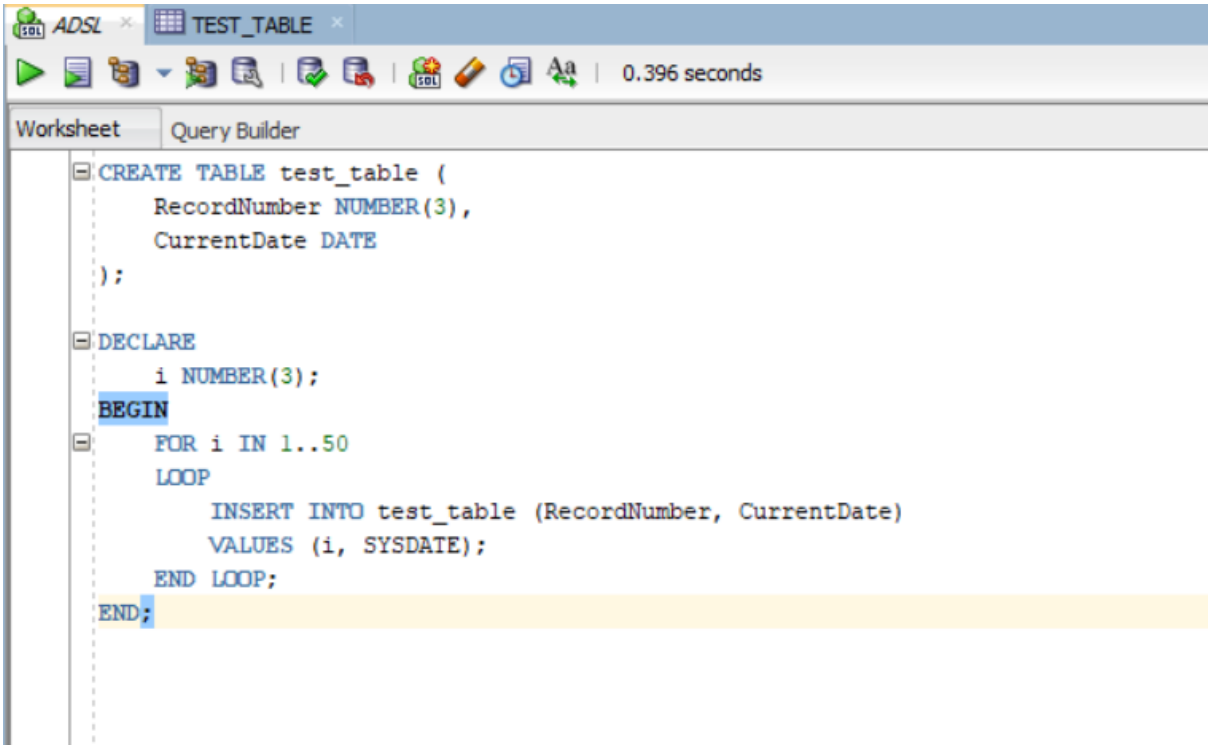
TY B.Tech. (CSE) – II [2022-23]

5CS372 : Advanced Database System Lab.

Assignment No. 1

I. PL / SQL Review :

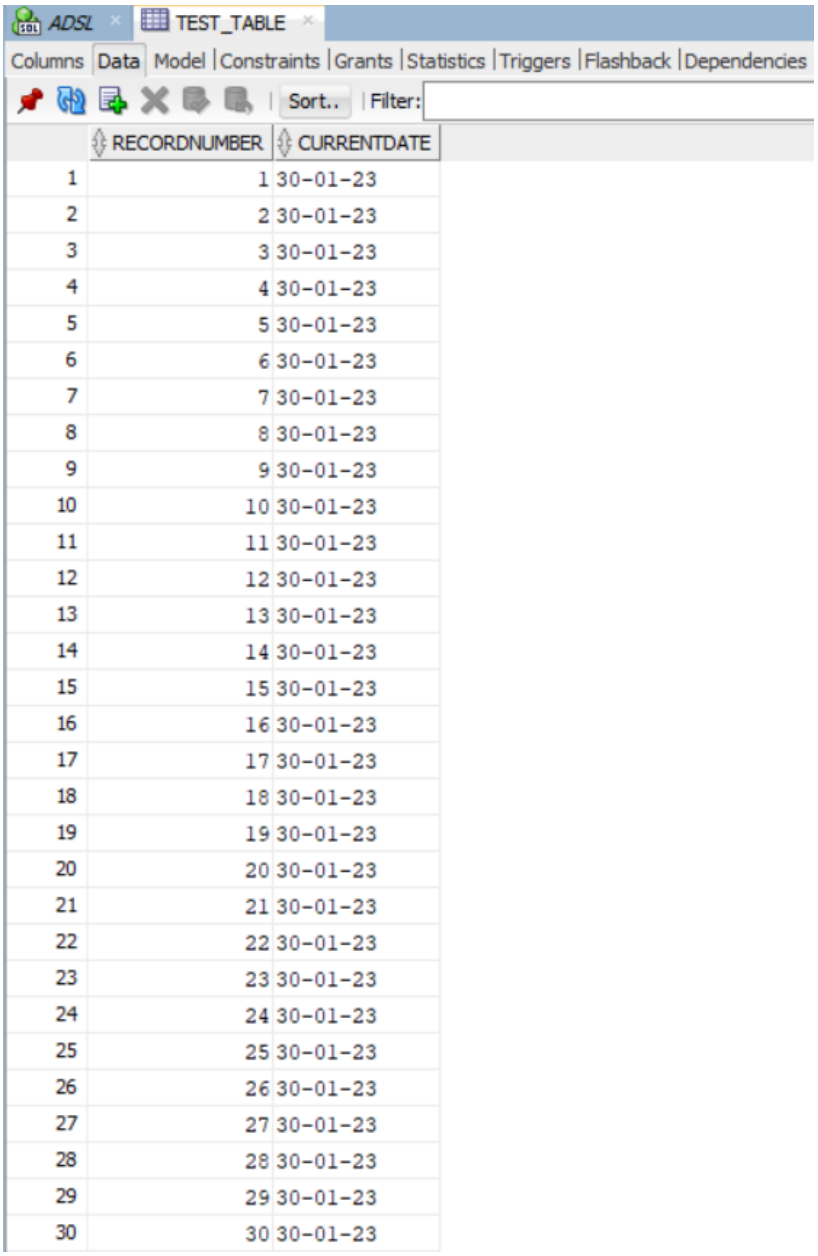
a) Create a table called test_table with 2 columns RecordNumber (type : Number(3)) and currentDate (type : Date)). Write PL/SQL block which will insert 50 records into test_table. Insert the current date value into the table.




```
CREATE TABLE test_table (
   RecordNumber NUMBER(3),
   CurrentDate DATE
);


DECLARE
   i NUMBER(3);
BEGIN
   FOR i IN 1..50
   LOOP
      INSERT INTO test_table (RecordNumber, CurrentDate)
```

```
    VALUES (i, SYSDATE);
  END LOOP;
END;
```

test_table data:

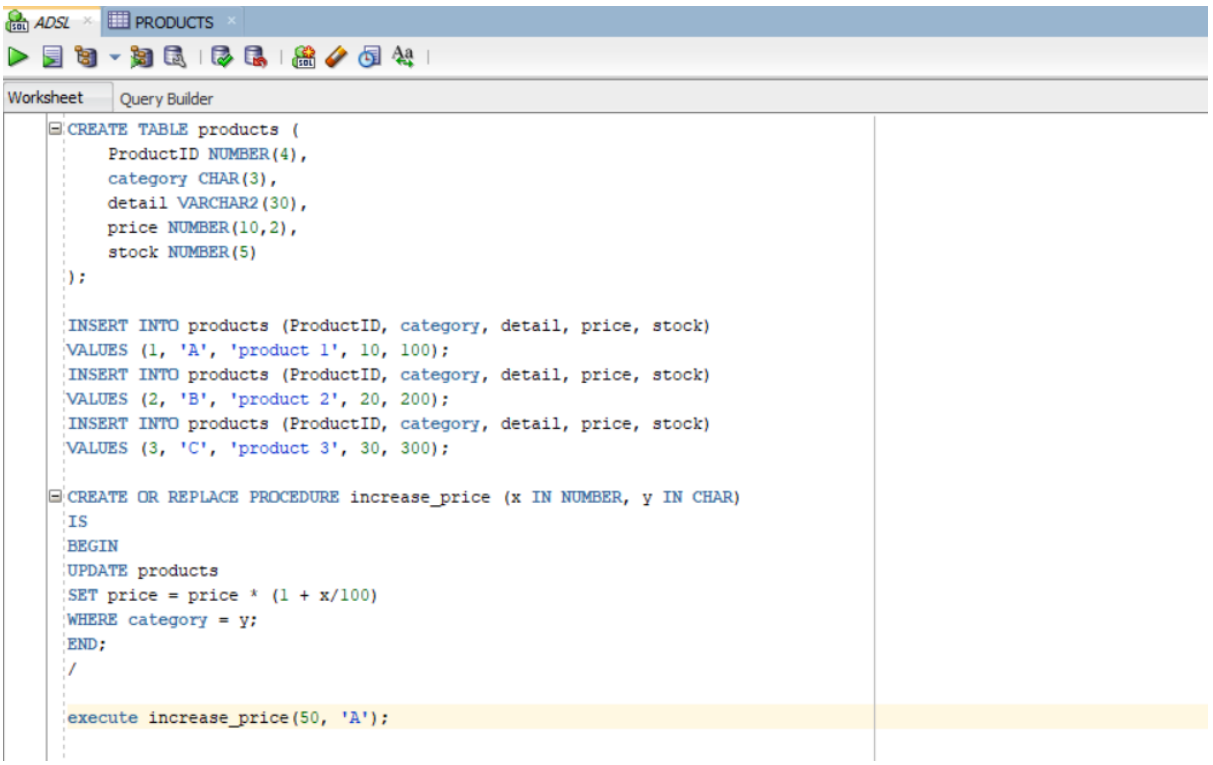


	RECORDNUMBER	CURRENTDATE
1	1	30-01-23
2	2	30-01-23
3	3	30-01-23
4	4	30-01-23
5	5	30-01-23
6	6	30-01-23
7	7	30-01-23
8	8	30-01-23
9	9	30-01-23
10	10	30-01-23
11	11	30-01-23
12	12	30-01-23
13	13	30-01-23
14	14	30-01-23
15	15	30-01-23
16	16	30-01-23
17	17	30-01-23
18	18	30-01-23
19	19	30-01-23
20	20	30-01-23
21	21	30-01-23
22	22	30-01-23
23	23	30-01-23
24	24	30-01-23
25	25	30-01-23
26	26	30-01-23
27	27	30-01-23
28	28	30-01-23
29	29	30-01-23
30	30	30-01-23

shorturl.at/iszBP

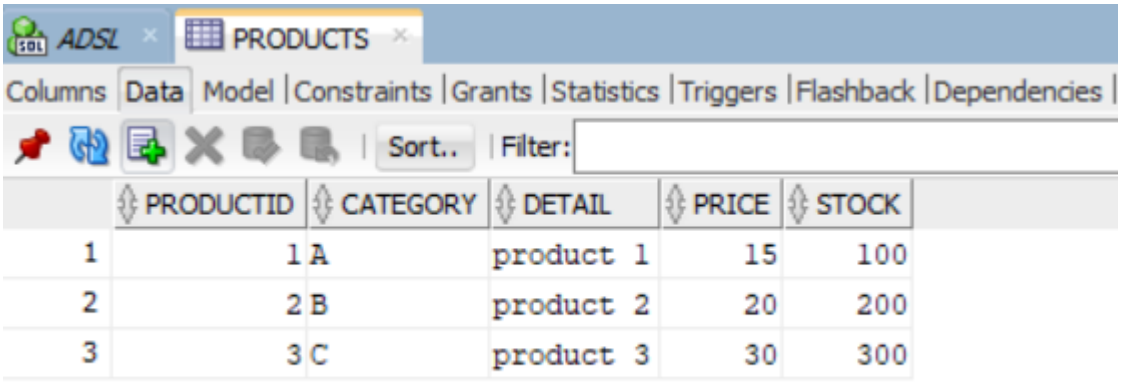
31	31	30-01-23
32	32	30-01-23
33	33	30-01-23
34	34	30-01-23
35	35	30-01-23
36	36	30-01-23
37	37	30-01-23
38	38	30-01-23
39	39	30-01-23
40	40	30-01-23
41	41	30-01-23
42	42	30-01-23
43	43	30-01-23
44	44	30-01-23
45	45	30-01-23
46	46	30-01-23
47	47	30-01-23
48	48	30-01-23
49	49	30-01-23
50	50	30-01-23

b) Create a 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 the 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, 'A', 'product 1', 10, 100);
INSERT INTO products (ProductID, category, detail, price, stock)
VALUES (2, 'B', 'product 2', 20, 200);
INSERT INTO products (ProductID, category, detail, price, stock)
VALUES (3, 'C', 'product 3', 30, 300);

CREATE OR REPLACE PROCEDURE increase_price (x IN NUMBER, y IN CHAR)
IS
BEGIN
UPDATE products
SET price = price * (1 + x/100)
WHERE category = y;
END;
/

execute increase_price(50, 'A');
```

Updated table:

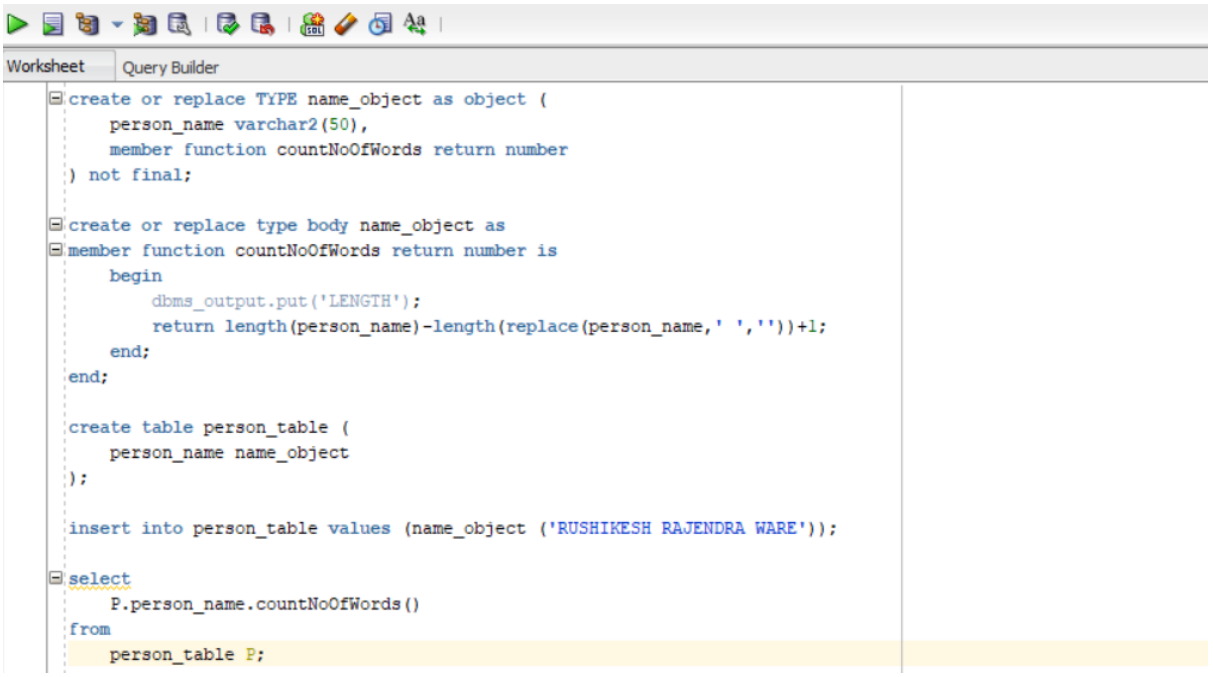


	PRODUCTID	CATEGORY	DETAIL	PRICE	STOCK
1	1	A	product 1	15	100
2	2	B	product 2	20	200
3	3	C	product 3	30	300

II. Object Relational Databases:

a) Create Object Table containing field “name” of size 50 characters and member function “countNoOfWords” which returns the no. of words in “name” field.

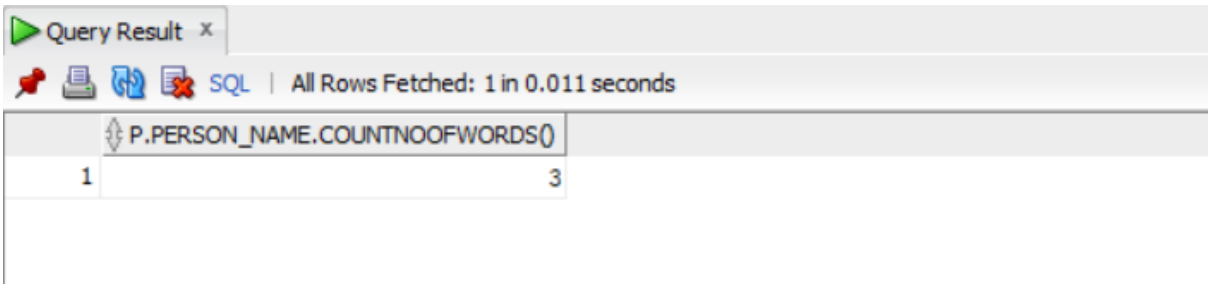
Demonstrate the working by entering different data.




```
create or replace TYPE name_object as object (
   person_name varchar2(50),
   member function countNoOfWords return number
) not final;
```

```
create or replace type body name_object as
member function countNoOfWords return number is
   begin
      dbms_output.put('LENGTH');
      return length(person_name)-length(replace(person_name,' ',''))+1;
   end;
end;

create table person_table (
   person_name name_object
);

insert into person_table values (name_object ('RUSHIKESH RAJENDRA WARE'));

select
   P.person_name.countNoOfWords()
from
   person_table P;
```

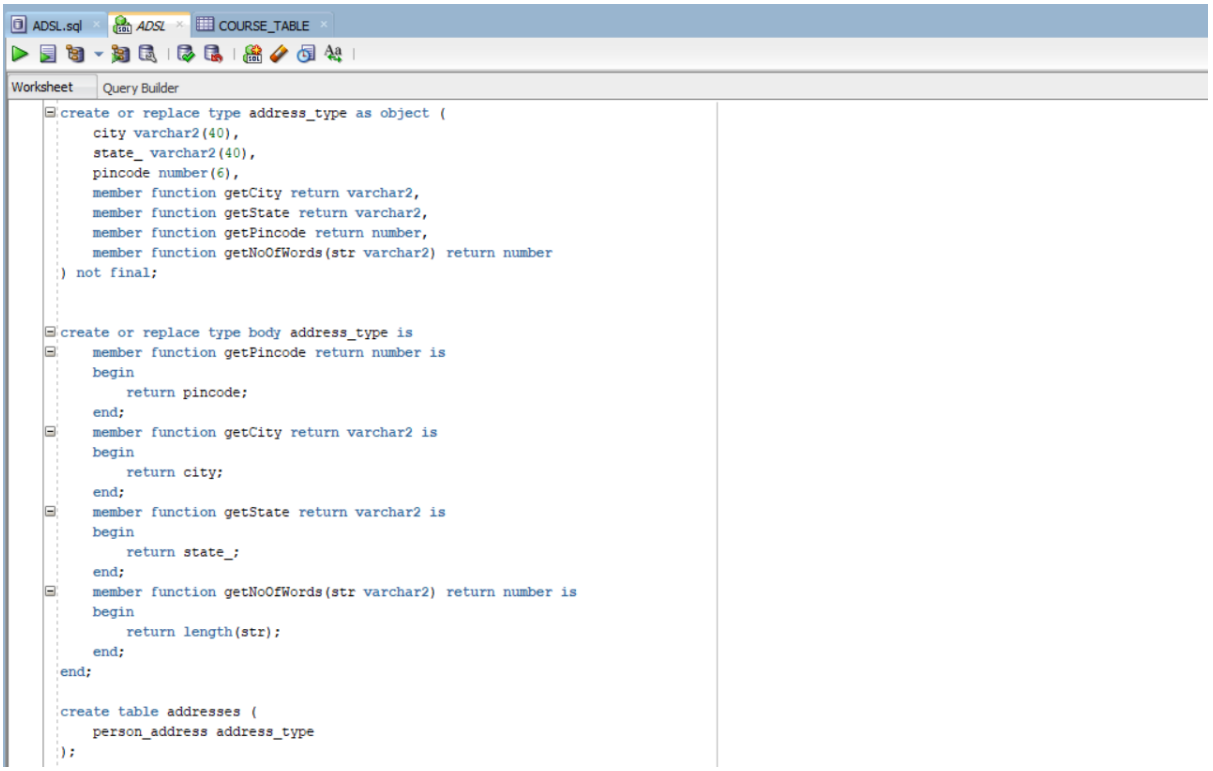
Output:

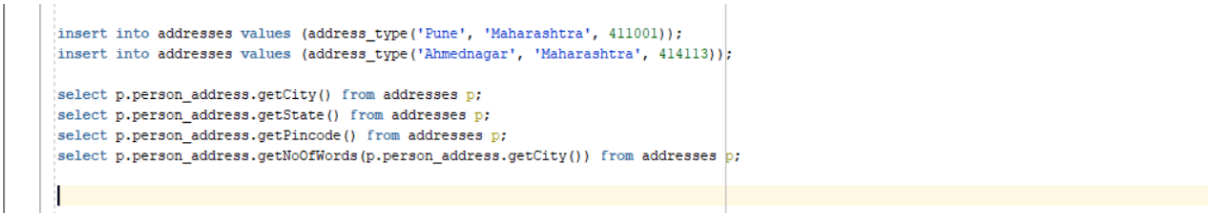


b) Create an address type with the following attributes : address, city, state & pincode. Include the following methods

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)

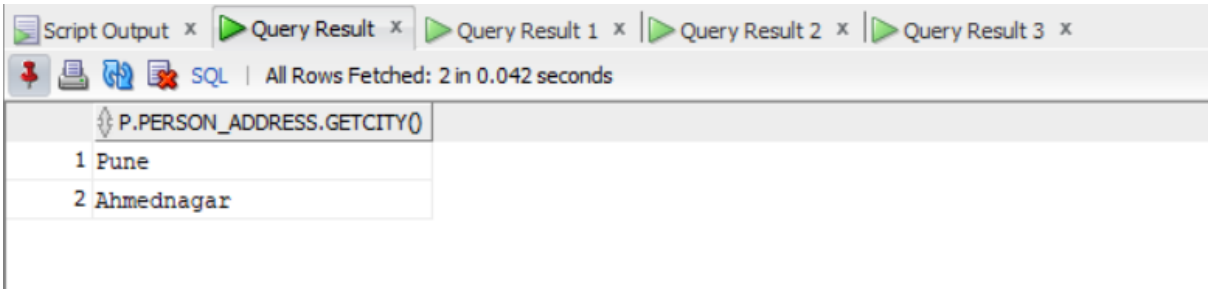





```
create or replace type address_type as object (
    city varchar2(40),
    state_ varchar2(40),
    pincode number(6),
    member function getCity return varchar2,
    member function getState return varchar2,
    member function getPincode return number,
    member function getNoOfWords(str varchar2) return number
) not final;
```

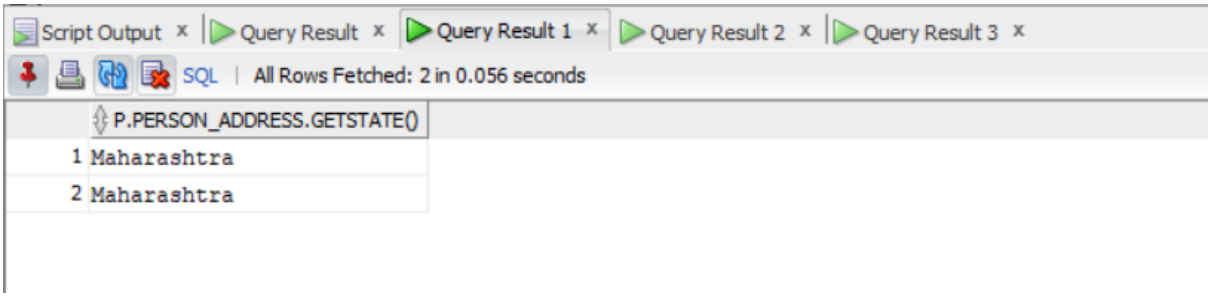
```
create or replace type body address_type is
   member function getPincode return number is
   begin
      return pincode;
   end;
   member function getCity return varchar2 is
   begin
      return city;
   end;
   member function getState return varchar2 is
   begin
      return state_;
   end;
   member function getNoOfWords(str varchar2) return number is
   begin
      return length(str);
   end;
end;

create table addresses (
   person_address address_type
);

insert into addresses values (address_type('Pune', 'Maharashtra', 411001));
insert into addresses values (address_type('Ahmednagar', 'Maharashtra', 414113));

select p.person_address.getCity() from addresses p;
select p.person_address.getState() from addresses p;
select p.person_address.getPincode() from addresses p;
select p.person_address.getNoOfWords(p.person_address.getCity()) from addresses p;
```

Output:


Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3

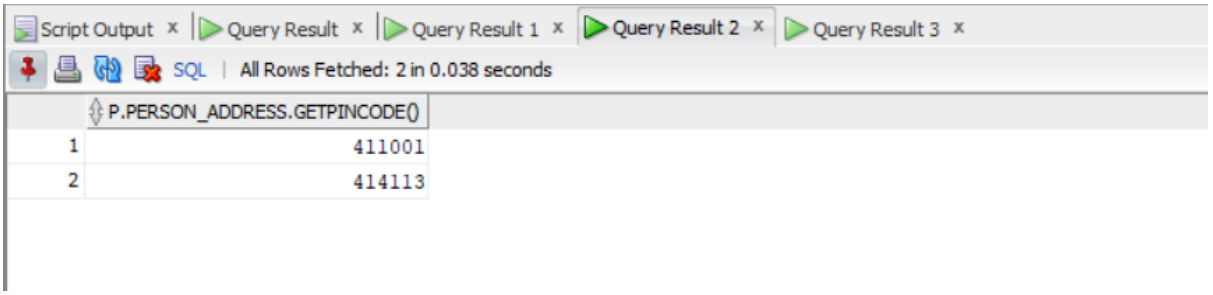
SQL | All Rows Fetched: 2 in 0.042 seconds

	P.PERSON_ADDRESS.GETCITY()
1	Pune
2	Ahmednagar


Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3

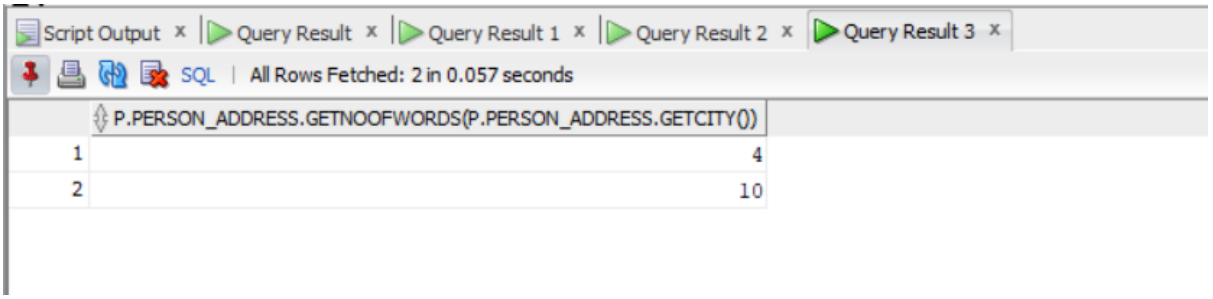
SQL | All Rows Fetched: 2 in 0.056 seconds

	P.PERSON_ADDRESS.GETSTATE()
1	Maharashtra
2	Maharashtra


Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3

SQL | All Rows Fetched: 2 in 0.038 seconds

	P.PERSON_ADDRESS.GETPINCODE()
1	411001
2	414113


Script Output | Query Result | Query Result 1 | Query Result 2 | Query Result 3

SQL | All Rows Fetched: 2 in 0.057 seconds

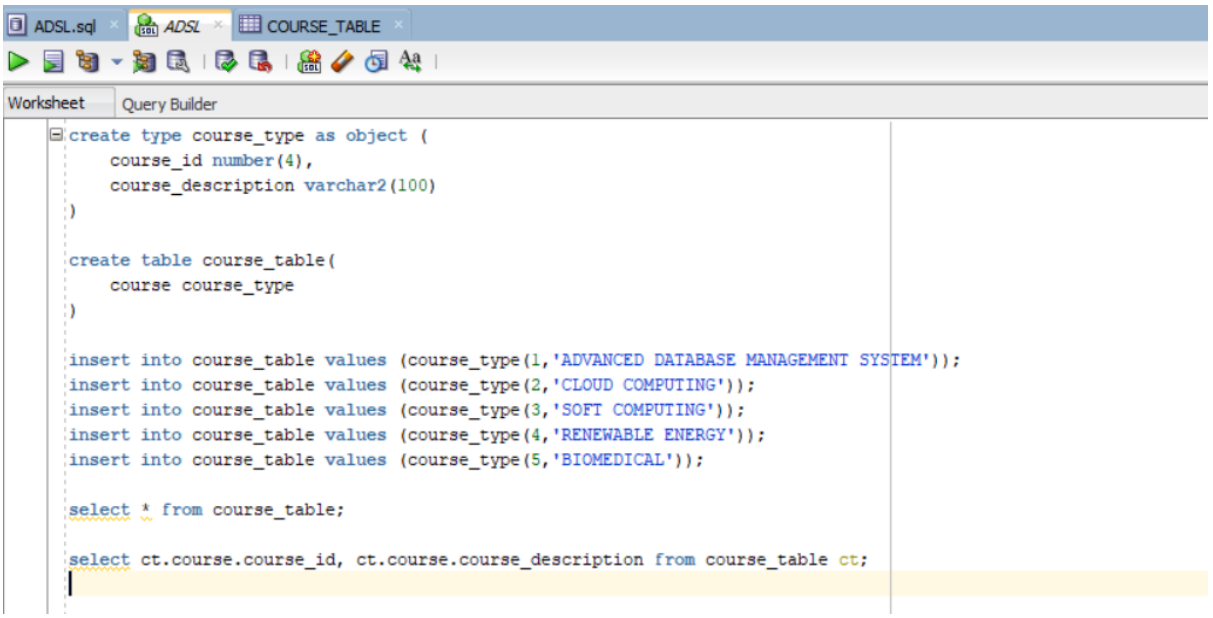
	P.PERSON_ADDRESS.GETNOOFWORDS(P.PERSON_ADDRESS.GETCITY())
1	4
2	10

c) Create a user defined data type course_Type with 2 attributes course_id, description :

i. Create an object table based on the type created.

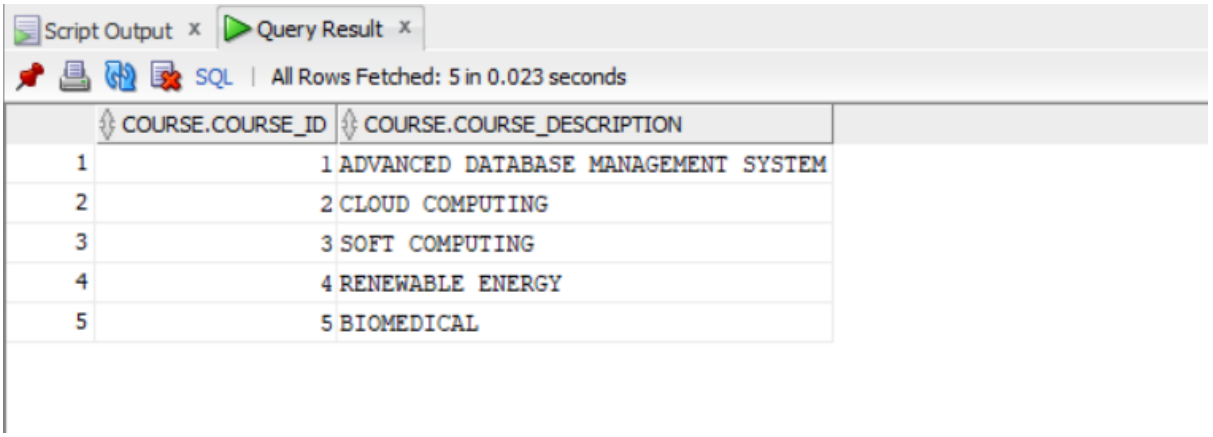
j. Insert rows into the table

Demonstrate the working with different data sets




```
create type course_type as object (
    course_id number(4),
    course_description varchar2(100)
)

create table course_table(
    course course_type
)

insert into course_table values (course_type(1,'ADVANCED DATABASE MANAGEMENT SYSTEM'));
insert into course_table values (course_type(2,'CLOUD COMPUTING'));
insert into course_table values (course_type(3,'SOFT COMPUTING'));
insert into course_table values (course_type(4,'RENEWABLE ENERGY'));
insert into course_table values (course_type(5,'BIOMEDICAL'));

select * from course_table;

select ct.course.course_id, ct.course.course_description from course_table ct;
```

Output:



	COURSE.COURSE_ID	COURSE.COURSE_DESCRIPTION
1	1	ADVANCED DATABASE MANAGEMENT SYSTEM
2	2	CLOUD COMPUTING
3	3	SOFT COMPUTING
4	4	RENEWABLE ENERGY
5	5	BIOMEDICAL