```
                         *** EXPERIMENT NO: 03 ***

-------------------------------------------------------------------------------
Author : Pradyot Patil

Roll No : 53 [6B]

Date : 28-February-2018

-------------------------------------------------------------------------------
*******************************************************************************
```

QUERY-01: Write SQL code to establish the schema (including enforcement of integrity constraints). Populate DEPT with indicated tuples. [Note: While creating DEPT do not enforce the referential integrity constraint]. You should use default value for HOD while inserting tuples in DEPT. Also populate first 6 tuples in STAFF

```
*******************************************************************************
CREATE TABLE DEPT (
             DNAME VARCHAR(25) NOT NULL,
             BRANCH CHAR(2) NOT NULL,
             INTAKE NUMBER(2) NOT NULL,
             YR_EST NUMBER(4) NOT NULL,
             HOD NUMBER(3) DEFAULT 101,
             CONSTRAINTS DEPT_PK_BRANCH PRIMARY KEY (BRANCH),
             CONSTRAINT DEPT_CK_BRANCH CHECK(BRANCH IN ('EN','IT','CS')),
             CONSTRAINT DEPT_CK_YR_EST CHECK(YR_EST >2005),
             CONSTRAINT DEPT_CK_INTAKE CHECK(INTAKE IN (20,30,40))
      );
Table created.

CREATE TABLE STAFF (
             SID NUMBER(3) NOT NULL,
             NAME VARCHAR(25) NOT NULL,
             BRANCH CHAR(2) NOT NULL,
             DESG VARCHAR(9) NOT NULL,
             JOIN_DT DATE NOT NULL,
             CONSTRAINTS STAFF_PK_SID PRIMARY KEY (SID),
             CONSTRAINT STAFF_CK_SID CHECK(SID>100),
             CONSTRAINT STAFF_CK_DESG CHECK(DESG IN
             ('Professor','Assistant','Associate')),
             CONSTRAINT STAFF_FK_BRANCH FOREIGN KEY(BRANCH) REFERENCES DEPT(BRANCH)
      );
Table created.
```

```
CREATE TABLE STUDENT(
             ROLL NUMBER(5) NOT NULL,
             LNAME VARCHAR(15) NOT NULL,
             FNAME VARCHAR(15) NOT NULL,
             EMAIL VARCHAR(25) UNIQUE,
             ENROLL CHAR(9) UNIQUE,
             SID NUMBER(3) ,
             CONSTRAINTS STUDENT_PK_ROLL PRIMARY KEY (ROLL),
             CONSTRAINT STUDENT_FK_SID FOREIGN KEY(SID) REFERENCES STAFF(SID)
      );
Table created.

INSERT INTO DEPT VALUES('Computer Science', 'CS', 40, 2006, 101);
1 row created.
INSERT INTO DEPT VALUES('Information Technology', 'IT', 20, 2007, 106) ;
1 row created.
INSERT INTO DEPT VALUES('Electronics Engineering', 'EN', 30, 2007, 107) ;
1 row created.

INSERT INTO STAFF VALUES(101, 'Kamalkant Marathe', 'CS', 'Professor', '12-Jun-
2005');
1 row created.

SQL> INSERT INTO STAFF VALUES(102, 'Adishesh Vidyarthi', 'CS', 'Associate', '22-
Jul-2006');
1 row created
```

****

QUERY-02 Create a sequence STAFF_SQ with appropriate starting value and maximum range such that you can use it to populate STAFF table with remaining tuples. [Use STAFF_SQ.NEXTVAL, STAFF_SQ.CURRVAL to access sequence values]. Verify if the sequence has been created [use USER_CONSTRAINTS table] along with other sequences on current schema tables. On populating STAFF, remove the sequence.

****

```
CREATE SEQUENCE STAFF_SQ
  2         INCREMENT BY 1
  3         START WITH 107
  4         MINVALUE 101
  5         MAXVALUE 110;
Sequence created.
```

```
SQL> INSERT INTO STAFF VALUES (STAFF_SQ.NEXTVAL, 'Ramanathan Arun', 'EN',
'Professor', '12-Aug-2005');
1 row created.

SQL> INSERT INTO STAFF VALUES (STAFF_SQ.NEXTVAL, 'Saifuddin Sheikh', 'EN',
'Associate', '19-Sep-2010');
1 row created.

SQL> INSERT INTO STAFF VALUES (STAFF_SQ.NEXTVAL, 'Babush Baltiwala', 'EN',
'Assistant', '12-Apr-2012');
1 row created.

SQL> INSERT INTO STAFF VALUES (STAFF_SQ.NEXTVAL, 'Christopher Kundu', 'CS',
'Assistant', '13-Mar-2013');
1 row created.

SELECT * FROM USER_SEQUENCES ;
SEQUENCE_NAME    MIN_VALUE  MAX_VALUE INCREMENT_BY C O CACHE_SIZE  LAST_NUMBER
---------------- ---------- ---------- ------------ - - ---------- -----------
STAFF_SQ                101        110            1 N N         20         111

DROP SEQUENCE STAFF_SQ ;
Sequence dropped.

*******************************************************************************
QUERY-03: Write SQL codes to populate STUDENT table with indicated tuples. Now,
enforce referential integrity constraint on DEPT [use ALTER TABLE ... ADD …]. You
will notice that HOD attribute for all departments is a default value of 101.
Rectify DEPT for the correct values of HOD attribute as indicated in sample data
[use UPDATE … SET … WHERE …].
*******************************************************************************
INSERT INTO STUDENT VALUES (3001, 'Agrawal', 'Aditi', 'agrawalaa8@rknec.edu',
'MT14CS001', 101);
1 row created.

INSERT INTO STUDENT VALUES (3002, 'Jadhao', 'Ankita', 'jadhaoar@rknec.edu',
'MT14CS002', 102);
1 row created.

ALTER TABLE DEPT ADD CONSTRAINT DEPT_FK_SID FOREIGN KEY(HOD) REFERENCES STAFF(SID)
;
Table altered.
```

```
SQL> SELECT * from DEPT ;

DNAME                     BR     INTAKE     YR_EST        HOD
------------------------- -- ---------- ---------- ----------
Computer Science          CS         40       2006        101
Information Technology    IT         20       2007        101
Electronics Engineering   EN         30       2007        101


update DEPT set HOD=106 where BRANCH='IT' ;
1 row updated.

update DEPT set HOD=107 where BRANCH='EN' ;
1 row updated.


DELETE FROM VENDOR WHERE V_CODE=21344 ;
```

****

QUERY-04: Write a SQL code that will create a temporary table (view) named STUDENT_VW on STUDENT table projecting the attributes ROLL, LNAME, FNAME, SID. List the contents of STUDENT_VW. Also list all the views for the current schema tables [use USER_VIEWS table].

****

```
CREATE OR REPLACE VIEW STUDENT_VW AS
  2         SELECT ROLL,LNAME,FNAME,SID FROM STUDENT;
View created.

SELECT * FROM STUDENT_VW ;
      ROLL LNAME          FNAME                 SID
---------- -------------- -------------- ----------
      3001 Agrawal        Aditi                 101
      3002 Jadhao         Ankita                102
      3003 Rathi          Charulata             101
      3004 Rathi          Divya                 101
      3005 Gadiya         Minal                 103
      3006 Naxane         Prajakta              101
      3007 Borele         Pranali               102
      3008 Kushwaha       Preeti                103
      3009 Mundada        Priya                 102
      3010 Agrawal        Ruchi                 110
      3011 Khatwani       Sneha                 110
```

```
ROLL LNAME           FNAME                  SID
---------- --------------- --------------- ----------
      3012 Pannase         Sonal                  110
      3013 Nikhar          Sonam                  103
      3014 Hardeniya       Tanvi                  110
      3015 Ninawe          Ujwala                 103
      3016 Bhogadhi        Vani                   101
      3017 Mal             Vishakha               110
      3018 Gowardhan       Yamini                 110
      3019 Rathi           Ankit                  103
      3020 Palaskar        Hanok                  101
      3021 Shahu           Ishankumar             103
      3022 Dongre          Rushikesh              102
      3023 Jain            Saurabh                102
      3024 Sathawane       Vishal                 103
      4001 Jain            Aarju                  104
      4002 Dixit           Ankita                 105
      4003 Tiwari          Ankita                 106
      4004 Shah            Arti                   105
      4005 Hinge           Ashwini                106
      4006 Singh           Asmita                 104
      4007 Chaudhari       Bhagyashree            106
      4008 Madan           Devyani                104
      4009 Bhojwani        Kanchan                105
      4010 Gandhi          Kripali                106
      5001 Nisal           Namita                 107
      5002 Pathan          Needa                  109
      5003 Agrawal         Nikita                 108
      5004 Kalra           Nikita                 108
      5005 Sharma          Roopa                  109
      5006 Adgurwar        Sayli                  108
      5007 Harode          Shivani                107
      5008 Thokal          Shweta                 108
      5009 Gupta           Suruchi                107
      5010 Sharma          Aashish                109
      5011 Ganediwal       Abhay                  109
      5012 Dixit           Abhishek               107
      4012 Tiwari          Abhishek               105
      4011 Parmar          Abhishek               104

48 rows selected.
```

```
SELECT VIEW_NAME, VIEW_TYPE, READ_ONLY FROM USER_VIEWS;

VIEW_NAME                      VIEW_TYPE                      R
------------------------------ ------------------------------ -
STUDENT_VW                                                    N
```

```
***********************************************************************************
```

QUERY-05: Write a SQL code to insert students Sheela Goenka and Vallabh Shastri with roll numbers 3025 and 5013 respectively, into STUDENT_VW and observe the effect on STUDENT table. Now create a view STU_AFFL_VW that will include attributes ROLL, LNAME, FNAME, BRANCH. Insert into STU_AFFL_VW, a tuple – 4013, Dawson, Ellis, IT. Observe the output and analyze the problems encountered.

```
***********************************************************************************
```

```
INSERT INTO STUDENT_VW (ROLL,LNAME,FNAME) VALUES (3025, 'Goenka', 'Sheela');
1 row created.
INSERT INTO STUDENT_VW (ROLL,LNAME,FNAME) VALUES (5013, 'Shastri', 'Vallabh');
1 row created.

SELECT * FROM STUDENT WHERE LNAME=’GOENKA’ OR LNAME=‘Shastri’ ;
   ROLL      LNAME           FNAME    EMAIL           ENROLL       SID
   ---- ---------- -------------- --------  ---------------     -----
   3025     Goenka          Sheela
   5013    Shastri         Vallabh
2 rows selected .

CREATE VIEW STU_AFFL_VW AS
  2         SELECT ROLL,LNAME,FNAME,BRANCH FROM STUDENT S
  3         JOIN STAFF D
  4         ON S.SID=D.SID ;
View created.

INSERT INTO STU_AFFL_VW (ROLL,LNAME,FNAME,BRANCH) VALUES (4013, 'Dawson',
'Ellis','IT');

INSERT INTO STU_AFFL_VW (ROLL,LNAME,FNAME,BRANCH) VALUES (4013, 'Dawson',
'Ellis','IT')
 *
ERROR at line 1:
ORA-01776: cannot modify more than one base table through a join view
```

****

QUERY-06: Write SQL code to create a view STUDENT_VW_RO on STUDENT table with READ ONLY option with same attribute set as in STUDENT_VW. List the contents of STUDENT_VW_RO. Now insert a student – Rory McLaren, 5015 – using STUDENT_VW_RO. Observe the effect

****

```
CREATE OR REPLACE VIEW STUDENT_VW_RO AS
  2        SELECT ROLL,LNAME,FNAME,SID FROM STUDENT
  3        WITH READ ONLY;
View created.

SELECT * FROM VIEW STUDENT_VW_RO ;
50 rows selected.

INSERT INTO STUDENT_VW_RO (ROLL,LNAME,FNAME) VALUES (5015, 'McLaren', 'Rory');
INSERT INTO STUDENT_VW_RO (ROLL,LNAME,FNAME) VALUES (5015, 'McLaren', 'Rory')
                           *
ERROR at line 1:
ORA-42399: cannot perform a DML operation on a read-only view
```

****

QUERY-07: Write SQL code to create a view STUDENT_VW_CK on STUDENT table with CHECK OPTION and CONSTRAINT with same attribute set as in STUDENT_VW but will include those tuples having advisors among 101, 106 and 107. Name the constraint as STUDENT_ADV_CK. List the contents of STUDENT_VW_CK. Now insert a student – Albert Lambda, 4014 – using STUDENT_VW_CK. Observe the effect.

****

```
CREATE OR REPLACE VIEW  STUDENT_VW_CK AS
  2   SELECT * FROM STUDENT WHERE SID IN (101, 106, 107)
  3   WITH CHECK OPTION CONSTRAINT TEST_ADV_CK ;
View created.

SELECT * FROM STUDENT_VW_CK ;

      ROLL LNAME           FNAME                  SID
---------- --------------- --------------- ----------
      3001 Agrawal         Aditi                  101
      3003 Rathi           Charulata              101
      3004 Rathi           Divya                  101
      3006 Naxane          Prajakta               101
      3016 Bhogadhi        Vani                   101
      3020 Palaskar        Hanok                  101
      4003 Tiwari          Ankita                 106
      4005 Hinge            Ashwini                106
      4007 Chaudhari        Bhagyashree            106
      4010 Gandhi           Kripali                106
      5001 Nisal            Namita                 107
      5007 Harode           Shivani                107
      5009 Gupta            Suruchi                107
      5012 Dixit            Abhishek               107

14 rows selected.
```

```
INSERT INTO STUDENT_VW_CK (ROLL,LNAME,FNAME) VALUES (4014, 'Lambda', 'Albert');
INSERT INTO STUDENT_VW_CK (ROLL,LNAME,FNAME) VALUES (4014, 'Lambda', 'Albert')
            *
ERROR at line 1:
ORA-01402: view WITH CHECK OPTION where-clause violation
```

****

QUERY-08: Write a SQL code to create a private synonym TEACHER_SN for STAFF. Use this synonym to show contents of STAFF. A faculty named Geoffrey Ball has been appointed as Associate in EN. Insert Ball record using TEACHER_SN. Observe contents of STAFF table.

****

```
CREATE OR REPLACE SYNONYM TEACHER_SN FOR STAFF;
Synonym created.

SELECT * FROM TEACHER_SN ;
  SID NAME                     BR DESG      JOIN_DT
---------- ------------------------- -- --------- ---------
       101 Kamalkant Marathe         CS Professor 12-JUN-05
       102 Adishesh Vidyarthi        CS Associate 22-JUL-06
       103 Aasawari Deodhar          CS Assistant 13-OCT-07
       104 Deo Narayan Mishra        IT Assistant 13-OCT-07
       105 Jasmine Paul              IT Associate 12-MAY-08
       106 Manishi Singh             IT Professor 11-NOV-09
       107 Ramanathan Arun           EN Professor 12-AUG-05
       108 Saifuddin Sheikh          EN Associate 19-SEP-10
       109 Babush Baltiwala          EN Assistant 12-APR-12
       110 Christopher Kundu         CS Assistant 13-MAR-13
10 rows selected.

INSERT INTO TEACHER_SN VALUES(111, 'Geoffrey Ball', 'EN', 'Associate', '20-Feb-
2018');
1 row created.
```

```
SELECT * FROM TEACHER_SN WHERE NAME=’ Geoffrey Ball’ ;

SID NAME                      BR DESG      JOIN_DT
---------- ------------------------ -- --------- ---------
       111 Geoffrey Ball            EN Associate 20-FEB-18
1 row selected.
```

****

QUERY-09: Change to User – SYSTEM. Write a SQL code to create a table STAFF (after ensuring object‘s non-existence) while excluding RI-constraint enforcement but enforcing domain constraint instead. Populate STAFF with first 8 tuples as indicated. Now, create a public synonym TEACHER_SNP for STAFF and verify its presence by inserting tuple-9 using TEACHER_SNP. [Allocate to user CS6XX the privilege to SELECT, INSERT on STAFF].

****

```
DISCONNECT
CONNECT system

DESC STAFF ;
ERROR:
ORA-04043: object STAFF  does not exist

CREATE TABLE STAFF (
  2   SID NUMBER(3) NOT NULL,
  3   NAME VARCHAR(25) NOT NULL,
  4   BRANCH CHAR(2) NOT NULL,
  5    DESG VARCHAR(9) NOT NULL,
  6    JOIN_DT DATE NOT NULL,
  7    CONSTRAINT STAFF_PK_SID PRIMARY KEY (SID),
  8    CONSTRAINT STAFF_CK_SID CHECK(SID>100),
  9    CONSTRAINT STAFF_CK_DESG CHECK(DESG IN ('Professor','Assistant','Associate'))
 10         );
Table created.

INSERT INTO STAFF VALUES(101, 'Kamalkant Marathe', 'CS', 'Professor', '12-Jun-2005');
1 row created.

INSERT INTO STAFF VALUES (108, 'Saifuddin Sheikh', 'EN', 'Associate', '19-Sep-2010');
1 row created.

CREATE PUBLIC SYNONYM TEACHER_SNP FOR STAFF;
Synonym created.
```

```
INSERT INTO TEACHER_SNP VALUES(109, 'Babush Baltiwala', 'EN', 'Assistant', '12-Apr-2012');
1 row created.

GRANT
  2    SELECT,
  3    INSERT
  4  ON
  5    STAFF
  6  TO
  7    CS653;
Grant succeeded.
```

****

QUERY-10: Change to User – CS6XX. Write a SQL code to display the contents of STAFF table of SYSTEM user with/without using TEACHER_SNP. Now insert tuple-10 and tuple-11 using TEACHER_SNP.

****

```
SELECT * FROM TEACHER_SNP ;
Or
SELECT * FROM SYSTEM.STAFF ;

SID NAME                      BR DESG      JOIN_DT
---------- ------------------------- -- --------- ---------
       101 Kamalkant Marathe         CS Professor 12-JUN-05
       102 Adishesh Vidyarthi        CS Associate 22-JUL-06
       103 Aasawari Deodhar          CS Assistant 13-OCT-07
       104 Deo Narayan Mishra        IT Assistant 13-OCT-07
       105 Jasmine Paul              IT Associate 12-MAY-08
       106 Manishi Singh             IT Professor 11-NOV-09
       107 Ramanathan Arun           EN Professor 12-AUG-05
       108 Saifuddin Sheikh          EN Associate 19-SEP-10
       109 Babush Baltiwala          EN Assistant 12-APR-12
9 rows selected.

INSERT INTO TEACHER_SNP VALUES(110, 'Christopher Kundu', 'CS', 'Assistant', '13-Mar-2013');
1 row created.
SQL> INSERT INTO TEACHER_SNP VALUES(111, 'Geoffrey Ball', 'EN', 'Associate', '20-Feb-2018');
1 row created.
```

****

QUERY-11: Write a SQL code to create a unique B-Tree index on LNAME attribute of STUDENT. Observe the output and report the problem(s). If it fails, create BTree index and test it to locate a certain customer by last name. Now create a concatenated B-tree index on (LNAME, FNAME) attributes of STUDENT and test the index. Also list all indexes for CS6XX for the current database schema [use USER_INDEXES table].

****

```
CREATE UNIQUE INDEX STUDENT_NDX_LNAME ON STUDENT(LNAME);
CREATE UNIQUE INDEX STUDENT_NDX_LNAME ON STUDENT(LNAME)
                                         *
ERROR at line 1:
ORA-01452: cannot CREATE UNIQUE INDEX; duplicate keys found

CREATE INDEX STUDENT_NDX_LNAME ON STUDENT(LNAME);
Index created.
SET AUTOTRACE ON ;
SELECT LNAME,FNAME,ENROLL FROM STUDENT WHERE LNAME='Sharma' ;

LNAME           FNAME           ENROLL
--------------- --------------- ---------
Sharma          Roopa           14MTEN010
Sharma          Aashish         14MTEN006

2 rows selected.


Execution Plan
----------------------------------------------------------
Plan hash value: 2942141181

--------------------------------------------------------------------------------
-----------------

| Id  | Operation                    | Name              | Rows | Bytes |Cost(%CPU)|
--------------------------------------------------------------------------------
|   0 | SELECT STATEMENT            |                   |    1 |    26 |     2 (0)|

|   1 |  TABLE ACCESS BY INDEX ROWID| STUDENT           |    1 |    26 |     2 (0)|

|   2 |   INDEX RANGE SCAN          | STUDENT_NDX_LNAME |    1 |       |     1 (0)|

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

```
Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("LNAME"='Sharma')
```

CREATE INDEX STUDENT_NDX_LNAME_FNAME ON STUDENT(FNAME|| ' ' ||LNAME);

```
Index created.
```

SELECT LNAME,FNAME,ENROLL FROM STUDENT WHERE (FNAME||' '||LNAME)='Aashish Sharma' ;

```
LNAME           FNAME           ENROLL
--------------- --------------- ---------
Sharma          Aashish         14MTEN006

1 row selected.

Execution Plan
----------------------------------------------------------
Plan hash value: 2604432115

--------------------------------------------------------------------------------
-----------------

|Id| Operation                   | Name                 |Rows|Bytes|Cost(%CPU)|
--------------------------------------------------------------------------------
| 0| SELECT STATEMENT            |                      |  1 |  41 |    2 (0)|

| 1| TABLE ACCESS BY INDEX ROWID| STUDENT               |  1 |  41 |    2 (0)|

| 2| INDEX RANGE SCAN           |STUDENT_NDX_LNAME_FNAME|  1 |     |    1 (0)|

-------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("FNAME"||' '||"LNAME"='Aashish Sharma')
```

SELECT INDEX_NAME, TABLE_NAME, UNIQUENESS, STATUS FROM USER_INDEXES ;

```
INDEX_NAME                     TABLE_NAME                     UNIQUENES STATUS
------------------------------ ------------------------------ --------- --------
STUDENT_PK_ROLL                STUDENT                        UNIQUE    VALID
```

```
SYS_C0011969                   STUDENT                        UNIQUE    VALID
SYS_C0011970                   STUDENT                        UNIQUE    VALID
STUDENT_NDX_LNAME              STUDENT                        NONUNIQUE VALID
STUDENT_NDX_LNAME_FNAME        STUDENT                        NONUNIQUE VALID
STAFF_PK_SID                   STAFF                          UNIQUE    VALID
DEPT_PK_BRANCH                 DEPT                           UNIQUE    VALID

7 rows selected.
```

```
********************************************************************************
```

QUERY-12: Write a SQL code to create a function-based index on LNAME attribute of students such that case-sensitivity is superseded by converting to uppercase/lowercase and test the index. Now create a concatenated functionbased index on (LNAME, FNAME) attributes of STUDENT and test the index. [Before testing the function-based index, the DBA must set the initialization parameter QUERY_REWRITE_ENABLED to true. Use…

CONNECT system/manager ALTER SYSTEM SET QUERY_REWRITE_ENABLED=TRUE;]

```
********************************************************************************
CREATE INDEX STUDENT_NDX_LNAME_FN ON STUDENT( UPPER(LNAME) );
Index created.

SELECT LNAME,FNAME,ENROLL FROM STUDENT WHERE UPPER(LNAME)='SHARMA' ;
LNAME           FNAME           ENROLL
--------------- --------------- ---------
Sharma          Roopa           14MTEN010
Sharma          Aashish         14MTEN006

2 rows selected.

CREATE INDEX STUDENT_NDX_LNAME_FNAME_FN ON STUDENT(LNAME,FNAME);
Index created.

SELECT LNAME,FNAME,ENROLL FROM STUDENT WHERE LNAME='Sharma' AND FNAME='Aashish' ;
LNAME           FNAME           ENROLL
--------------- --------------- ---------
Sharma          Aashish         14MTEN006

1 row selected.
Execution Plan
----------------------------------------------------------
Plan hash value: 648059635

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

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

|Id| Operation                    | Name                        |Rows|Bytes|Cost(%CPU)|
--------------------------------------------------------------------------------
| 0| SELECT STATEMENT             |                             |  1 |  26 |    2 (0)|

| 1| TABLE ACCESS BY INDEX ROWID| STUDENT                     |  1 |  26 |    2 (0)|

| 2| INDEX RANGE SCAN             |STUDENT_NDX_LNAME_FNAME_FN|  1 |     |    1 (0)|

-------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("FNAME"||' '||"LNAME"='Aashish Sharma')
```

****

QUERY-13: Write a SQL script that will (a) insert a student Krishh Umredkar with roll number 5014; (b) Assign Sheela Goenka to advisor 110; (c) Assign Vallabh Shashri to advisor 109. Before insert create a savepoint SP_NONE. After insert create savepoint SP_KRISHH. Create savepoints SP_SHEELA and SP_VALLABH after mentioned updates in sequence.

****

```
SAVEPOINT SP_NONE ;
Savepoint created.

INSERT INTO STUDENT (ROLL,LNAME,FNAME) VALUES (5014, 'Umredkar', 'Krishh');
1 row created.

SAVEPOINT SP_KRISHH ;
Savepoint created.

UPDATE STUDENT SET SID=110 WHERE (FNAME|| ' ' ||LNAME)='Sheela Goenka' ;
1 row updated.

SAVEPOINT SP_SHEELA ;
Savepoint created.

UPDATE STUDENT SET SID=109 WHERE (FNAME|| ' ' ||LNAME)='Vallabh Shastri' ;
1 row updated.

SAVEPOINT SP_VALLABH ;
Savepoint created.
```

```
SELECT LNAME,FNAME,SID FROM STUDENT
WHERE (FNAME|| ' ' ||LNAME)='Sheela Goenka'
OR (FNAME|| ' ' ||LNAME)='Vallabh Shastri'
OR (FNAME|| ' ' ||LNAME)='Krishh Umredkar' ;

LNAME           FNAME                 SID
--------------- --------------- ----------
Goenka          Sheela                110
Krishh          Umredkar
Shastri         Vallabh               109
3 rows selected.
```

****

QUERY-14: Write SQL code to reinstate to database state after executing Query-13(a). Now revert to regain the database state before executing Query-13.

****

```
ROLLBACK TO SP_KRISHH ;
Rollback complete.

SELECT LNAME,FNAME,SID FROM STUDENT
WHERE (FNAME|| ' ' ||LNAME)='Sheela Goenka'
OR (FNAME|| ' ' ||LNAME)='Vallabh Shastri'
OR (FNAME|| ' ' ||LNAME)='Krishh Umredkar' ;

LNAME           FNAME                 SID
--------------- --------------- ----------
Goenka          Sheela
Krishh          Umredkar
Shastri         Vallabh
3 rows selected.

ROLLBACK TO SP_NONE ;
Rollback complete.

SELECT LNAME,FNAME,SID FROM STUDENT
WHERE LNAME='Krishh' ;

no rows selected
```