Date:31.01.2022

Third Year B. Tech., Sem VI 2021-22

Advanced Database System Lab

Assignment submission

PRN No: 2019BTECS00064

Full name: Kunal Santosh Kadam

Batch: T2

Assignment: 2

Title of assignment: PL/SQL Review And Object Relational Database

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.

Ans

Create table test_table:

2019BTECS00064

Worksheet | Query Builder

```
CREATE TABLE test_table
(
    recordnumber NUMBER(3),
    currentate DATE
);
```

Script Output

Task completed in 0.238 seconds

```
Error starting at line : 5 in command -
)
Error report -
Unknown Command


Table TEST_TABLE created.
```

PL/SQL statement to add 50 entries:

2019BTECS00064

2019BTECS00064

Worksheet | Query Builder

```
DECLARE
    i NUMBER(2);
    currentdate DATE;
BEGIN
    FOR i IN 1 .. 50 LOOP
        INSERT INTO test_table VALUES(i,sysdate);
    END LOOP;
END;
```

Script Output

Task completed in 0.531 seconds

```
Error starting at line : 5 in command -
)
Error report -
Unknown Command


Table TEST_TABLE created.


PL/SQL procedure successfully completed.
```

Table Values:

2019BTECS00064

Worksheet | Query Builder

```
SELECT * FROM test_table;
```

Script Output | Query Result

SQL | All Rows Fetched: 50 in 0.191 seconds

	RECORDNUMBER	CURRENTATE
1	1	05-02-22
2	2	05-02-22
3	3	05-02-22
4	4	05-02-22
5	5	05-02-22
6	6	05-02-22
7	7	05-02-22
8	8	05-02-22
9	9	05-02-22
10	10	05-02-22
11	11	05-02-22
12	12	05-02-22
13	13	05-02-22
14	14	05-02-22
15	15	05-02-22
16	16	05-02-22
17	17	05-02-22
18	18	05-02-22
19	19	05-02-22
20	20	05-02-22
21	21	05-02-22
22	22	05-02-22
23	23	05-02-22
24	24	05-02-22
25	25	05-02-22
26	26	05-02-22
27	27	05-02-22
28	28	05-02-22
29	29	05-02-22
30	30	05-02-22
31	31	05-02-22
32	32	05-02-22
33	33	05-02-22

Script Output × Query Result ×

SQL | All Rows Fetched: 50 in 0.191 seconds

	RECORDNUMBER	CURRENTATE
31	31	05-02-22
32	32	05-02-22
33	33	05-02-22
34	34	05-02-22
35	35	05-02-22
36	36	05-02-22
37	37	05-02-22
38	38	05-02-22
39	39	05-02-22
40	40	05-02-22
41	41	05-02-22
42	42	05-02-22
43	43	05-02-22
44	44	05-02-22
45	45	05-02-22
46	46	05-02-22
47	47	05-02-22
48	48	05-02-22
49	49	05-02-22
50	50	05-02-22

b. 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.

Ans

Create Products table:

2019BTECS00064

Worksheet | Query Builder

```
CREATE TABLE products
(
    product_id NUMBER(4),
    category VARCHAR(3),
    price NUMBER(10),
    stocknumber NUMBER(5)
);
```

Script Output

Task completed in 0.188 seconds

```
Table PRODUCTS created.
```

Table Data:

2019BTECS00064

Worksheet | Query Builder

```
SELECT * FROM products;
```

Script Output | Query Result

SQL | All Rows Fetched: 15 in 0.009 seconds

	PRODUCT_ID	CATEGORY	PRICE	STOCKNUMBER
1	1	A	123	12
2	2	B	125	10
3	3	C	134	45
4	4	D	160	23
5	5	A	20	23
6	6	D	200	43
7	7	B	344	56
8	8	C	56	6
9	9	A	344	2
10	10	B	790	45
11	11	C	34	56
12	12	A	23	67
13	13	D	345	34
14	14	B	156	23
15	15	D	134	56

SQL Procedure to update price:

2019BTECS00064

Worksheet | Query Builder

```
CREATE OR REPLACE PROCEDURE updatePrice(x IN NUMBER, y IN VARCHAR2) IS
BEGIN
    UPDATE products
    SET price = price + (price * x) / 100
    WHERE category = y;
END;
```

Query Result | Script Output

Task completed in 1.138 seconds

```
Procedure UPDATEPRICE compiled
```

Invoking updatePrice:

2019BTECS00064

Worksheet | Query Builder

```
DECLARE
    x NUMBER;
    y VARCHAR(3);
BEGIN
    x := 10;
    y := 'B';
    updatePrice(x,y);
    dbms_output.put_line('Prices updated');
END;
```

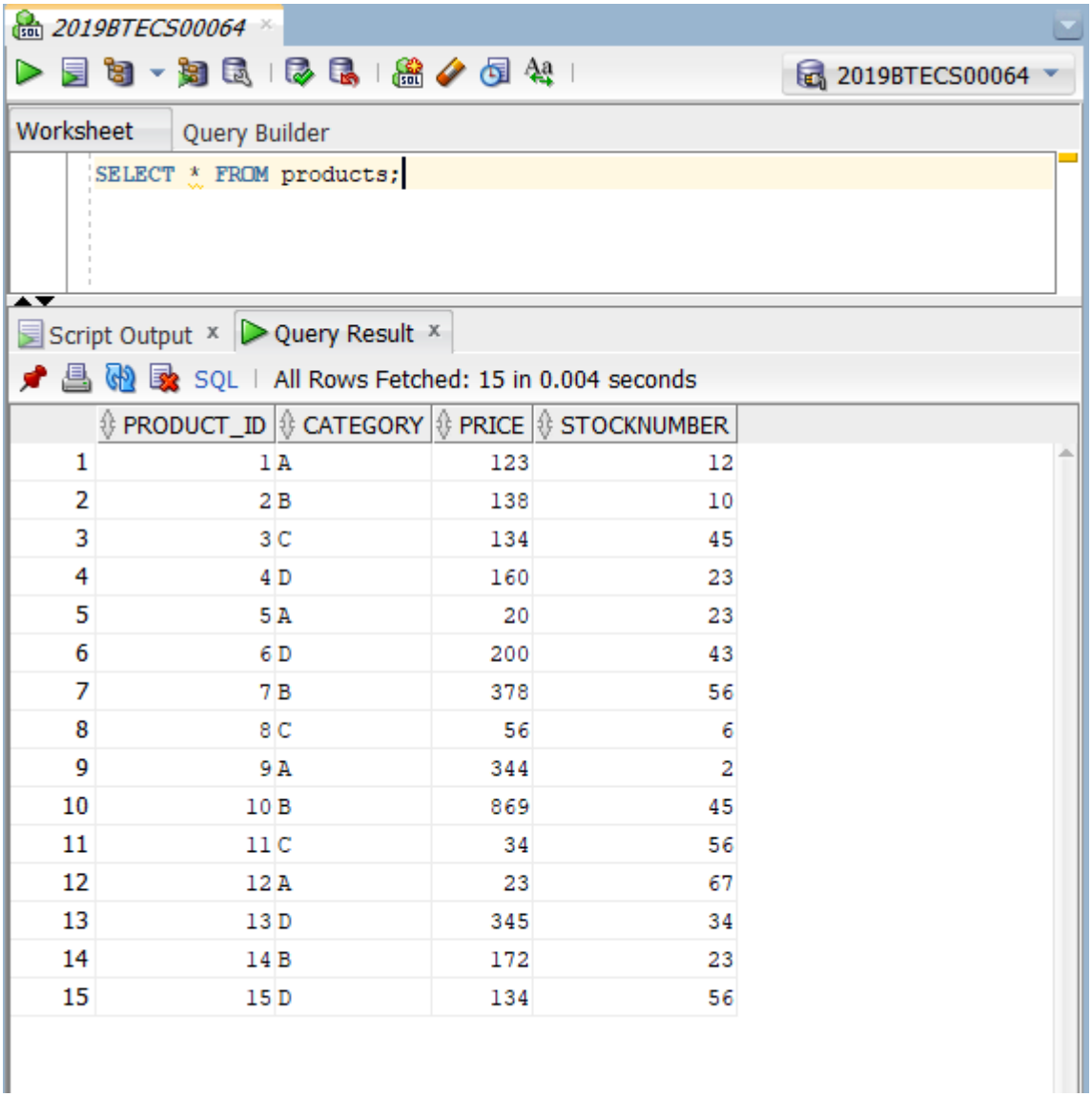
Query Result | Script Output

Task completed in 0.167 seconds

```
Procedure UPDATEPRICE compiled


PL/SQL procedure successfully completed.
```

Table after updating prices:



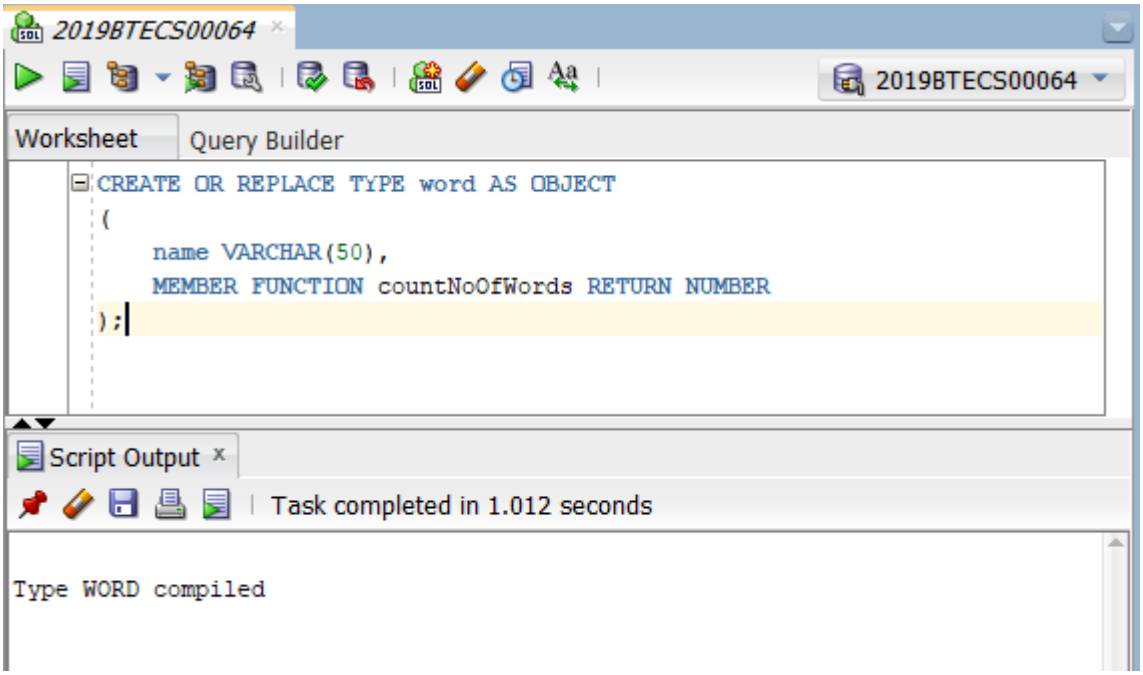
	PRODUCT_ID	CATEGORY	PRICE	STOCKNUMBER
1	1	A	123	12
2	2	B	138	10
3	3	C	134	45
4	4	D	160	23
5	5	A	20	23
6	6	D	200	43
7	7	B	378	56
8	8	C	56	6
9	9	A	344	2
10	10	B	869	45
11	11	C	34	56
12	12	A	23	67
13	13	D	345	34
14	14	B	172	23
15	15	D	134	56

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.

Ans:

Create the type “word”:



Create the body of the type “word”:

2019BTECS00064

Worksheet | Query Builder

```
CREATE OR REPLACE TYPE BODY word AS
    MEMBER FUNCTION countNoOfWords
    RETURN NUMBER IS
    wordcount NUMBER(20):= 1;
    ch char;
    BEGIN
        FOR i IN 1 .. Length(name) LOOP
            ch := SUBSTR(name, i, 1);
            IF ch = ' ' or ch = '.' or ch = ',' then
                wordcount := wordcount + 1;
            END IF;
        END LOOP;
        RETURN wordcount;
    END countNoOfWords;
END;
```

Script Output

Task completed in 0.245 seconds

```
Type WORD compiled


Type Body WORD compiled

LINE/COL  ERROR
--------- -------------------------------------------------------------
7/9       PL/SQL: Statement ignored
7/23      PLS-00201: identifier 'L' must be declared
Errors: check compiler log

Type Body WORD compiled
```

Create table of word type:

2019BTECS00064

Worksheet | Query Builder

```
CREATE TABLE word_t of word;
INSERT INTO word_t VALUES(('Kunal Kadam'));
INSERT INTO word_t VALUES(('2019BTECS00064'));
INSERT INTO word_t VALUES(('I am studing at Third Year'));
INSERT INTO word_t VALUES(('Course is Advanced Database System'));
INSERT INTO word_t VALUES(('Example of creating object table'));
```

Script Output

Task completed in 0.443 seconds

```
Type WORD compiled


Type Body WORD compiled

LINE/COL  ERROR
--------- -------------------------------------------------------------
7/9       PL/SQL: Statement ignored
7/23      PLS-00201: identifier 'L' must be declared
Errors: check compiler log

Type Body WORD compiled


Table WORD_T created.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.
```

Query to get count of words:

2019BTECS00064

Worksheet | Query Builder

```
SELECT
VALUE(wordtable) .name AS sentence,
VALUE(wordtable) .countNoOfWords() as wordcount
FROM word_t wordtable;
```

Script Output | Query Result

All Rows Fetched: 5 in 0.024 seconds

	SENTENCE	WORDCOUNT
1	Kunal Kadam	2
2	2019BTECS00064	1
3	I am studing at Third Year	6
4	Course is Advanced Database System	5
5	Example of creating object table	5

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

   i. To extract the addresses based on given keyword.

   ii. To return the no. of words in each given field (method should accept the name of attribute/field)

Ans

Create Type "address":

2019BTECS00064

Worksheet | Query Builder

```oracle_plsql
CREATE TYPE address AS OBJECT
(
    short_address VARCHAR2(100),
    city VARCHAR2(20),
    state VARCHAR2(20),
    pincode NUMBER(7),

    MEMBER FUNCTION getAddress(key_val VARCHAR) RETURN NUMBER,
    MEMBER FUNCTION no_of_words(type VARCHAR) RETURN NUMBER
);

CREATE OR REPLACE TYPE BODY address AS
    MEMBER FUNCTION getAddress(key_val in VARCHAR)
    RETURN NUMBER IS
    no_word NUMBER(20) := 0;
    s CHAR := ' ';
    str VARCHAR(20) := ' ';
    flg NUMBER(1) := 0;
    j NUMBER(10) := 0;
```

Script Output

Task completed in 0.119 seconds

```
Type ADDRESS compiled


Type Body ADDRESS compiled
```

Create body for the “address”:

2019BTECS00064

Worksheet | Query Builder

```
CREATE OR REPLACE TYPE BODY address AS
    MEMBER FUNCTION getAddress(key_val in VARCHAR)
    RETURN NUMBER IS
    no_word NUMBER(20) := 0;
    s CHAR := ' ';
    str VARCHAR(20) := ' ';
    flg NUMBER(1) := 0;
    j NUMBER(10) := 0;
    k NUMBER(10) := 0;
    BEGIN
        FOR i IN 1 .. Length(short_address) LOOP
            s := SUBSTR(short_address, i, 1);
            IF s=' ' OR s='.' OR s=',' THEN
                str := SUBSTR(short_address, j, k);
                j := i + 1;
                k := -1;
                IF str = key_val THEN
                    DBMS_OUTPUT.PUT_LINE('Address: '||short_address);
                    flg := 1;
                    return flg;
                END IF;
            END IF;
            k := k+1;
        END LOOP;
        RETURN flg;
    END getAddress;
    MEMBER FUNCTION no_of_words(type IN VARCHAR)
    RETURN NUMBER IS
    wordcount NUMBER(20) := 1;
    target VARCHAR(100);
    ch CHAR;
    BEGIN
        IF TYPE = 'short_address' THEN
            target := short_address;
        END IF;
        IF TYPE = 'city' THEN
            target := city;
        END IF;
        IF TYPE = 'state' THEN
            target := state;
        END IF;
```

2019BTECS00064

Worksheet | Query Builder

```
    BEGIN
        FOR i IN 1 .. Length(short_address) LOOP
            s := SUBSTR(short_address, i, 1);
            IF s=' ' OR s='.' OR s=',' THEN
                str := SUBSTR(short_address, j, k);
                j := i + 1;
                k := -1;
                IF str = key_val THEN
                    DBMS_OUTPUT.PUT_LINE('Address: '||short_address);
                    flg := 1;
                    return flg;
                END IF;
            END IF;
            k := k+1;
        END LOOP;
        RETURN flg;
    END getAddress;
    MEMBER FUNCTION no_of_words(type IN VARCHAR)
    RETURN NUMBER IS
    wordcount NUMBER(20) := 1;
    target VARCHAR(100);
    ch CHAR;
    BEGIN
        IF TYPE = 'short_address' THEN
            target := short_address;
        END IF;
        IF TYPE = 'city' THEN
            target := city;
        END IF;
        IF TYPE = 'state' THEN
            target := state;
        END IF;
        FOR i in 1 .. Length(target) LOOP
            ch := SUBSTR(target, i, 1);
            IF ch = ' ' OR ch = '.' OR ch = ',' THEN
                wordcount := wordcount + 1;
            END IF;
        END LOOP;
        RETURN wordcount;
    END no_of_words;
END;
```

Table Entries:

2019BTECS00064

Worksheet | Query Builder

```
CREATE TABLE address_table of address;
DESC address_table;

INSERT INTO address_table VALUES('Ankush Nagar','Nagpur','Maharashtra', 415002);
INSERT INTO address_table VALUES('Friends Colony','Beed','Maharashtra', 415102);
INSERT INTO address_table VALUES('Sayaji Nagar','Kolhapur','Maharashtra', 415012);
INSERT INTO address_table VALUES('Mahada Colony','Nagpur','Maharashtra', 415104);
INSERT INTO address_table VALUES('Shivaj Nagar','Kolhapur','Maharashtra', 415013);
```

Script Output

Task completed in 0.076 seconds

```
Table ADDRESS_TABLE created.

Name           Null? Type
-------------- ----- -------------
SHORT_ADDRESS        VARCHAR2(100)
CITY                 VARCHAR2(20)
STATE                VARCHAR2(20)
PINCODE              NUMBER(7)

1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.
```

Word count based on attribute name:

2019BTECS00064

Worksheet | Query Builder

```
SELECT ad.short_address
FROM address_table ad WHERE
VALUE (ad).getAddress('Nagar') = 2;

SELECT ad.city,
VALUE(ad).no_of_words('city') as wordCount
FROM address_table ad;
```

Script Output | Query Result 1

All Rows Fetched: 5 in 0.043 seconds

	CITY	WORDCOUNT
1	Nagpur	1
2	Beed	1
3	Kolhapur	1
4	Nagpur	1
5	Kolhapur	1

2019BTECS00064

Worksheet | Query Builder

```
SELECT ad.short_address
FROM address_table ad WHERE
VALUE (ad).getAddress('Nagar') = 2;

SELECT ad.short_address,
VALUE(ad).no_of_words('short_address') as wordCount
FROM address_table ad;
```

Script Output | Query Result 1 | Query Result 2

All Rows Fetched: 5 in 0.008 seconds

	SHORT_ADDRESS	WORDCOUNT
1	Ankush Nagar	2
2	Friends Colony	2
3	Sayaji Nagar	2
4	Mahada Colony	2
5	Shivaj Nagar	2

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

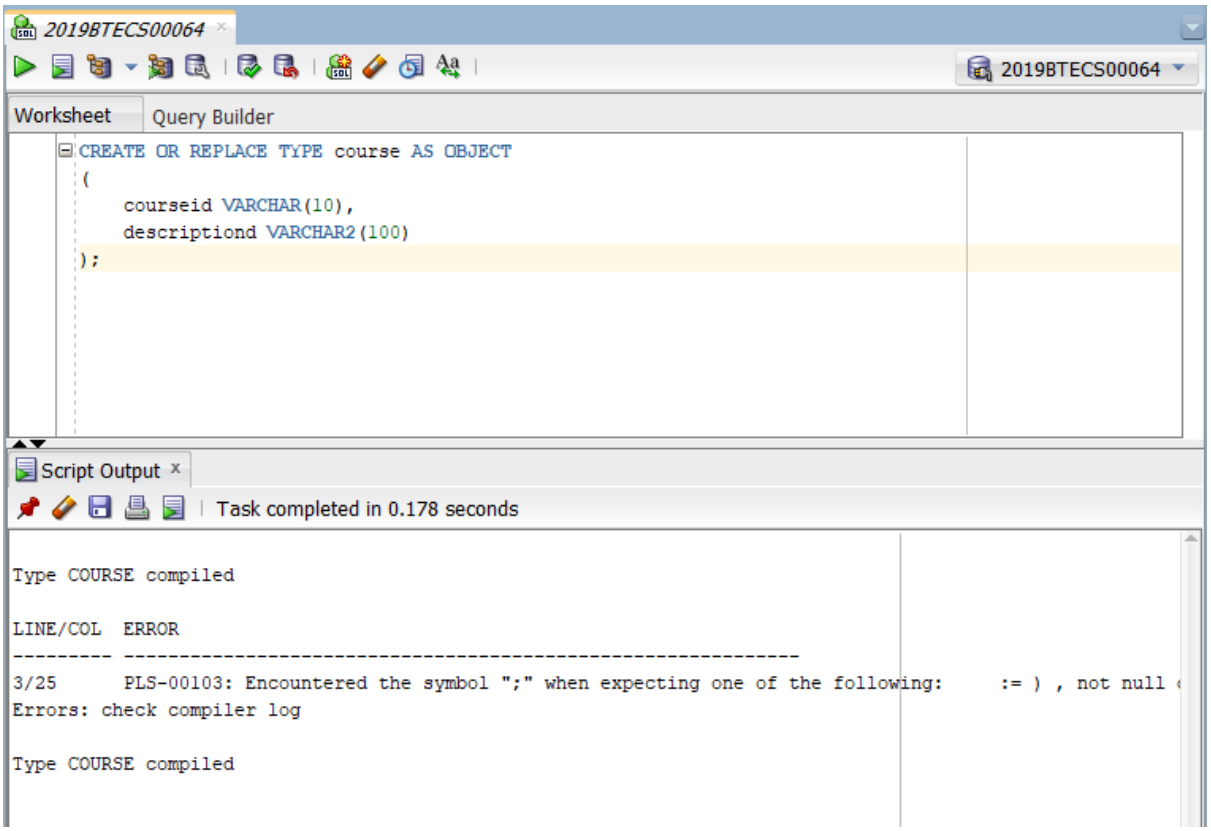
   i. Create an object table based on the type created.

   ii. Insert rows into the table.

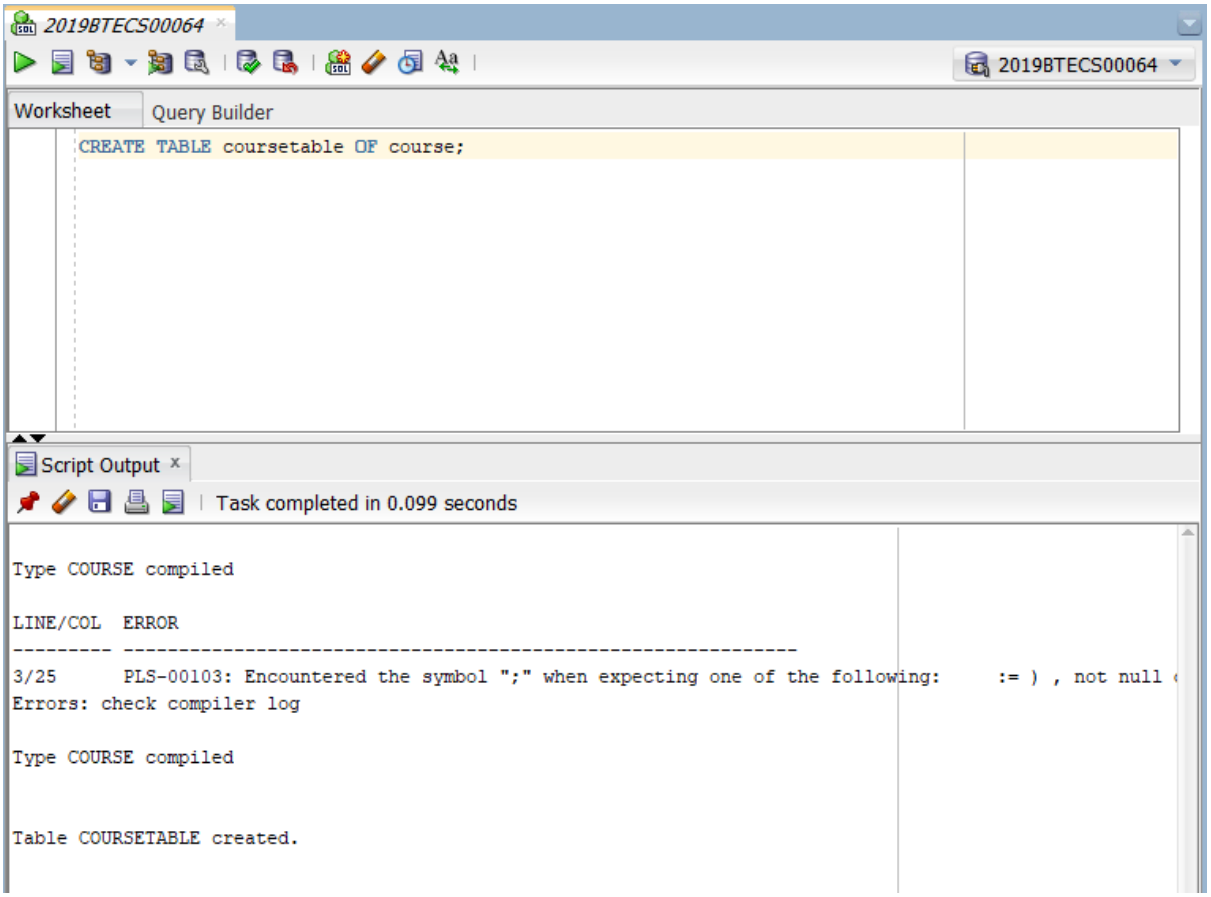
   Demonstrate the working with different data sets

Ans

Course Type:



Creating Table of Type Course:



Insert Values into Table:

2019BTECS00064

Worksheet | Query Builder

```
INSERT INTO coursetable VALUES('4CS301','Computer Networks');
INSERT INTO coursetable VALUES('4CS302','Data Structures');
INSERT INTO coursetable VALUES('4CS303','Design and Analysis of Algorithms');
INSERT INTO coursetable VALUES('4CS304','Advanced Database Engineering');
```

Script Output | Task completed in 0.07 seconds

```
Type COURSE compiled

LINE/COL  ERROR
--------- -------------------------------------------------------------
3/25      PLS-00103: Encountered the symbol ";" when expecting one of the following:     := ) , not null
Errors: check compiler log

Type COURSE compiled


Table COURSETABLE created.


1 row inserted.


1 row inserted.


1 row inserted.


1 row inserted.
```

2019BTECS00064

Worksheet | Query Builder

```
SELECT * from coursetable;
```

Script Output | Query Result

SQL | All Rows Fetched: 4 in 0.006 seconds

	COURSEID	DESCRIPTIOND
1	4CS301	Computer Networks
2	4CS302	Data Structures
3	4CS303	Design and Analysis of Algorithms
4	4CS304	Advanced Database Engineering