PROCEDURES:

1.Create a procedure to display the n number of records from students table.

sol:

```
SQL> CREATE TABLE students (
  2      student_id NUMBER PRIMARY KEY,
  3      student_name VARCHAR2(100),
  4      age NUMBER
  5  );

Table created.

SQL>
SQL> -- Insert some sample data into the students table
SQL> INSERT INTO students (student_id, student_name, age) VALUES (1, 'John Doe', 20);

1 row created.

SQL> INSERT INTO students (student_id, student_name, age) VALUES (2, 'Jane Smith', 22);

1 row created.

SQL> INSERT INTO students (student_id, student_name, age) VALUES (3, 'Michael Johnson', 25);

1 row created.

SQL> -- Create the procedure
SQL> CREATE OR REPLACE PROCEDURE display_students(n IN NUMBER) AS
  2      CURSOR student_cursor IS
  3          SELECT *
  4          FROM students
  5          WHERE ROWNUM <= n;
  6  BEGIN
  7      FOR student_record IN student_cursor LOOP
  8          -- Display the student information
  9          DBMS_OUTPUT.PUT_LINE('Student ID: ' || student_record.student_id || ', Student Name: ' || stu
dent_record.student_name || ', Age: ' || student_record.age);
 10      END LOOP;
 11  END;
 12  /

Procedure created.

SQL> DECLARE
  2      num_records NUMBER := 2; -- Replace this value with the desired number of records to display
  3  BEGIN
  4      display_students(num_records);
  5  END;
  6  /
Student ID: 1, Student Name: John Doe, Age: 20
Student ID: 2, Student Name: Jane Smith, Age: 22

PL/SQL procedure successfully completed.
```

2.Create a procedure that prints the max percentage obtained by a student.

sol:

```
SQL> CREATE TABLE students (
  2      student_id NUMBER PRIMARY KEY,
  3      student_name VARCHAR2(100),
  4      subject VARCHAR2(100),
  5      percentage NUMBER
  6  );

Table created.

SQL>
SQL> -- Insert some sample data into the students table
SQL> INSERT INTO students (student_id, student_name, subject, percentage) VALUES (1, 'John Doe', '
Math', 80);

1 row created.

SQL> INSERT INTO students (student_id, student_name, subject, percentage) VALUES (2, 'Jane Smith',
 'Science', 90);

1 row created.

SQL> INSERT INTO students (student_id, student_name, subject, percentage) VALUES (3, 'Michael Joh
nson', 'History', 75);

1 row created.

SQL> INSERT INTO students (student_id, student_name, subject, percentage) VALUES (4, 'Alice Walker
', 'English', 85);

1 row created.

SQL> -- Create the procedure
SQL> CREATE OR REPLACE PROCEDURE print_max_percentage AS
  2      max_percentage NUMBER;
  3  BEGIN
  4      -- Find the maximum percentage using the MAX function
  5      SELECT MAX(percentage) INTO max_percentage
  6      FROM students;
  7
  8      -- Display the result
  9      DBMS_OUTPUT.PUT_LINE('Maximum Percentage: ' || max_percentage);
 10  END;
 11  /

Procedure created.

SQL> -- Enable DBMS_OUTPUT
SQL> SET SERVEROUTPUT ON;
SQL>
SQL> -- Call the procedure to print the maximum percentage
SQL> BEGIN
```

```
  2      print_max_percentage;
  3  END;
  4  /
Maximum Percentage: 90

PL/SQL procedure successfully completed.
```

3.Create a procedure that prints the greatest of 3 numbers.

sol:

```
SQL> CREATE OR REPLACE PROCEDURE print_greatest_number(
  2      num1 IN NUMBER,
  3      num2 IN NUMBER,
  4      num3 IN NUMBER
  5  ) AS
  6      greatest_num NUMBER;
  7  BEGIN
  8      -- Find the greatest number using the GREATEST function
  9      greatest_num := GREATEST(num1, num2, num3);
 10
 11       -- Display the result
 12       DBMS_OUTPUT.PUT_LINE('The greatest number is: ' || greatest_num);
 13  END;
 14  /

Procedure created.

SQL> -- Enable DBMS_OUTPUT
SQL> SET SERVEROUTPUT ON;
SQL>
SQL> -- Call the procedure to find the greatest number
SQL> BEGIN
  2      print_greatest_number(10, 25, 15);
  3  END;
  4  /
The greatest number is: 25

PL/SQL procedure successfully completed.
```

4.Create a procedure to display the n number of records from students table.

sol:

```
SQL> CREATE TABLE students (
  2      student_id NUMBER PRIMARY KEY,
  3      student_name VARCHAR2(100),
  4      age NUMBER
  5  );

Table created.
```

```
SQL>
SQL> -- Insert some sample data into the students table
SQL> INSERT INTO students (student_id, student_name, age) VALUES (1, 'John Doe', 20);

1 row created.

SQL> INSERT INTO students (student_id, student_name, age) VALUES (2, 'Jane Smith', 22);

1 row created.

SQL> INSERT INTO students (student_id, student_name, age) VALUES (3, 'Michael Johnson', 25);

1 row created.

SQL> -- Create the procedure
SQL> CREATE OR REPLACE PROCEDURE display_students(n IN NUMBER) AS
  2     TYPE student_record_type IS RECORD (
  3         student_id NUMBER,
  4         student_name VARCHAR2(100),
  5         age NUMBER
  6     );
  7
  8     TYPE student_cursor_type IS REF CURSOR;
  9     student_cursor student_cursor_type;
 10
 11     student_info student_record_type;
 12  BEGIN
 13     -- Open the cursor to fetch n records from students
 14     OPEN student_cursor FOR
 15         SELECT student_id, student_name, age
 16         FROM students
 17         WHERE ROWNUM <= n;
 18
 19     -- Fetch and display each record from the cursor
 20     LOOP
 21         FETCH student_cursor INTO student_info;
 22         EXIT WHEN student_cursor%NOTFOUND;
 23
 24         -- Display the student information
 25         DBMS_OUTPUT.PUT_LINE('Student ID: ' || student_info.student_id || ', Student Name: ' || stude
nt_info.student_name || ', Age: ' || student_info.age);
 26     END LOOP;
 27
 28     -- Close the cursor
 29     CLOSE student_cursor;
 30  END;
 31  /

Procedure created.

SQL> -- Enable DBMS_OUTPUT
SQL> SET SERVEROUTPUT ON;
SQL>
SQL> -- Call the procedure to display the first 2 records
SQL> BEGIN
```

```
  2      display_students(2);
  3  END;
  4  /
Student ID: 1, Student Name: John Doe, Age: 20
Student ID: 2, Student Name: Jane Smith, Age: 22

PL/SQL procedure successfully completed.
```