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

Aim : To facilitate creation of views, synonyms, sequences, indexes and savepoints on underlying database and to demonstrate their usage through queries on the database. [group by]

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 : Bhavesh Kewalramani

Roll No : 025 [5A]

Date : 26-July-2021

****

QUERY-02: Write SQL code to roll number, print first name, last name, advisor name for **your** roll number.

****

```
SELECT S.ROLL, S.FNAME, S.LNAME, ST.NAME AS ADVISOR
FROM STUDENT S JOIN STAFF ST
ON S.ADVISOR=ST.SID
WHERE S.ROLL = 25;
```

```
no rows 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] along with other sequences on current schema tables. After populating STAFF table, remove the sequence.
****

```
             CREATE SEQUENCE STAFF_SQ
               2  START WITH 106
               3  INCREMENT BY 1
               4  MAXVALUE 110
               5  MINVALUE 106
               6  NOCYCLE;
```

```
Sequence created.
```

```
           SELECT STAFF_SQ.NEXTVAL FROM DUAL;
```

```
   NEXTVAL
----------
       106
```

```
           INSERT INTO STAFF
             2  VALUES (STAFF_SQ.CURRVAL,'Deo Narayan
           Mishra','DAT','Assistant','13-OCT-2013');
```

```
1 row created.
```

```
           INSERT INTO STAFF
             2  VALUES (STAFF_SQ.NEXTVAL,'Sanjeev
           Bamireddy','CSEC','Associate','12-MAY-2018');
```

```
1 row created.
```

```
             INSERT INTO STAFF
               2  VALUES (STAFF_SQ.NEXTVAL,'Jasmine Arora','CSE','Assistant','11-
             AUG-2017');
```

```
1 row created.
```

```
             INSERT INTO STAFF
```

```
  2  VALUES (STAFF_SQ.NEXTVAL,'Vallabh Pai','CSE','Assistant','17-SEP-2018');

1 row created.

INSERT INTO STAFF
  2  VALUES (STAFF_SQ.NEXTVAL,'Harmeet Kullar','AIML','Assistant','17-MAR-2019');

1 row created.

SELECT *
  2  FROM USER_SEQUENCES;

SEQUENCE_NAME    MIN_VALUE  MAX_VALUE INCREMENT_BY C O CACHE_SIZE LAST_NUMBER
--------------- ---------- ---------- ------------ - - ---------- -----------
STAFF_SQ               106        110            1 N N         20         111


DROP SEQUENCE STAFF_SQ;

Sequence dropped.

SELECT *
  2  FROM USER_SEQUENCES;

no rows selected
```

****

QUERY-04: 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.

****

```
SELECT CONSTRAINT_NAME, TABLE_NAME, CONSTRAINT_TYPE
  2  FROM USER_CONSTRAINTS
  3  WHERE TABLE_NAME LIKE 'STUDENT' AND CONSTRAINT_TYPE IN ('P','R');

CONSTRAINT_NAME                TABLE_NAME                     C
------------------------------ ------------------------------ -
STUDENT_PK_ROLL                STUDENT                        P

1 row selected.
```

```
             ALTER TABLE STUDENT
               2  ADD
               3  CONSTRAINT STUDENT_FK_STAFF_SID FOREIGN KEY (ADVISOR) REFERENCES
             STAFF(SID);

Table altered.

             SELECT CONSTRAINT_NAME, TABLE_NAME, CONSTRAINT_TYPE
               2  FROM USER_CONSTRAINTS
               3  WHERE TABLE_NAME LIKE 'STUDENT' AND CONSTRAINT_TYPE IN ('P','R');

CONSTRAINT_NAME                TABLE_NAME                     C
------------------------------ ------------------------------ -
STUDENT_PK_ROLL                STUDENT                        P
STUDENT_FK_STAFF_SID           STUDENT                        R

2 rows selected.


             SELECT CONSTRAINT_NAME, TABLE_NAME, CONSTRAINT_TYPE
               2  FROM USER_CONSTRAINTS
               3  WHERE TABLE_NAME LIKE 'STAFF' AND CONSTRAINT_TYPE IN ('P','R');

CONSTRAINT_NAME                TABLE_NAME                     C
------------------------------ ------------------------------ -
STAFF_PK_SID                   STAFF                          P
STAFF_FK_DEPT                  STAFF                          R

2 rows selected.

             SELECT CONSTRAINT_NAME, TABLE_NAME, CONSTRAINT_TYPE
               2  FROM USER_CONSTRAINTS
               3  WHERE TABLE_NAME LIKE 'DEPT' AND CONSTRAINT_TYPE IN ('P','R');

CONSTRAINT_NAME                TABLE_NAME                     C
------------------------------ ------------------------------ -
DEPT_PK_BRANCH                 DEPT                           P
SYS_C007487                    DEPT                           R

2 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, REG_DT, ADVISOR. List the contents of STUDENT_VW.

****

```
CREATE OR REPLACE VIEW STUDENT_VW
  2  AS SELECT ENROLL,LNAME,FNAME,ROLL,REG_DT,ADVISOR
  3  FROM STUDENT;
```

```
View created.
```

```
SELECT *
  2  FROM STUDENT_VW;
```

```
ENROLL        LNAME           FNAME                 ROLL REG_DT       ADVISOR
------------- --------------- --------------- ---------- --------- ----------
18CSU2001CSU2 Sayed           Afra                     1 20-JUL-18        101
18CSU2002CSU2 Wasalu          Akansha                  2 20-JUL-18        104
18CSU2003CSU2 Rajendran       Anjali                   3 19-JUL-18        108
18CSU2009CSU2 Menghal         Aradhita                 4 07-JUL-18        109
18CSU2023CSU2 Deshmukh        Ritul                   11 18-JUL-18        101
18CSU2024CSU2 Nema            Sakshi                  12 07-JUL-18        104
18CSU2025CSU2 Agnihotri       Shreya                  13 07-JUL-18        108
18CSU2026CSU2 Shukla          Shrishti                14 19-JUL-18        109
18CSU2010CSU2 Muley           Aayush                  31 19-JUL-18        101
18CSU2011CSU2 Chohan          Abhishek                32 07-JUL-18        104
18CSU2012CSU2 Kotgirwar       Adesh                   33 20-JUL-18        108
18CSU2013CSU2 Nawghare        Adhney                  34 08-AUG-18        109
18CSU2019CSU2 Gupta           Ayush                   41 12-JUL-18        101
18CSU2020CSU2 Kapre           Chaitanya               42 25-JUL-18        104
18CSU2021CSU2 Paliwal         Dev                     43 21-JUL-18        108
18CSU2029CSU2 Shukla          Gaurav                  44 17-JUL-18        109
18CSU2032CSU2 Sharma          Keshubh                 53 20-JUL-18        109
18CSU2033CSU2 Thorane         Kunal                   54 08-AUG-18        108
18CSU2034CSU2 Khandhadiya     Mehul                   55 19-JUL-18        104
18CSU2035CSU2 Tiwari          Nikhil                  56 04-JUL-18        101
18CSU2042CSU2 Kale            Rishikesh               63 07-JUL-18        104
18CSU2043CSU2 Parashar        Ritik                   64 19-JUL-18        108

ENROLL        LNAME           FNAME                 ROLL REG_DT       ADVISOR
------------- --------------- --------------- ---------- --------- ----------
```

```
18CSU2044CSU2 Chandani        Rohit                   65 08-AUG-18        101
18CSU2051CSU2 Jha             Shubham                 78 12-JUL-18        109
18CSU2052CSU2 Kushwah         Yaman                   79 17-JUL-18        108
18CSU2053CSU2 Bhageriya       Yash                    80 19-JUL-18        104
16CSU2095CSU2 Soni            Renuka                  30 25-JUL-16        109
16CSU2094CSU2 Rangari         Mayank                  87 25-JUL-16        108
18CSU2004CSU2 Fadnavis        Ketki                    5 14-JUL-18        102
18CSU2005CSU2 Sharma          Lalita                   6 10-JUL-18        110
18CSU2027CSU2 Baheti          Simran                  15 20-JUL-18        102
18CSU2028CSU2 Negi            Urvi                    16 19-JUL-18        110
18CSU2014CSU2 Chandak         Akshat                  35 20-JUL-18        102
18CSU2060CSU2 Chole           Amey                    36 08-AUG-18        110
18CSU2058CSU2 Virdi           Gursewak                45 07-JUL-18        110
19CSU2206CSU2 Khandagale      Saurabh                 46 10-AUG-19        102
18CSU2036CSU2 Dandekar        Paritosh                57 14-JUL-18        102
18CSU2037CSU2 Gupta           Pavankumar              58 03-JUL-18        110
18CSU2045CSU2 Parikh          Rushil                  71 07-JUL-18        110
18CSU2046CSU2 Pandey          Sankalp                 72 07-JUL-18        102
18CSU2054CSU2 Daware          Yash                    81 20-JUL-18        102
18CSU2059CSU2 Roy             Yash                    82 07-JUL-18        110
17CSU2093CSU2 Sharnagat       Love                    68 25-JUL-17        110
18CSU2006CSU2 Gupta           Muskan                   7 19-JUL-18        103

ENROLL        LNAME           FNAME                 ROLL REG_DT       ADVISOR
------------- --------------- --------------- ---------- --------- ----------
18CSU2007CSU2 Devikar         Prateeksha               8 13-JUL-18        106
19CSU2201CSU2 Pathe           Deepali                 17 10-AUG-19        106
19CSU2202CSU2 Bhanuse         Prachi                  18 11-AUG-19        103
18CSU2015CSU2 Ray             Amit                    37 20-JUL-18        103
18CSU2016CSU2 Pandharipande   Aryan                   38 07-JUL-18        106
19CSU2204CSU2 Thakur          Ganesh                  47 22-AUG-19        106
19CSU2205CSU2 Pardhi          Manishkumar             48 23-AUG-19        103
18CSU2038CSU2 Agrawal         Rahul                   59 16-JUL-18        103
18CSU2039CSU2 Chandak         Rajat                   60 20-JUL-18        106
18CSU2047CSU2 Sushir          Saurabh                 73 07-JUL-18        103
17CSU2052CSU2 Nimbalkar       Shardul                 74 28-JUL-17        106
18CSU2055CSU2 Dhamecha        Yash                    83 21-JUL-18        106
18CSU2056CSU2 Jain            Yash                    84 03-JUL-18        103
17CSU2092CSU2 Soni            Anujesh                 67 25-JUL-17        103
18CSU2008CSU2 Taori           Priyal                   9 19-JUL-18        105
18CSU2022CSU2 Chouksey        Rashi                   10 08-AUG-18        107
19CSU2203CSU2 Tripathi        Siddhi                  19 31-AUG-19        107
```

```
18CSU2017CSU2 Uplanchiwar     Atharva                 39 07-JUL-18        105
18CSU2018CSU2 Paliwal         Atharva                 40 20-JUL-18        107
18CSU2030CSU2 Karwa           Harsh                   51 11-JUL-18        105
18CSU2031CSU2 Kapse           Jayesh                  52 08-AUG-18        107
18CSU2040CSU2 Agrawal         Ram                     61 19-JUL-18        107

ENROLL        LNAME           FNAME                 ROLL REG_DT       ADVISOR
------------- --------------- --------------- ---------- --------- ----------
18CSU2041CSU2 Khandelwal      Raunak                  62 19-JUL-18        105
18CSU2048CSU2 Tapas           Shashank                75 07-JUL-18        105
18CSU2049CSU2 Bagadia         Shivam                  76 20-JUL-18        107
18CSU2050CSU2 Nemani          Shreyas                 77 20-JUL-18        105
18CSU2057CSU2 Siral           Yogesh                  85 21-JUL-18        105
17CSU2047CSU2 Pandey          Shapath                 86 27-JUL-17        107
17CSU2091CSU2 Singh           Ayush                   66 27-JUL-17        107
```

```
DESC STUDENT_VW
```

```
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ENROLL                                             CHAR(13)
 LNAME                                     NOT NULL VARCHAR2(15)
 FNAME                                     NOT NULL VARCHAR2(15)
 ROLL                                      NOT NULL NUMBER(3)
 REG_DT                                    NOT NULL DATE
 ADVISOR                                   NOT NULL NUMBER(3)
```

****

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
  2  VALUES (NULL, 'Namjoshi','Naveen',88,'14-AUG-2019',109);

1 row created.
```

```
             INSERT INTO STUDENT_VW
               2  VALUES (NULL, 'Tipnis','Tushar',89,'14-AUG-2019',110);

1 row created.

             SELECT *
               2  FROM STUDENT
               3  WHERE ROLL = 88 OR ROLL = 89;

      ROLL LNAME      FNAME      EMAIL      ENROLL            ADVISOR
---------- ---------- ---------- ---------- -------------- ----------
     PHONE REG_DT
---------- ---------
        88 Namjoshi   Naveen                                      109
           14-AUG-19

        89 Tipnis     Tushar                                      110
           14-AUG-19


2 rows selected.
```

The records have been added to the STUDENT table but on the places of EMAIL,ENROLL, PHONE as no input was given so they were set to Null. This view was editable and updatable so the record was added successfully in it.

****

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
               2  AS SELECT ENROLL,LNAME,FNAME,ROLL,REG_DT,ADVISOR
               3  FROM STUDENT
               4  WITH READ ONLY
               5  CONSTRAINT STUDENT_RO_VW;

View created.
```

```
            DESC STUDENT_VW_RO

Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
ENROLL                                             CHAR(13)
LNAME                                     NOT NULL VARCHAR2(15)
FNAME                                     NOT NULL VARCHAR2(15)
ROLL                                      NOT NULL NUMBER(3)
REG_DT                                    NOT NULL DATE
ADVISOR                                   NOT NULL NUMBER(3)
```

```
            SELECT *
              2  FROM STUDENT_VW_RO;
```

```
ENROLL          LNAME           FNAME                 ROLL REG_DT       ADVISOR
--------------- --------------- --------------- ---------- --------- ----------
18CSU2001CSU2   Sayed           Afra                     1 20-JUL-18        101
18CSU2002CSU2   Wasalu          Akansha                  2 20-JUL-18        104
18CSU2003CSU2   Rajendran       Anjali                   3 19-JUL-18        108
18CSU2009CSU2   Menghal         Aradhita                 4 07-JUL-18        109
18CSU2023CSU2   Deshmukh        Ritul                   11 18-JUL-18        101
18CSU2024CSU2   Nema            Sakshi                  12 07-JUL-18        104
18CSU2025CSU2   Agnihotri       Shreya                  13 07-JUL-18        108
18CSU2026CSU2   Shukla          Shrishti                14 19-JUL-18        109
18CSU2010CSU2   Muley           Aayush                  31 19-JUL-18        101
18CSU2011CSU2   Chohan          Abhishek                32 07-JUL-18        104
18CSU2012CSU2   Kotgirwar       Adesh                   33 20-JUL-18        108
18CSU2013CSU2   Nawghare        Adhney                  34 08-AUG-18        109
18CSU2019CSU2   Gupta           Ayush                   41 12-JUL-18        101
18CSU2020CSU2   Kapre           Chaitanya               42 25-JUL-18        104
18CSU2021CSU2   Paliwal         Dev                     43 21-JUL-18        108
18CSU2029CSU2   Shukla          Gaurav                  44 17-JUL-18        109
18CSU2032CSU2   Sharma          Keshubh                 53 20-JUL-18        109
18CSU2033CSU2   Thorane         Kunal                   54 08-AUG-18        108
18CSU2034CSU2   Khandhadiya     Mehul                   55 19-JUL-18        104
18CSU2035CSU2   Tiwari          Nikhil                  56 04-JUL-18        101
18CSU2042CSU2   Kale            Rishikesh               63 07-JUL-18        104
18CSU2043CSU2   Parashar        Ritik                   64 19-JUL-18        108
```

```
ENROLL          LNAME           FNAME                 ROLL REG_DT       ADVISOR
--------------- --------------- --------------- ---------- --------- ----------
18CSU2044CSU2   Chandani        Rohit                   65 08-AUG-18        101
18CSU2051CSU2   Jha             Shubham                 78 12-JUL-18        109
18CSU2052CSU2   Kushwah         Yaman                   79 17-JUL-18        108
18CSU2053CSU2   Bhageriya       Yash                    80 19-JUL-18        104
16CSU2095CSU2   Soni            Renuka                  30 25-JUL-16        109
16CSU2094CSU2   Rangari         Mayank                  87 25-JUL-16        108
18CSU2004CSU2   Fadnavis        Ketki                    5 14-JUL-18        102
18CSU2005CSU2   Sharma          Lalita                   6 10-JUL-18        110
18CSU2027CSU2   Baheti          Simran                  15 20-JUL-18        102
18CSU2028CSU2   Negi            Urvi                    16 19-JUL-18        110
18CSU2014CSU2   Chandak         Akshat                  35 20-JUL-18        102
18CSU2060CSU2   Chole           Amey                    36 08-AUG-18        110
18CSU2058CSU2   Virdi           Gursewak                45 07-JUL-18        110
19CSU2206CSU2   Khandagale      Saurabh                 46 10-AUG-19        102
18CSU2036CSU2   Dandekar        Paritosh                57 14-JUL-18        102
18CSU2037CSU2   Gupta           Pavankumar              58 03-JUL-18        110
18CSU2045CSU2   Parikh          Rushil                  71 07-JUL-18        110
18CSU2046CSU2   Pandey          Sankalp                 72 07-JUL-18        102
18CSU2054CSU2   Daware          Yash                    81 20-JUL-18        102
18CSU2059CSU2   Roy             Yash                    82 07-JUL-18        110
17CSU2093CSU2   Sharnagat       Love                    68 25-JUL-17        110
18CSU2006CSU2   Gupta           Muskan                   7 19-JUL-18        103

ENROLL          LNAME           FNAME                 ROLL REG_DT       ADVISOR
--------------- --------------- --------------- ---------- --------- ----------
18CSU2007CSU2   Devikar         Prateeksha               8 13-JUL-18        106
19CSU2201CSU2   Pathe           Deepali                 17 10-AUG-19        106
19CSU2202CSU2   Bhanuse         Prachi                  18 11-AUG-19        103
18CSU2015CSU2   Ray             Amit                    37 20-JUL-18        103
18CSU2016CSU2   Pandharipande   Aryan                   38 07-JUL-18        106
19CSU2204CSU2   Thakur          Ganesh                  47 22-AUG-19        106
19CSU2205CSU2   Pardhi          Manishkumar             48 23-AUG-19        103
18CSU2038CSU2   Agrawal         Rahul                   59 16-JUL-18        103
18CSU2039CSU2   Chandak         Rajat                   60 20-JUL-18        106
18CSU2047CSU2   Sushir          Saurabh                 73 07-JUL-18        103
17CSU2052CSU2   Nimbalkar       Shardul                 74 28-JUL-17        106
18CSU2055CSU2   Dhamecha        Yash                    83 21-JUL-18        106
18CSU2056CSU2   Jain            Yash                    84 03-JUL-18        103
17CSU2092CSU2   Soni            Anujesh                 67 25-JUL-17        103
18CSU2008CSU2   Taori           Priyal                   9 19-JUL-18        105
```

```
18CSU2022CSU2   Chouksey        Rashi                   10 08-AUG-18        107
19CSU2203CSU2   Tripathi        Siddhi                  19 31-AUG-19        107
18CSU2017CSU2   Uplanchiwar     Atharva                 39 07-JUL-18        105
18CSU2018CSU2   Paliwal         Atharva                 40 20-JUL-18        107
18CSU2030CSU2   Karwa           Harsh                   51 11-JUL-18        105
18CSU2031CSU2   Kapse           Jayesh                  52 08-AUG-18        107
18CSU2040CSU2   Agrawal         Ram                     61 19-JUL-18        107

ENROLL          LNAME           FNAME                 ROLL REG_DT       ADVISOR
--------------- --------------- --------------- ---------- --------- ----------
18CSU2041CSU2   Khandelwal      Raunak                  62 19-JUL-18        105
18CSU2048CSU2   Tapas           Shashank                75 07-JUL-18        105
18CSU2049CSU2   Bagadia         Shivam                  76 20-JUL-18        107
18CSU2050CSU2   Nemani          Shreyas                 77 20-JUL-18        105
18CSU2057CSU2   Siral           Yogesh                  85 21-JUL-18        105
17CSU2047CSU2   Pandey          Shapath                 86 27-JUL-17        107
17CSU2091CSU2   Singh           Ayush                   66 27-JUL-17        107
                Namjoshi        Naveen                  88 14-AUG-19        109
                Tipnis          Tushar                  89 14-AUG-19        110

75 rows selected.
```

```
INSERT INTO STUDENT_VW_RO
  2  VALUES (NULL,'Goldsmith','Cinderella',91,'18-AUG-2019',101);
```

```
INSERT INTO STUDENT_VW_RO
*
ERROR at line 1:
ORA-42399: cannot perform a DML operation on a read-only view
```

The view does not allow the insertion of the record as it has a constraint READ ONLY which means it is only for viewing and not for editing or updating.

****

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-2019 - into STUDENT_VW_CK. Observe the effect.
****

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

```
View created.
```

```
DESC STUDENT_VW_CK
```

```
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ENROLL                                             CHAR(13)
 LNAME                                     NOT NULL VARCHAR2(15)
 FNAME                                     NOT NULL VARCHAR2(15)
 ROLL                                      NOT NULL NUMBER(3)
 REG_DT                                    NOT NULL DATE
 ADVISOR                                   NOT NULL NUMBER(3)
```

```
SELECT *
     FROM STUDENT_VW_CK;
```

```
ENROLL          LNAME           FNAME                 ROLL REG_DT       ADVISOR
--------------- --------------- --------------- ---------- --------- ----------
18CSU2001CSU2   Sayed           Afra                     1 20-JUL-18        101
18CSU2003CSU2   Rajendran       Anjali                   3 19-JUL-18        108
18CSU2009CSU2   Menghal         Aradhita                 4 07-JUL-18        109
18CSU2023CSU2   Deshmukh        Ritul                   11 18-JUL-18        101
18CSU2025CSU2   Agnihotri       Shreya                  13 07-JUL-18        108
```

```
18CSU2026CSU2   Shukla          Shrishti                14 19-JUL-18        109
18CSU2010CSU2   Muley           Aayush                  31 19-JUL-18        101
18CSU2012CSU2   Kotgirwar       Adesh                   33 20-JUL-18        108
18CSU2013CSU2   Nawghare        Adhney                  34 08-AUG-18        109
18CSU2019CSU2   Gupta           Ayush                   41 12-JUL-18        101
18CSU2021CSU2   Paliwal         Dev                     43 21-JUL-18        108
18CSU2029CSU2   Shukla          Gaurav                  44 17-JUL-18        109
18CSU2032CSU2   Sharma          Keshubh                 53 20-JUL-18        109
18CSU2033CSU2   Thorane         Kunal                   54 08-AUG-18        108
18CSU2035CSU2   Tiwari          Nikhil                  56 04-JUL-18        101
18CSU2043CSU2   Parashar        Ritik                   64 19-JUL-18        108
18CSU2044CSU2   Chandani        Rohit                   65 08-AUG-18        101
18CSU2051CSU2   Jha             Shubham                 78 12-JUL-18        109
18CSU2052CSU2   Kushwah         Yaman                   79 17-JUL-18        108
16CSU2095CSU2   Soni            Renuka                  30 25-JUL-16        109
16CSU2094CSU2   Rangari         Mayank                  87 25-JUL-16        108
18CSU2006CSU2   Gupta           Muskan                   7 19-JUL-18        103

ENROLL          LNAME           FNAME                 ROLL REG_DT       ADVISOR
--------------- --------------- --------------- ---------- --------- ----------
19CSU2202CSU2   Bhanuse         Prachi                  18 11-AUG-19        103
18CSU2015CSU2   Ray             Amit                    37 20-JUL-18        103
19CSU2205CSU2   Pardhi          Manishkumar             48 23-AUG-19        103
18CSU2038CSU2   Agrawal         Rahul                   59 16-JUL-18        103
18CSU2047CSU2   Sushir          Saurabh                 73 07-JUL-18        103
18CSU2056CSU2   Jain            Yash                    84 03-JUL-18        103
17CSU2092CSU2   Soni            Anujesh                 67 25-JUL-17        103
18CSU2008CSU2   Taori           Priyal                   9 19-JUL-18        105
18CSU2017CSU2   Uplanchiwar     Atharva                 39 07-JUL-18        105
18CSU2030CSU2   Karwa           Harsh                   51 11-JUL-18        105
18CSU2041CSU2   Khandelwal      Raunak                  62 19-JUL-18        105
18CSU2048CSU2   Tapas           Shashank                75 07-JUL-18        105
18CSU2050CSU2   Nemani          Shreyas                 77 20-JUL-18        105
18CSU2057CSU2   Siral           Yogesh                  85 21-JUL-18        105
                Namjoshi        Naveen                  88 14-AUG-19        109

37 rows selected.
```

```
INSERT INTO STUDENT_VW_CK
  2  VALUES(NULL,'Ford','Sebastian',92,'18-AUG-2019',104);
```

```
INSERT INTO STUDENT_VW_CK
```

```
            *
ERROR at line 1:
ORA-01402: view WITH CHECK OPTION where-clause violation
```

As the STUDENT_VW_CK only allows the insertion when ADVISOR has ID 101,103,105,108,109 but while inserting the record here the ID is 104 which is not in the specified IDS which violates the CHECK CONSTRAINT and throws error.

****

QUERY-09: 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.

****

```
SELECT VIEW_NAME,TEXT
  2  FROM USER_VIEWS
  3  WHERE VIEW_NAME LIKE 'STUDENT%';
```

```
VIEW_NAME
------------------------------
TEXT
--------------------------------------------------------------------------------
STUDENT_VW
SELECT ENROLL,LNAME,FNAME,ROLL,REG_DT,ADVISOR
FROM STUDENT

STUDENT_VW_CK
SELECT ENROLL,LNAME,FNAME,ROLL,REG_DT,ADVISOR
FROM STUDENT
WHERE ADVISOR IN (101

STUDENT_VW_RO
SELECT ENROLL,LNAME,FNAME,ROLL,REG_DT,ADVISOR
FROM STUDENT
WITH READ ONLY


3 rows selected.
```

```
                   SELECT B.VIEW_NAME,A.CONSTRAINT_NAME,A.CONSTRAINT_TYPE
                     2  FROM USER_CONSTRAINTS A JOIN USER_VIEWS B
                     3  ON B.VIEW_NAME=A.TABLE_NAME
                     4  WHERE B.VIEW_NAME LIKE 'STUDENT%';

VIEW_NAME                      CONSTRAINT_NAME                C
------------------------------ ------------------------------ -
STUDENT_VW_CK                  STUDENT_ADV_CK                 V
STUDENT_VW_RO                  STUDENT_RO_VW                  O

2 rows selected.
```

****

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
                     2  FOR STAFF;

Synonym created.


                   DESC FACULTY_SN


 Name                                     Null?    Type
 ---------------------------------------- -------- ----------------------------
 SID                                      NOT NULL NUMBER(3)
 NAME                                     NOT NULL VARCHAR2(25)
 BRANCH                                   NOT NULL VARCHAR2(4)
 DESG                                     NOT NULL VARCHAR2(9)
 JOIN_DT                                  NOT NULL DATE
```

```
                    SELECT *
                      2  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-18
       108 Jasmine Arora             CSE  Assistant 11-AUG-17
       109 Vallabh Pai               CSE  Assistant 17-SEP-18
       110 Harmeet Kullar            AIML Assistant 17-MAR-19

10 rows selected.
```

```
                    INSERT INTO FACULTY_SN
                      2  VALUES (111,'Dhawal Giri','AIML','Assistant','29-JUL-
                    2021');

1 row created.
```

```
                    DESC STAFF
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SID                                       NOT NULL NUMBER(3)
 NAME                                      NOT NULL VARCHAR2(25)
 BRANCH                                    NOT NULL VARCHAR2(4)
 DESG                                      NOT NULL VARCHAR2(9)
 JOIN_DT                                   NOT NULL DATE
```

```
                    SELECT *
                      2  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-18
       108 Jasmine Arora            CSE  Assistant 11-AUG-17
       109 Vallabh Pai              CSE  Assistant 17-SEP-18
       110 Harmeet Kullar           AIML Assistant 17-MAR-19
       111 Dhawal Giri              AIML Assistant 29-JUL-21

11 rows selected.
```

The recorded has been successfully added in the STAFF using FACULTY_SN. Synonym just means the other name and here FACULTY_SN is the other name of STAFF.

****

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 FNAME_KEY
  2  ON STUDENT(FNAME);
```

```
ON STUDENT(FNAME)
   *
ERROR at line 2:
ORA-01452: cannot CREATE UNIQUE INDEX; duplicate keys found
```

As many students have same FNAME there are many duplicate keys present and we are not able to create a unique index as FNAME should be unique and different for all students.

```
CREATE INDEX FNAME_IDX
  2  ON STUDENT(FNAME);
```

```
Index created.

             SELECT *
               2  FROM STUDENT
               3  WHERE FNAME='Atharva';


      ROLL LNAME           FNAME           EMAIL           ENROLL
---------- --------------- --------------- --------------- ---------------
   ADVISOR      PHONE REG_DT
---------- ---------- ---------
        39 Uplanchiwar     Atharva         uplanchiwarad@r 18CSU2017CSU2
                                           knec.edu
       105 9860320604 07-JUL-18

        40 Paliwal         Atharva         paliwalap@rknec 18CSU2018CSU2
                                           .edu
       107 7218385709 20-JUL-18


2 rows selected.


Execution Plan
----------------------------------------------------------
Plan hash value: 2420807299

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

| Id  | Operation                   | Name      | Rows  | Bytes | Cost (%CPU)| T
ime     |

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

|   0 | SELECT STATEMENT            |           |     2 |   144 |     2   (0)| 0
0:00:01 |

|   1 |  TABLE ACCESS BY INDEX ROWID| STUDENT   |     2 |   144 |     2   (0)| 0
0:00:01 |
|*  2 |   INDEX RANGE SCAN          | FNAME_IDX |     2 |       |     1   (0)| 0
0:00:01 |

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


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

   2 - access("FNAME"='Atharva')


Statistics
----------------------------------------------------------
          8  recursive calls
          0  db block gets
          6  consistent gets
          0  physical reads
          0  redo size
       1203  bytes sent via SQL*Net to client
        524  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
          2  rows processed



                    CREATE INDEX FNAME_LNAME_IDX
                      2  ON STUDENT(FNAME,LNAME);

Index created.

                    SELECT *
                      2  FROM STUDENT
                      3  WHERE FNAME='Atharva' and LNAME='Paliwal';

      ROLL LNAME           FNAME           EMAIL           ENROLL
---------- --------------- --------------- --------------- ---------------
   ADVISOR      PHONE REG_DT
---------- ---------- ---------
```

```
        40 Paliwal         Atharva         paliwalap@rknec 18CSU2018CSU2
                                           .edu
       107 7218385709 20-JUL-18


1 row selected.



Execution Plan
----------------------------------------------------------
Plan hash value: 4239860490

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

| Id  | Operation                   | Name            | Rows  | Bytes | Cost (%C
PU)| Time     |

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

|   0 | SELECT STATEMENT            |                 |     1 |    72 |     2
(0)| 00:00:01 |

|   1 |  TABLE ACCESS BY INDEX ROWID| STUDENT         |     1 |    72 |     2
(0)| 00:00:01 |

|*  2 |   INDEX RANGE SCAN          | FNAME_LNAME_IDX |     1 |       |     1
(0)| 00:00:01 |

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


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

   2 - access("FNAME"='Atharva' AND "LNAME"='Paliwal')


Statistics
```

```
----------------------------------------------------------
          8  recursive calls
          0  db block gets
          5  consistent gets
          0  physical reads
          0  redo size
       1068  bytes sent via SQL*Net to client
        524  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
          1  rows processed
```

```
                    SELECT INDEX_NAME, INDEX_TYPE, UNIQUENESS
                      2  FROM USER_INDEXES
                      3  WHERE TABLE_NAME IN ('STUDENT','STAFF','DEPT');

INDEX_NAME                     INDEX_TYPE                  UNIQUENES
------------------------------ --------------------------- ---------
DEPT_PK_BRANCH                 NORMAL                      UNIQUE
STAFF_PK_SID                   NORMAL                      UNIQUE
FNAME_LNAME_IDX                NORMAL                      NONUNIQUE
FNAME_IDX                      NORMAL                      NONUNIQUE
SYS_C007486                    NORMAL                      UNIQUE
SYS_C007485                    NORMAL                      UNIQUE
STUDENT_PK_ROLL                NORMAL                      UNIQUE

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 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;

*********************************************************************************

                    CREATE INDEX LNAME_FN_IDX
                      2  ON STUDENT(UPPER(LNAME));

Index created.


                    SELECT *
                      2  FROM STUDENT
                      3  WHERE UPPER(LNAME)='PALIWAL';

      ROLL LNAME           FNAME           EMAIL           ENROLL
---------- --------------- --------------- --------------- ---------------
   ADVISOR      PHONE REG_DT
---------- ---------- ---------
        43 Paliwal         Dev             paliwald@rknec. 18CSU2021CSU2
                                           edu
       108 9665745065 21-JUL-18

        40 Paliwal         Atharva         paliwalap@rknec 18CSU2018CSU2
                                           .edu
       107 7218385709 20-JUL-18


2 rows selected.


Execution Plan
----------------------------------------------------------
Plan hash value: 869780340

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

| Id  | Operation                    | Name         | Rows  | Bytes | Cost (%CPU)
| Time     |

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

```
|   0 | SELECT STATEMENT            |              |     1 |    81 |     1   (0)
| 00:00:01 |

|   1 |  TABLE ACCESS BY INDEX ROWID| STUDENT      |     1 |    81 |     1   (0)
| 00:00:01 |

|*  2 |   INDEX RANGE SCAN          | LNAME_FN_IDX |     1 |       |     1   (0)
| 00:00:01 |

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


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

   2 - access(UPPER("LNAME")='PALIWAL')


Statistics
----------------------------------------------------------
          0  recursive calls
          0  db block gets
          4  consistent gets
          0  physical reads
          0  redo size
       1182  bytes sent via SQL*Net to client
        524  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
          2  rows processed
```

```
                    CREATE INDEX FNAME_LNAME_FN_IDX
                      2  ON STUDENT(UPPER(FNAME),UPPER(LNAME));

Index created.



                    SELECT *
                      2  FROM STUDENT
                      3  WHERE UPPER(FNAME)='ATHARVA' AND UPPER(LNAME)='PALIWAL';

      ROLL LNAME           FNAME           EMAIL           ENROLL
---------- --------------- --------------- --------------- ---------------
   ADVISOR      PHONE REG_DT
---------- ---------- ---------
        40 Paliwal         Atharva         paliwalap@rknec 18CSU2018CSU2
                                           .edu
       107 7218385709 20-JUL-18


1 row selected.


Execution Plan
----------------------------------------------------------
Plan hash value: 869780340

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

| Id  | Operation                   | Name          | Rows  | Bytes | Cost (%CPU)
| Time     |

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

|   0 | SELECT STATEMENT            |               |     1 |    85 |     1   (0)
| 00:00:01 |

|*  1 |  TABLE ACCESS BY INDEX ROWID| STUDENT       |     1 |    85 |     1   (0)
| 00:00:01 |
|*  2 |   INDEX RANGE SCAN          | LNAME_FN_IDX |     1 |       |     1   (0)
| 00:00:01 |

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


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

   1 - filter(UPPER("FNAME")='ATHARVA')
   2 - access(UPPER("LNAME")='PALIWAL')


Statistics
----------------------------------------------------------
         12  recursive calls
          0  db block gets
          7  consistent gets
          0  physical reads
          0  redo size
       1064  bytes sent via SQL*Net to client
        524  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          0  sorts (memory)
          0  sorts (disk)
          1  rows processed
```

****

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.

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.

****

a)

```
                    SAVEPOINT SP_NONE;

Savepoint created.

                    INSERT INTO STUDENT
                      2  VALUES (91,'Goldsmith','Cinderella',NULL,NULL,101,NULL,'18-AUG-2019');

1 row created.

                    INSERT INTO STUDENT
                      2  VALUES (92,'Ford','Sebastian',NULL,NULL,104,NULL,'18-AUG-2019');

1 row created.

                    SAVEPOINT SP_FORD;

Savepoint created.
```

b)

```
                    UPDATE STUDENT
                      2  SET ADVISOR=108
                      3  WHERE FNAME='Naveen' and LNAME='Namjoshi';

1 row updated.

                    SAVEPOINT SP_NAV;

Savepoint created.
```

c)

```
                    UPDATE STUDENT
```

```
                      2  SET ADVISOR=111
                      3  WHERE FNAME='Tushar' and LNAME='Tipnis';

1 row updated.

                    SAVEPOINT SP_TUS;


Savepoint created.



                    SELECT *
                      2  FROM STUDENT
                      3  WHERE ROLL BETWEEN 88 AND 93;

      ROLL LNAME           FNAME           EMAIL
---------- --------------- --------------- -------------------------
ENROLL           ADVISOR      PHONE REG_DT
------------- ---------- ---------- ---------
        88 Namjoshi        Naveen
                     108            14-AUG-19


        89 Tipnis          Tushar
                     111            14-AUG-19


        91 Goldsmith       Cinderella
                     101            18-AUG-19



      ROLL LNAME           FNAME           EMAIL
---------- --------------- --------------- -------------------------
ENROLL           ADVISOR      PHONE REG_DT
------------- ---------- ---------- ---------
        92 Ford            Sebastian
                     104            18-AUG-19
```

****

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 *
                      2  FROM STUDENT
                      3  WHERE ROLL BETWEEN 88 AND 93;

      ROLL LNAME           FNAME           EMAIL
---------- --------------- --------------- ------------------------
ENROLL           ADVISOR      PHONE REG_DT
------------- ---------- ---------- ---------
        88 Namjoshi        Naveen
                     109            14-AUG-19


        89 Tipnis          Tushar
                     110            14-AUG-19


        91 Goldsmith       Cinderella
                     101            18-AUG-19



      ROLL LNAME           FNAME           EMAIL
---------- --------------- --------------- ------------------------
ENROLL           ADVISOR      PHONE REG_DT
------------- ---------- ---------- ---------
        92 Ford            Sebastian
                     104            18-AUG-19




                    ROLLBACK TO SP_NONE;


Rollback complete.
```

```
                    SELECT *
                      2  FROM STUDENT
                      3  WHERE ROLL BETWEEN 88 AND 93;

      ROLL LNAME           FNAME           EMAIL
---------- --------------- --------------- ------------------------
ENROLL           ADVISOR      PHONE REG_DT
------------- ---------- ---------- ---------
        88 Namjoshi        Naveen
                     109            14-AUG-19

        89 Tipnis          Tushar
                     110            14-AUG-19
```

Conclusion:

Views are virtual tables that do not exist in the actual world. The views are built on top of an existing table or tables. Views are built for commonly used queries, complicated queries, and join queries. The view performs well when running complicated searches or joining queries. DML activities cannot be performed while in view. A view cab is designed to provide security. If the table on which the view is built is dropped, the view becomes inactive.

On most databases, it is usually better to utilise the newer identity choices, although sequences are unavoidable in Oracle.

A synonym is a database item that allows you to construct aliases for other objects. They are useful for simplifying your searches and enhancing the way data and objects are accessed.

Indexes are schema objects that are optionally connected with tables. Indexes are added to tables to enhance query speed. An Oracle Database index, similar to a guide's index, allows you to easily seek certain information in a table. A table can have as many indexes as it needs. After you build an index, the database will automatically maintain and use it. Changes to a table's contents or structure, such as adding new rows, changing rows, or removing rows, are integrated into all relevant indexes automatically. This is obvious to the user. Some indexes are produced implicitly as a result of restrictions put on a table. For example, the database generates an index on the columns of a primary key constraint or unique key constraint automatically.

A SAVEPOINT is a transaction marker that allows for a partial rewind. As modifications are made in a transaction, SAVEPOINTs can be created to indicate different points in the process. If an issue occurs, we may rollback to a SAVEPOINT or all the way back to the start of the transaction.

Viva Questions:

1. **How does a simple view differ from a complex view?**

   In SQL, a simple view is one that is generated by involving only one table. In other words, there is just one basis table in the case of a Simple View in SQL. Complex View, on the other hand, is formed by involving more than one table, i.e., several tables are projected in Complex View.

   Because just one table is in context in the case of a Simple View, no important associations need to be made in the case of this view in SQL. In the case of Complex View, however, many tables exist in the context, thus generic connections such as join conditions, a group by phrase, and an order by clause must be used.

   As Simple View just has one table, we can't utilise group methods like MAX(), COUNT(), and so on. In the case of a Complex View, however, we may utilise a variety of group functions owing to the many tables.
   DML operations were simple to conduct in Simple View.
   However, DML procedures could not always be executed in the case of a Complex view.

   As previously stated, DML operations like as INSERT, DELETE, and UPDATE are directly available. However, in the case of a Complex view, we cannot use INSERT, DELETE, or UPDATE.

   NOT NULL columns from the base table cannot be included in Simple View. NOT NULL columns, on the other hand, can be included in Complex view.

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

   Yes, they are changed whenever we use them. Views are not cached by default. When we SELECT from a view, the database must execute the query contained in the view to obtain the result set for your statement. The data that we see in a view is not really saved anywhere and is created on the fly from the tables. As a result, we should use caution when running views that are quite complicated. We should always remember that the view must be run before the result set can be retrieved.

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

As sequence numbers are produced independently of tables, the same sequence number can be used for one or more tables. Individual sequence numbers may appear to be skipped because they were created and utilised in a transaction that was afterwards rolled back. Furthermore, one user may be unaware that other users are pulling from the same series.

We can utilise sequences to produce primary key values automatically. When a sequence number is produced, it is increased regardless of whether the transaction commits or rolls back. If two users concurrently increase the same sequence, the sequence numbers acquired by each user may have gaps since the other user is generating sequence numbers. A user will never be able to obtain the sequence number created by another user. When one user generates a sequence value, that user can continue to access that value regardless of whether the series is incremented by another user.

4. How does a synonym differ from an alias?

When a table or tablespace is deleted, SYNONYM is also deleted. Only the creator has access to synonyms. ALIAS, on the other hand, is preserved even if a table or tablespace is deleted. Even if the table does not exist, ALIAS can be constructed. It is mostly used in dispersed environments to conceal program location information. Alias is a global object that is exposed to everyone.

5. Do you need to remove savepoints explicitly?

The SAVEPOINT statement is used to create a SQL savepoint and therefore the beginning of a sub-transaction within a transaction, as well as to name this SQL savepoint. No, we do not need to explicitly delete savepoints. A subsequent ROLLBACK TO command with the SQL savepoint name reverses all modifications done in the meanwhile, without impacting database activities performed within the transaction prior to the commencement of this sub-transaction. In the process, the SQL savepoint is also removed. SQL savepoints can also be deleted using the RELEASE SAVEPOINT command.