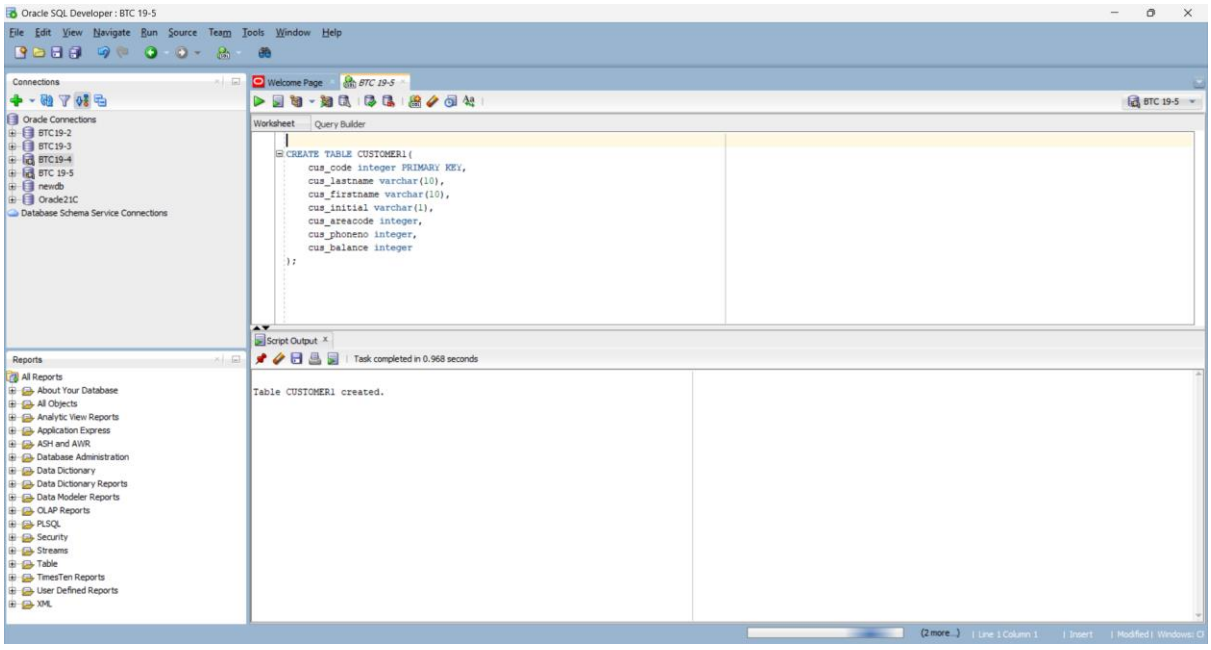
EXPERIMENT 5

Name:- Omkar Kore **Class**: Btech C

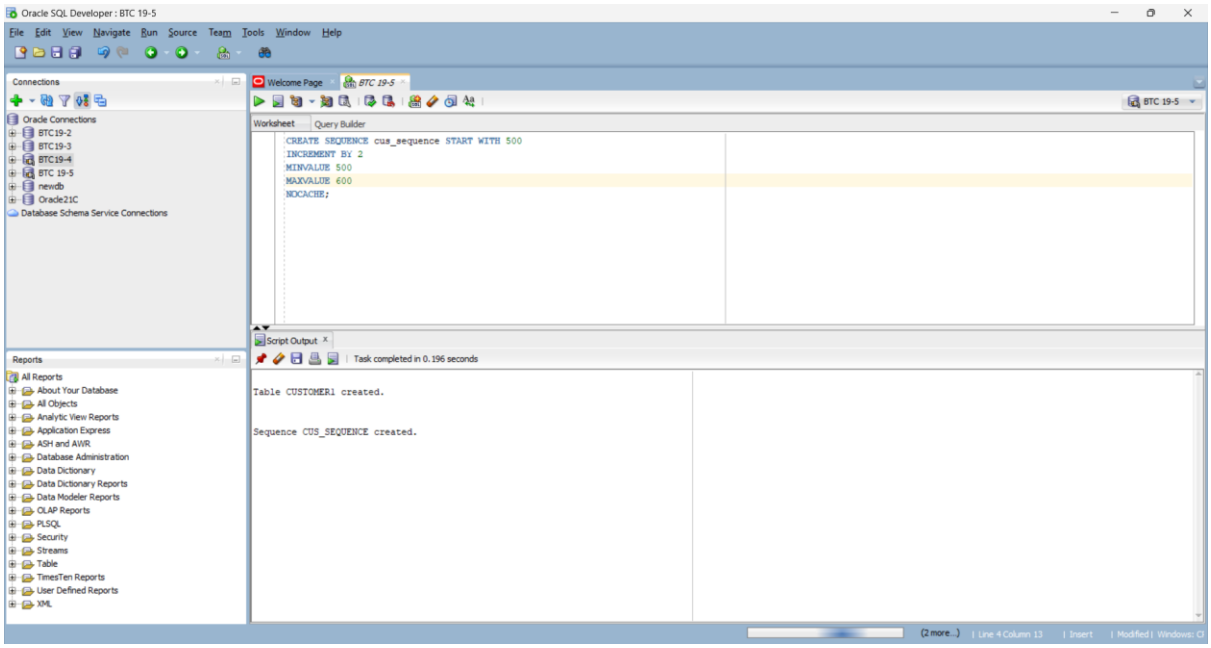
Roll Number:- BTC19 **Batch**: C1

--PART A]ORACLE SEQUENCES

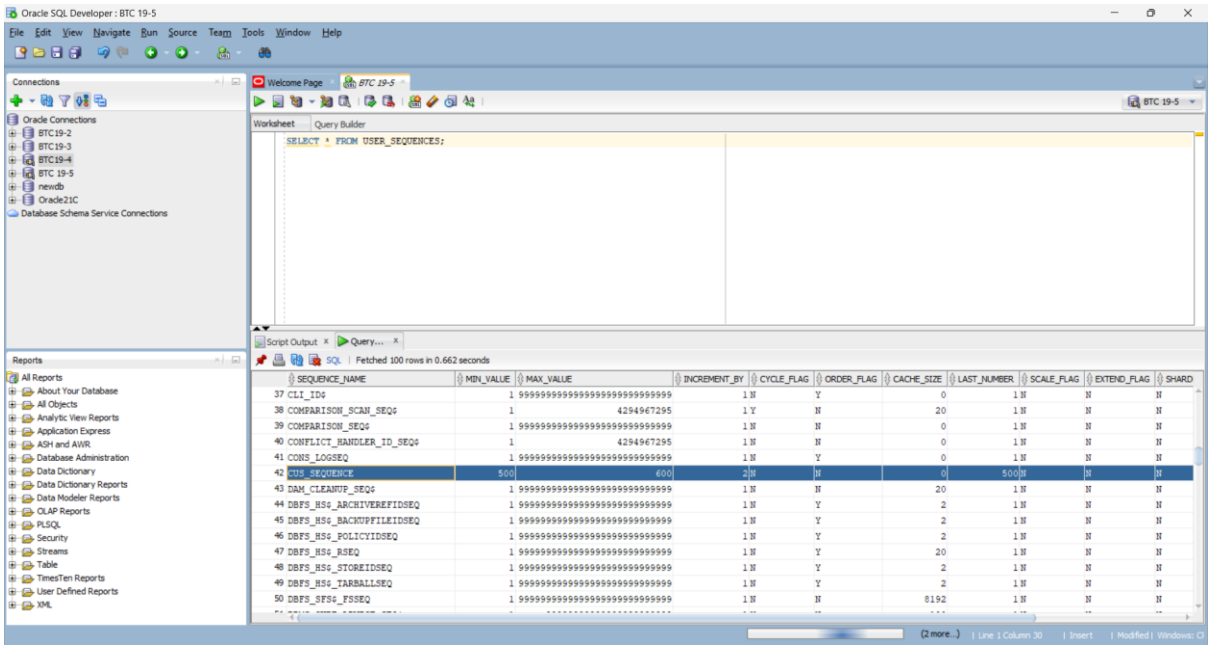
```
CREATE TABLE CUSTOMER1(
    cus_code integer PRIMARY KEY,
    cus_lastname varchar(10),
    cus_firstname varchar(10),
    cus_initial varchar(1),
    cus_areacode integer,
    cus_phoneno integer,
    cus_balance integer
);
```




```
CREATE SEQUENCE cus_sequence START WITH 500
INCREMENT BY 2
MINVALUE 500
MAXVALUE 600
NOCACHE;
```



SELECT * FROM USER_SEQUENCES;



INSERT INTO CUSTOMER1 VALUES(cus_sequence.NEXTVAL, 'Kore', 'Omkar', 'S', '1234', '7896541236', '30000');

INSERT INTO CUSTOMER1 VALUES(cus_sequence.NEXTVAL, 'Patil', 'Rudra', 'D', '8965', '7893214561', '40000');

INSERT INTO CUSTOMER1 VALUES(cus_sequence.NEXTVAL, 'Karale', 'Aarav', 'A', '2365', '9658741236', '60000');

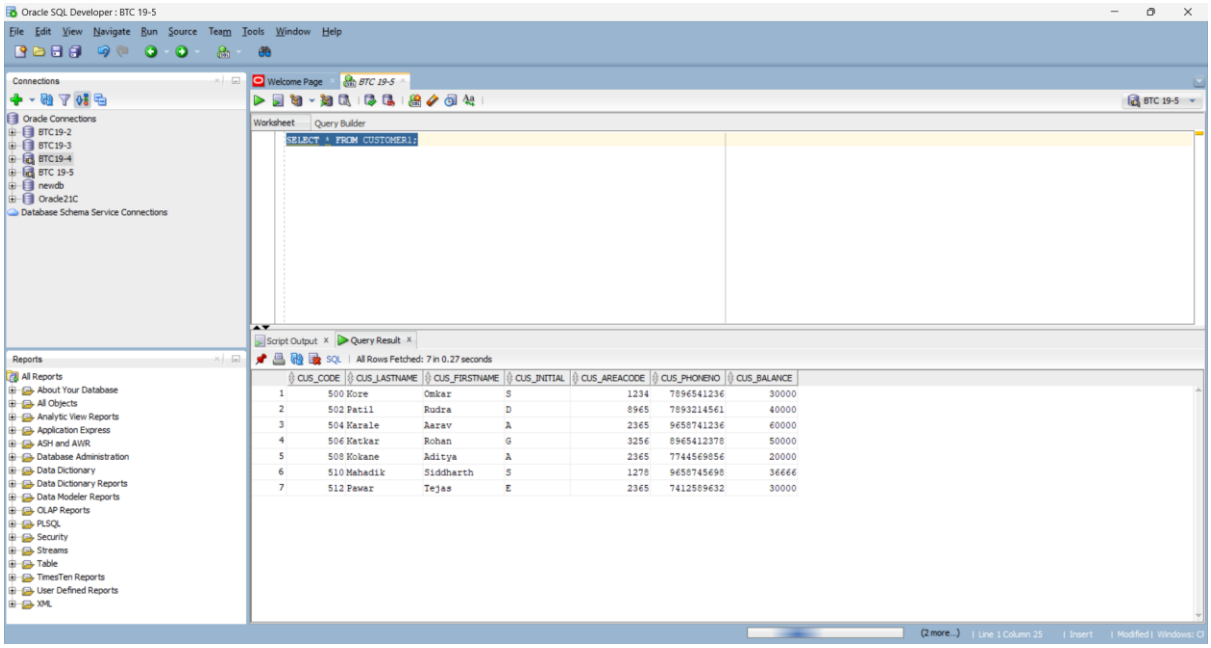
INSERT INTO CUSTOMER1 VALUES(cus_sequence.NEXTVAL, 'Katkar', 'Rohan', 'G', '3256', '8965412378', '50000');

INSERT INTO CUSTOMER1 VALUES(cus_sequence.NEXTVAL, 'Kokane', 'Aditya', 'A', '2365', '7744569856', '20000');

INSERT INTO CUSTOMER1 VALUES(cus_sequence.NEXTVAL, 'Mahadik', 'Siddharth', 'S', '1278', '9658745698', '36666');

INSERT INTO CUSTOMER1 VALUES(cus_sequence.NEXTVAL, 'Pawar', 'Tejas', 'E', '2365', '7412589632', '30000');

SELECT * FROM CUSTOMER1;



--PART B]TRIGGERS

CREATE TABLE Student_reports (

T_id INT *PRIMARY KEY*,

student_name VARCHAR(30),

subject1 INT,

subject2 INT,

subject3 INT,

total_marks INT,

percentage INT

);

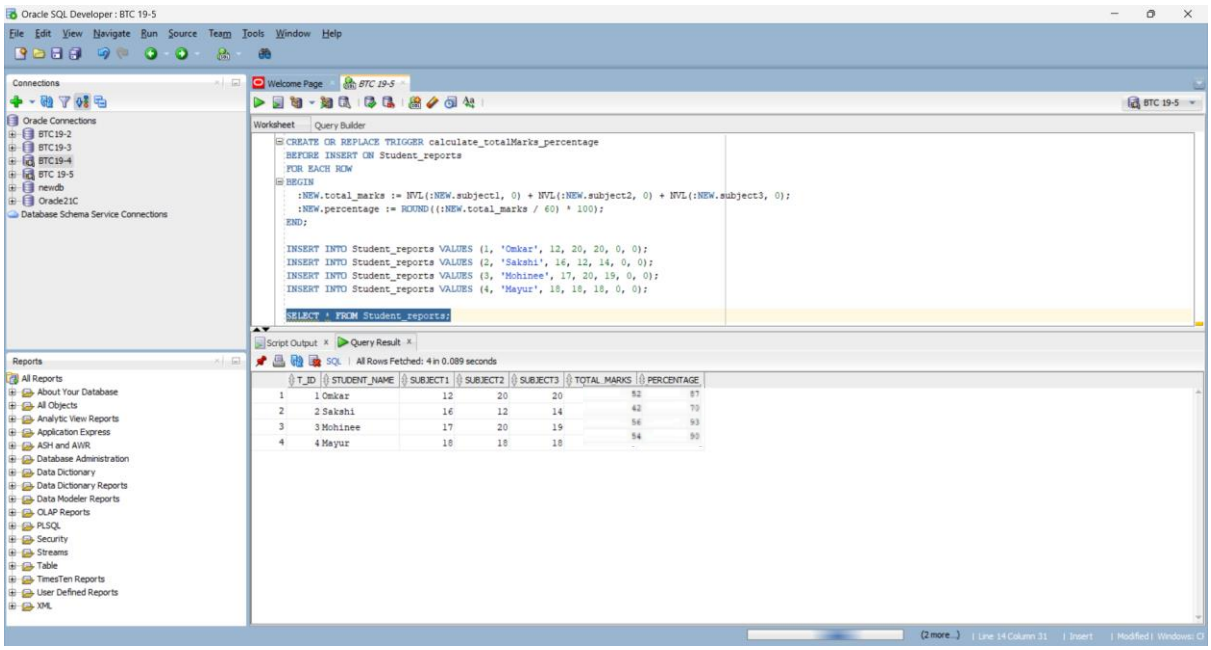
CREATE OR REPLACE TRIGGER calculate_totalMarks_percentage

BEFORE INSERT ON Student_reports

FOR EACH ROW

```
BEGIN
  :NEW.total_marks := NVL(:NEW.subject1, 0) + NVL(:NEW.subject2, 0) +
NVL(:NEW.subject3, 0);
  :NEW.percentage := ROUND((:NEW.total_marks / 60) * 100);
END;

INSERT INTO Student_reports VALUES (1, 'Omkar', 12, 20, 20, 0, 0);
INSERT INTO Student_reports VALUES (2, 'Sakshi', 16, 12, 14, 0, 0);
INSERT INTO Student_reports VALUES (3, 'Mohinee', 17, 20, 19, 0, 0);
INSERT INTO Student_reports VALUES (4, 'Mayur', 18, 18, 18, 0, 0);
SELECT * FROM Student_reports;
```

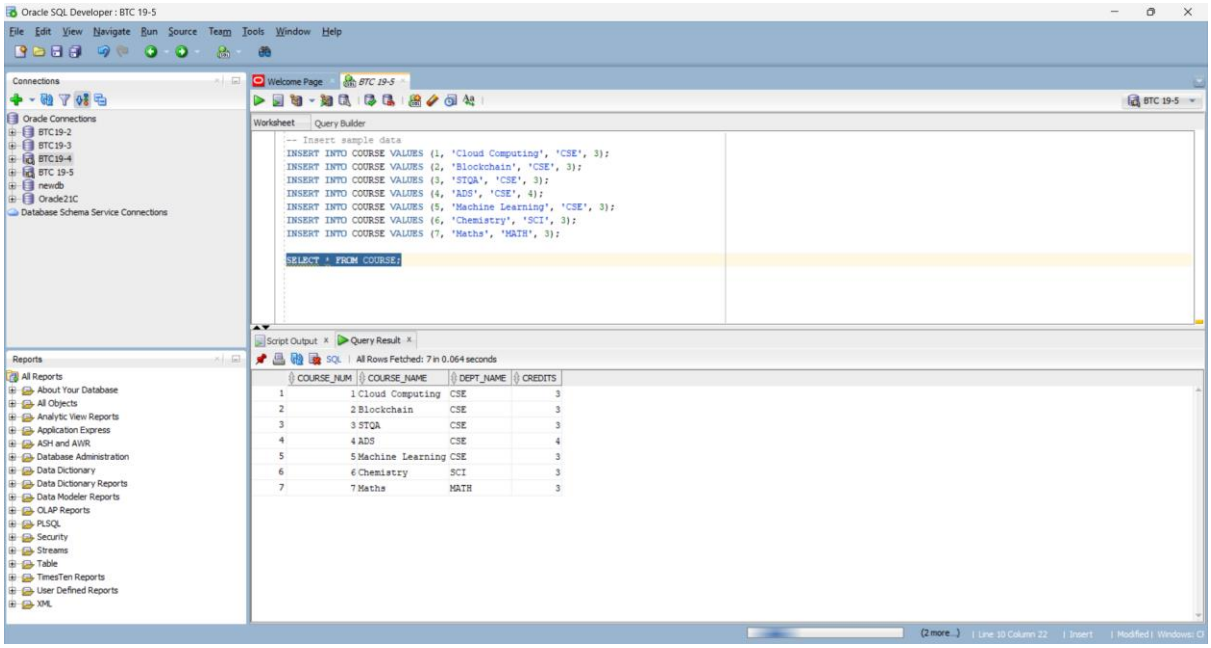


--PART C|PROCEDURE AND CURSOR

```
CREATE TABLE COURSE (
    course_num   INTEGER PRIMARY KEY,
    course_name  VARCHAR2(20),
    dept_name    VARCHAR2(15),
    credits      INTEGER
);
```

```
-- Insert sample data
INSERT INTO COURSE VALUES (1, 'Cloud Computing', 'CSE', 3);
INSERT INTO COURSE VALUES (2, 'Blockchain', 'CSE', 3);
INSERT INTO COURSE VALUES (3, 'STQA', 'CSE', 3);
INSERT INTO COURSE VALUES (4, 'ADS', 'CSE', 4);
INSERT INTO COURSE VALUES (5, 'Machine Learning', 'CSE', 3);
INSERT INTO COURSE VALUES (6, 'Chemistry', 'SCI', 3);
INSERT INTO COURSE VALUES (7, 'Maths', 'MATH', 3);

SELECT * FROM COURSE;
```

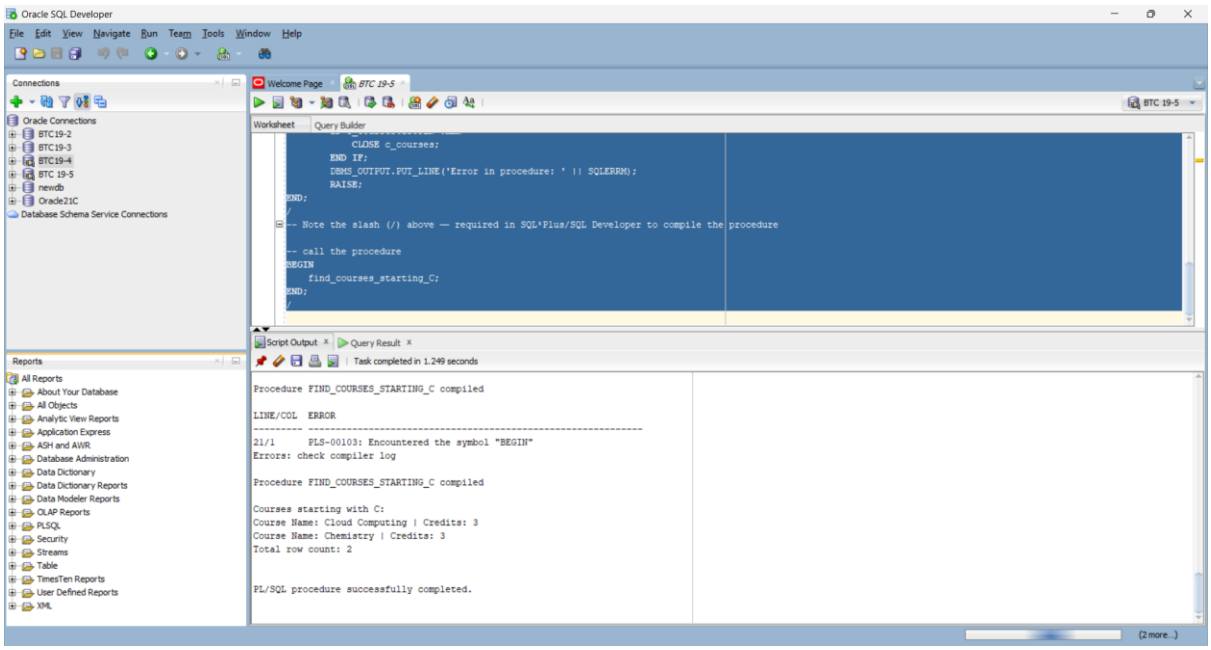



```
-- Procedure 1: Find courses starting with 'C'
SET SERVEROUTPUT ON;
CREATE OR REPLACE PROCEDURE find_courses_starting_C IS
    CURSOR c_courses IS
        SELECT course_name, credits
        FROM COURSE
        WHERE course_name LIKE 'C%';

    v_course_name   COURSE.course_name%TYPE;
    v_credits       COURSE.credits%TYPE;
```

```
BEGIN
    DBMS_OUTPUT.PUT_LINE('Courses starting with C:');
    OPEN c_courses;
    LOOP
        FETCH c_courses INTO v_course_name, v_credits;
        EXIT WHEN c_courses%NOTFOUND;

        DBMS_OUTPUT.PUT_LINE('Course Name: ' || v_course_name || ' | Credits: ' ||
v_credits);
    END LOOP;
     DBMS_OUTPUT.PUT_LINE('Total row count:' || c_courses%ROWCOUNT);
    CLOSE c_courses;
END;
BEGIN
    find_courses_starting_C;
END;
```



```
-- Procedure 2: Find courses from CSE department
CREATE OR REPLACE PROCEDURE find_courses_from_CSE IS
   CURSOR c_courses_cse IS
      SELECT course_name
      FROM COURSE
      WHERE dept_name = 'CSE';

   v_course_name COURSE.course_name%TYPE;
   v_count       NUMBER := 0;
BEGIN
   DBMS_OUTPUT.PUT_LINE('-----------------------------------------');
   DBMS_OUTPUT.PUT_LINE('Courses from CSE department:');
   DBMS_OUTPUT.PUT_LINE('-----------------------------------------');

   OPEN c_courses_cse;
   LOOP
      FETCH c_courses_cse INTO v_course_name;
      EXIT WHEN c_courses_cse%NOTFOUND;
      v_count := v_count + 1;
      DBMS_OUTPUT.PUT_LINE('Course Name: ' || v_course_name);
   END LOOP;

   CLOSE c_courses_cse;

   DBMS_OUTPUT.PUT_LINE('-----------------------------------------');
   DBMS_OUTPUT.PUT_LINE('Total row count: ' || v_count);
   DBMS_OUTPUT.PUT_LINE('-----------------------------------------');
EXCEPTION
   WHEN OTHERS THEN
      IF c_courses_cse%ISOPEN THEN
```

```
        CLOSE c_courses_cse;
    END IF;
    DBMS_OUTPUT.PUT_LINE('Error in find_courses_from_CSE: ' || SQLERRM);
END;
/

-------------------------------------------------------------
-- Execute Procedure 2
-------------------------------------------------------------
BEGIN
   find_courses_from_CSE;
END;
/

-------------------------------------------------------------
```