

Term: 2024S FSDM

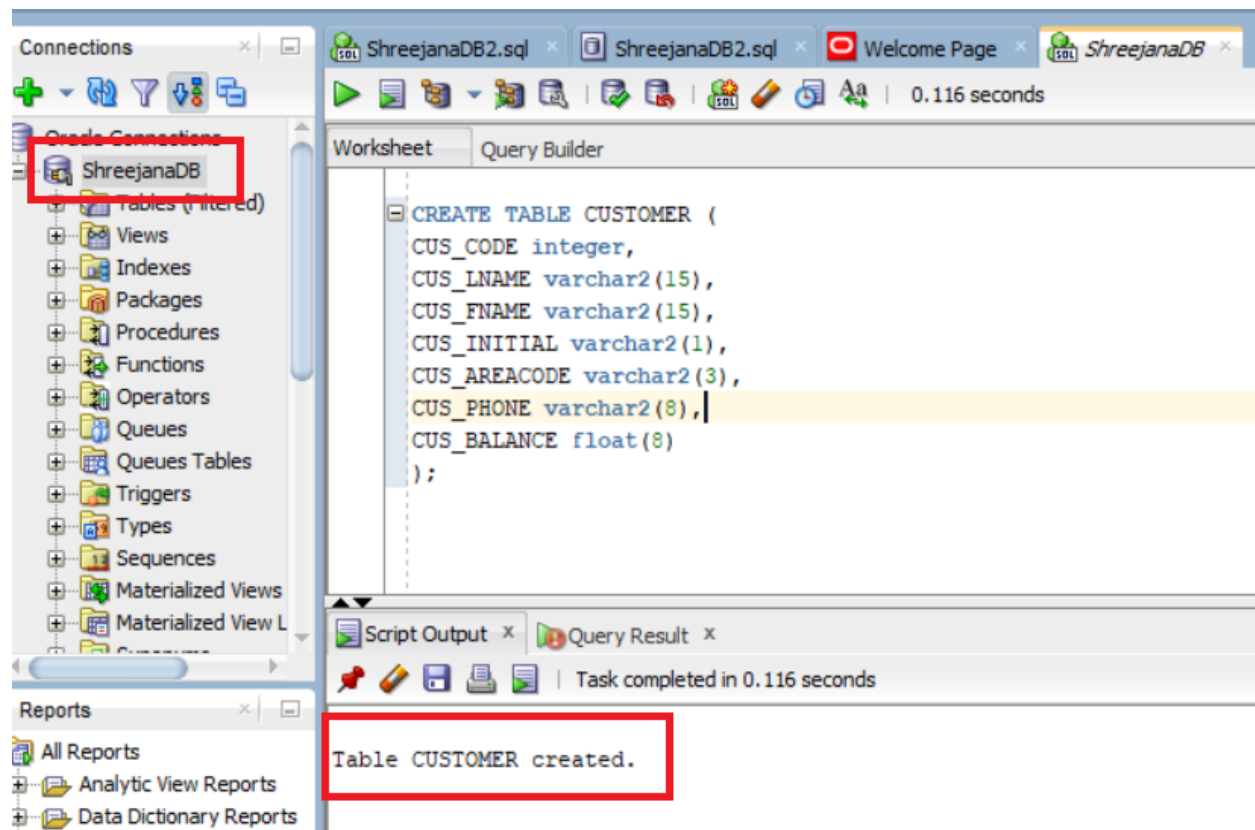
Course: Database Programming

## Practical Exercise #2

Student Name: C0930321

Student Number: Shreejana Shrestha

### 1. Customer Table creation



### 2. Table data insertion

Connections

ShreejanaDB2.sql x ShreejanaDB2.sql x Welcome Page x ShreejanaDB x

0.228 seconds

ShreejanaDB

Worksheet Query Builder

```
INSERT INTO CUSTOMER VALUES('10010','Ramas','Alfred','A','615','844-2573','0');
INSERT INTO CUSTOMER VALUES('10011','Dunne','Leona','K','713','894-1238','0');
INSERT INTO CUSTOMER VALUES('10012','Smith','Kathy','W','615','894-2285','896.54');
INSERT INTO CUSTOMER VALUES('10013','Olowski','Paul','F','615','894-2180','1285.19');
INSERT INTO CUSTOMER VALUES('10014','Orlando','Myron','','615','222-1672','673.21');
INSERT INTO CUSTOMER VALUES('10015','O'Brian','Amy','B','713','442-3381','1014.56');
INSERT INTO CUSTOMER VALUES('10016','Brown','James','G','615','297-1228','0');
INSERT INTO CUSTOMER VALUES('10017','Williams','George','','615','290-2556','0');
INSERT INTO CUSTOMER VALUES('10018','Farriass','Anne','G','713','382-7185','0');
INSERT INTO CUSTOMER VALUES('10019','Smith','Olette','K','615','297-3809','453.98');
```

Script Output x Query Result x

Task completed in 0.228 seconds

Reports

All Reports

- Analytic View Reports
- Data Dictionary Reports
- Data Modeler Reports
- OLAP Reports
- TimesTen Reports
- User Defined Reports

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

### 3. Making copy of CUSTOMER table

The screenshot shows the SQL Developer interface. On the left, the 'Connections' pane is open, and 'ShreejanaDB' is selected. The main window displays a SQL script in the 'Query Builder' tab. The script is as follows:

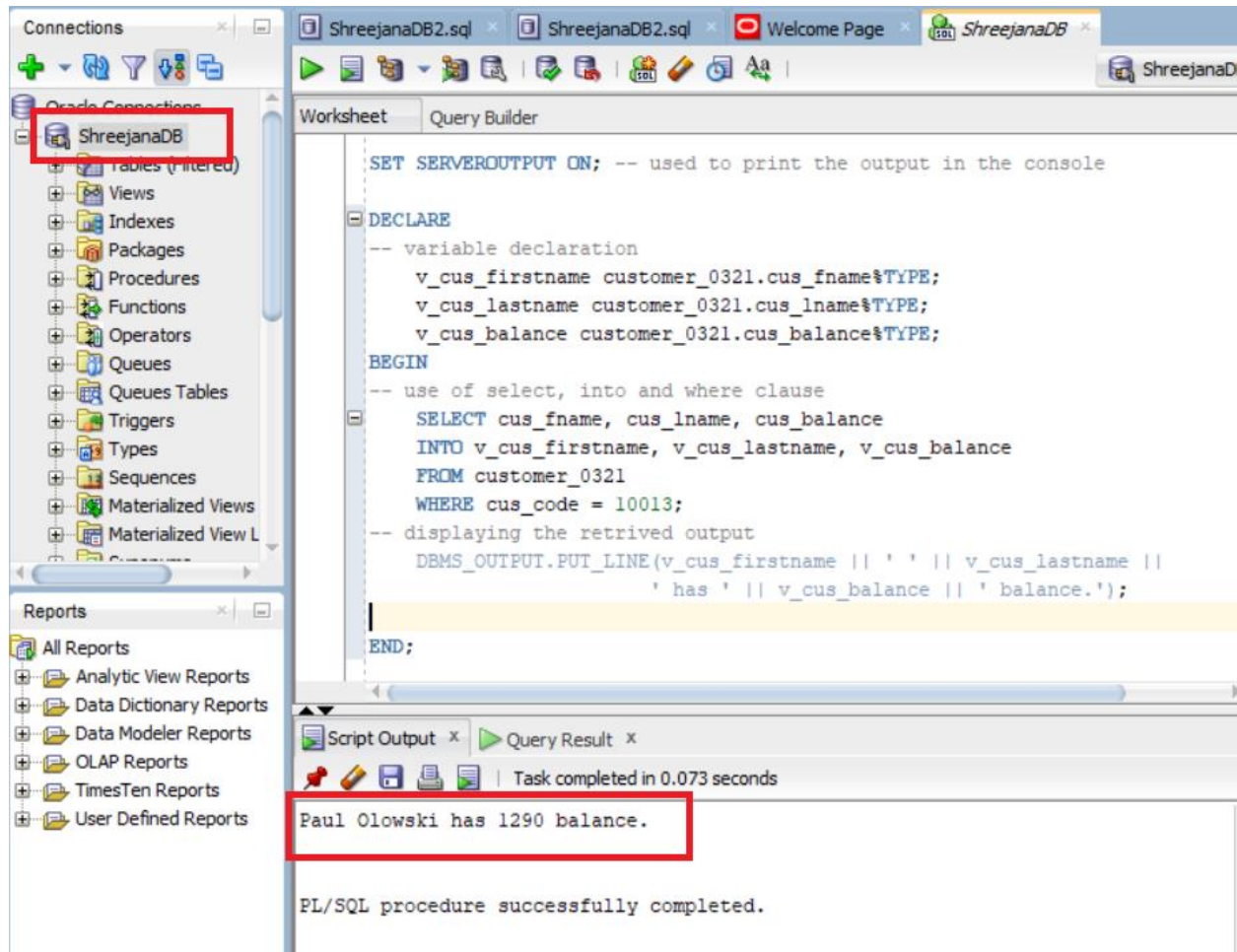
```
-- Making a copy of a table CUSTOMER with table name CUSTOMER_0321  
CREATE TABLE CUSTOMER_0321 AS SELECT * FROM CUSTOMER;
```

Below the script, the 'Script Output' pane shows the message: 'Table CUSTOMER\_0321 created.' The execution time is 0.069 seconds.

The screenshot shows the 'Query Result' pane displaying the data of the newly created table. The data is as follows:

	CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_BALANCE
1	10010	Ramas	Alfred	A	615	844-2573	0
2	10011	Dunne	Leona	K	713	894-1238	0
3	10012	Smith	Kathy	W	615	894-2285	897
4	10013	Olawski	Paul	F	615	894-2180	1290
5	10014	Orlando	Myron	(null)	615	222-1672	673
6	10015	O'Brian	Amy	B	713	442-3381	1010
7	10016	Brown	James	G	615	297-1228	0
8	10017	Williams	George	(null)	615	290-2556	0
9	10018	Farriss	Anne	G	713	382-7185	0
10	10019	Smith	Olette	K	615	297-3809	454

4. Write a PL/SQL script to demonstrate the use of %TYPE, SELECT INTO using where clause, and display output.



The screenshot displays the Oracle SQL Developer interface. On the left, the 'Connections' pane shows 'ShreejanaDB' selected. The main window shows a PL/SQL script in the 'Worksheet' tab. The script is as follows:

```
SET SERVEROUTPUT ON; -- used to print the output in the console

DECLARE
-- variable declaration
  v_cus_firstname customer_0321.cus_fname%TYPE;
  v_cus_lastname customer_0321.cus_lname%TYPE;
  v_cus_balance customer_0321.cus_balance%TYPE;
BEGIN
-- use of select, into and where clause
  SELECT cus_fname, cus_lname, cus_balance
  INTO v_cus_firstname, v_cus_lastname, v_cus_balance
  FROM customer_0321
  WHERE cus_code = 10013;
-- displaying the retrived output
  DBMS_OUTPUT.PUT_LINE(v_cus_firstname || ' ' || v_cus_lastname ||
    ' has ' || v_cus_balance || ' balance.');
```

The 'Script Output' pane at the bottom shows the result of the execution:

```
Paul Olowski has 1290 balance.

PL/SQL procedure successfully completed.
```

## 5. Write a PL/SQL script to demonstrate the error “no data found”.

The screenshot displays the Oracle SQL Developer interface. On the left, the 'Connections' pane shows 'ShreejanaDB' selected. The main window is divided into a 'Worksheet' and a 'Query Builder'. The 'Worksheet' contains a PL/SQL script designed to demonstrate the 'no data found' error. The script declares three variables, selects data from the 'customer\_0321' table where 'cus\_code' is 10020, and uses 'DBMS\_OUTPUT.PUT\_LINE' to display the results. The 'Query Result' pane at the bottom shows the execution output, which includes an error report indicating that no data was found for the specified criteria.

```
DECLARE
-- variable declaration
v_cus_firstname customer_0321.cus_fname%TYPE;
v_cus_lastname customer_0321.cus_lname%TYPE;
v_cus_balance customer_0321.cus_balance%TYPE;
BEGIN
-- use of select, into and where clause
SELECT cus_fname, cus_lname, cus_balance
INTO v_cus_firstname, v_cus_lastname, v_cus_balance
FROM customer_0321
WHERE cus_code = 10020;
-- displaying the retrived output
DBMS_OUTPUT.PUT_LINE(v_cus_firstname || ' ' || v_cus_lastname ||
' has ' || v_cus_balance || ' balance.');
```

**Demonstrating the error "no data found" by using the code to retrieve the customer information with customer code 10020.**

Task completed in 0.142 seconds

```
WHERE cus_code = 10020;
-- displaying the retrived output
DBMS_OUTPUT.PUT_LINE(v_cus_firstname || ' ' || v_cus_lastname ||
' has ' || v_cus_balance || ' balance.');
```

**Error report -**  
ORA-01403: no data found  
ORA-06512: at line 8  
01403. 00000 - "no data found"  
\*Cause: No data was found from the objects.  
\*Action: There was no data from the objects which may be due to end of fetch.

6. Write a PL/SQL script to demonstrate the error “ multiple rows returned”.

The screenshot displays the Oracle SQL Developer interface. On the left, the 'Connections' pane shows 'ShreejanaDB' selected. The main workspace is divided into a 'Worksheet' and a 'Query Builder'. The 'Worksheet' contains a PL/SQL script with a variable declaration and a SELECT statement. The 'Query Builder' pane shows the same script. The 'Script Output' pane at the bottom displays the error message: 'ORA-01422: exact fetch returns more than requested number of rows'. A red box highlights the error message, and another red box highlights the variable declaration and SELECT statement in the script.

```
DECLARE
-- variable declaration
v_cus_balance customer_0321.cus_balance%TYPE;
BEGIN

SELECT cus_balance
INTO v_cus_balance
FROM customer_0321;
-- displaying the retrived output

DBMS_OUTPUT.PUT_LINE('Balance is: ' || v_cus_balance);

END;
```

Demonstrating the use of "multiple rows returned" error as we are fetching whole cus\_balance column and trying to return those multiple values in v\_cus\_balance variable using PL/SQL.

```
SELECT cus_balance
INTO v_cus_balance
FROM customer_0321;
-- displaying the retrived output

DBMS_OUTPUT.PUT_LINE('Balance is: ' || v_cus_balance);

END;
```

Error report -  
ORA-01422: exact fetch returns more than requested number of rows  
ORA-06512: at line 6  
01422. 00000 - "exact fetch returns more than requested number of rows"  
\*Cause: The number specified in exact fetch is less than the rows returned.  
\*Action: Rewrite the query or change number of rows requested



## 7. Write a PL/SQL script to demonstrate the use of group function.

The screenshot displays the Oracle SQL Developer interface. On the left, the 'Connections' pane shows 'ShreejanaDB' selected. The main workspace contains a PL/SQL script in the 'Query Builder' tab. The script declares a variable, selects the sum of 'cus\_balance' from the 'customer\_0321' table, and outputs the result using 'DBMS\_OUTPUT.PUT\_LINE'. The 'Script Output' pane at the bottom shows the task completed in 0.066 seconds and the output: 'The total Balance of all customers is: 4324'. A red box highlights the 'SUM(cus\_balance)' function in the script, and another red box highlights the output line. A text box on the right explains the use of the SUM group function.

```
DECLARE
-- variable declaration
  v_cus_balance NUMBER(10, 2);
BEGIN
  SELECT SUM(cus_balance) -- group function
  INTO v_cus_balance
  FROM customer_0321;
  -- displaying the retrived output

  DBMS_OUTPUT.PUT_LINE('The total Balance of all customers is: '
    || v_cus_balance);
END;
```

**Making use of group function (SUM) to return a value from customer\_0321 table.**

Task completed in 0.066 seconds

The total Balance of all customers is: 4324

PL/SQL procedure successfully completed.