

## Database Programming

### Practical Exercise #5

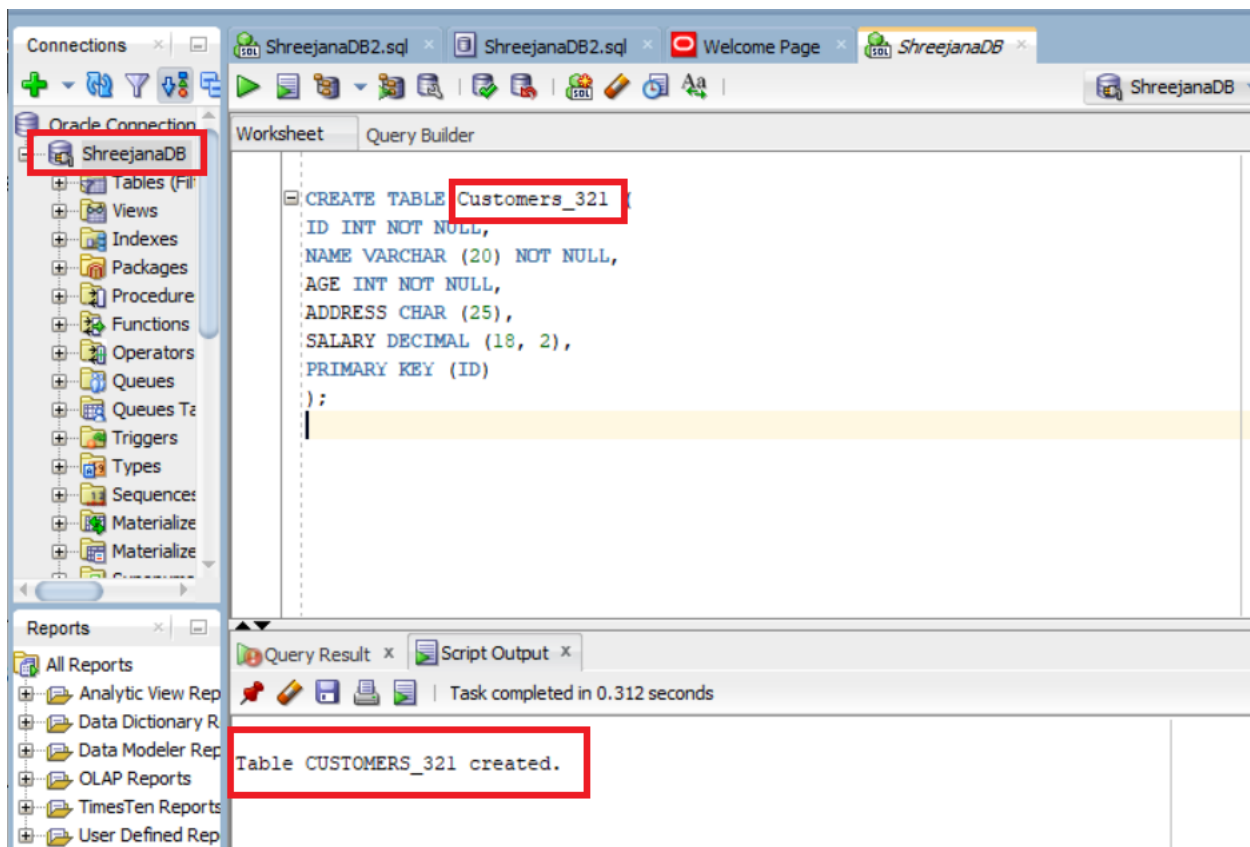
Student Name : Shreejana Shrestha

Student Id : C0930321

---

## # Question 1

### Table Creation



## # Question 2

### Stored Procedure

The screenshot displays the Oracle SQL Developer interface. On the left, the 'Connections' pane shows 'ShreejanaDB' selected. The main workspace is the 'Query Builder' tab, containing the following SQL code:

```
/*create a Stored Procedures to insert data to the table.
Stored Procedure name : sp_insert789
*/
CREATE OR REPLACE PROCEDURE sp_insert321 -- procedure name with parameters
-- use of IN parameters which represents the mode and is default if not specified
(p_id IN customers_321.id%TYPE,
 p_name customers_321.name%TYPE,
 p_age customers_321.age%TYPE,
 p_address customers_321.address%TYPE,
 p_salary customers_321.salary%TYPE)
IS
BEGIN
    INSERT INTO customers_321(id, name, age, address, salary)
    VALUES(p_id, p_name, p_age, p_address, p_salary);

    DBMS_OUTPUT.PUT_LINE('Inserted ' || SQL%ROWCOUNT || ' row');
    DBMS_OUTPUT.PUT_LINE('Inserted ' || 'ID ' || p_id);
    DBMS_OUTPUT.PUT_LINE('Inserted ' || 'Name ' || p_name);
    DBMS_OUTPUT.PUT_LINE('Inserted ' || 'Age ' || p_age);
    DBMS_OUTPUT.PUT_LINE('Inserted ' || 'Address ' || p_address);
    DBMS_OUTPUT.PUT_LINE('Inserted ' || 'Salary ' || p_salary);
END;
```

Below the code, the 'Query Result' pane shows the message: 'Table CUSTOMERS\_321 created.'

The 'Script Output' pane at the bottom shows the message: 'Procedure SP\_INSERT321 compiled'.

Two red boxes highlight key elements: the procedure name 'sp\_insert321' in the code, and the 'Script Output' message.

Here, we have created a stored procedure named **sp\_insert321** with parameters. Parameters are indicated by the prefix **p\_**. We have created this procedure to insert the record to the customers\_321 table. We can simply pass the values while calling the stored procedure to insert the record.

Connections x | ShreejanaDB2.sql x | ShreejanaDB2.sql x | Welcome Page x | ShreejanaDB x

0.053 seconds

Worksheet | Query Builder

```
DESC sp_insert321;
```

Script Output x | Query Result x

Task completed in 0.053 seconds

PROCEDURE sp\_insert321

Argument	Name	Type	In/Out	Default?
P_ID		NUMBER (38)	IN	
P_NAME		VARCHAR2 (20)	IN	
P_AGE		NUMBER (38)	IN	
P_ADDRESS		CHAR (25)	IN	
P_SALARY		NUMBER (18,2)	IN	

# Inserting data invoking stored procedure

The screenshot displays the SQL Developer interface with the 'ShreejanaDB' connection selected. The 'Query Builder' tab is active, showing a PL/SQL block that invokes the stored procedure `sp_insert321` with six sets of data. The 'Script Output' window shows the execution results, including the inserted ID, name, age, address, and salary for each row. A red box highlights the procedure call in the script, and another red box highlights the output details. A black box contains a red text annotation explaining the procedure call.

**Query Builder:**

```
-- Invoking the procedure to insert the given set of data

BEGIN
    sp_insert321(2, 'Arshdeep', 25, 'Delhi', 15000.00);
    sp_insert321(3, 'Navpreet', 23, 'Kota', 20000.00);
    sp_insert321(4, 'Shubham', 45, 'Mumbai', 60500.00);
    sp_insert321(5, 'Satyam', 27, 'Bhopal', 85000.00);
    sp_insert321(6, 'Rahul', 35, 'MP', 45000.00);
END;
```

**Script Output:**

```
Task completed in 0.067 seconds

Inserted ID 4
Inserted Name Shubham
Inserted Age 45
Inserted Address Mumbai
Inserted Salary 60500
Inserted 1 row
Inserted ID 5
Inserted Name Satyam
Inserted Age 27
Inserted Address Bhopal
Inserted Salary 85000
Inserted 1 row
Inserted ID 6
Inserted Name Rahul
Inserted Age 35
Inserted Address MP
Inserted Salary 45000

PL/SQL procedure successfully completed.
```

**Annotation:**

Here, we called the stored procedure `sp_insert321` created above to insert the set of records. The data get inserted to the table and displayed the inserted data info in the console output.

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

Oracle Connection ShreejanaDB

- Tables (Fil
- Views
- Indexes
- Packages
- Procedure
- Functions
- Operators
- Queues
- Queues Tz
- Triggers
- Types
- Sequences
- Materialize
- Materialize

Worksheet Query Builder

```
select * from Customers_321;
```

Script Output x Query Result x

SQL | All Rows Fetched: 6 in 0.003 seconds

	ID	NAME	AGE	ADDRESS	SALARY
1	1	Kanav	32	Ahmedabad	20000
2	2	Arshdeep	25	Delhi	15000
3	3	Navpreet	23	Kota	20000
4	4	Shubham	45	Mumbai	60500
5	5	Satyam	27	Bhopal	85000
6	6	Rahul	35	MP	45000

Reports x

- All Reports
- Analytic View Rep
- Data Dictionary R
- Data Modeler Rep
- OLAP Reports
- TimesTen Reports
- User Defined Rep

## # Question 3

The screenshot displays the Oracle SQL Developer interface. On the left, the 'Connections' pane shows 'ShreejanaDB' selected. The main workspace is the 'Query Builder' tab, containing the following PL/SQL code:

```
CREATE OR REPLACE PROCEDURE query_loan_approval_check321 -- procedure name
IS
    CURSOR customer_cursor_321 IS -- cursor declaration
        SELECT *
        FROM customers_321
        WHERE salary > 40000 and age >=35;
    -- variables declarations
    v_id customers_321.id%TYPE;
    v_name customers_321.name%TYPE;
    v_age customers_321.age%TYPE;
    v_address customers_321.address%TYPE;
    v_salary customers_321.salary%TYPE;
BEGIN
    OPEN customer_cursor_321; -- -- OPEN cursor created above
    LOOP
        -- fetch current row from cursor
        FETCH customer_cursor_321 INTO v_id, v_name, v_age, v_address, v_salary;
        -- Test for end of the cursor using %NOTFOUND
        EXIT WHEN customer_cursor_321%NOTFOUND;
        -- use of %ROWCOUNT to find the rows fetched
        DBMS_OUTPUT.PUT_LINE('Rows fetched so far '
            || TO_CHAR(customer_cursor_321%ROWCOUNT));

        DBMS_OUTPUT.PUT_LINE('Customer ID: ' || v_id);
        DBMS_OUTPUT.PUT_LINE('Name: ' || v_name);
        DBMS_OUTPUT.PUT_LINE('Age: ' || v_age);
        DBMS_OUTPUT.PUT_LINE('Salary: ' || v_salary);
        DBMS_OUTPUT.PUT_LINE('Loan Approved');
    END LOOP;
    CLOSE customer_cursor_321; -- close cursor
END;
```

At the bottom, the 'Script Output' pane shows the message: 'Task completed in 0.062 seconds' and 'Procedure QUERY\_LOAN\_APPROVAL\_CHECK321 compiled'.

Here, we have created a procedure `query_loan_approval_check321` to identify and list customers who meet the above loan approval criteria. Basically, it selects customers whose salary is greater than 40,000 and who are aged 35 or older. And we are using cursor to fetch multiple rows of data.



# execution of loan approval check procedure  
invoking it

The screenshot displays the Oracle SQL Developer environment. The 'Connections' pane on the left shows 'ShreejanaDB' selected. The 'Worksheet' pane contains the following SQL code:

```
-- executing the loan_approval procedure invoking it  
  
BEGIN  
    query_loan_approval_check321;  
END;
```

The 'Script Output' pane shows the execution results:

```
Procedure QUERY_LOAN_APPROVAL_CHECK321 compiled  
  
Customer ID: 4  
Name: Shubham  
Age: 45  
Salary: 60500  
Loan Approved  
  
Customer ID: 6  
Name: Rahul  
Age: 35  
Salary: 45000  
Loan Approved  
  
PL/SQL procedure successfully completed.
```

The output is presented in a table-like format with two rows of data, each representing a customer whose loan was approved.

Customer ID	Name	Age	Salary	Loan Status
4	Shubham	45	60500	Approved
6	Rahul	35	45000	Approved