

EXPERIMENT 5

Name:- Omkar Kore

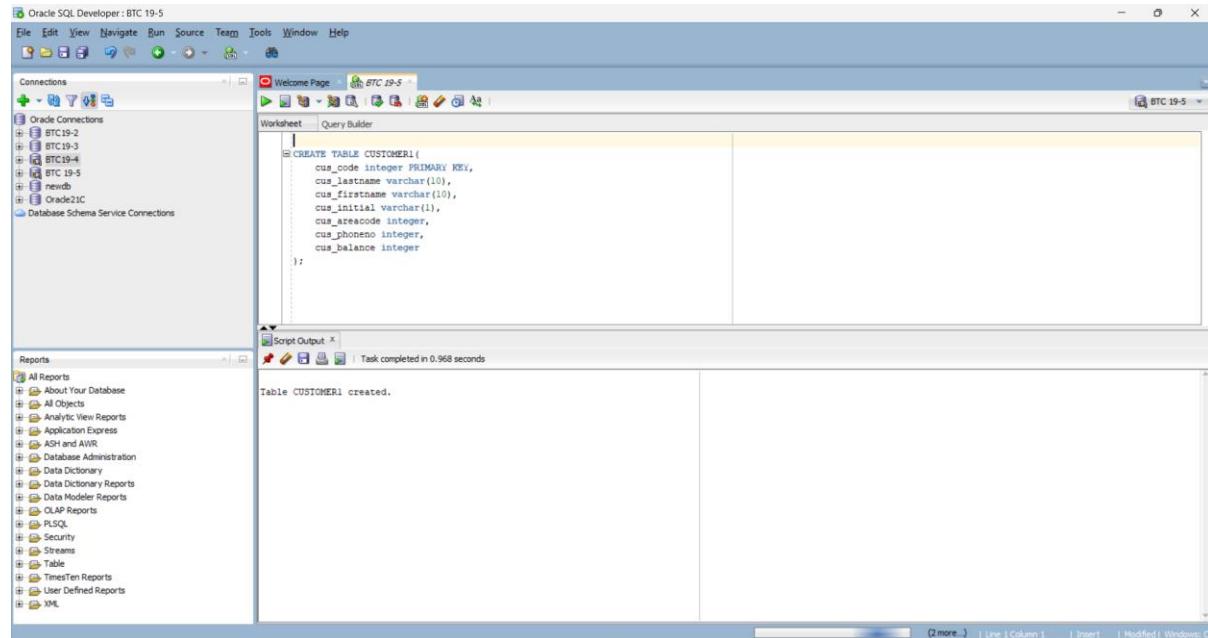
Class: Btech C

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;

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

Table CUSTOMER1 created.
Sequence CUS_SEQUENCE created.

SELECT * FROM USER_SEQUENCES;

SEQUENCE_NAME	MIN_VALUE	MAX_VALUE	INCREMENT_BY	CYCLE_FLAG	ORDER_FLAG	CACHE_SIZE	LAST_NUMBER	SCALE_FLAG	EXTEND_FLAG	SHARD
CLT_ID	1	99999999999999999999999999999999	1	N	Y	0	1M	N	N	
COMPARISON_SCAN_SEQ0	1	4254967295	1	Y	N	20	1M	N	N	
COMPARISON_SEQ0	1	99999999999999999999999999999999	1	N	N	0	1M	N	N	
CONFFLICT_HANDLER_ID_SEQ0	1	4254967295	1	N	N	0	1M	N	N	
CONS_LOGSEQ	1	99999999999999999999999999999999	1	Y	Y	0	1M	N	N	
CUS_SEQUENCE	500	600	2	N	N	0	500M	N	N	
DAM_CLEANUP_SEQ0	1	99999999999999999999999999999999	1	N	N	20	1M	N	N	
DBFS_HSG_ARCHIVEREFIDSEQ	1	99999999999999999999999999999999	1	N	Y	2	1M	N	N	
DBFS_HSG_BACKUPREFIDSEQ	1	99999999999999999999999999999999	1	N	Y	2	1M	N	N	
DBFS_HSG_POLICYIDSEQ	1	99999999999999999999999999999999	1	N	Y	2	1M	N	N	
DBFS_HSG_RSEQ	1	99999999999999999999999999999999	1	N	Y	20	1M	N	N	
DBFS_HSG_STOREIDSEQ	1	99999999999999999999999999999999	1	N	Y	2	1M	N	N	
DBFS_HSG_TARBALSEQ	1	99999999999999999999999999999999	1	N	Y	2	1M	N	N	
DBFS_SFSE_F5SEQ	1	99999999999999999999999999999999	1	N	N	6192	1M	N	N	

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;

```

The screenshot shows the Oracle SQL Developer interface. In the top-left pane, there are connections listed: BTC19-2, BTC19-3, BTC19-4, BTC 19-5, newdb, and Oracle21C. The central workspace has a 'Worksheet' tab open with the query: 'SELECT * FROM CUSTOMER1;'. Below it, the 'Query Result' tab displays the following data:

CUS_CODE	CUS_LASTNAME	CUS_FIRSTNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONENO	CUS_BALANCE
1	500 Kore	Omkar	S	1234	7896541236	30000
2	502 Patil	Rudra	D	8965	7893214561	40000
3	504 Karale	Aarav	A	2365	9659741236	60000
4	506 Kekre	Roban	G	3256	8965412378	50000
5	508 Kokane	Aditya	A	2365	7744569856	20000
6	510 Mahadik	Siddharth	S	1278	9658745698	36666
7	512 Pawar	Tejas	E	2365	7412589632	30000

--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;

```

The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar lists several database connections, including 'BTC19-2', 'BTC19-3', 'BTC19-4', 'BTC19-5', 'neirdb', and 'Oracle21C'. The 'Worksheet' tab contains the trigger creation script:

```

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;

```

Below the script, the following SQL statements are listed:

```

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;

```

The 'Script Output' tab shows the execution results:

T_ID	STUDENT_NAME	SUBJECT1	SUBJECT2	SUBJECT3	TOTAL_MARKS	PERCENTAGE
1	Omkar	12	20	20	52	87
2	Sakshi	16	12	14	42	70
3	Mohinee	17	20	19	56	93
4	Mayur	18	18	18	54	90

--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;

The screenshot shows the Oracle SQL Developer interface. The 'Connections' sidebar lists several connections, including 'BTC19-2', 'BTC19-3', 'BTC19-4', 'BTC19-5', 'Database Schema Service Connections', and 'Oracle21C'. The 'Reports' sidebar contains various report types like 'About Your Database', 'All Objects', 'Analytic View Reports', etc. The 'Worksheet' tab in the center has the sample data insertion script and the query 'SELECT * FROM COURSE;'. The 'Script Output' tab shows the execution results, including the message 'All Rows Fetched: 7 in 0.064 seconds' and a table with the following data:

COURSE_NUM	COURSE_NAME	DEPT_NAME	CREDITS
1	Cloud Computing	CSE	3
2	Blockchain	CSE	3
3	STQA	CSE	3
4	ADS	CSE	4
5	Machine Learning	CSE	3
6	Chemistry	SCI	3
7	Maths	MATH	3

-- 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;

```

The screenshot shows the Oracle SQL Developer interface. The top menu bar includes File, Edit, View, Navigate, Run, Team, Tools, Window, and Help. The left sidebar contains sections for Connections (with entries like BTC19-2, BTC19-3, BTC19-4, BTC19-5, nevdbs, and Oracle21C), Database Schema Service Connections, Reports (with All Reports, About Your Database, All Objects, Analytic View Reports, Application Express, ASH and AWR, Database Administration, Data Dictionary, Data Dictionary Reports, Data Modeler Reports, OLAP Reports, PLSQL, Security, Streams, Table, TimesTen Reports, User Defined Reports, and XML), and a Task bar with Welcome Page, BTC 19-5, and BTC 19-5.

The main workspace has two tabs: Worksheet and Query Builder. The Worksheet tab displays the following PL/SQL code:

```

  CLOSE c_courses;
END IF;
DBMS_OUTPUT.PUT_LINE('Error in procedure: ' || SQLERRM);
RAISE;
END;
-- Note the slash (/) above -- required in SQL*Plus/SQL Developer to compile the procedure
-- call the procedure
BEGIN
  find_courses_starting_C;
END;
/

```

The Script Output tab shows the results of the compilation and execution:

```

Procedure FIND_COURSES_STARTING_C compiled
LINE/COL  ERROR
-----
21/1      PLS-00103: Encountered the symbol "BEGIN"
Errors: check compiler log

Procedure FIND_COURSE_STARTING_C compiled

Courses starting with C:
Course Name: Cloud Computing | Credits: 3
Course Name: Chemistry | Credits: 3
Total row count: 2

PL/SQL procedure successfully completed.

```

```

-- 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;
```

```
/
```

