**Question 1**

CREATE TABLE called supplier with the following fields .

supplier\_id numeric(10) - > Primary key with constraint name

supplier\_name varchar2(50) - >unique name

contact\_name varchar2(50)

phone\_no varchar2(10)- >unique name

city varchar2(10)

Region –> should accept only ('N', 'NW', 'NE', 'S', 'SE', 'SW', 'W', 'E')

1. Insert 5 records
2. Display the details of the supplier who comes from Florida and their supplier id 500;
3. Add phone number in the supplier table using DDL command
4. Delete the unused column in the supplier table
5. Write a sql command to delete supplier table.
6. Create a view named supplier\_contact . Include supplier\_id,supplier\_name,phone\_no

/\* QUESTION 01 \*/

CREATE TABLE supplier

(supplier\_id NUMBER(10),

supplier\_name VARCHAR2(50) UNIQUE,

contact\_name VARCHAR2(50),

phone\_no VARCHAR2(10) UNIQUE,

city VARCHAR2(10),

region VARCHAR2(2),

CONSTRAINT supplier\_supplierid\_pk PRIMARY KEY(supplier\_id),

CONSTRAINT supplier\_region\_ck

CHECK (region IN ('N', 'NW', 'NE', 'S', 'SE', 'SW', 'W', 'E')));



--#1

INSERT INTO supplier VALUES (500, 'Kyle Lowry', 'Ayahna Cornish-Lowry', '123-1234', 'Florida', 'S');

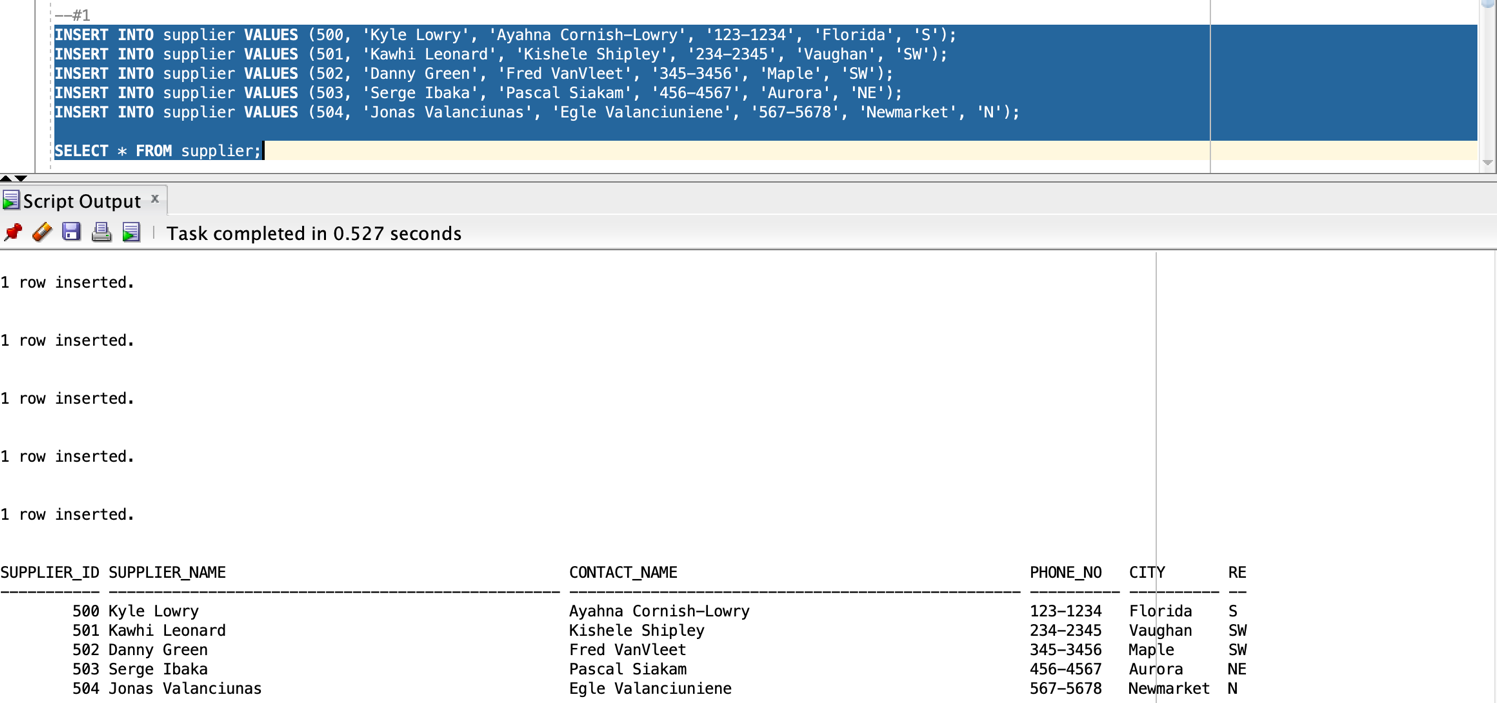
INSERT INTO supplier VALUES (501, 'Kawhi Leonard', 'Kishele Shipley', '234-2345', 'Vaughan', 'SW');

INSERT INTO supplier VALUES (502, 'Danny Green', 'Fred VanVleet', '345-3456', 'Maple', 'SW');

INSERT INTO supplier VALUES (503, 'Serge Ibaka', 'Pascal Siakam', '456-4567', 'Aurora', 'NE');

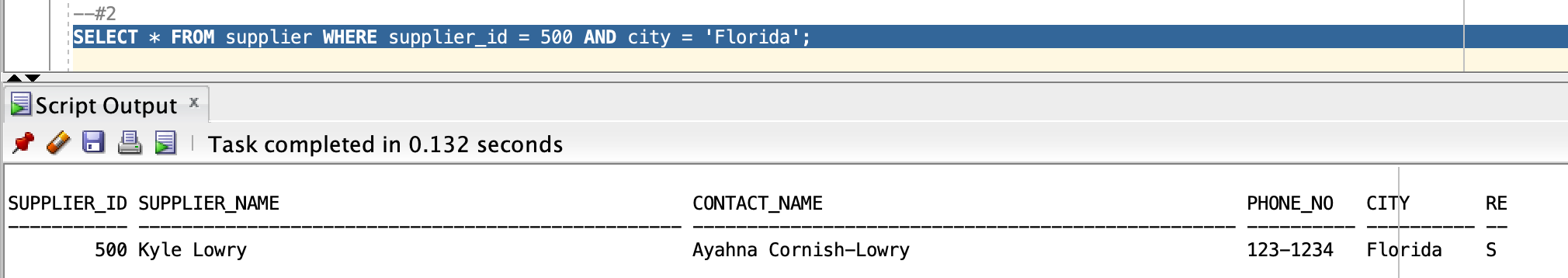
INSERT INTO supplier VALUES (504, 'Jonas Valanciunas', 'Egle Valanciuniene', '567-5678', 'Newmarket', 'N');

SELECT \* FROM supplier;



--#2

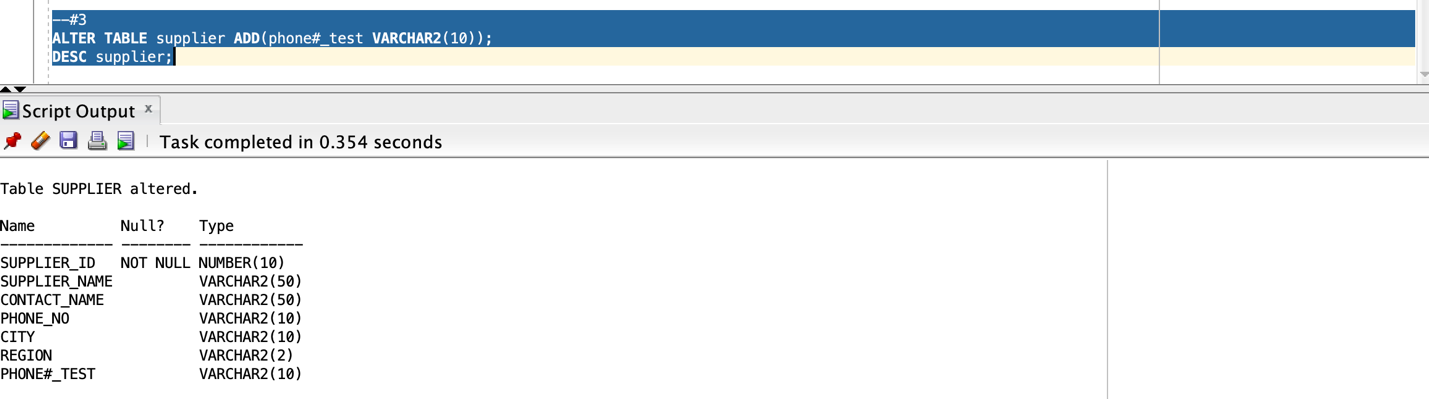
SELECT \* FROM supplier WHERE supplier\_id = 500 AND city = 'Florida';



--#3

ALTER TABLE supplier ADD(phone#\_test VARCHAR2(10));

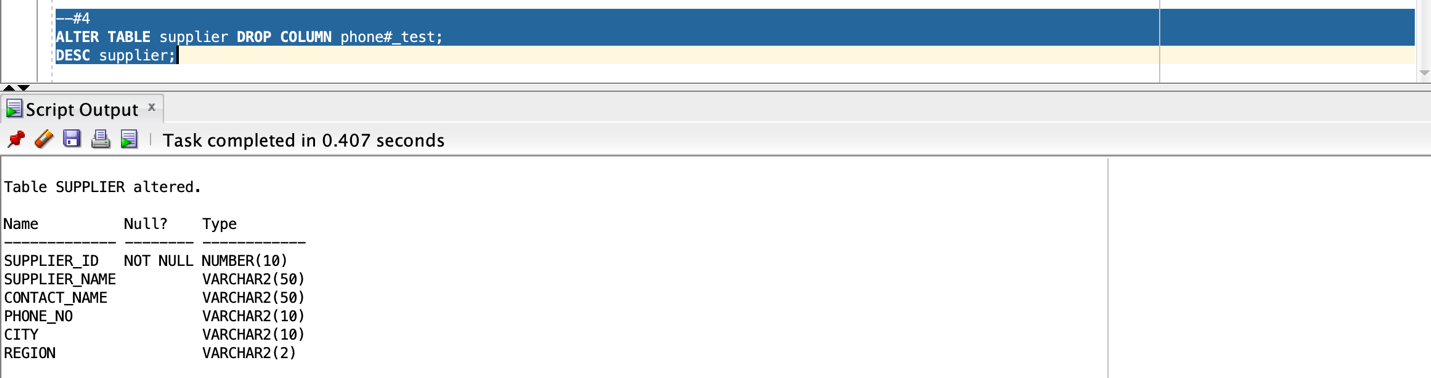
DESC supplier;



--#4

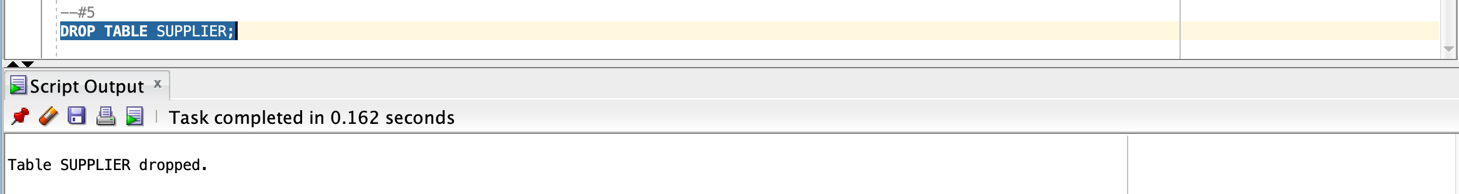
ALTER TABLE supplier DROP COLUMN phone#\_test;

DESC supplier;



--#5

DROP TABLE SUPPLIER;



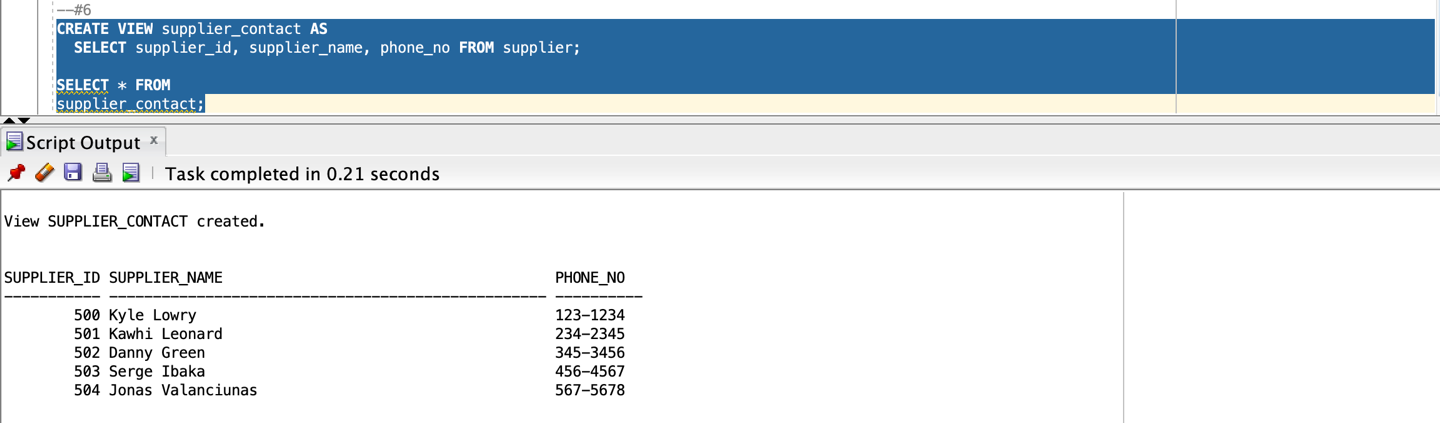
--#6

CREATE VIEW supplier\_contact AS

SELECT supplier\_id, supplier\_name, phone\_no FROM supplier;

SELECT \* FROM

supplier\_contact;

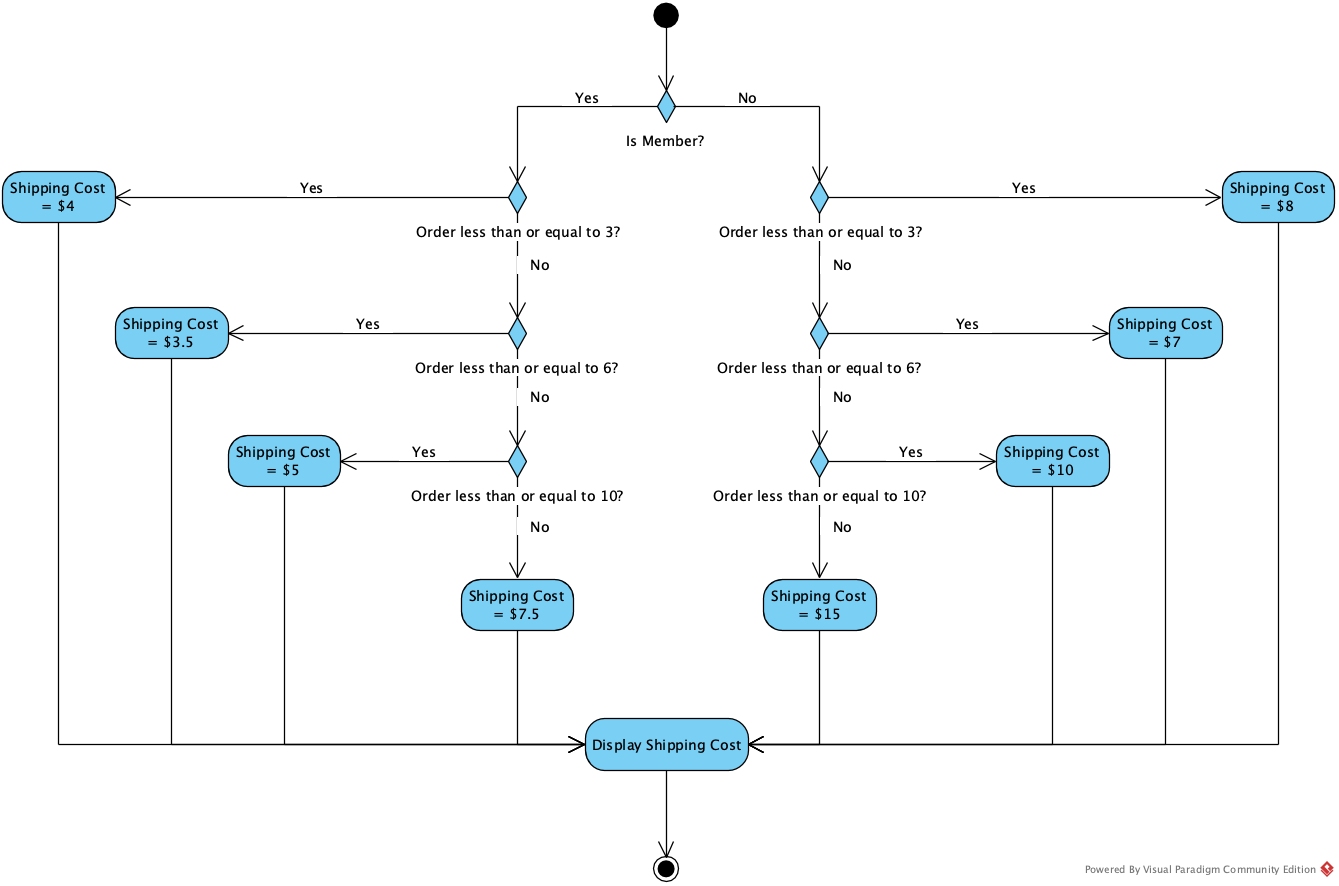


**Question 2**

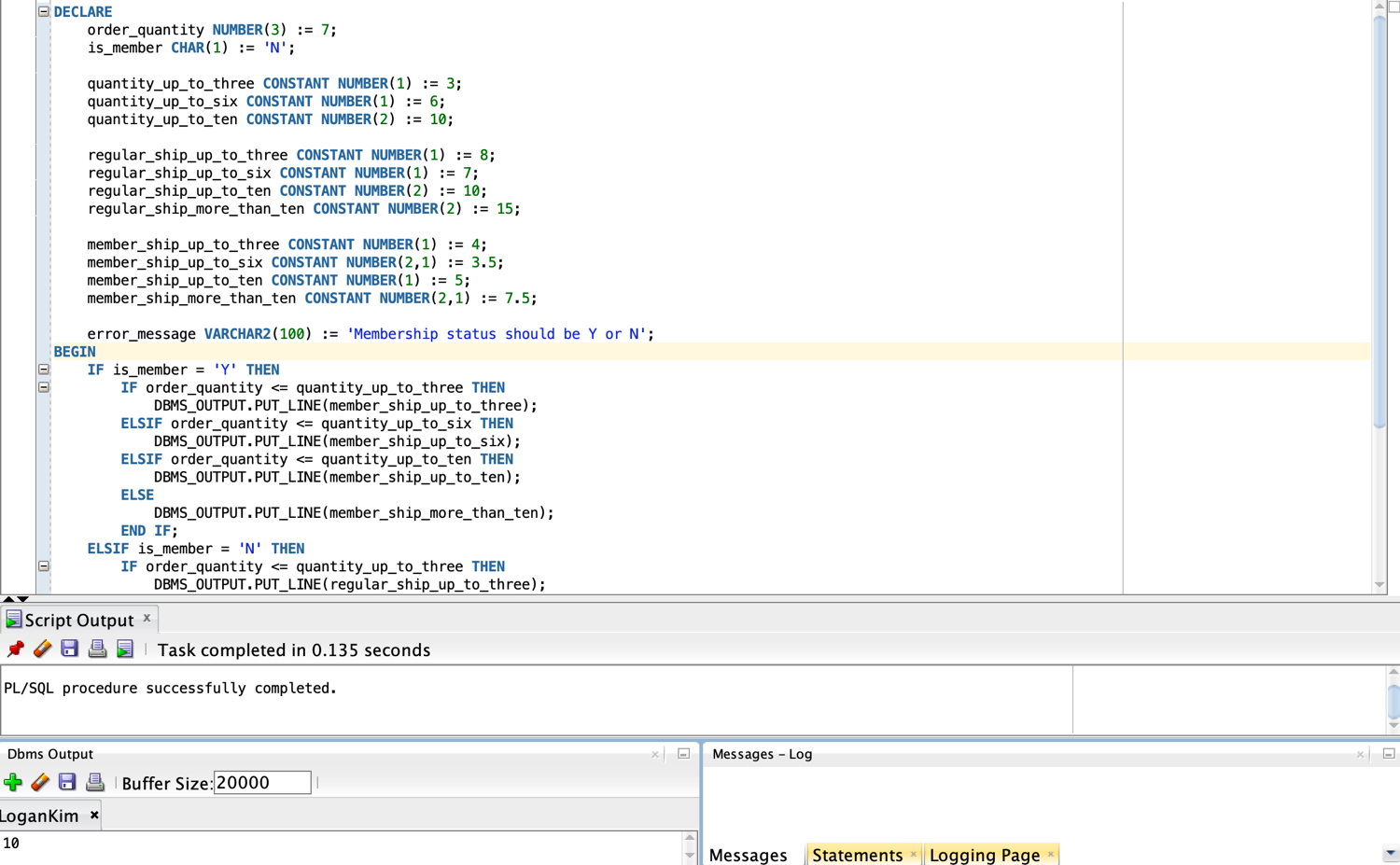
Ecommerce site determines shipping cost based on the products ordered and membership. The valid rates are displayed in the following table:

|  |  |  |
| --- | --- | --- |
| QUANTITY | REGULAR SHIPPING COST | MEMBERS SHIPPING COST |
| Up to 3 | $ 8.00 | $ 4.00 |
| 4-6 | $7.00 | $ 3.50 |
| 7-10 | $10.00 | $5.00 |
| >10 | $15.00 | $7.50 |

1. Create a **flowchart** to outline the processing steps in order to handle this calculation.



1. Create a pl/sql block to complete the above task. Include variable that holds Y OR N to include membership status and a variable to denote the number of items purchased. Verify with different values
2. DECLARE
3. order\_quantity NUMBER(3) := 7;
4. is\_member CHAR(1) := 'N';
6. quantity\_up\_to\_three CONSTANT NUMBER(1) := 3;
7. quantity\_up\_to\_six CONSTANT NUMBER(1) := 6;
8. quantity\_up\_to\_ten CONSTANT NUMBER(2) := 10;
10. regular\_ship\_up\_to\_three CONSTANT NUMBER(1) := 8;
11. regular\_ship\_up\_to\_six CONSTANT NUMBER(1) := 7;
12. regular\_ship\_up\_to\_ten CONSTANT NUMBER(2) := 10;
13. regular\_ship\_more\_than\_ten CONSTANT NUMBER(2) := 15;
14. member\_ship\_up\_to\_three CONSTANT NUMBER(1) := 4;
15. member\_ship\_up\_to\_six CONSTANT NUMBER(2,1) := 3.5;
16. member\_ship\_up\_to\_ten CONSTANT NUMBER(1) := 5;
17. member\_ship\_more\_than\_ten CONSTANT NUMBER(2,1) := 7.5;
19. error\_message VARCHAR2(100) := 'Membership status should be Y or N';
20. BEGIN
21. IF is\_member = 'Y' THEN
22. IF order\_quantity <= quantity\_up\_to\_three THEN
23. DBMS\_OUTPUT.PUT\_LINE(member\_ship\_up\_to\_three);
24. ELSIF order\_quantity <= quantity\_up\_to\_six THEN
25. DBMS\_OUTPUT.PUT\_LINE(member\_ship\_up\_to\_six);
26. ELSIF order\_quantity <= quantity\_up\_to\_ten THEN
27. DBMS\_OUTPUT.PUT\_LINE(member\_ship\_up\_to\_ten);
28. ELSE
29. DBMS\_OUTPUT.PUT\_LINE(member\_ship\_more\_than\_ten);
30. END IF;
31. ELSIF is\_member = 'N' THEN
32. IF order\_quantity <= quantity\_up\_to\_three THEN
33. DBMS\_OUTPUT.PUT\_LINE(regular\_ship\_up\_to\_three);
34. ELSIF order\_quantity <= quantity\_up\_to\_six THEN
35. DBMS\_OUTPUT.PUT\_LINE(regular\_ship\_up\_to\_six);
36. ELSIF order\_quantity <= quantity\_up\_to\_ten THEN
37. DBMS\_OUTPUT.PUT\_LINE(regular\_ship\_up\_to\_ten);
38. ELSE
39. DBMS\_OUTPUT.PUT\_LINE(regular\_ship\_more\_than\_ten);
40. END IF;
41. ELSE
42. DBMS\_OUTPUT.PUT\_LINE(error\_message);
43. END IF;
44. END;
45. /



**Question 3**

Run below script to create the table

DROP TABLE messages;

CREATE TABLE messages (results NUMBER(3));

a) Insert the numbers 1 through 10, excluding 6 and 8.

b) Commit before the end of the block.

c) Execute a SELECT statement to verify that your PL/SQL block worked.

DROP TABLE messages;

CREATE TABLE messages(results NUMBER(3));

BEGIN

FOR i in 1..10 LOOP

IF i = 6 or i = 8 THEN

null;

ELSE

INSERT INTO messages(results)

VALUES (i);

END IF;

END LOOP;

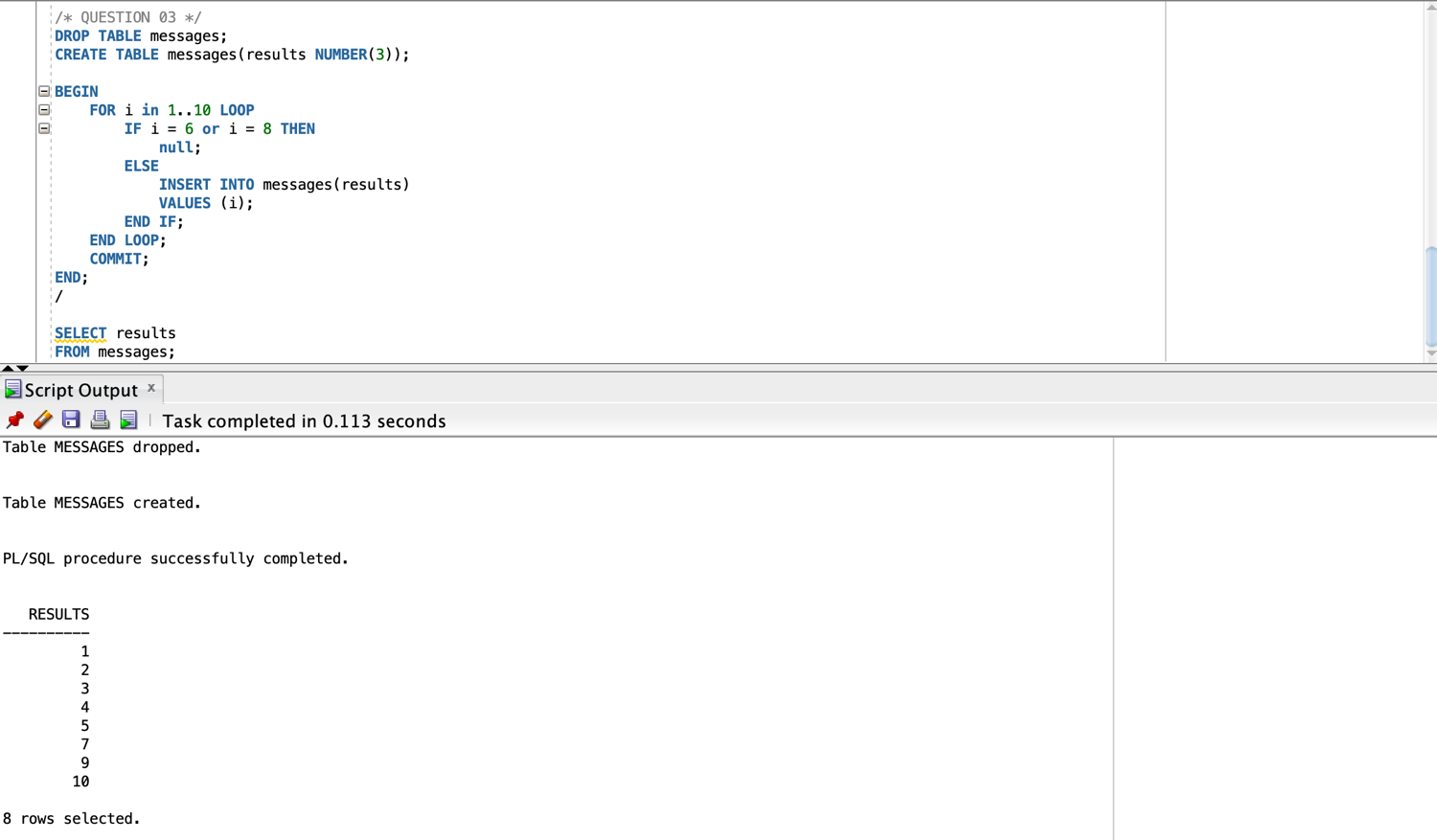
COMMIT;

END;

/

SELECT results

FROM messages;



**Submission:**

* Copy your code to a MS-word file
* Include a screenshot of the output of each code segment.
* This assignment should be done individually
* Submit your work to e-centennial
* Email submission will be ignored