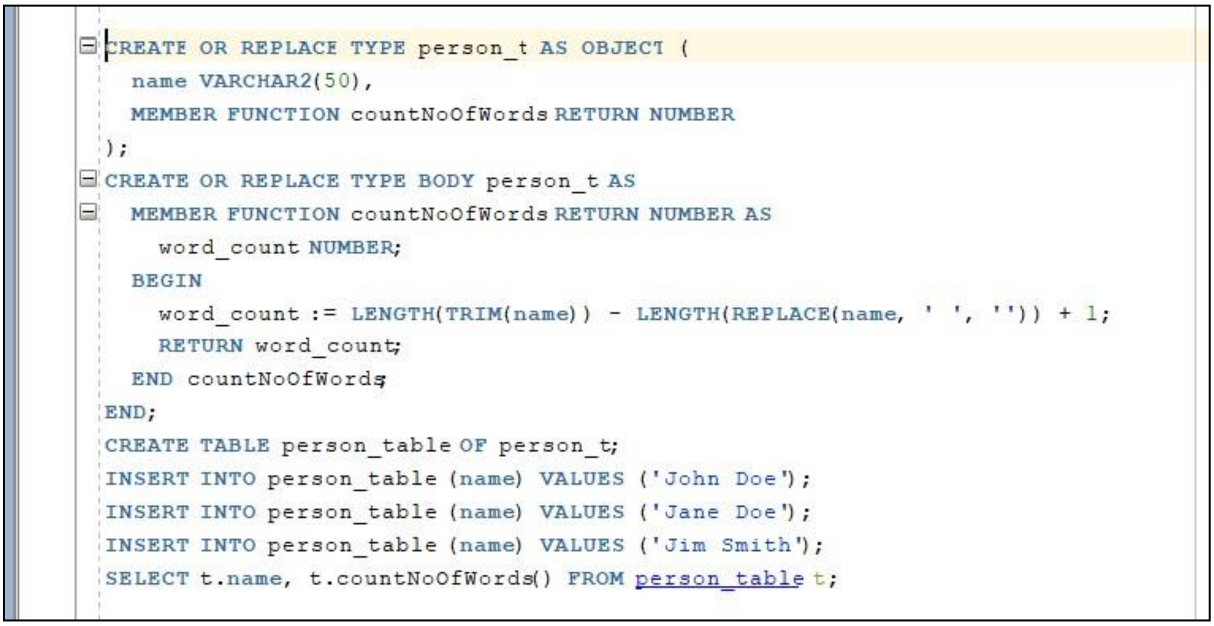


All Rows Fetched: 4 in 0.132 seconds

	PRODUCTID	CATEGORY	DETAIL	PRICE	STOCK
1	5	ELE	LG TV	1500	1
2	4	ELE	LG Buds	300	2
3	3	ELE	LG AC	9000	3
4	1	CSE	Volvo ...	1000	1

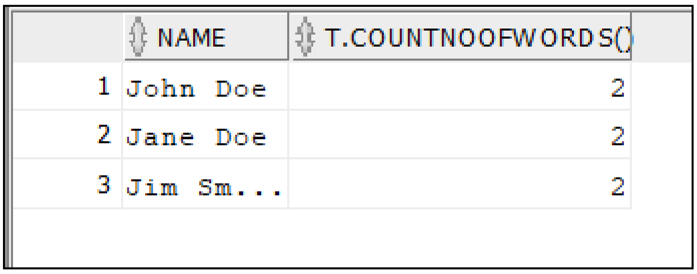
- # Object Relational Databsaes:

1. **Create Object Table containing field “name” of size 50 characters and member function “countNoOfWords” which returns the no. of words in “name” field. Demonstrate the working by entering different data.**



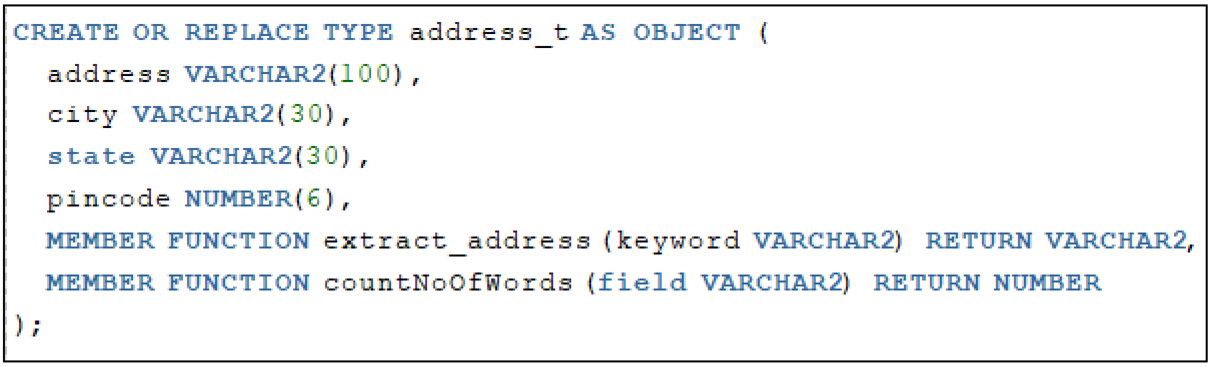
```
CREATE OR REPLACE TYPE person_t AS OBJECT (
  name VARCHAR2(50),
  MEMBER FUNCTION countNoOfWords RETURN NUMBER
);
CREATE OR REPLACE TYPE BODY person_t AS
  MEMBER FUNCTION countNoOfWords RETURN NUMBER AS
    word_count NUMBER;
  BEGIN
    word_count := LENGTH(TRIM(name)) - LENGTH(REPLACE(name, ' ', '')) + 1;
    RETURN word_count;
  END countNoOfWords;
END;
CREATE TABLE person_table OF person_t;
INSERT INTO person_table (name) VALUES ('John Doe');
INSERT INTO person_table (name) VALUES ('Jane Doe');
INSERT INTO person_table (name) VALUES ('Jim Smith');
SELECT t.name, t.countNoOfWords() FROM person_table t;
```

Result :

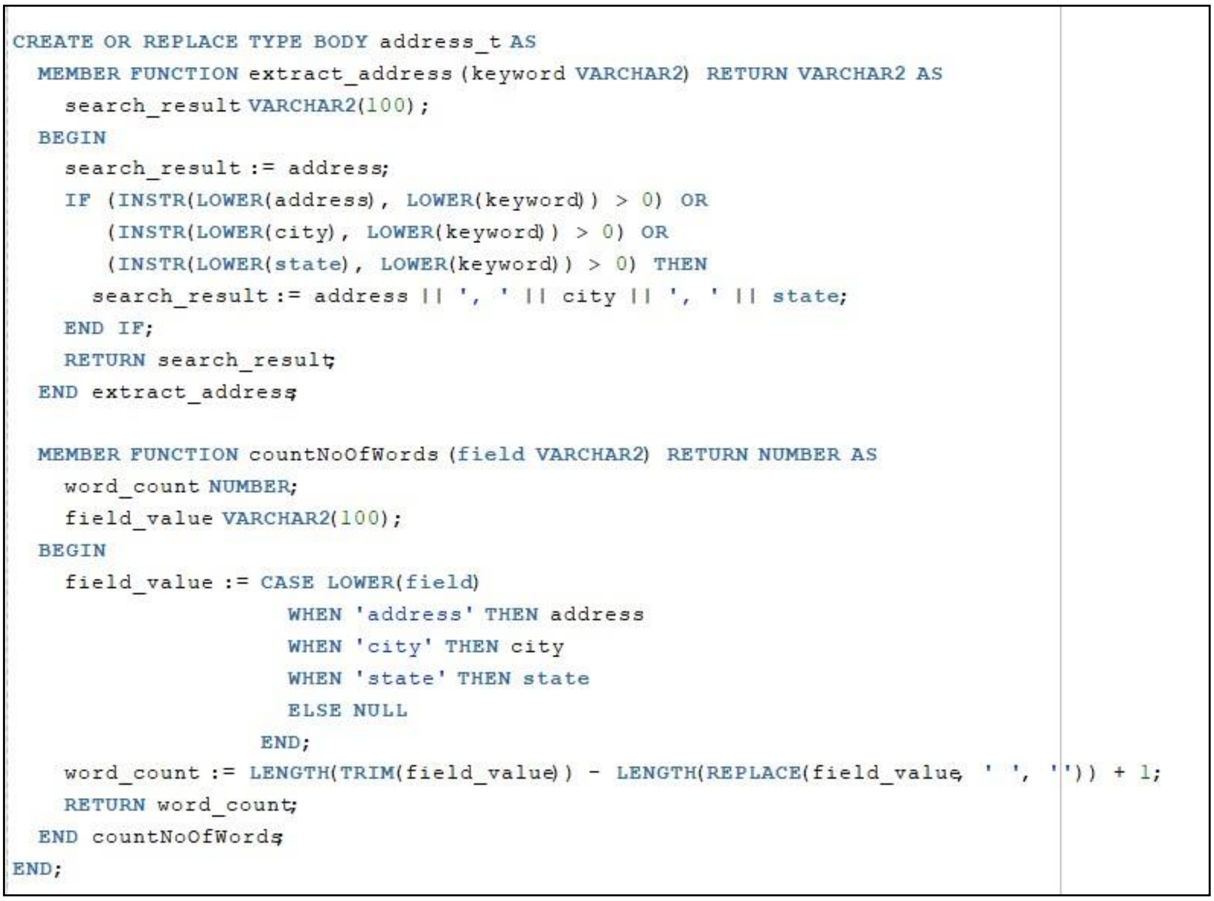


	NAME	T.COUNTNOOFWORDS()
1	John Doe	2
2	Jane Doe	2
3	Jim Sm...	2

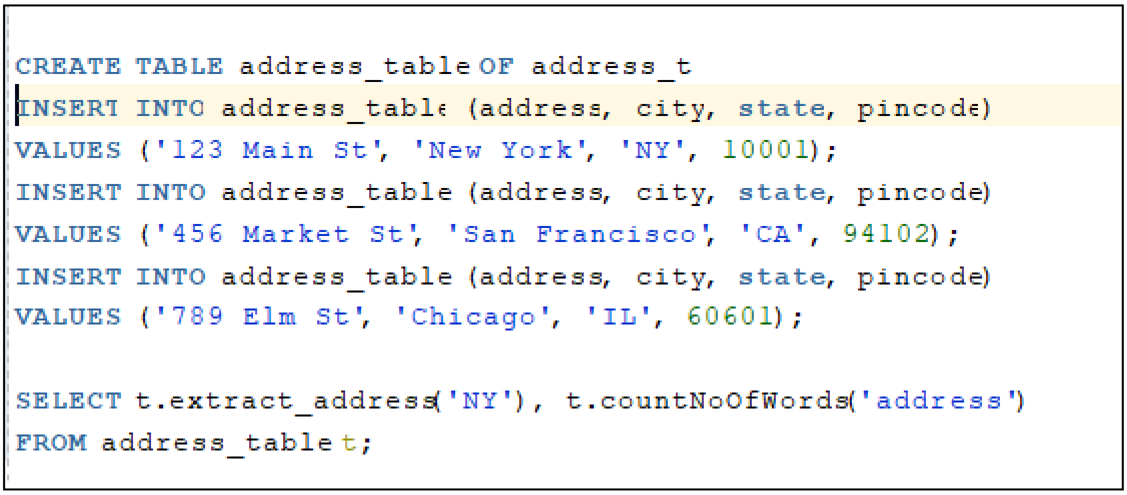
2. **Create an address type with the following attributes : address, city, state & pincode. Include the following methods i. to extract the addresses based on given keyword. j. to return the no. of words in each given field (method should accept the name of attribute/field)**



```
CREATE OR REPLACE TYPE address_t AS OBJECT (
  address VARCHAR2(100),
  city VARCHAR2(30),
  state VARCHAR2(30),
  pincode NUMBER(6),
  MEMBER FUNCTION extract_address(keyword VARCHAR2) RETURN VARCHAR2,
  MEMBER FUNCTION countNoOfWords (field VARCHAR2) RETURN NUMBER
);
```

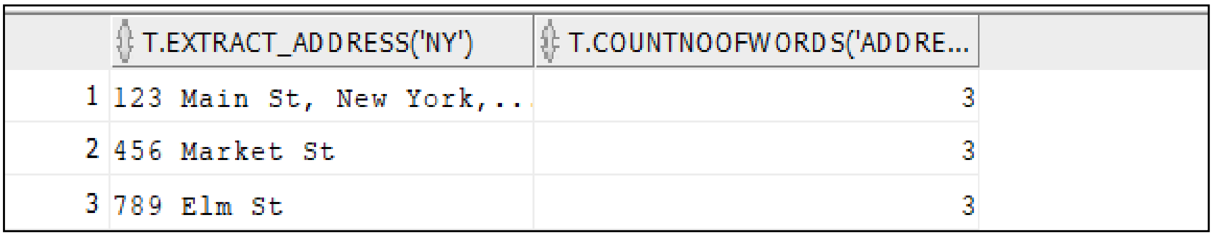


```
CREATE OR REPLACE TYPE BODY address_t AS
  MEMBER FUNCTION extract_address(keyword VARCHAR2) RETURN VARCHAR2 AS
    search_result VARCHAR2(100);
  BEGIN
    search_result := address;
    IF (INSTR(LOWER(address), LOWER(keyword)) > 0) OR
       (INSTR(LOWER(city), LOWER(keyword)) > 0) OR
       (INSTR(LOWER(state), LOWER(keyword)) > 0) THEN
      search_result := address || ', ' || city || ', ' || state;
    END IF;
    RETURN search_result;
  END extract_address;

  MEMBER FUNCTION countNoOfWords (field VARCHAR2) RETURN NUMBER AS
    word_count NUMBER;
    field_value VARCHAR2(100);
  BEGIN
    field_value := CASE LOWER(field)
                     WHEN 'address' THEN address
                     WHEN 'city' THEN city
                     WHEN 'state' THEN state
                     ELSE NULL
                   END;
    word_count := LENGTH(TRIM(field_value)) - LENGTH(REPLACE(field_value, ' ', '')) + 1;
    RETURN word_count;
  END countNoOfWords;
END;
```



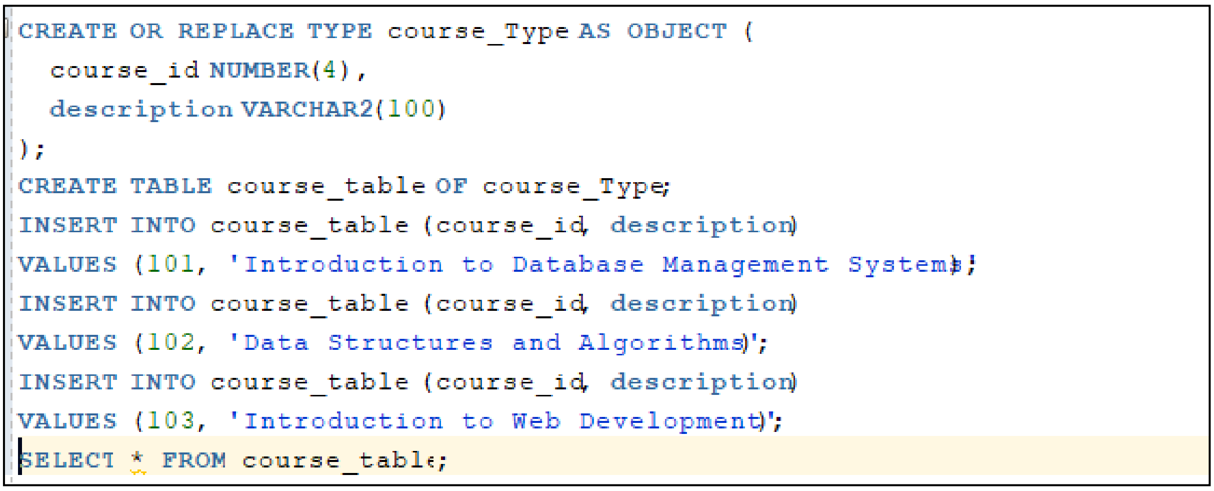
```
CREATE TABLE address_table OF address_t
INSERT INTO address_table (address, city, state, pincode)
VALUES ('123 Main St', 'New York', 'NY', 10001);
INSERT INTO address_table (address, city, state, pincode)
VALUES ('456 Market St', 'San Francisco', 'CA', 94102);
INSERT INTO address_table (address, city, state, pincode)
VALUES ('789 Elm St', 'Chicago', 'IL', 60601);

SELECT t.extract_address('NY'), t.countNoOfWords('address')
FROM address_table t;
```

Result :



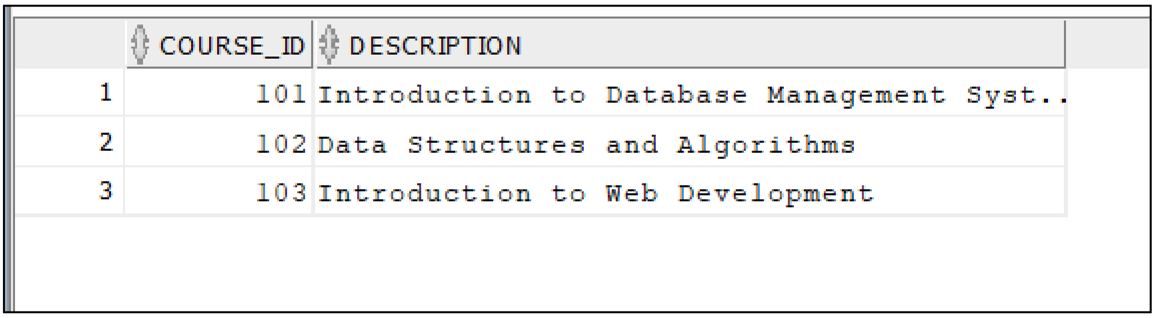
	T.EXTRACT_ADDRESS('NY')	T.COUNTNOOFWORDS('ADDRE...
1	123 Main St, New York,...	3
2	456 Market St	3
3	789 Elm St	3

3. **Create a user defined data type course_Type with 2 attributes course_id, description : i. Create an object table based on the type created. j. Insert rows into the table Demonstrate the working with different data sets**



```
CREATE OR REPLACE TYPE course_Type AS OBJECT (
  course_id NUMBER(4),
  description VARCHAR2(100)
);
CREATE TABLE course_table OF course_Type;
INSERT INTO course_table (course_id, description)
VALUES (101, 'Introduction to Database Management Systems');
INSERT INTO course_table (course_id, description)
VALUES (102, 'Data Structures and Algorithms');
INSERT INTO course_table (course_id, description)
VALUES (103, 'Introduction to Web Development');
SELECT * FROM course_table;
```

Result :



	COURSE_ID	DESCRIPTION
1	101	Introduction to Database Management Syst..
2	102	Data Structures and Algorithms
3	103	Introduction to Web Development