Name: Aryan Mangrule

Roll No.: BTB46

PRN: 2122000220

Experiment No.: 6

Problem Statement

Oracle Sequences:

Consider table customer with primary key(cus_code))

Field Type	Data Type
cus_code	Integer
cus_lname	varchar2(10)
cus_fname	varchar2(10)
cus_initial	varchar2(1)
cus_areacode	INTEGER
cus_phone	INTEGER
cus_balance)	number(10,2

i) **Create sequence on cus_code**

ANS:CREATE SEQUENCE CUS_SEQUENCES
START WITH 500 noCache;
drop sequence CUS_SEQUENCES;
select * from USER_SEQUENCES;

ii) Display user sequences

ANS:

Script Output × | Query Result × | Query Result 1 ×

SQL | All Rows Fetched: 7 in 0.145 seconds

	SEQUENCE_NAME	MIN_VALUE	MAX_VALUE	INCREMENT_BY	CYCLE_FLAG	ORDER_FLAG	CACHE_SIZE	LAST_NUMBER	SCALE_FLAG	EXTEND_FLAG	SHARDED_FLAG	SESSION_FLAG	KEEP_VALUE
1	LOGMNR_DIDS$	1	9999999999999999999999999999	1	N	Y	0	1	N	N	N	N	N
2	LOGMNR_EVOLVE_SEQ$	1	9999999999999999999999999999	1	N	Y	0	1	N	N	N	N	N
3	LOGMNR_SEQ$	1	9999999999999999999999999999	1	N	Y	0	1	N	N	N	N	N
4	LOGMNR_UIDS$	100	99999	1	Y	Y	0	100	N	N	N	N	N
5	MVIEW$_ADVSEQ_GENERIC	1	4294967295	1	N	N	50	1	N	N	N	N	N
6	MVIEW$_ADVSEQ_ID	1	4294967295	1	N	N	20	1	N	N	N	N	N
7	ROLLING_EVENT_SEQ$	1	9999999999999999999999999999	1	N	Y	0	1	N	N	N	N	N

iii) Insert values into customer using created sequence

ANS:

insert into customer values(CUS_SEQUENCES.nextval,'Sujay','Gagan',null,'615','7878448841',1000.00);
insert into customer values(CUS_SEQUENCES.nextval,'Ram','Patil',null,'616','8956231245',1100.00);
insert into customer values(CUS_SEQUENCES.nextval,'Sham','Jadhav',null,'617','9865451283',600.00);
insert into customer values(CUS_SEQUENCES.nextval,'Rohan','Yadav',null,'618','9235628945',1100.00);

insert into customer values(CUS_SEQUENCES.nextval,'Pushkaraj','Yadav',null,'619','897844884',1800.00);

select *from customer;

iv) Display customer records

v) **ANS:s**

Script Output × | Query Result × | Query Result 1 × | Query Result 2 ×

SQL | All Rows Fetched: 5 in 0.027 seconds

	CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_BALANCE
1	505	Sujay	Gagan	(null)	615	7878448841	1000
2	506	Ram	Patil	(null)	616	8956231245	1100
3	507	Sham	Jadhav	(null)	617	9865451283	600
4	508	Rohan	Yadav	(null)	618	9235628945	1100
5	509	Pushkaraj	Yadav	(null)	619	897844884	1800

Trigger:

Consider Student Report table, in which student marks assessment is recorded. In such schema, create a trigger so that the total and percentage of specified marks is automatically inserted whenever a record is inserting. Initial insert 0 for total and per attributes. Maximum marks should be 20 for each subject

Field | Type | Null | Key |

+------+-----+---------+----------+

| tid | int(4) | NO | PRI |

| name | varchar(30) | YES |

| subj1 | int(2) | YES |

| subj2 | int(2) | YES |

| subj3 | int(2) | YES |

| total | int(3) | YES |

| per | int(3) | YES |

-- Creating the trigger to calculate total and percentage before insert

```
create table
  student_report ( tid
  number(4) primary key,
```

```
  name varchar2(30),
  subj1 number(2) check (subj1 > 0 and subj1 <=
  20),subj2 number(2) check (subj2 > 0 and subj2
  <= 20), subj3 number(2) check (subj3 > 0 and
  subj3 <= 20),total number(3) default 0,
  per number(3) default 0
);
create or replace trigger calc_total_perc
before insert or update on
student_reportfor each row

  :new.total := nvl(:new.subj1, 0) + nvl(:new.subj2, 0) + nvl(:new.subj3, 0);
  :new.per := (:new.total * 100) /
60;end;

insert into student_report (tid, name, subj1, subj2,
subj3,total,per)values (1, 'Alice', 18, 15, 17,0,0);
insert into student_report (tid, name, subj1, subj2,
subj3,total,per)values (2, 'bob', 18, 15, 17,0,0);
```

ANS:

```
--Checking constraint
insert into student_report (tid, name, subj1, subj2,
subj3)values (2, 'Bob', 0, 10, 12);
```

```
Script Output x   Query Result x
Task completed in 0.15 seconds
Error starting at line : 61 in command -
insert into student_report (tid, name, subj1, subj2, subj3)
values (2, 'Bob', 0, 10, 12)
Error report -
ORA-02290: check constraint (SYSTEM.SYS_C008351) violated
```

--Checking triggered or not

select * from

student_report;

Script Output x Query Result x

SQL | All Rows Fetched: 2 in 0.006 seconds

	TID	NAME	SUBJ1	SUBJ2	SUBJ3	TOTAL	PER
1	1	Alice	18	15	17	50	83
2	2	bob	18	15	17	50	83

Procedure and Cursor:

Consider Course Table with course_num as primary key.

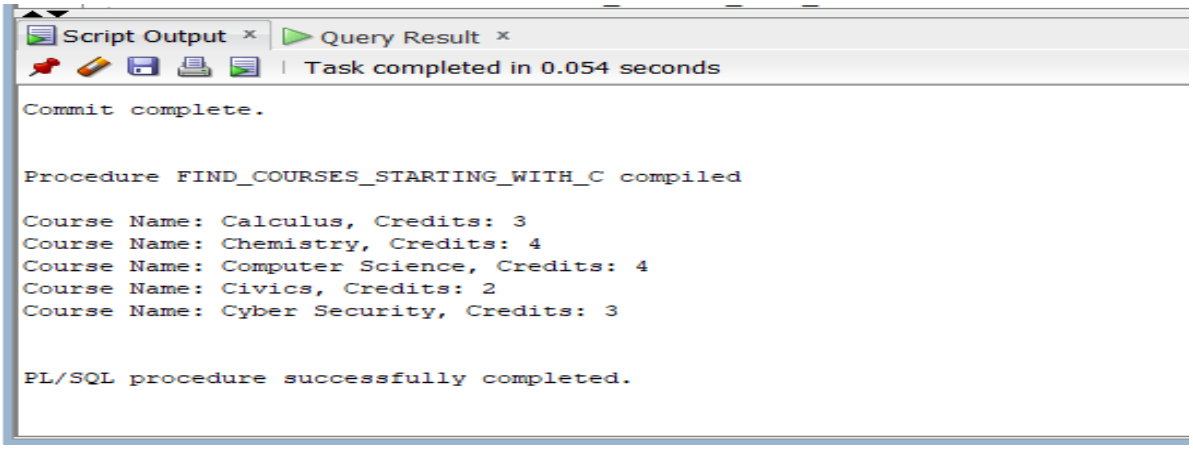
Field Type	Data Type
course_num	Integer
course_name	varchar2(20)
dept_name	varchar2(15)
credits	Integer

1.)Write a procedure which includes cursors: Find course_name and credits where course name starts with 'C'

```
create table Course(
course_num integer primary key,
course_name    varchar2(20),
dept_name varchar2(15),
credits integer
)
drop table course;
INSERT INTO Course (course_num, course_name, dept_name, credits) VALUES (101, 'Calculus', 'MATH', 3);
```

```
INSERT INTO Course (course_num, course_name, dept_name, credits) VALUES (102, 'Chemistry', 'SCIENCE', 4);

INSERT INTO Course (course_num, course_name, dept_name, credits) VALUES (103, 'Computer Science', 'CSE', 4);

INSERT INTO Course (course_num, course_name, dept_name, credits) VALUES (104, 'Biology', 'SCIENCE', 3);

INSERT INTO Course (course_num, course_name, dept_name, credits) VALUES (105, 'Civics', 'ARTS', 2);

INSERT INTO Course (course_num, course_name, dept_name, credits) VALUES (106, 'Physics', 'SCIENCE', 4);

INSERT INTO Course (course_num, course_name, dept_name, credits) VALUES (107, 'Cyber Security', 'CSE', 3);

CREATE OR REPLACE PROCEDURE find_courses_start_with_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

  -- Opening and fetching cursor data

  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;
```

```
    CLOSE c_courses;
END;
SET SERVEROUTPUT ON;

BEGIN
    find_courses_starting_with_C;
END;
```



2.) Write a procedure which includes cursors: Find course names 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;
BEGIN
    OPEN c_courses_cse;

    LOOP
        -- Fetch data from the cursor into variable
```

```
        FETCH c_courses_cse INTO v_course_name;

        EXIT WHEN c_courses_cse%NOTFOUND;

        DBMS_OUTPUT.PUT_LINE('Course Name: ' || v_course_name);
    END LOOP;

    CLOSE c_courses_cse;
END;

SET SERVEROUTPUT ON;

BEGIN
    find_courses_from_CSE;
END;
```

ANS: