**\*\*\*\*\*\*\*\*\*\* EXPERIMENT: 03 \*\*\*\*\*\*\*\*\*\***

**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)

**Author** : Mehul Y Khandhadiya

**Roll no** : 55

**Date** : 27-Aug-2020

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* QUERY – 02 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Write SQL code to roll number, print first name, last name, advisor name for your roll number.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**SELECT ROLL, FNAME, LNAME, ADVISOR, NAME AS ADV\_NAME**

**FROM STUDENT ST, STAFF S**

**WHERE ROLL = 55**

**AND ST.ADVISOR = S.SID;**

ROLL FNAME LNAME ADVISOR ADV\_NAME

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

55 Mehul Khandhadiya 104 Aasawari Deodhar

1 row selected.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 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**

**Verify whether the sequence has been created [use USER\_SEQUENCES view]**

**alongwith other sequences on current schema tables. After populating STAFF table, remove the sequence.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**CREATE SEQUENCE STAFF\_SQ**

**START WITH 106**

**INCREMENT BY 1**

**MINVALUE 101**

**MAXVALUE 150**

**NOCYCLE;**

Sequence created.

**SELECT STAFF\_SQ.NEXTVAL FROM DUAL;**

NEXTVAL

----------

106

**INSERT INTO STAFF (SID, NAME, BRANCH, DESG, JOIN\_DT) VALUES (STAFF\_SQ.CURRVAL,'DEO NARAYAN MISHRA','DAT','Assistant','13-OCT-2013');**

1 row created.

**INSERT INTO STAFF (SID, NAME, BRANCH, DESG, JOIN\_DT) VALUES (STAFF\_SQ.NEXTVAL,'SANJEEV BAMIREDDY','DAT','Associate','12-MAY-2018');**

1 row created.

**INSERT INTO STAFF (SID, NAME, BRANCH, DESG, JOIN\_DT) VALUES (STAFF\_SQ.NEXTVAL,'JASMINE ARORA','CSE','Assistant','11-AUG-2017');**

1 row created.

**INSERT INTO STAFF (SID, NAME, BRANCH, DESG, JOIN\_DT) VALUES (STAFF\_SQ.NEXTVAL,'VALLABH PAI','CSE', 'Assistant','17-SEP-2018');**

1 row created.

**INSERT INTO STAFF (SID, NAME, BRANCH, DESG, JOIN\_DT) VALUES (STAFF\_SQ.NEXTVAL,'HARMEET KHULLAR','AIML', 'Assistant','17-MAR-2019');**

1 row created.

**DESC USER\_SEQUENCES;**

Name Null? Type

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

SEQUENCE\_NAME NOT NULL VARCHAR2(30)

MIN\_VALUE NUMBER

MAX\_VALUE NUMBER

INCREMENT\_BY NOT NULL NUMBER

CYCLE\_FLAG VARCHAR2(1)

ORDER\_FLAG VARCHAR2(1)

CACHE\_SIZE NOT NULL NUMBER

LAST\_NUMBER NOT NULL NUMBER

**SELECT \* FROM USER\_SEQUENCES;**

SEQUENCE\_NAME MIN\_VALUE MAX\_VALUE INCREMENT\_BY C O CACHE\_SIZE LAST\_NUMBER

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

STAFF\_SQ 101 150 1 N N 20 126

**SELECT \* FROM STAFF;**

SID NAME BRAN DESG JOIN\_DT

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

106 DEO NARAYAN MISHRA DAT Assistant 13-OCT-13

107 SANJEEV BAMIREDDY DAT Associate 12-MAY-18

108 JASMINE ARORA CSE Assistant 11-AUG-17

109 VALLABH PAI CSE Assistant 17-SEP-18

110 HARMEET KHULLAR AIML Assistant 17-MAR-19

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

10 rows selected.

**DROP SEQUENCE STAFF\_SQ;**

Sequence dropped.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* QUERY – 04 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

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

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**SELECT TABLE\_NAME, CONSTRAINT\_NAME, CONSTRAINT\_TYPE**

**FROM USER\_CONSTRAINTS**

**WHERE TABLE\_NAME IN ('DEPT');**

TABLE\_NAME CONSTRAINT\_NAME C

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

DEPT SYS\_C0011820 C

DEPT SYS\_C0011821 C

DEPT SYS\_C0011822 C

DEPT SYS\_C0011823 C

DEPT SYS\_C0011824 C

DEPT DEPT\_CK\_BRANCH C

DEPT DEPT\_CK\_INTAKE C

DEPT DEPT\_CK\_YR\_EST C

DEPT DEPT\_PK\_BRANCH P

DEPT SYS\_C0011846 R

10 rows selected.

**SELECT TABLE\_NAME, CONSTRAINT\_NAME, CONSTRAINT\_TYPE**

**FROM USER\_CONSTRAINTS**

**WHERE TABLE\_NAME IN ('STAFF');**

TABLE\_NAME CONSTRAINT\_NAME C

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

STAFF SYS\_C0011829 C

STAFF SYS\_C0011830 C

STAFF SYS\_C0011831 C

STAFF SYS\_C0011832 C

STAFF SYS\_C0011833 C

STAFF STAFF\_CK\_SID C

STAFF STAFF\_CK\_DESG C

STAFF STAFF\_PK\_SID P

STAFF STAFF\_FK\_DEPT R

9 rows selected.

**SELECT TABLE\_NAME, CONSTRAINT\_NAME, CONSTRAINT\_TYPE**

**FROM USER\_CONSTRAINTS**

**WHERE TABLE\_NAME IN ('STUDENT');**

TABLE\_NAME CONSTRAINT\_NAME C

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

STUDENT SYS\_C0011838 C

STUDENT SYS\_C0011839 C

STUDENT SYS\_C0011840 C

STUDENT SYS\_C0011841 C

STUDENT STUDENT\_CK\_ROLL C

STUDENT STUDENT\_PK\_ROLL P

STUDENT SYS\_C0011844 U

STUDENT SYS\_C0011845 U

STUDENT SYS\_C0011846 C

9 rows selected.

**ALTER TABLE STUDENT**

**ADD CONSTRAINT STUDENT\_FK\_STAFF FOREIGN KEY (ADVISOR) REFERENCES**

**STAFF (SID);**

Table altered.

**SELECT TABLE\_NAME, CONSTRAINT\_NAME, CONSTRAINT\_TYPE**

**FROM USER\_CONSTRAINTS**

**WHERE TABLE\_NAME IN ('STUDENT');**

TABLE\_NAME CONSTRAINT\_NAME C

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

STUDENT SYS\_C0011838 C

STUDENT SYS\_C0011839 C

STUDENT SYS\_C0011840 C

STUDENT SYS\_C0011841 C

STUDENT STUDENT\_CK\_ROLL C

STUDENT STUDENT\_PK\_ROLL P

STUDENT SYS\_C0011844 U

STUDENT SYS\_C0011845 U

STUDENT STUDENT\_FK\_STAFF R

9 rows selected.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* QUERY – 05 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

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

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**CREATE VIEW STUDENT\_VW**

**AS**

**SELECT ENROLL, LNAME, FNAME, ROLL, ADVISOR, REG\_DT**

**FROM STUDENT;**

View created.

**SELECT VIEW\_NAME, VIEW\_TYPE, READ\_ONLY**

**FROM USER\_VIEWS**

**WHERE VIEW\_NAME = 'STUDENT\_VW';**

VIEW\_NAME VIEW\_TYPE R

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

STUDENT\_VW N

**SELECT CONSTRAINT\_NAME, TABLE\_NAME, CONSTRAINT\_TYPE**

**FROM USER\_CONSTRAINTS**

**WHERE TABLE\_NAME = 'STUDENT\_VW';**

no rows selected

**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

18CSU2033CSU2 Thorane Kunal 54 108 08-AUG-18

18CSU2034CSU2 Khandhadiya Mehul 55 104 19-JUL-18

18CSU2035CSU2 Tiwari Nikhil 56 101 04-JUL-18

16CSU2094CSU2 Rangari Mayank 87 108 25-JUL-16

18CSU2036CSU2 Dandekar Paritosh 57 102 14-JUL-18

18CSU2056CSU2 Jain Yash 84 103 03-JUL-18

18CSU2057CSU2 Siral Yogesh 85 105 21-JUL-18

17CSU2047CSU2 Pandey Shapath 86 107 27-JUL-17

73 rows selected.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* QUERY – 06 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**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.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**INSERT INTO STUDENT\_VW VALUES (NULL,'NAMJOSHI','NAVEEN', 88, 109,'14-AUG-19');**

1 row created.

**INSERT INTO STUDENT\_VW VALUES (NULL,'TIPNIS','TUSHAR', 89, 110,'14-AUG-19');**

1 row created.

**SELECT ENROLL, LNAME, FNAME, ROLL, ADVISOR, REG\_DT**

**FROM STUDENT;**

ENROLL LNAME FNAME ROLL ADVISOR REG\_DT

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

TIPNIS TUSHAR 89 110 14-AUG-19

NAMJOSHI NAVEEN 88 109 14-AUG-19

18CSU2001CSU2 Sayed Afra 1 101 20-JUL-18

18CSU2002CSU2 Wasalu Akansha 2 104 20-JUL-18

18CSU2003CSU2 Rajendran Anjali 3 108 19-JUL-18

18CSU2033CSU2 Thorane Kunal 54 108 08-AUG-18

18CSU2034CSU2 Khandhadiya Mehul 55 104 19-JUL-18

18CSU2035CSU2 Tiwari Nikhil 56 101 04-JUL-18

16CSU2094CSU2 Rangari Mayank 87 108 25-JUL-16

18CSU2036CSU2 Dandekar Paritosh 57 102 14-JUL-18

17CSU2047CSU2 Pandey Shapath 86 107 27-JUL-17

75 rows selected.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* QUERY – 07 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**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.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**CREATE OR REPLACE VIEW STUDENT\_VW\_RO**

**AS SELECT ENROLL, LNAME, FNAME, ROLL, ADVISOR, REG\_DT**

**FROM STUDENT**

**WITH READ ONLY;**

View created.

**SELECT VIEW\_NAME, VIEW\_TYPE, READ\_ONLY**

**FROM USER\_VIEWS**

**WHERE VIEW\_NAME = 'STUDENT\_VW\_RO';**

VIEW\_NAME VIEW\_TYPE R

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

STUDENT\_VW\_RO Y

**SELECT CONSTRAINT\_NAME, TABLE\_NAME, CONSTRAINT\_TYPE**

**FROM USER\_CONSTRAINTS**

**WHERE TABLE\_NAME = 'STUDENT\_VW\_RO';**

CONSTRAINT\_NAME TABLE\_NAME C

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

SYS\_C0011938 STUDENT\_VW\_RO O

**SELECT \* FROM STUDENT\_VW\_RO;**

ENROLL LNAME FNAME ROLL ADVISOR REG\_DT

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

TIPNIS TUSHAR 89 110 14-AUG-19

NAMJOSHI NAVEEN 88 109 14-AUG-19

18CSU2001CSU2 Sayed Afra 1 101 20-JUL-18

18CSU2002CSU2 Wasalu Akansha 2 104 20-JUL-18

18CSU2003CSU2 Rajendran Anjali 3 108 19-JUL-18

18CSU2033CSU2 Thorane Kunal 54 108 08-AUG-18

18CSU2034CSU2 Khandhadiya Mehul 55 104 19-JUL-18

18CSU2035CSU2 Tiwari Nikhil 56 101 04-JUL-18

16CSU2094CSU2 Rangari Mayank 87 108 25-JUL-16

18CSU2036CSU2 Dandekar Paritosh 57 102 14-JUL-18

17CSU2047CSU2 Pandey Shapath 86 107 27-JUL-17

75 rows selected.

**INSERT INTO STUDENT\_VW\_RO VALUES(NULL,'GOLDSMITH','CINDRELLA', 91, 101,'18-AUG-19');**

INSERT INTO STUDENT\_VW\_RO VALUES(NULL,'GOLDSMITH','CINDRELLA', 91, 101,'18-AUG-19')

\*

ERROR at line 1:

ORA-42399: cannot perform a DML operation on a read-only view

**SELECT ENROLL, LNAME, FNAME, ROLL, ADVISOR, REG\_DT**

**FROM STUDENT;**

ENROLL LNAME FNAME ROLL ADVISOR REG\_DT

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

TIPNIS TUSHAR 89 110 14-AUG-19

NAMJOSHI NAVEEN 88 109 14-AUG-19

18CSU2001CSU2 Sayed Afra 1 101 20-JUL-18

18CSU2002CSU2 Wasalu Akansha 2 104 20-JUL-18

18CSU2003CSU2 Rajendran Anjali 3 108 19-JUL-18

18CSU2033CSU2 Thorane Kunal 54 108 08-AUG-18

18CSU2034CSU2 Khandhadiya Mehul 55 104 19-JUL-18

18CSU2035CSU2 Tiwari Nikhil 56 101 04-JUL-18

16CSU2094CSU2 Rangari Mayank 87 108 25-JUL-16

18CSU2036CSU2 Dandekar Paritosh 57 102 14-JUL-18

17CSU2047CSU2 Pandey Shapath 86 107 27-JUL-17

75 rows selected

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* QUERY – 08 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**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-2919 into**

**STUDENT\_VW\_CK. Observe the effect.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**CREATE OR REPLACE VIEW STUDENT\_VW\_CK**

**AS SELECT ENROLL, LNAME, FNAME, ROLL, ADVISOR, REG\_DT**

**FROM STUDENT**

**WHERE ADVISOR IN (101,103,105,108,109)**

**WITH CHECK OPTION**

**CONSTRAINT STUDENT\_ADV\_CK;**

View created.

**SELECT VIEW\_NAME, VIEW\_TYPE, READ\_ONLY**

**FROM USER\_VIEWS**

**WHERE VIEW\_NAME = 'STUDENT\_VW\_CK';**

VIEW\_NAME VIEW\_TYPE R

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

STUDENT\_VW\_CK N

**SELECT CONSTRAINT\_NAME, TABLE\_NAME, CONSTRAINT\_TYPE**

**FROM USER\_CONSTRAINTS**

**WHERE TABLE\_NAME = 'STUDENT\_VW\_CK';**

CONSTRAINT\_NAME TABLE\_NAME C

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

STUDENT\_ADV\_CK STUDENT\_VW\_CK V

**SELECT \* FROM STUDENT\_VW\_CK;**

ENROLL LNAME FNAME ROLL ADVISOR REG\_DT

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

NAMJOSHI NAVEEN 88 109 14-AUG-19

18CSU2001CSU2 Sayed Afra 1 101 20-JUL-18

18CSU2003CSU2 Rajendran Anjali 3 108 19-JUL-18

18CSU2033CSU2 Thorane Kunal 54 108 08-AUG-18

18CSU2035CSU2 Tiwari Nikhil 56 101 04-JUL-18

16CSU2094CSU2 Rangari Mayank 87 108 25-JUL-16

18CSU2056CSU2 Jain Yash 84 103 03-JUL-18

18CSU2057CSU2 Siral Yogesh 85 105 21-JUL-18

37 rows selected.

**INSERT INTO STUDENT\_VW\_CK VALUES (NULL,'FORD','SEBASTIAN', 92, 104,'18-AUG-19');**

INSERT INTO STUDENT\_VW\_CK VALUES (NULL,'FORD','SEBASTIAN', 92, 104,'18-AUG-19')

\*

ERROR at line 1:

ORA-01402: view WITH CHECK OPTION where-clause violation

**REASON:** Here, since we have restricted advisors to 101, 103, 105, 108, 109 and the tuple which we are trying to insert contains 104 as ADVISOR, hence there is a violation.

**SELECT ENROLL, LNAME, FNAME, ROLL, ADVISOR, REG\_DT**

**FROM STUDENT;**

ENROLL LNAME FNAME ROLL ADVISOR REG\_DT

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

TIPNIS TUSHAR 89 110 14-AUG-19

NAMJOSHI NAVEEN 88 109 14-AUG-19

18CSU2001CSU2 Sayed Afra 1 101 20-JUL-18

18CSU2002CSU2 Wasalu Akansha 2 104 20-JUL-18

18CSU2003CSU2 Rajendran Anjali 3 108 19-JUL-18

18CSU2033CSU2 Thorane Kunal 54 108 08-AUG-18

18CSU2034CSU2 Khandhadiya Mehul 55 104 19-JUL-18

18CSU2035CSU2 Tiwari Nikhil 56 101 04-JUL-18

16CSU2094CSU2 Rangari Mayank 87 108 25-JUL-16

18CSU2036CSU2 Dandekar Paritosh 57 102 14-JUL-18

17CSU2047CSU2 Pandey Shapath 86 107 27-JUL-17

75 rows selected.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* QUERY – 09 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**List all the views tor the current schema tables [use USER\_VIEWS table]. List the constraints (include constraint type) on the views in Academic Schema.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**SELECT VIEW\_NAME, VIEW\_TYPE, READ\_ONLY**

**FROM USER\_VIEWS;**

VIEW\_NAME VIEW\_TYPE R

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

STUDENT\_VW N

STUDENT\_VW\_CK N

STUDENT\_VW\_RO Y

3 rows selected.

**SELECT CONSTRAINT\_NAME, TABLE\_NAME, CONSTRAINT\_TYPE**

**FROM USER\_CONSTRAINTS**

**WHERE TABLE\_NAME LIKE 'STUDENT\_VW%';**

CONSTRAINT\_NAME TABLE\_NAME C

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

STUDENT\_ADV\_CK STUDENT\_VW\_CK V

SYS\_C0011938 STUDENT\_VW\_RO O

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* QUERY – 10 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**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.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**CREATE SYNONYM FACULTY\_SN FOR STAFF;**

Synonym created.

**SELECT \* FROM FACULTY\_SN;**

SID NAME BRAN DESG JOIN\_DT

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

106 DEO NARAYAN MISHRA DAT Assistant 13-OCT-13

107 SANJEEV BAMIREDDY DAT Associate 12-MAY-18

108 JASMINE ARORA CSE Assistant 11-AUG-17

109 VALLABH PAI CSE Assistant 17-SEP-18

110 HARMEET KHULLAR AIML Assistant 17-MAR-19

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

10 rows selected.

**INSERT INTO FACULTY\_SN VALUES (111,'DHAWAL GIRI','AIML','Assistant', SYSDATE);**

**SELECT \* FROM STAFF;**

SID NAME BRAN DESG JOIN\_DT

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

106 DEO NARAYAN MISHRA DAT Assistant 13-OCT-13

107 SANJEEV BAMIREDDY DAT Associate 12-MAY-18

108 JASMINE ARORA CSE Assistant 11-AUG-17

109 VALLABH PAI CSE Assistant 17-SEP-18

110 HARMEET KHULLAR AIML Assistant 17-MAR-19

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

111 DHAWAL GIRI AIML Assistant 26-AUG-20

11 rows selected.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* QUERY – 11 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**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].**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**CREATE UNIQUE INDEX STUDENT\_NDX\_FNAME\_UQ**

**ON STUDENT (FNAME);**

ON STUDENT (FNAME)

\*

ERROR at line 2:

ORA-01452: cannot CREATE UNIQUE INDEX; duplicate keys found

**REASON**: A CREATE UNIQUE INDEX statement specified one or more columns that currently contain duplicate values. All values in the indexed columns must be unique by row to create a UNIQUE INDEX. If the entries need not be unique, remove the keyword UNIQUE from the CREATE INDEX statement, then re-execute the statement. If the entries must be unique, as in a primary key, then remove duplicate values before creating the UNIQUE index.

**CREATE INDEX STUDENT\_NDX\_FNAME**

**ON STUDENT (FNAME);**

Index created.

**SELECT ENROLL, LNAME, FNAME, ROLL, ADVISOR, REG\_DT**

**FROM STUDENT**

**WHERE FNAME = 'Mehul';**

ENROLL LNAME FNAME ROLL ADVISOR REG\_DT

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

18CSU2034CSU2 Khandhadiya Mehul 55 104 19-JUL-18

**CREATE INDEX STUDENT\_NDX\_LNAME\_FNAME**

**ON STUDENT (LNAME, FNAME);**

Index created.

**SELECT ENROLL, LNAME, FNAME, ROLL, ADVISOR, REG\_DT**

**FROM STUDENT**

**WHERE LNAME = 'Khandhadiya' AND FNAME = 'Mehul';**

ENROLL LNAME FNAME ROLL ADVISOR REG\_DT

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

18CSU2034CSU2 Khandhadiya Mehul 55 104 19-JUL-18

**SELECT INDEX\_NAME, TABLE\_NAME, UNIQUENESS, STATUS**

**FROM USER\_INDEXES**

**WHERE TABLE\_NAME IN ('STUDENT','STAFF','DEPT')**

**ORDER BY INDEX\_NAME;**

INDEX\_NAME TABLE\_NAME UNIQUENES STATUS

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

DEPT\_PK\_BRANCH DEPT UNIQUE VALID

STAFF\_PK\_SID STAFF UNIQUE VALID

STUDENT\_NDX\_FNAME STUDENT NONUNIQUE VALID

STUDENT\_NDX\_LNAME\_FNAME STUDENT NONUNIQUE VALID

STUDENT\_PK\_ROLL STUDENT UNIQUE VALID

SYS\_C0011844 STUDENT UNIQUE VALID

SYS\_C0011845 STUDENT UNIQUE VALID

7 rows selected.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* QUERY – 12 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**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 at 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;**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**CREATE INDEX STUDENT\_NDX\_LNAME**

**ON STUDENT (UPPER (LNAME));**

Index created.

**SELECT ENROLL, LNAME, FNAME, ROLL, ADVISOR, REG\_DT**

**FROM STUDENT**

**WHERE UPPER (LNAME) = 'KHANDHADIYA';**

ENROLL LNAME FNAME ROLL ADVISOR REG\_DT

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

18CSU2034CSU2 Khandhadiya Mehul 55 104 19-JUL-18

**CREATE INDEX STUDENT\_NDXX\_LNAME\_FNAME**

**ON STUDENT(UPPER(LNAME),UPPER(FNAME));**

Index created.

**SELECT ENROLL, LNAME, FNAME, ROLL, ADVISOR, REG\_DT**

**FROM STUDENT**

**WHERE UPPER (LNAME) = 'KHANDHADIYA' AND UPPER (FNAME) = 'MEHUL' ;**

ENROLL LNAME FNAME ROLL ADVISOR REG\_DT

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

18CSU2034CSU2 Khandhadiya Mehul 55 104 19-JUL-18

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* QUERY – 13 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**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.**

**e) 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.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

DEPT#

----------

4

STAFF#

----------

11

STUDENT#

----------

75

ENROLL LNAME FNAME ROLL ADVISOR REG\_DT

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

GOLDSMITH CINDRELLA 91 101 18-AUG-19

ENROLL LNAME FNAME ROLL ADVISOR REG\_DT

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

FORD SEBASTIAN 92 104 18-AUG-19

LNAME FNAME ROLL ADVISOR

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

NAMJOSHI NAVEEN 88 108

LNAME FNAME ROLL ADVISOR

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

TIPNIS TUSHAR 88 111

DATABASE CONTAINS..

DEPT#

----------

4

STAFF#

----------

11

STUDENT#

----------

77

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* QUERY – 14 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Write SQL code to recover the database state as it was after executing 13(a).**

**Now, regain the database state to the one before executing Query-13.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**ROLLBACK TO SP\_FORD;**

Rollback complete.

**SELECT ENROLL, LNAME, FNAME, ROLL, ADVISOR, REG\_DT**

**FROM STUDENT**

**WHERE ROLL = 91;**

ENROLL LNAME FNAME ROLL ADVISOR REG\_DT

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

GOLDSMITH CINDRELLA 91 101 18-AUG-19

**SELECT ENROLL, LNAME, FNAME, ROLL, ADVISOR, REG\_DT**

**FROM STUDENT**

**WHERE ROLL = 92;**

ENROLL LNAME FNAME ROLL ADVISOR REG\_DT

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

FORD SEBASTIAN 92 104 18-AUG-19

**SELECT LNAME, FNAME, ROLL, ADVISOR**

**FROM STUDENT**

**WHERE ROLL = 88;**

LNAME FNAME ROLL ADVISOR

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

NAMJOSHI NAVEEN 88 109

**SELECT LNAME, FNAME, ROLL, ADVISOR**

**FROM STUDENT**

**WHERE ROLL = 88;**

LNAME FNAME ROLL ADVISOR

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

TIPNIS TUSHAR 88 110

**ROLLBACK TO SP\_NONE;**

Rollback complete.

**SELECT COUNT(\*) "DEPT#" FROM DEPT;**

DEPT#

----------

4

**SELECT COUNT(\*) "STAFF#" FROM STAFF;**

STAFF#

----------

11

**SELECT COUNT(\*) "STUDENT#" FROM STUDENT;**

STUDENT#

----------

75

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* VIVA-VOCE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q1 – How does a simple view differ from a complex view?**

Views in SQL is nothing but the logical table created from one or more tables. We can use the views to fetch the columns from one or more different tables at a time. In real life specifically views are used in Reporting purpose. To create a report we need data from different tables and need to show it on a single dashboard so we are using the views to fetch the data from different tables. View can contain all rows from the table or selected rows from the table.

There are 2 types of views in SQL:

1. Simple View-Simple view is view created on single table
2. Complex View-Complex view is view created on more than 1 tables

**Simple View**:

When user wants data or some columns from same table then simple view is used. For an example if there is employee table which has Employee\_num, Employee\_name, salary columnns and we just need to see Employee\_num, Employee\_name then user can create a simple view.

**Complex View**:

Complex view is created on using more than one tables. When user wants to retrieve data from more than 1 table then we have to use complex views. To create complex view there should be relation between 2 tables else cartesian product will come by joining 2 tables. There should be some joining conditions, some filters needs to be considered while creating complex views.

**Q2 – What effect does altering parent table(s) have on views created on them?**

A view is nothing more than a SQL statement that is stored in the database with an associated name. A view is actually a composition of a table in the form of a predefined SQL query.

A view can contain all rows of a table or select rows from a table. A view can be created from one or many tables which depends on the written SQL query to create a view.

Views, which are a type of virtual tables allow users to do the following –

1. Structure data in a way that users or classes of users find natural or intuitive.
2. Restrict access to the data in such a way that a user can see and (sometimes) modify exactly what they need and no more.
3. Summarize data from various tables which can be used to generate reports.

**Q3 – Can a sequence be reused? What will happen if it were enforced on the**

**EI-columns?**

When the instance is under memory pressure, and multiple concurrent connections request sequence values from the same sequence object, duplicate sequence values may be generated. In addition, a unique or primary key (PK) violation error occurs when the duplicate sequence value is inserted into a table.

A sequence is a user defined schema bound object that generates a sequence of numeric values.

Sequences are frequently used in many databases because many applications require each row in a table to contain a unique value and sequences provides an easy way to generate them.

The sequence of numeric values is generated in an ascending or descending order at defined intervals and can be configured to restart when exceeds max value.

**Q4 – How does a synonym differ from an alias?**

In English, synonym and alias have nearly the same meanings. But in databases those are two different things. Especially in ORACLE databases, both of their usage is different. Synonyms are used to refer objects of a schema or a database from another schema. So synonym is a database object type. But aliases are coming in a different way. That means; they are not database objects. Aliases are used to refer tables, views and columns inside queries.

These are a type of database objects. They refer to other objects in the database. The most common usage of synonym is, to refer an object of a separate schema by using another name. But synonyms can be created to refer the objects of another database, as well (in distributed databases, using database links). Tables, views, functions, procedures, packages, sequences, materialized views, java class objects and triggers can be used as references for the synonyms. There are two types of synonyms.

Private synonyms (can be used only by the user who created them.)

Public synonyms (can be used by all users who have the appropriate privileges)

**Q5 – Do you need to remove savepoints explicitly?**

A savepoint is a named entity that represents the state of data and schemas at a particular point within a unit of work. You can create savepoints within a transaction. If the transaction rolls back, changes are undone to the specified savepoint, rather than to the beginning of the transaction.If the savepoint represents a unique milestone that should not be moved with another SAVEPOINT statement, specify the UNIQUE keyword. This prevents the accidental reuse of the name that can occur by invoking a stored procedure that uses the identical savepoint name in a SAVEPOINT statement.

Savepoints are released using the RELEASE SAVEPOINT statement. If a RELEASE SAVEPOINT statement is not used to explicitly release a savepoint, it is released at the end of the current savepoint level or at the end of the transaction.

Savepoints are released when the transaction is committed or rolled back. Once the savepoint name is released, a rollback to the savepoint name is no longer possible. The COMMIT or ROLLBACK statement releases all savepoint names established within a transactions. Since all savepoint names are released within the transaction, all savepoint names can be reused following a commit or rollback.

**INFERENCE**: In this practical, we learnt about creation, updating and usage of

Oracle objects - views, synonyms, sequences, indexes and savepoints.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* END \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***