```
                ****************Experiment no:-03*****************
------------------------------------------------------------------------
                          Author:Saurabh Khandagale
                          Roll No:46
                          Date :27-August-2020
------------------------------------------------------------------------
```

EXPERIMENT:-03

AIM:-

TO ESTABLISH A MULTI-RELATION DATABASE AND EXECUTE SQL QUERIES INVOLVING INSERTIONS,DELETIONS AND UPDATE ON IT.

PROBLEM STATEMENT:

Establish the Academic Database schema, for demonstrating creation, updating and usage of Oracle objects – views, synonyms, indexes, sequences and savepoints. Execute queries based on the logical schemata given below...

STUDENT (ROLL, LNAME, FNAME, EMAIL, ENROLL, ADVISOR, PHONE, REG_DT)
STAFF (SID, NAME, BRANCH, DESG, JOIN_DT) DEPT (DNAME, BRANCH, INTAKE, YR_EST, HOD)

Relation: STUDENT
Attribute Name ROLL LNAME FNAME EMAIL ENROLL ADVISOR PHONE REG_DT
Data Description NUMBER (3) VARCHAR (15) VARCHAR (15) VARCHAR (25) CHAR (13) NUMBER (3) NUMBER (10) DATE
Remarks El-column, [1, 300] Required Required Unique Unique RI-Column, Required, Ref STAFF (SID)
Required

Relation: STAFF
Attribute Name SID NAME
BRANCH DESG JOIN_DT
Data Description NUMBER (3) VARCHAR (25) VARCHAR (4) VARCHAR (9) DATE
Remarks El-column, Starts 101 Required RI-Column, Required Required, [Professor, Associate, Assistant] Required

Relation: DEPT
Attribute Name DNAME BRANCH INTAKE YR_EST HOD
Data Description VARCHAR (25) VARCHAR (4) NUMBER (2) NUMBER (4) NUMBER (3)
Remarks Required El-column, [CSE, DAT, AIML, CSEC] Required, [20, 30, 40] Required, YR_EST > 2004 RI-column, Ref STAFF(SID), default 101

The SQL script including Academic Database configuration, constraints enforcement and populating the tables will be shared on Google Classroom. [**academic.sql]**

```
=================================Query-02====================================
      Write SQL code to roll number, print first name, last name, advisor
      name for your roll number.
=============================================================================
```

```
  SELECT STUDENT.ROLL ,STUDENT.FNAME,STUDENT.LNAME,STAFF.NAME
  FROM STUDENT INNER JOIN STAFF
  ON STAFF.SID=STUDENT. ADVISOR WHERE ROLL=46;

      ROLL FNAME           LNAME           NAME
    ------ --------------- --------------- -------------------
        46 Saurabh         Khandagale      Adishesh Vidyarthi
```

```
 =================================Query-03===============================
Create a sequence STAFF_SQ with appropriate starting value and maximum range
such that you can use it to populate STAFF table the tuples listed below. [Use
STAFF_SQ. NEXTVAL, STAFF_SQ.CURRVAL to access sequence values].
 106, DAT, Deo Narayan Mishra, Assistant, 13-Oct-2013
 107, CSEC, Sanjeev Bamireddy, Associate, 12-May-2018
 108, CSE, Jasmine Arora, Assistant, 11-Aug-2017
 109, CSE, Vallabh Pai, Assistant, 17-Sep-2018
 110, AIML, Harmeet Khullar, Assistant, 17-Mar-2019
======================================================================
```

```
      SQL> CREATE SEQUENCE STAFF_SQ START WITH 106
           INCREMENT BY 1
           MAXVALUE 110
           NOCYCLE CACHE 20;

      Sequence created.


      SQL> insert into staff
      values(staff_sq.nextval,'&name','&bran','&desg','&join_dt');
      Enter value for name: Deo Narayan Mishra
      Enter value for bran: DAT
      Enter value for desg: Assistant
      Enter value for join_dt: 13-Oct-2013
      old   1: insert into staff
      values(staff_sq.nextval,'&name','&bran','&desg','&join_dt')
      new   1: INSERT  INTO STAFF VALUES
      (staff_sq.nextval,'Deo Narayan Mishra','DAT','Assistant','13-Oct-2013')

      1 row created.

      SQL> INSERT  INTO STAFF VALUES (staff_sq.nextval,'Sanjeev
      Bamireddy','CSEC','Associate','12-MAY-19');

      1 row created.

      SQL> INSERT  INTO STAFF VALUES (staff_sq.nextval,'Jasmine
      Arora','CSE','Assistant','11-Aug-17');
```

```
SQL> INSERT  INTO STAFF VALUES
 (staff_sq.nextval,'Vallabh Pai','CSE','Assistant','17-SEP-18');

SQL> INSERT  INTO STAFF VALUES
 (staff_sq.nextval,'Harmeet Khullar','AIML','Assistant','17-Mar-19');


SELECT * FROM STAFF;

       SID NAME                      BRAN DESG      JOIN_DT
---------- ------------------------- ---- --------- ---------
       101 Kamalkant Marathe         CSE  Professor 12-JUN-05
       102 Adishesh Vidyarthi        AIML Associate 22-JUL-06
       103 Manishi Singh             DAT  Professor 10-NOV-07
       104 Aasawari Deodhar          CSE  Associate 13-OCT-08
       105 Geetika Goenka            CSEC Professor 15-NOV-09
       106 Deo Narayan Mishra        DAT  Assistant 13-OCT-13
       107 Sanjeev Bamireddy         CSEC Associate 12-MAY-19
       108 Jasmine Arora             CSE  Assistant 11-AUG-17
       109 Vallabh Pai               CSE  Assistant 17-SEP-18
       110 Harmeet Khullar           AIML Assistant 17-MAR-19

10 rows selected.



SQL> SELECT * FROM USER_SEQUENCES;

SEQUENCE_NAME MIN_VALUE MAX_VALUE INCREMENT C O CACHE_SIZ LAST_NUMBER
----------- ---------- ---------- --------- - - --------- -----
STAFF_SQ          1        110        1     N N    20      111


SQL> DROP SEQUENCE STAFF_SQ;

Sequence dropped.
```

==============================Query4==
While Academic Database was configured few constraints were not enforced
as mentioned in the logical schema. Identify (by listing them table-by-table)
these constraints (PK & FK) and enforce them.
===

```
      [BEFORE]

SQL> SELECT  TABLE_NAME,CONSTRAINT_NAME,CONSTRAINT_TYPE
     FROM  user_constraints
     WHERE  (CONSTRAINT_TYPE='R' OR CONSTRAINT_TYPE='P')
         AND
     (TABLE_NAME='STUDENT' OR TABLE_NAME='STAFF' OR TABLE_NAME='DEPT');

      TABLE_NAME                     CONSTRAINT_NAME                C
      ------------------------------ ------------------------------ -
      DEPT                           DEPT_PK_BRANCH                 P
      DEPT                           SYS_C007491                    R
      STAFF                          STAFF_PK_SID                   P
      STAFF                          STAFF_FK_DEPT                  R
      STUDENT                        STUDENT_PK_ROLL                P

      5 rows selected.
      [AFTER]

SQL> ALTER TABLE ADD
     CONSTRAINTS STUDENT_FK_STAFF
     FOREIGN KEY(ADVISOR) REFERENCES STAFF(SID);


SQL> SELECT  TABLE_NAME,CONSTRAINT_NAME,CONSTRAINT_TYPE
         FROM  user_constraints
         WHERE  (CONSTRAINT_TYPE='R' OR CONSTRAINT_TYPE='P')
         AND
  (TABLE_NAME='STUDENT' OR TABLE_NAME='STAFF' OR TABLE_NAME='DEPT');

TABLE_NAME                     CONSTRAINT_NAME                C
------------------------------ ------------------------------ -
DEPT                           DEPT_PK_BRANCH                 P
DEPT                           SYS_C007491                    R
STAFF                          STAFF_PK_SID                   P
STAFF                          STAFF_FK_DEPT                  R
STUDENT                        STUDENT_PK_ROLL                P
STUDENT                        STUDENT_FK_STAFF               R

6 rows selected.


============================Query05==========================================

Write SQL code that will create a temporary table (view) named STUDENT_VW on
STUDENT table projecting the attributes ENROLL, LNAME, FNAME, ROLL, ADVISOR.
List the contents of STUDENT_VW.
=============================================================================

SQL> CREATE VIEW STUDENT_VW
       AS
     (SELECT ENROLL,LNAME,FNAME,ROLL,ADVISOR,REG_DT FROM STUDENT);
View created.
SQL> SELECT * FROM STUDENT_VW;

ENROLL        LNAME           FNAME                 ROLL    ADVISOR REG_DT
------------- --------------- --------------- ---------- ---------- ---------
18CSU2001CSU2 Sayed           Afra                     1        101 20-JUL-18
18CSU2002CSU2 Wasalu          Akansha                  2        104 20-JUL-18
18CSU2003CSU2 Rajendran       Anjali                   3        108 19-JUL-18
19CSU2206CSU2 Khandagale      Saurabh                 46        102 10-AUG-19
18CSU2036CSU2 Dandekar        Paritosh                57        102 14-JUL-18
17CSU2047CSU2 Pandey          Shapath                 86        107 27-JUL-17
17CSU2091CSU2 Singh           Ayush                   66        107 27-JUL-17
```

```
==================================Query06======================================

6) Two students Naveen Namjoshi (88) and Tushar Tipnis (89) were admitted on
August 14, 2019 and were assigned to staff members 109 and 110 respectively.
Write SQL code to insert these student records into STUDENT_VW and observe the
effect on STUDENT table.
==============================================================================

EXPLAIN:-[FOR SIMPLE VIEW]
VIEW MAINTAINS LOGICAL VIEW OF DATA AS WELL AS DYNAMIC DATA AND
INSERT,UPDATE,DELETE IS AN PHYSICAL QUERY ,SO CHANGES MADE IN LOGICAL DATA
WILL REFLECT ON TABLE AS WELL AS CHANGES MADE IN TABLE WILL REFLECT ON VIEW.


SQL> INSERT INTO STUDENT_VW  (LNAME,FNAME,ROLL,ADVISOR,REG_DT)
        VALUES
     ('NAMJOSHI','NAVEEN',88,109,'14-AUG-2019');

1 row created.

SQL> INSERT INTO STUDENT_VW (LNAME,FNAME,ROLL,ADVISOR,REG_DT)
         VALUES
     ('TUSHAR','TIPNIS',89,110,'14-AUG-2019');

1 row created.

SQL> COMMIT;
SQL> SELECT LNAME,FNAME,ROLL,ADVISOR,REG_DT FROM STUDENT WHERE  ROLL=88 OR
ROLL=89;

LNAME           FNAME                 ROLL    ADVISOR REG_DT
--------------- --------------- ---------- ---------- ---------
NAMJOSHI        NAVEEN                  88        109 14-AUG-19
TUSHAR          TIPNIS                  89        110 14-AUG-19
```

```
==================================Query07=====================================
 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 record
- 91, Cinderella Goldsmith, 101, 18-Aug-2019 - into
STUDENT_VW_RO. Observe the effect.

==============================================================================
SQL> SELECT LNAME,FNAME,ROLL,ADVISOR,REG_DT FROM STUDENT_VW_RO ;

LNAME           FNAME                 ROLL    ADVISOR REG_DT
--------------- --------------- ---------- ---------- ---------
Sayed           Afra                     1        101 20-JUL-18
Wasalu          Akansha                  2        104 20-JUL-18
Rajendran       Anjali                   3        108 19-JUL-18
NAMJOSHI        NAVEEN                  88        109 14-AUG-19
TUSHAR          TIPNIS                  89        110 14-AUG-19
[NO CHANGES]


SQL>INSERT INTO STUDENT_VW_RO
(LNAME,FNAME,ROLL,ADVISOR,REG_DT)VALUES('CINDERELLA','GOLDSMITH',91,101,'18-
AUG-2019')

ERROR at line 1:
ORA-42399: cannot perform a DML operation on a read-only view

EXPLAIN:-WHEN WE APPLY DML ON SIMPLE VIEW THE CHANGES IS REFLECTED , BUT
THERE ARE SOME CASE WHERE DATABASE USER HAVE TO RESTRICT THE UPDATATION SO FOR
THAT ORACLE PROVIDE ADDITIONAL CONSTRAINTS ON VIEW AS READ ONLY ,WHICH DOES
NOT ALLOW UNAUTHORIZED USER TO MANIPULATE DATA.



======================================Query08=================================

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, 103, 105, 108 and 109. Name
the constraint as STUDENT_ADV_CK. List the contents of STUDENT_VW_CK. Now,
insert a record - 92, Sebastian Ford, 104, 18-Aug-2019 - into STUDENT_VW_CK.
Observe the  effect.
==============================================================================

SQL> CREATE OR REPLACE  VIEW STUDENT_VW_CK AS  SELECT
ENROLL,LNAME,FNAME,ROLL,ADVISOR,REG_DT FROM STUDENT WHERE ADVISOR=101 OR
ADVISOR=103 OR ADVISOR=105 OR ADVISOR=108 OR ADVISOR=109     WITH    CHECK
OPTION CONSTRAINT STUDENT_ADV_CK ;

View created.

SQL> COMMIT;
```

```
INSERT INTO STUDENT_VW_CK
(ROLL,LNAME,FNAME,ADVISOR,REG_DT)VALUES
(92,'SEBASTIAN','FORD',104,'18-AUG-2019')
            *
ERROR at line 1:
ORA-01402: view WITH CHECK OPTION where-clause violation
```

EXPLAIN:-ORACLE PROVIDE A CONSTRAINT ON VIEW THAT IS **CHECK CONSTRAINT** IN WHICH USER WILL ONLY ADD SPECIFIED DATA ONLY,IF WE TRY TO ADD INVALID DATA THEN ORCALE GIVE AS ERROR(where-clause violation).

```
======================================Query09================================
List all the views for the current schema tables (use USER_VIEWS table]. List
the constraints (include constraint type) on the views in Academic Schema.

=============================================================================

SQL> SELECT VIEW_NAME FROM USER_VIEWS;

VIEW_NAME
--------------------
STUDENT_VW
STUDENT_VW1
STUDENT_VW_CK
STUDENT_VW_RO
VW_PRODUCT

SQL> SELECT VIEW_NAME,VIEW_TYPE,READ_ONLY,EDITIONING_VIEW FROM USER_VIEWS;

VIEW_NAME            VIEW_TYPE       R E
-------------------- --------------- - -
STUDENT_VW                           N N
STUDENT_VW_RO                        Y N
STUDENT_VW_CK                        N N
VW_PRODUCT                           N N
STUDENT_VW1                          N N
```

```
======================================Query10================================
Write a SQL code to create a private synonym FACULTY_SN for STAFF. Use
this synonym to show contents of STAFF. A faculty named Dhawal Giri has been
appointed as Assistant in AIML. Insert this record using FACULTY_SN. Observe
contents of STAFF table.
=============================================================================

SQL> CREATE SYNONYM FACULTY_SN FOR STAFF;
SQL> SELECT * FROM  FACULTY_SN;

       SID NAME                      BRAN DESG      JOIN_DT
---------- ------------------------- ---- --------- ---------
       101 Kamalkant Marathe         CSE  Professor 12-JUN-05
       102 Adishesh Vidyarthi        AIML Associate 22-JUL-06
       103 Manishi Singh             DAT  Professor 10-NOV-07
       104 Aasawari Deodhar          CSE  Associate 13-OCT-08
       105 Geetika Goenka            CSEC Professor 15-NOV-09
       106 Deo Narayan Mishra        DAT  Assistant 13-OCT-13
       107 Sanjeev Bamireddy         CSEC Associate 12-MAY-19
       108 Jasmine Arora             CSE  Assistant 11-AUG-17
       109 Vallabh Pai               CSE  Assistant 17-SEP-18
       110 Harmeet Khullar           AIML Assistant 17-MAR-19

10 rows selected.



SQL> INSERT INTO FACULTY_SN
      VALUES
     (111,'Dhawal Giri','AIML','Assistant','25-AUG-20');

1 row created.


SQL> SELECT * FROM STAFF;

       SID NAME                      BRAN DESG      JOIN_DT
---------- ------------------------- ---- --------- ---------
       101 Kamalkant Marathe         CSE  Professor 12-JUN-05
       102 Adishesh Vidyarthi        AIML Associate 22-JUL-06
       103 Manishi Singh             DAT  Professor 10-NOV-07
       104 Aasawari Deodhar          CSE  Associate 13-OCT-08
       105 Geetika Goenka            CSEC Professor 15-NOV-09
       106 Deo Narayan Mishra        DAT  Assistant 13-OCT-13
       107 Sanjeev Bamireddy         CSEC Associate 12-MAY-19
       108 Jasmine Arora             CSE  Assistant 11-AUG-17
       109 Vallabh Pai               CSE  Assistant 17-SEP-18
       110 Harmeet Khullar           AIML Assistant 17-MAR-19
       111 Dhawal Giri               AIML Assistant 25-AUG-20

11 rows selected.
```

```
======================================Query11=================================
Write SQL code to create a unique B-Tree index on FNAME attribute of
STUDENT table. Observe the output and report the problem(s). If it fails,
create B-Tree index and test it to locate a certain student by first name.
Now, create a concatenated B-tree index on (LNAME, FNAME) attributes of
STUDENT table and test the index. Also list all indexes for CS5XX for the
current database schema [use USER_INDEXES table].
==============================================================================
```

```
SQL> CREATE UNIQUE INDEX UN_IDX1 ON STUDENT(FNAME);
CREATE UNIQUE INDEX UN_IDX1 ON STUDENT(FNAME)
                               *
ERROR at line 1:
ORA-01452: cannot CREATE UNIQUE INDEX; duplicate keys found
```

EXPLAIN:-WE CAN CREATE UNIQUE INDEX ONLY ON ATTRIBUTE HAVING UNIQUE VALUES,IN OUR SCHEMA FNAME CONTAINS DUPLICATE VALUES SO BECAUSE OF THAT WE CANNOT CREATE **UNIQUE** INDEX ON FNAME ATTRIBUTE.

```
SQL> CREATE  INDEX FNAME_IDX1 ON STUDENT(FNAME);

Index created.



SQL> SELECT FNAME FROM STUDENT WHERE FNAME='Saurabh';

FNAME
---------------
Saurabh
Saurabh


Execution Plan
----------------------------------------------------------
Plan hash value: 3472840140

------------------------------------------------------------------------------
-
| Id  | Operation        | Name       | Rows  | Bytes | Cost (%CPU)| Time
|
------------------------------------------------------------------------------
-
|   0 | SELECT STATEMENT |            |     1 |     8 |     1   (0)| 00:00:01
|
|*  1 |  INDEX RANGE SCAN| FNAME_IDX1 |     1 |     8 |     1   (0)| 00:00:01
|
------------------------------------------------------------------------------
-

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

   1 - access("FNAME"='Saurabh')
```

```
SQL> CREATE UNIQUE INDEX COMPOSITE_IDX1 ON STUDENT(LNAME,FNAME);

Index created.

SQL> SELECT INDEX_NAME , INDEX_TYPE,TABLE_TYPE ,TABLE_NAME,TABLE_OWNER
     FROM USER_INDEXES
     WHERE (TABLE_OWNER='CSB46')
      AND
 (TABLE_NAME='STUDENT' OR TABLE_NAME='STAFF' OR TABLE_NAME='DEPT');


INDEX_NAME           INDEX_TYPE      TABLE_TYPE  TABLE_NAME      TABLE_OWNER
-------------------- --------------- ----------- --------------- ------------
DEPT_PK_BRANCH       NORMAL          TABLE       DEPT            CSB46
STAFF_PK_SID         NORMAL          TABLE       STAFF           CSB46
COMPOSITE_IDX1       NORMAL          TABLE       STUDENT         CSB46
FNAME_IDX1           NORMAL          TABLE       STUDENT         CSB46
SYS_C007490          NORMAL          TABLE       STUDENT         CSB46
SYS_C007489          NORMAL          TABLE       STUDENT         CSB46
STUDENT_PK_ROLL      NORMAL          TABLE       STUDENT         CSB46

NOTE:-ORACLE ENGINE IMPLICITYLY CREATE A INDEX ON PRIMARY KEY,OUTPUT WILL BE
IN SORTED ORDER BECAUSE(SEARCH & STORE);

***************************************QUES12*********************************

Write 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 function-
based 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. CONNECT system/system
ALTER SYSTEM SET QUERY_REWRITE_ENABLED=TRUE;
******************************************************************************
SQL> CREATE INDEX IDX_LNAME ON STUDENT(UPPER(LNAME));

Index created.

SQL> SELECT LNAME FROM STUDENT WHERE upper(LNAME)='KHANDAGALE';

LNAME
---------------
Khandagale


SQL> CREATE INDEX IDX_FUNC ON STUDENT(UPPER(LNAME),UPPER(FNAME));

Index created.


SQL> SELECT LNAME,FNAME FROM STUDENT WHERE UPPER(LNAME)='KHANDAGALE' AND
UPPER(FNAME)='SAURABH';

LNAME           FNAME
--------------- ---------------
Khandagale      Saurabh
```

```
SQL> SELECT INDEX_NAME , INDEX_TYPE,TABLE_TYPE ,TABLE_NAME,TABLE_OWNER
 FROM USER_INDEXES
WHERE TABLE_NAME='STUDENT'
 OR TABLE_NAME='STAFF'
 OR TABLE_NAME='DEPT';

INDEX_NAME           INDEX_TYPE            TABLE_TYPE  TABLE_NAME     TABLE_OWNE
-------------------- -------------------- -------- -------------- ---------
DEPT_PK_BRANCH        NORMAL                 TABLE       DEPT           CSB46
STAFF_PK_SID          NORMAL                 TABLE       STAFF          CSB46
IDX_FUNC              FUNCTION-BASED NORMAL  TABLE       STUDENT        CSB46
IDX_LNAME             FUNCTION-BASED NORMAL  TABLE       STUDENT        CSB46
COMPOSITE_IDX1        NORMAL                 TABLE       STUDENT        CSB46
FNAME_IDX1            NORMAL                 TABLE       STUDENT        CSB46
SYS_C007490           NORMAL                 TABLE       STUDENT        CSB46
SYS_C007489           NORMAL                 TABLE       STUDENT        CSB46
STUDENT_PK_ROLL       NORMAL                 TABLE       STUDENT        CSB46


***************************************QUES13*********************************

 Write SQL script that will
a) Add a student records
91, Cinderella Goldsmith, 101, 18-Aug-2019
92, Sebastian Ford, 104, 18-Aug-2019
b) Naveen Namjoshi has a new advisor, 108.
c) Tushar Tipnis has a new advisor, 111.
Before executing 13(a) create a savepoint SP_NONE. On adding records for roll
numbers 91 and 92, create a savepoint SP_FORD. Create savepoints
SP_NAV and SP_TUS after updating in 13(b) and 13(c) respectively.
******************************************************************************

SAVEPOINT SP_NONE;

Savepoint created.


INSERT INTO STUDENT(ROLL,LNAME,FNAME,ADVISOR,REG_DT)
VALUES(91,'Goldsmith','Cinderella',101,'18-AUG-2019');

1 row created.

INSERT INTO STUDENT(ROLL,LNAME,FNAME,ADVISOR,REG_DT)
VALUES(92,'Ford','Sebstain',104,'18-AUG-2019');
SAVEPOINT SP_FORD;
Savepoint created.

SQL> UPDATE STUDENT SET ADVISOR=108 WHERE ROLL=88;
1 row updated.

SQL> SAVEPOINT SP_NAV;
Savepoint created.
```

```
SQL> UPDATE STUDENT SET ADVISOR=111 WHERE ROLL=89;
1 row updated.

SQL> SAVEPOINT SP_TUS;
Savepoint created.
```

```
***************************************QUES14*********************************

 Write SQL code to recover the database state as it was after
executing13(a). Now, regain the database state to the one before executing
Query-13.

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

```
SQL> SELECT ROLL, FNAME,LNAME,ADVISOR,REG_DT FROM STUDENT;

      ROLL FNAME           LNAME              ADVISOR REG_DT
---------- --------------- --------------- ---------- ---------
         1 Afra            Sayed                  101 20-JUL-18
         2 Akansha         Wasalu                 104 20-JUL-18
         3 Anjali          Rajendran              108 19-JUL-18
        86 Shapath         Pandey                 107 27-JUL-17
        66 Ayush           Singh                  107 27-JUL-17
        92 Sebstain        Ford                   104 18-AUG-19
        88 NAVEEN          NAMJOSHI               108 14-AUG-19
        89 TIPNIS          TUSHAR                 111 14-AUG-19
        91 Cinderella      Goldsmith              101 18-AUG-19
        92 Ford            Sebstain               104 18-AUG-19
SQL> ROLLBACK TO SP_FORD;

Rollback complete.
```

```
SQL> SELECT ROLL, FNAME,LNAME,ADVISOR,REG_DT FROM STUDENT;

      ROLL FNAME           LNAME              ADVISOR REG_DT
---------- --------------- --------------- ---------- ---------
         1 Afra            Sayed                  101 20-JUL-18
         2 Akansha         Wasalu                 104 20-JUL-18
         3 Anjali          Rajendran              108 19-JUL-18
        85 Yogesh          Siral                  105 21-JUL-18
        86 Shapath         Pandey                 107 27-JUL-17
        66 Ayush           Singh                  107 27-JUL-17
        88 NAVEEN          NAMJOSHI               109 14-AUG-19
        89 TIPNIS          TUSHAR                 110 14-AUG-19
        91 Cinderella      Goldsmith              101 18-AUG-19
        92 Sebstain        Ford                   104 18-AUG-19


ROLLBACK TO SP_NONE;

Rollback complete.
```

```
SQL> SELECT ROLL,FNAME,LNAME,ADVISOR,REG_DT FROM STUDENT;

      ROLL FNAME           LNAME              ADVISOR REG_DT
---------- --------------- --------------- ---------- ---------
         1 Afra            Sayed                  101 20-JUL-18
         2 Akansha         Wasalu                 104 20-JUL-18
         3 Anjali          Rajendran              108 19-JUL-18
        85 Yogesh          Siral                  105 21-JUL-18
        86 Shapath         Pandey                 107 27-JUL-17
        66 Ayush           Singh                  107 27-JUL-17
        88 NAVEEN          NAMJOSHI               109 14-AUG-19
        89 TIPNIS          TUSHAR                 110 14-AUG-19
```

Viva Voice
==========

```
*********************************Question-1*******************************
```
1. How does a simple view differ from a complex view

Ans:-

SIMPLE VIEW

- Contains only one single base table or is created from only one table.
- We cannot use group functions like MAX(), COUNT(), etc.
- Does not contain groups of data
- INSERT, DELETE and UPDATE are directly possible on a simple view.

COMPLEX VIEW

- Contains more than one base tables or is created from more than one tables
- We can use group functions.
- It can contain groups of data
- DML operations could not always be performed through a complex view

```
*********************************Question-2*******************************
```
2. What effect does altering parent table(s) have on a view(s) created on them?.

ANS:-Yes the data will be reflected because view constains the dynamic data.

```
*********************************Question-3*******************************
```
3. Can a sequence be reused? What will happen if it were enforced on the El-columns?

Ans:

Case 1:- if we use cycle

If we use cycle while creating the sequence then it can be re used

Case 2:- if we not use cycle

If we do not use cycle then sequence will not be reuse if it reaches to max value

Now if we were enforcing sequence on the El-columns then it will work fine in Case 2 but it will not work for case 1 because we need only unique values in EL-column.

```
*********************************Question-4*******************************
```
4. How does a synonym differ from an alias?

Synonym:-

- Synonyms are a database object type
- Synonyms can be created for tables, views, functions, procedures, packages,
- sequences, materialized views, java class object types and triggers.
- Since synonyms are a database object, they are valid inside the schema (private synonym) or inside the database (public synonym

Alias

- aliases are just a name to refer a table, view or a column inside a query.
- Not a database object.
- aliases are used only for views, tables and their columns.
- aliases valid inside the query where they are being used.

***********************************Question-4*********************************
5. Do you need to remove savepoints explicitly?
NO,IT AUTOMATICALLY GETS REMOVE BY ORACLE .

****

Inference:-
===========

1. View stores the dynamic data.
2.Index are created by default by oracle for primary key.
3. By default the index are private.
4.View is gives logical representation of data.
5.ORACLE ENGINE IMPLICITYLY CREATE A INDEX ON PRIMARY KEY.