PRACTICAL: 3

Name: Arjun Doye
RollNo: 48

```
-- tablespace
CREATE TABLESPACE TSA1 DATAFILE 'C:\Desktop\tspa1.dbf' SIZE 10M;

CREATE TABLESPACE TSA2 DATAFILE 'C:\Desktop\tspa2.dbf' SIZE 10M;

CREATE TABLESPACE TSA3 DATAFILE 'C:\Desktop\tspa3.dbf' SIZE 10M;

CREATE TABLESPACE TSA4 DATAFILE 'C:\Desktop\tspa4.dbf' SIZE 10M;

-- QUESTION 1
/*
Write a query to create range portioned table:
1. Creates a table named- Sales consisting of four partitions, one for each quarter of sales.
 The columns sale_year, sale_month, and sale_day are the partitioning columns,
 while their values constitute the partitioning key of a specific row.
2. Each partition is given a name (sales_q1, sales_q2, ...), and each partition is
contained in a separate tablespace (tsa, tsb, ...)
3. The columns for table must be prod_id, cust_id, promo_id, quantify sold,
 amount_sold – all in number format and time_id.
*/


CREATE TABLE SALES
      (PROD_ID NUMBER(6),
      CUST_ID NUMBER(6),
      TIME_ID DATE,
      PROMO_ID NUMBER(6),
      QTY_SOLD NUMBER(6),
      AMT_SOLD NUMBER(4,2)
      )
      PARTITION BY RANGE(TIME_ID)
      (PARTITION SALES_Q1 VALUES LESS THAN ('01-APR-2018') TABLESPACE TSA1,
      PARTITION SALES_Q2 VALUES LESS THAN ('01-JUL-2018') TABLESPACE TSA2,
      PARTITION SALES_Q3 VALUES LESS THAN ('01-OCT-2018') TABLESPACE TSA3,
      PARTITION SALES_Q4 VALUES LESS THAN ('01-JAN-2019') TABLESPACE TSA4
      );

      INSERT INTO SALES VALUES ('123','1234','01-JAN-2018','11',23,34.5);


      INSERT INTO SALES VALUES ('125','1234','15-DEC-2018','21',23,34.5);
```

```
INSERT INTO SALES VALUES ('128','1234','29-APR-2018','31',23,34.5);

INSERT INTO SALES VALUES ('193','1234','23-SEP-2018','41',23,34.5);

exec dbms_stats.gather_table_stats(USER,'SALES');

SELECT TABLE_NAME,TABLESPACE_NAME,HIGH_VALUE,NUM_ROWS FROM USER_TAB_PARTITIONS WHERE
TABLE_NAME='SALES';
/*
TABLE_NAME                     TABLESPACE_NAME                HIGH_VALUE                                                                                          NUM_ROWS
------------------------------ ------------------------------ --------------------------------------------------------------------------------------------------- ----------
SALES                          TSA1                           TO_DATE(' 2018-04-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA        1
SALES                          TSA2                           TO_DATE(' 2018-07-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA        1
SALES                          TSA3                           TO_DATE(' 2018-10-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA        1
SALES                          TSA4                           TO_DATE(' 2019-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA        1
*/

SELECT * FROM SALES PARTITION(SALES_Q1);
/*
  PROD_ID    CUST_ID TIME_ID    PROMO_ID   QTY_SOLD   AMT_SOLD
---------- ---------- --------- ---------- ---------- ----------
     123       1234 01-JAN-18         11         23       34.5
*/

SELECT * FROM SALES PARTITION(SALES_Q2);
/*
  PROD_ID    CUST_ID TIME_ID    PROMO_ID   QTY_SOLD   AMT_SOLD
---------- ---------- --------- ---------- ---------- ----------
     128       1234 29-APR-18         31         23       34.5
*/

SELECT * FROM SALES PARTITION(SALES_Q3);
/*
  PROD_ID    CUST_ID TIME_ID    PROMO_ID   QTY_SOLD   AMT_SOLD
---------- ---------- --------- ---------- ---------- ----------
     193       1234 23-SEP-18         41         23       34.5
*/

SELECT * FROM SALES PARTITION(SALES_Q4);

/*
  PROD_ID    CUST_ID TIME_ID    PROMO_ID   QTY_SOLD   AMT_SOLD
---------- ---------- --------- ---------- ---------- ----------
     125       1234 15-DEC-18         21         23       34.5
*/


--QUESTION 2:
```

```
/*
Create the same table as in Q1. With a different name with ENABLE ROW MOVEMENT
*/
      CREATE TABLE SALES1
      (PROD_ID NUMBER(6),
      CUST_ID NUMBER(6),
      TIME_ID DATE,
      PROMO_ID NUMBER(6),
      QTY_SOLD NUMBER(6),
      AMT_SOLD NUMBER(4,2)
      )
      PARTITION BY RANGE(TIME_ID)
      (PARTITION SALES_Q1 VALUES LESS THAN ('01-APR-2018') TABLESPACE TSA1,
      PARTITION SALES_Q2 VALUES LESS THAN ('01-JUL-2018') TABLESPACE TSA2,
      PARTITION SALES_Q3 VALUES LESS THAN ('01-OCT-2018') TABLESPACE TSA3,
      PARTITION SALES_Q4 VALUES LESS THAN ('01-JAN-2019') TABLESPACE TSA4
      )
      ENABLE ROW MOVEMENT;

      INSERT INTO SALES1 VALUES ('123','1234','01-JAN-2018','11',23,34.5);


      INSERT INTO SALES1 VALUES ('125','1234','15-DEC-2018','21',23,34.5);


      INSERT INTO SALES1 VALUES ('128','1234','29-APR-2018','31',23,34.5);


      INSERT INTO SALES1 VALUES ('193','1234','23-SEP-2018','41',23,34.5);

      SELECT * FROM SALES1 PARTITION(SALES_Q1);

      SELECT * FROM SALES1 PARTITION(SALES_Q2);

      SELECT * FROM SALES1 PARTITION(SALES_Q3);

      SELECT * FROM SALES1 PARTITION(SALES_Q4);

/*
SQL> SELECT * FROM SALES1 PARTITION(SALES_Q1);
  PROD_ID    CUST_ID TIME_ID    PROMO_ID   QTY_SOLD   AMT_SOLD
---------- ---------- --------- ---------- ---------- ----------
     123       1234 01-JAN-18         11         23       34.5
SQL>
SQL> SELECT * FROM SALES1 PARTITION(SALES_Q2);
  PROD_ID    CUST_ID TIME_ID    PROMO_ID   QTY_SOLD   AMT_SOLD
---------- ---------- --------- ---------- ---------- ----------
     128       1234 29-APR-18         31         23       34.5
SQL>
SQL> SELECT * FROM SALES1 PARTITION(SALES_Q3);
  PROD_ID    CUST_ID TIME_ID    PROMO_ID   QTY_SOLD   AMT_SOLD
---------- ---------- --------- ---------- ---------- ----------
     193       1234 23-SEP-18         41         23       34.5
```

```
SQL>
SQL> SELECT * FROM SALES1 PARTITION(SALES_Q4);
  PROD_ID    CUST_ID TIME_ID    PROMO_ID   QTY_SOLD   AMT_SOLD
---------- ---------- --------- ---------- ---------- ----------
     125       1234 15-DEC-18        21        23       34.5
*/

      UPDATE SALES1 SET TIME_ID='14-APR-2018' WHERE PROD_ID='123';


      /*
SQL> SELECT * FROM SALES1 PARTITION(SALES_Q1);
no rows selected
SQL>
SQL> SELECT * FROM SALES1 PARTITION(SALES_Q2);
  PROD_ID    CUST_ID TIME_ID    PROMO_ID   QTY_SOLD   AMT_SOLD
---------- ---------- --------- ---------- ---------- ----------
     128       1234 29-APR-18        31        23       34.5
     123       1234 14-APR-18        11        23       34.5
SQL>
SQL> SELECT * FROM SALES1 PARTITION(SALES_Q3);
  PROD_ID    CUST_ID TIME_ID    PROMO_ID   QTY_SOLD   AMT_SOLD
---------- ---------- --------- ---------- ---------- ----------
     193       1234 23-SEP-18        41        23       34.5
SQL>
SQL> SELECT * FROM SALES1 PARTITION(SALES_Q4);
  PROD_ID    CUST_ID TIME_ID    PROMO_ID   QTY_SOLD   AMT_SOLD
---------- ---------- --------- ---------- ---------- ----------
     125       1234 15-DEC-18        21        23       34.5
      */


      exec dbms_stats.gather_table_stats('RAKSHIT_74','SALES1');

      SELECT TABLE_NAME,TABLESPACE_NAME,HIGH_VALUE,NUM_ROWS FROM USER_TAB_PARTITIONS WHERE
TABLE_NAME='SALES1';
/*
TABLE_NAME                     TABLESPACE_NAME                HIGH_VALUE                                                                                      NUM_ROWS
------------------------------ ------------------------------ ----------------------------------------------------------------------------------------------- ----------
SALES1                         TSA1                           TO_DATE(' 2018-04-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA         0
SALES1                         TSA2                           TO_DATE(' 2018-07-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA         2
SALES1                         TSA3                           TO_DATE(' 2018-10-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA         1
SALES1                         TSA4                           TO_DATE(' 2019-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA         1
*/

-- QUESTION 3:
/*
CREATE TABLE WIH LIST PARITIONING.
*/
```

```
CREATE TABLE SALES_BY_LIST
     (DEPTNO NUMBER,
      DEPTNAME VARCHAR2(20),
      QUARTERLY_SALES NUMBER(10, 2),
      STATE VARCHAR2(2))
   PARTITION BY LIST (STATE)
     (PARTITION Q1_NORTHWEST VALUES ('OR', 'WA'),
      PARTITION Q1_SOUTHWEST VALUES ('AZ', 'UT', 'NM'),
      PARTITION Q1_NORTHEAST VALUES  ('NY', 'VM', 'NJ'),
         PARTITION Q1_SOUTHEAST VALUES  ('FL','GA'),
         PARTITION Q1_NORTHCENTRAL VALUES('SD','WI'),
     PARTITION Q1_SOUTHCENTRAL VALUES ('OK', 'TX'));

        INSERT INTO SALES_BY_LIST VALUES (10, 'ACCOUNTING', 100, 'WA');
        INSERT INTO SALES_BY_LIST VALUES (20, 'RND', 150, 'OR');
        INSERT INTO SALES_BY_LIST VALUES (30, 'SALES', 100, 'FL');
        INSERT INTO SALES_BY_LIST VALUES (40, 'HR', 10, 'TX');
        INSERT INTO SALES_BY_LIST VALUES (50, 'SYSTEMS_ENG', 10, 'CA');
/*
 INSERT INTO SALES_BY_LIST VALUES (50, 'SYSTEMS_ENG', 10, 'CA')
            *
ERROR at line 1:
ORA-14400: inserted partition key does not map to any partition
*/

        SELECT * FROM SALES_BY_LIST PARTITION (Q1_NORTHWEST);

        SELECT * FROM SALES_BY_LIST PARTITION (Q1_SOUTHWEST);

        SELECT * FROM SALES_BY_LIST PARTITION (Q1_NORTHEAST);

        SELECT * FROM SALES_BY_LIST PARTITION (Q1_SOUTHEAST);

        SELECT * FROM SALES_BY_LIST PARTITION (Q1_NORTHCENTRAL);

        SELECT * FROM SALES_BY_LIST PARTITION (Q1_SOUTHCENTRAL);

/*
 SELECT * FROM SALES_BY_LIST PARTITION (Q1_NORTHWEST);
    DEPTNO DEPTNAME            QUARTERLY_SALES ST
---------- -------------------- --------------- --
        10 ACCOUNTING                      100 WA
        20 RND                          150 OR
  SELECT * FROM SALES_BY_LIST PARTITION (Q1_SOUTHWEST);
no rows selected
  SELECT * FROM SALES_BY_LIST PARTITION (Q1_NORTHEAST);
no rows selected
  SELECT * FROM SALES_BY_LIST PARTITION (Q1_SOUTHEAST);
    DEPTNO DEPTNAME            QUARTERLY_SALES ST
---------- -------------------- --------------- --
        30 SALES                        100 FL
  SELECT * FROM SALES_BY_LIST PARTITION (Q1_NORTHCENTRAL);
no rows selected
  SELECT * FROM SALES_BY_LIST PARTITION (Q1_SOUTHCENTRAL);
```

```
   DEPTNO DEPTNAME           QUARTERLY_SALES ST
--------- -------------------- --------------- --
       40 HR                           10 TX
*/

        ALTER TABLE SALES_BY_LIST ADD PARTITION Q1_NEW VALUES('CA');



        INSERT INTO SALES_BY_LIST VALUES (50, 'SYSTEMS_ENG', 10, 'CA');

       --1 row created.

        ALTER TABLE SALES_BY_LIST ADD PARTITION Q1_NEW_DEF VALUES(DEFAULT);

        INSERT INTO SALES_BY_LIST VALUES (51, 'SYSTEMS_ENG', 10, 'AB');

        exec dbms_stats.gather_table_stats(USER,'SALES_BY_LIST');

       SELECT TABLE_NAME,TABLESPACE_NAME,HIGH_VALUE,NUM_ROWS FROM USER_TAB_PARTITIONS WHERE
TABLE_NAME='SALES_BY_LIST';

/*
TABLE_NAME                     TABLESPACE_NAME                HIGH_VALUE                                                                                   NUM_ROWS
------------------------------ ------------------------------ -------------------------------------------------------------------------------------------- ----------
SALES_BY_LIST                  USERS                          'OR', 'WA'                                                                                        2
SALES_BY_LIST                  USERS                          'AZ', 'UT', 'NM'                                                                                  0
SALES_BY_LIST                  USERS                          'NY', 'VM', 'NJ'                                                                                  0
SALES_BY_LIST                  USERS                          'FL', 'GA'                                                                                      1
SALES_BY_LIST                  USERS                          'SD', 'WI'                                                                                      0
SALES_BY_LIST                  USERS                          'OK', 'TX'                                                                                      1
SALES_BY_LIST                  USERS                          'CA'                                                                                            1
SALES_BY_LIST                  USERS                          DEFAULT                                                                                           1
8 rows selected.
*/
-- QUESTION 4:

/*
CREATE TABLE WITH HASH PARTITIONS, NO OF PARTITIONS SHOULD BE 5. DEMONSTRATE USING SYSTEM AND USER DEFINED
PARTITIONS.
*/

       -- SYSTEM DEFINED PARTITIONS
              CREATE TABLE EMPLOYEE_HASH
                   (EMP_NO NUMBER(6),
                   EMP_JOB VARCHAR(2),
                   EMP_SAL NUMBER(6),
                   EMP_DEPTNO NUMBER(6))
                   PARTITION BY HASH(EMP_NO)
                   PARTITIONS 5;

                   INSERT INTO EMPLOYEE_HASH VALUES(1116,'AB',1,11);
                   INSERT INTO EMPLOYEE_HASH VALUES(1212,'AX',1,12);
                   INSERT INTO EMPLOYEE_HASH VALUES(1390,'AC',1,13);
```

```
INSERT INTO EMPLOYEE_HASH VALUES(1413,'AD',1,14);
INSERT INTO EMPLOYEE_HASH VALUES(1582,'AE',1,15);


 exec dbms_stats.gather_table_stats(USER,'EMPLOYEE_HASH');

SELECT TABLE_NAME,PARTITION_NAME,HIGH_VALUE,NUM_ROWS FROM USER_TAB_PARTITIONS WHERE
TABLE_NAME='EMPLOYEE_HASH';
/*
TABLE_NAME                PARTITION_NAME            HIGH_VALUE                                                                                    NUM_ROWS
------------------------- ------------------------- ----------------------------------------------------------------------------------------------- ----------
EMPLOYEE_HASH             SYS_P31                                                                                                                    2
EMPLOYEE_HASH             SYS_P32                                                                                                                    2
EMPLOYEE_HASH             SYS_P33                                                                                                                    1
EMPLOYEE_HASH             SYS_P34                                                                                                                    0
EMPLOYEE_HASH             SYS_P35                                                                                                                    0
*/

-- USER PARTITION

CREATE TABLE EMPLOYEE_HASH_USER
(EMP_NO NUMBER(6),
EMP_JOB VARCHAR(2),
EMP_SAL NUMBER(6),
EMP_DEPTNO NUMBER(6))
PARTITION BY HASH(EMP_NO)
(PARTITION P1,
PARTITION P2,
PARTITION P3,
PARTITION P4,
PARTITION P5
);


INSERT INTO EMPLOYEE_HASH_USER VALUES(1116,'AB',1,11);
INSERT INTO EMPLOYEE_HASH_USER VALUES(1212,'AX',1,12);
INSERT INTO EMPLOYEE_HASH_USER VALUES(1390,'AC',1,13);
INSERT INTO EMPLOYEE_HASH_USER VALUES(1413,'AD',1,14);
INSERT INTO EMPLOYEE_HASH_USER VALUES(1582,'AE',1,15);

 exec dbms_stats.gather_table_stats(USER,'EMPLOYEE_HASH_USER');

SELECT TABLE_NAME,PARTITION_NAME,HIGH_VALUE,NUM_ROWS FROM USER_TAB_PARTITIONS WHERE
TABLE_NAME='EMPLOYEE_HASH_USER';
/*
TABLE_NAME                PARTITION_NAME            HIGH_VALUE                                                                                    NUM_ROWS
------------------------- ------------------------- ----------------------------------------------------------------------------------------------- ----------
EMPLOYEE_HASH_USER        P1                                                                                                                         2
EMPLOYEE_HASH_USER        P2                                                                                                                         2
EMPLOYEE_HASH_USER        P3                                                                                                                         1
EMPLOYEE_HASH_USER        P4                                                                                                                         0
EMPLOYEE_HASH_USER        P5                                                                                                                         0
*/
```

```
--QUESTION  5:
/*
CREATE MULTI- COLUMN RANGE PARTTIONED TABLE
1. DATE WITH MONTH, YEAR, DAY AND AMOUNT SOLD
2. INSERT THE NECESSARY AND GIVEN VALUES AND PROVIDE CONCLUSION OVER IT
*/

CREATE TABLE DATE_TABLE
(
	YEAR NUMBER(4),
	MONTH NUMBER(2),
	DAY NUMBER(2),
	AMT_SOLD NUMBER(5)
)
PARTITION BY RANGE(YEAR,MONTH)
(
	PARTITION P1 VALUES LESS THAN (2001,1),
	PARTITION P2 VALUES LESS THAN (2001,4),
	PARTITION P3 VALUES LESS THAN (2001,7),
	PARTITION P4 VALUES LESS THAN (2001,10),
	PARTITION P5 VALUES LESS THAN (2002,1),
	PARTITION P6 VALUES LESS THAN (MAXVALUE,MAXVALUE)
);

INSERT INTO DATE_TABLE VALUES(2001,3,17,11);

INSERT INTO DATE_TABLE VALUES(2001,11,1,33);

INSERT INTO DATE_TABLE VALUES(2021,3,17,11);

INSERT INTO DATE_TABLE VALUES(2002,1,1,11);

 exec dbms_stats.gather_table_stats(USER,'DATE_TABLE');

	SELECT TABLE_NAME,PARTITION_NAME,HIGH_VALUE,NUM_ROWS FROM USER_TAB_PARTITIONS WHERE
TABLE_NAME='DATE_TABLE';
/*
TABLE_NAME                     PARTITION_NAME                 HIGH_VALUE                                                                                       NUM_ROWS
------------------------------ ------------------------------ ------------------------------------------------------------------------------------------------ ----------
DATE_TABLE                     P1                             2001, 1                                                                                               0
DATE_TABLE                     P2                             2001, 4                                                                                               1
DATE_TABLE                     P3                             2001, 7                                                                                               0
DATE_TABLE                     P4                             2001, 10                                                                                              0
DATE_TABLE                     P5                             2002, 1                                                                                               1
DATE_TABLE                     P6                             MAXVALUE, MAXVALUE                                                                                    2
*/


	SELECT * FROM DATE_TABLE PARTITION(P1);

	SELECT * FROM DATE_TABLE PARTITION(P2);

	SELECT * FROM DATE_TABLE PARTITION(P3);
```

```
	SELECT * FROM DATE_TABLE PARTITION(P4);

	SELECT * FROM DATE_TABLE PARTITION(P5);

	SELECT * FROM DATE_TABLE PARTITION(P6);

/*
SQL> SELECT * FROM DATE_TABLE PARTITION(P1);
no rows selected
SQL> SELECT * FROM DATE_TABLE PARTITION(P2);
    YEAR      MONTH       DAY   AMT_SOLD
---------- ---------- ---------- ----------
    2001         3        17        11
SQL> SELECT * FROM DATE_TABLE PARTITION(P3);
no rows selected
SQL> SELECT * FROM DATE_TABLE PARTITION(P4);
no rows selected
SQL> SELECT * FROM DATE_TABLE PARTITION(P5);
    YEAR      MONTH       DAY   AMT_SOLD
---------- ---------- ---------- ----------
    2001        11         1        33
SQL> SELECT * FROM DATE_TABLE PARTITION(P6);
    YEAR      MONTH       DAY   AMT_SOLD
---------- ---------- ---------- ----------
    2021         3        17        11
    2002         1         1        11
*/

	-- QUESTION 6:

/*
CREATE MULTI COLUMN PARTITIONED INDEX SUCH THAT ALL PARTITIONS ARE EQUAL SIZED
*/
	CREATE TABLE SUPPLIER
	(
		SUP_ID NUMBER(6),
		P_NUM NUMBER(6),
		AMT_SOLD NUMBER(6)
	)
	PARTITION BY RANGE(SUP_ID,P_NUM)
	(
		PARTITION P1 VALUES LESS THAN (5,100),
		PARTITION P2 VALUES LESS THAN (5,200),
		PARTITION P3 VALUES LESS THAN (10,50),
		PARTITION P4 VALUES LESS THAN (10,200),
		PARTITION P5_DEF VALUES LESS THAN (MAXVALUE,MAXVALUE)
	);

	INSERT INTO SUPPLIER VALUES (5,5,1000);

	INSERT INTO SUPPLIER VALUES (5,150,1000);
```

```
INSERT INTO SUPPLIER VALUES (10,100,1000);


SELECT * FROM SUPPLIER PARTITION (P1);

SELECT * FROM SUPPLIER PARTITION (P2);

SELECT * FROM SUPPLIER PARTITION (P3);

SELECT * FROM SUPPLIER PARTITION (P4);

SELECT * FROM SUPPLIER PARTITION (P5_DEF);

/*
SQL> SELECT * FROM SUPPLIER PARTITION (P1);
   SUP_ID     P_NUM   AMT_SOLD
---------- ---------- ----------
        5         5      1000
SQL> SELECT * FROM SUPPLIER PARTITION (P2);
   SUP_ID     P_NUM   AMT_SOLD
---------- ---------- ----------
        5       150      1000
SQL> SELECT * FROM SUPPLIER PARTITION (P3);
no rows selected
SQL> SELECT * FROM SUPPLIER PARTITION (P4);
   SUP_ID     P_NUM   AMT_SOLD
---------- ---------- ----------
       10       100      1000
SQL> SELECT * FROM SUPPLIER PARTITION (P5_DEF);
no rows selected
*/

exec dbms_stats.gather_table_stats(USER,'SUPPLIER');

        SELECT TABLE_NAME,PARTITION_NAME,HIGH_VALUE,NUM_ROWS FROM USER_TAB_PARTITIONS WHERE
TABLE_NAME='SUPPLIER';

/*
TABLE_NAME                     PARTITION_NAME                 HIGH_VALUE                                                                                          NUM_ROWS
------------------------------ ------------------------------ --------------------------------------------------------------------------------------------------- ----------
SUPPLIER                       P1                             5, 100                                                                                                     1
SUPPLIER                       P2                             5, 200                                                                                                     1
SUPPLIER                       P3                             10, 50                                                                                                     0
SUPPLIER                       P4                             10, 200                                                                                                    1
SUPPLIER                       P5_DEF                             MAXVALUE, MAXVALUE                                                                                         0
*/

-- QUESTION 7:
/*
CREATE INTERVAL PARTITIONED TABLE. PERFORM PARTITION WITH THE INTERVAL OF 1 MONTHS
*/

CREATE TABLE SALES_INT
(PROD_ID NUMBER(6),
```

```
    CUST_ID NUMBER(6),
    TIME_ID DATE,
    PROMO_ID NUMBER(6),
    QTY_SOLD NUMBER(6),
    AMT_SOLD NUMBER(4,2)
    )
    PARTITION BY RANGE(TIME_ID)
    INTERVAL (NUMTOYMINTERVAL(1,'MONTH'))
    (PARTITION SALES_Q1 VALUES LESS THAN ('01-APR-2018') ,
    PARTITION SALES_Q2 VALUES LESS THAN ('01-JUL-2018'),
    PARTITION SALES_Q3 VALUES LESS THAN ('01-OCT-2018') ,
    PARTITION SALES_Q4 VALUES LESS THAN ('01-JAN-2019')
    );

    INSERT INTO SALES_INT VALUES ('123','1234','01-JAN-2018','11',23,34.5);


    INSERT INTO SALES_INT VALUES ('125','1234','15-DEC-2018','21',23,34.5);


    INSERT INTO SALES_INT VALUES ('128','1234','29-APR-2018','31',23,34.5);


    INSERT INTO SALES_INT VALUES ('193','1234','23-SEP-2018','41',23,34.5);

    INSERT INTO SALES_INT VALUES ('198','1239','23-FEB-2019','41',23,34.5);


        exec dbms_stats.gather_table_stats(USER,'SALES_INT');

        SELECT TABLE_NAME,PARTITION_NAME,HIGH_VALUE,NUM_ROWS FROM USER_TAB_PARTITIONS WHERE
TABLE_NAME='SALES_INT';
/*
TABLE_NAME                     PARTITION_NAME                 HIGH_VALUE
                NUM_ROWS
------------------------------ ------------------------------ ---------------------------------------------------------------------------------------- ----------
SALES_INT                      SALES_Q1                       TO_DATE(' 2018-04-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA        1
SALES_INT                      SALES_Q2                       TO_DATE(' 2018-07-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA        1
SALES_INT                      SALES_Q3                       TO_DATE(' 2018-10-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA        1
SALES_INT                      SALES_Q4                       TO_DATE(' 2019-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA        1
SALES_INT                      SYS_P61                       TO_DATE(' 2019-03-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA        1
*/

    SELECT * FROM SALES_INT PARTITION (SYS_P61);
/*
  PROD_ID    CUST_ID TIME_ID    PROMO_ID   QTY_SOLD   AMT_SOLD
---------- ---------- --------- ---------- ---------- ----------
     198       1239 23-FEB-19         41         23       34.5
*/
```

```
    INSERT INTO SALES_INT VALUES ('190','1239','23-MAR-2019','41',23,34.5);
/*
TABLE_NAME                     PARTITION_NAME                 HIGH_VALUE
       NUM_ROWS
------------------------------ ------------------------------ ------------------------------------------------------------------------------------ ----------
SALES_INT                      SALES_Q1                       TO_DATE(' 2018-04-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA
1
SALES_INT                      SALES_Q2                       TO_DATE(' 2018-07-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA
1
SALES_INT                      SALES_Q3                       TO_DATE(' 2018-10-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA
1
SALES_INT                      SALES_Q4                       TO_DATE(' 2019-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA
1
SALES_INT                      SYS_P61                        TO_DATE(' 2019-03-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA
1
SALES_INT                      SYS_P62                        TO_DATE(' 2019-04-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA
*/

    SELECT * FROM SALES_INT PARTITION (SYS_P62);
/*
  PROD_ID    CUST_ID TIME_ID    PROMO_ID   QTY_SOLD   AMT_SOLD
---------- ---------- --------- ---------- ---------- ----------
     190       1239 23-MAR-19         41         23       34.5
*/



-- QUESTION 8:
/*
PERFORM AND DEMONSTRATE ALL CASES OF DELETION IN REFRENCED AND CHILD TABLE
*/
CREATE TABLE ORDERS(
    ORDER_ID NUMBER(4) PRIMARY KEY,
    ORDER_DATE DATE NOT NULL,
    CUST_ID NUMBER(4),
    SHIP_ID NUMBER(4)
    )
    PARTITION BY RANGE(ORDER_DATE)
    (
    PARTITION ORDERS_Q1 VALUES LESS THAN ('01-APR-2018') ,
    PARTITION ORDERS_Q2 VALUES LESS THAN ('01-JUL-2018'),
    PARTITION ORDERS_Q3 VALUES LESS THAN ('01-OCT-2018') ,
    PARTITION ORDERS_Q4 VALUES LESS THAN ('01-JAN-2019')
    );


    CREATE TABLE ORDER_ITEMS(
```

```
	ITEM_ID NUMBER(4) PRIMARY KEY,
	ORDER_ID NUMBER(4) NOT NULL,
	PROD_ID NUMBER(4),
	PRICE NUMBER(4),
	QTY NUMBER(4) ,
	CONSTRAINT FK_ITEMS FOREIGN KEY(ORDER_ID) REFERENCES ORDERS
	)
	PARTITION BY REFERENCE (FK_ITEMS);


	INSERT INTO ORDERS VALUES (123,'12-FEB-2018',34,89);

	INSERT INTO ORDERS VALUES (124,'15-DEC-2018',34,909);

	exec dbms_stats.gather_table_stats(USER,'ORDERS');

	SELECT TABLE_NAME,PARTITION_NAME,HIGH_VALUE,NUM_ROWS FROM USER_TAB_PARTITIONS WHERE
TABLE_NAME='ORDERS';

/*
TABLE_NAME                     PARTITION_NAME                 HIGH_VALUE                                                                          NUM_ROWS
------------------------------ ------------------------------ ----------------------------------------------------------------------------------- ----------
ORDERS                         ORDERS_Q1                      TO_DATE(' 2018-04-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA         1
ORDERS                         ORDERS_Q2                      TO_DATE(' 2018-07-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA         0
ORDERS                         ORDERS_Q3                      TO_DATE(' 2018-10-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA         0
ORDERS                         ORDERS_Q4                      TO_DATE(' 2019-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA         1
*/


		INSERT INTO ORDER_ITEMS VALUES (111,123,456,78,90);

		INSERT INTO ORDER_ITEMS VALUES (112,124,456,78,90);


	exec dbms_stats.gather_table_stats(USER,'ORDER_ITEMS');

	SELECT TABLE_NAME,PARTITION_NAME,HIGH_VALUE,NUM_ROWS FROM USER_TAB_PARTITIONS WHERE
TABLE_NAME='ORDER_ITEMS';
/*
TABLE_NAME                     PARTITION_NAME                 HIGH_VALUE                                                                          NUM_ROWS
------------------------------ ------------------------------ ----------------------------------------------------------------------------------- ----------
ORDER_ITEMS                    ORDERS_Q1                                                                                                                 1
ORDER_ITEMS                    ORDERS_Q2                                                                                                                 0
ORDER_ITEMS                    ORDERS_Q3                                                                                                                 0
ORDER_ITEMS                    ORDERS_Q4                                                                                                                 1
*/

SELECT TABLE_NAME, PARTITION_NAME
    FROM USER_TAB_PARTITIONS WHERE TABLE_NAME IN ('ORDERS', 'ORDER_ITEMS');
/*
```

```
TABLE_NAME                PARTITION_NAME
------------------------------ ------------------------------
ORDERS                    ORDERS_Q1
ORDERS                    ORDERS_Q2
ORDERS                    ORDERS_Q3
ORDERS                    ORDERS_Q4
ORDER_ITEMS               ORDERS_Q1
ORDER_ITEMS               ORDERS_Q2
ORDER_ITEMS               ORDERS_Q3
ORDER_ITEMS               ORDERS_Q4
*/

ALTER TABLE ORDERS DROP PARTITION ORDERS_Q1;

SELECT TABLE_NAME, PARTITION_NAME
      FROM USER_TAB_PARTITIONS WHERE TABLE_NAME IN ('ORDERS', 'ORDER_ITEMS');

/*
TABLE_NAME                PARTITION_NAME
------------------------------ ------------------------------
ORDERS                    ORDERS_Q2
ORDERS                    ORDERS_Q3
ORDERS                    ORDERS_Q4
ORDER_ITEMS               ORDERS_Q2
ORDER_ITEMS               ORDERS_Q3
ORDER_ITEMS               ORDERS_Q4
*/


-- QUESTION 9:
/*
IMPLEMENT VIRTUAL COLUMN BASED PARTITIONING
*/

      CREATE TABLE EMPLOYEE (
            EMP_ID NUMBER(4) PRIMARY KEY,
            EMP_NAME VARCHAR2(20),
            FIX_SAL NUMBER(4),
            VAR_SAL NUMBER(4),
            TOTAL_SAL NUMBER(6)
                  GENERATED ALWAYS AS (
                              FIX_SAL + VAR_SAL
                  )VIRTUAL
            )
            PARTITION BY RANGE(TOTAL_SAL)
            (
                  PARTITION EMP_Q1 VALUES LESS THAN (25000),
                  PARTITION EMP_Q2 VALUES LESS THAN (50000),
                  PARTITION EMP_Q3 VALUES LESS THAN (75000),
                  PARTITION EMP_Q4 VALUES LESS THAN (MAXVALUE)
            );


            INSERT INTO EMPLOYEE (EMP_ID,EMP_NAME,FIX_SAL,VAR_SAL) VALUES (123,'SARTHAK',10,20);
```

```
        INSERT INTO EMPLOYEE (EMP_ID,EMP_NAME,FIX_SAL,VAR_SAL) VALUES (124,'RAKSHIT',100,2);


        SELECT * FROM EMPLOYEE;
/*
   EMP_ID EMP_NAME             FIX_SAL    VAR_SAL  TOTAL_SAL
---------- -------------------- ---------- ---------- ----------
     123 SARTHAK                 10         20         30
     124 RAKSHIT                100         2        102
*/



-- QUERY 10:

/*
DEMONSTRATE COMPOSITE PARTITIONING WITH ALL POSSIBLE SCENERIOS
*/

-- RANGE LIST PARTITION

    DROP TABLE CUSTOMER;
    CREATE TABLE CUSTOMER(
        CUST_ID NUMBER(4) PRIMARY KEY,
        CUST_NAME VARCHAR2(20),
        CUST_STATE VARCHAR2(20),
        TIME_ID DATE
    )
    PARTITION BY RANGE (TIME_ID)
        SUBPARTITION BY LIST (CUST_STATE)
            SUBPARTITION TEMPLATE
                (
                SUBPARTITION WEST VALUES ('MH','GJ'),
                SUBPARTITION SOUTH VALUES ('TN','AP'),
                SUBPARTITION NORTH VALUES ('UP','HP'),
                SUBPARTITION UN_KNOWN VALUES (DEFAULT)
                )
        (
                    PARTITION CUST_RG_1 VALUES LESS THAN ('01-JAN-2005'),
                    PARTITION CUST_RG_2 VALUES LESS THAN ('01-JAN-2010'),
                    PARTITION CUST_RG_3 VALUES LESS THAN ('01-JAN-2015'),
                    PARTITION CUST_RG_4 VALUES LESS THAN (MAXVALUE)
        )
    ;


    INSERT INTO CUSTOMER VALUES (123,'ABC','MH','01-JAN-2011');
    INSERT INTO CUSTOMER VALUES (124,'BCD','MH','01-FEB-2019');

    INSERT INTO CUSTOMER VALUES (125,'ABCD','AP','01-DEC-2001');
    INSERT INTO CUSTOMER VALUES (126,'AAAA','UP','01-DEC-2011');
    INSERT INTO CUSTOMER VALUES (127,'AAAB','UP','01-FEB-2011');
    INSERT INTO CUSTOMER VALUES (128,'AAC','CK','01-FEB-2015');
    INSERT INTO CUSTOMER VALUES (129,'POO','MH','04-NOV-2019');
```

```
exec dbms_stats.gather_table_stats(USER,'CUSTOMER');


        SELECT TABLE_NAME,PARTITION_NAME, COMPOSITE, HIGH_VALUE,NUM_ROWS
FROM USER_TAB_PARTITIONS WHERE TABLE_NAME='CUSTOMER';


/*
TABLE_NAME                     PARTITION_NAME                 COM HIGH_VALUE                                                                      NUM_ROWS
------------------------------ ------------------------------ --- ---------------------------------------------------------------------------------- ----------
CUSTOMER                       CUST_RG_1                      YES TO_DATE(' 2005-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA         1
CUSTOMER                       CUST_RG_2                      YES TO_DATE(' 2010-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA         0
CUSTOMER                       CUST_RG_3                      YES TO_DATE(' 2015-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA         3
CUSTOMER                       CUST_RG_4                      YES MAXVALUE                                                                                3
*/


      SELECT TABLE_NAME,PARTITION_NAME, SUBPARTITION_NAME, NUM_ROWS
FROM USER_TAB_SUBPARTITIONS WHERE TABLE_NAME='CUSTOMER';

/*
TABLE_NAME                     PARTITION_NAME                 SUBPARTITION_NAME              NUM_ROWS
------------------------------ ------------------------------ ------------------------------ ----------
CUSTOMER                       CUST_RG_1                      CUST_RG_1_WEST                 0
CUSTOMER                       CUST_RG_1                      CUST_RG_1_SOUTH                1
CUSTOMER                       CUST_RG_1                      CUST_RG_1_NORTH                0
CUSTOMER                       CUST_RG_1                      CUST_RG_1_UN_KNOWN             0
CUSTOMER                       CUST_RG_2                      CUST_RG_2_WEST                 0
CUSTOMER                       CUST_RG_2                      CUST_RG_2_SOUTH                0
CUSTOMER                       CUST_RG_2                      CUST_RG_2_NORTH                0
CUSTOMER                       CUST_RG_2                      CUST_RG_2_UN_KNOWN             0
CUSTOMER                       CUST_RG_3                      CUST_RG_3_WEST                 1
CUSTOMER                       CUST_RG_3                      CUST_RG_3_SOUTH                0
CUSTOMER                       CUST_RG_3                      CUST_RG_3_NORTH                2
TABLE_NAME                     PARTITION_NAME                 SUBPARTITION_NAME              NUM_ROWS
------------------------------ ------------------------------ ------------------------------ ----------
CUSTOMER                       CUST_RG_3                      CUST_RG_3_UN_KNOWN             0
CUSTOMER                       CUST_RG_4                      CUST_RG_4_WEST                 2
CUSTOMER                       CUST_RG_4                      CUST_RG_4_SOUTH                0
CUSTOMER                       CUST_RG_4                      CUST_RG_4_NORTH                0
CUSTOMER                       CUST_RG_4                      CUST_RG_4_UN_KNOWN             1
16 rows selected.
*/



-- QUERY 11:
-- RANGE(TIME_ID) - RANGE(CUST_ID) PARTITION

      DROP TABLE CUSTOMER;
```

```
CREATE TABLE CUSTOMER(
    CUST_ID NUMBER(4) PRIMARY KEY,
    CUST_NAME VARCHAR2(20),
    CUST_STATE VARCHAR2(20),
    TIME_ID DATE
)
PARTITION BY RANGE (TIME_ID)
    SUBPARTITION BY RANGE (CUST_ID)
        SUBPARTITION TEMPLATE
            (
            SUBPARTITION CUST_SUB_ID_1 VALUES LESS THAN (124),
            SUBPARTITION CUST_SUB_ID_2 VALUES LESS THAN (126),
            SUBPARTITION CUST_SUB_ID_3 VALUES LESS THAN (128),
            SUBPARTITION CUST_SUB_ID_4 VALUES LESS THAN (MAXVALUE)
            )
    (
            PARTITION CUST_RG_1 VALUES LESS THAN ('01-JAN-2005'),
            PARTITION CUST_RG_2 VALUES LESS THAN ('01-JAN-2010'),
            PARTITION CUST_RG_3 VALUES LESS THAN ('01-JAN-2015'),
            PARTITION CUST_RG_4 VALUES LESS THAN (MAXVALUE)
    )
;


INSERT INTO CUSTOMER VALUES (123,'ABC','MH','01-JAN-2011');
INSERT INTO CUSTOMER VALUES (124,'BCD','MH','01-FEB-2019');

INSERT INTO CUSTOMER VALUES (125,'ABCD','AP','01-DEC-2001');
INSERT INTO CUSTOMER VALUES (126,'AAAA','UP','01-DEC-2011');
INSERT INTO CUSTOMER VALUES (127,'AAAB','UP','01-FEB-2011');
INSERT INTO CUSTOMER VALUES (128,'AAC','CK','01-FEB-2015');
INSERT INTO CUSTOMER VALUES (129,'POO','MH','04-NOV-2019');

exec dbms_stats.gather_table_stats(USER,'CUSTOMER');

SELECT TABLE_NAME,PARTITION_NAME, COMPOSITE, HIGH_VALUE,NUM_ROWS
    FROM USER_TAB_PARTITIONS WHERE TABLE_NAME='CUSTOMER';

/*
TABLE_NAME                     PARTITION_NAME                 COM HIGH_VALUE                                                                      NUM_ROWS
------------------------------ ------------------------------ --- ---------------------------------------------------------------------------------- ----------
CUSTOMER                       CUST_RG_1                      YES TO_DATE(' 2005-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA         1
CUSTOMER                       CUST_RG_2                      YES TO_DATE(' 2010-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA         0
CUSTOMER                       CUST_RG_3                      YES TO_DATE(' 2015-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA         3
CUSTOMER                       CUST_RG_4                      YES MAXVALUE                                                                                3
*/

exec dbms_stats.gather_schema_stats(USER);
SELECT TABLE_NAME,PARTITION_NAME, SUBPARTITION_NAME, NUM_ROWS
    FROM USER_TAB_SUBPARTITIONS WHERE TABLE_NAME='CUSTOMER';
```

```
/*
TABLE_NAME                     PARTITION_NAME                 SUBPARTITION_NAME              NUM_ROWS
------------------------------ ------------------------------ ------------------------------ ----------
CUSTOMER                       CUST_RG_1                      CUST_RG_1_CUST_SUB_ID_1
CUSTOMER                       CUST_RG_1                      CUST_RG_1_CUST_SUB_ID_2
CUSTOMER                       CUST_RG_1                      CUST_RG_1_CUST_SUB_ID_3
CUSTOMER                       CUST_RG_1                      CUST_RG_1_CUST_SUB_ID_4
CUSTOMER                       CUST_RG_2                      CUST_RG_2_CUST_SUB_ID_1
CUSTOMER                       CUST_RG_2                      CUST_RG_2_CUST_SUB_ID_2
CUSTOMER                       CUST_RG_2                      CUST_RG_2_CUST_SUB_ID_3
CUSTOMER                       CUST_RG_2                      CUST_RG_2_CUST_SUB_ID_4
CUSTOMER                       CUST_RG_3                      CUST_RG_3_CUST_SUB_ID_1
CUSTOMER                       CUST_RG_3                      CUST_RG_3_CUST_SUB_ID_2
CUSTOMER                       CUST_RG_3                      CUST_RG_3_CUST_SUB_ID_3
TABLE_NAME                     PARTITION_NAME                 SUBPARTITION_NAME              NUM_ROWS
------------------------------ ------------------------------ ------------------------------ ----------
CUSTOMER                       CUST_RG_3                      CUST_RG_3_CUST_SUB_ID_4
CUSTOMER                       CUST_RG_4                      CUST_RG_4_CUST_SUB_ID_1
CUSTOMER                       CUST_RG_4                      CUST_RG_4_CUST_SUB_ID_2
CUSTOMER                       CUST_RG_4                      CUST_RG_4_CUST_SUB_ID_3
CUSTOMER                       CUST_RG_4                      CUST_RG_4_CUST_SUB_ID_4
16 rows selected.
*/


-- QUERY 12:
-- RANGE (TIME_ID) - HASH (CUST_ID) PARTITION


    DROP TABLE CUSTOMER;

    CREATE TABLE CUSTOMER(
        CUST_ID NUMBER(4) PRIMARY KEY,
        CUST_NAME VARCHAR2(20),
        CUST_STATE VARCHAR2(20),
        TIME_ID DATE
    )
    PARTITION BY RANGE (TIME_ID)
        SUBPARTITION BY HASH (CUST_ID)
            SUBPARTITIONS 4
        (
                    PARTITION CUST_RG_1 VALUES LESS THAN ('01-JAN-2005'),
                    PARTITION CUST_RG_2 VALUES LESS THAN ('01-JAN-2010'),
                    PARTITION CUST_RG_3 VALUES LESS THAN ('01-JAN-2015'),
                    PARTITION CUST_RG_4 VALUES LESS THAN (MAXVALUE)
        )
    ;


    INSERT INTO CUSTOMER VALUES (123,'ABC','MH','01-JAN-2011');
    INSERT INTO CUSTOMER VALUES (124,'BCD','MH','01-FEB-2019');

    INSERT INTO CUSTOMER VALUES (125,'ABCD','AP','01-DEC-2001');
    INSERT INTO CUSTOMER VALUES (126,'AAAA','UP','01-DEC-2011');
```

```
INSERT INTO CUSTOMER VALUES (127,'AAAB','UP','01-FEB-2011');
INSERT INTO CUSTOMER VALUES (128,'AAC','CK','01-FEB-2015');
INSERT INTO CUSTOMER VALUES (129,'POO','MH','04-NOV-2019');

exec dbms_stats.gather_table_stats('CHAMPION_DWM','CUSTOMER');

SELECT TABLE_NAME,PARTITION_NAME, COMPOSITE, HIGH_VALUE,NUM_ROWS
    FROM USER_TAB_PARTITIONS WHERE TABLE_NAME='CUSTOMER';
/*
TABLE_NAME                     PARTITION_NAME                 COM HIGH_VALUE                                                                      NUM_ROWS
------------------------------ ------------------------------ --- ---------------------------------------------------------------------------------- ----------
CUSTOMER                       CUST_RG_1                      YES TO_DATE(' 2005-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA         1
CUSTOMER                       CUST_RG_2                      YES TO_DATE(' 2010-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA         0
CUSTOMER                       CUST_RG_3                      YES TO_DATE(' 2015-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIA         3
CUSTOMER                       CUST_RG_4                      YES MAXVALUE                                                                                 3
*/


SELECT TABLE_NAME,PARTITION_NAME, SUBPARTITION_NAME, NUM_ROWS
    FROM USER_TAB_SUBPARTITIONS WHERE TABLE_NAME='CUSTOMER';
/*
TABLE_NAME                     PARTITION_NAME                 SUBPARTITION_NAME              NUM_ROWS
------------------------------ ------------------------------ ------------------------------ ----------
CUSTOMER                       CUST_RG_1                      SYS_SUBP41
CUSTOMER                       CUST_RG_1                      SYS_SUBP42
CUSTOMER                       CUST_RG_1                      SYS_SUBP43
CUSTOMER                       CUST_RG_1                      SYS_SUBP44
CUSTOMER                       CUST_RG_2                      SYS_SUBP45
CUSTOMER                       CUST_RG_2                      SYS_SUBP46
CUSTOMER                       CUST_RG_2                      SYS_SUBP47
CUSTOMER                       CUST_RG_2                      SYS_SUBP48
CUSTOMER                       CUST_RG_3                      SYS_SUBP49
CUSTOMER                       CUST_RG_3                      SYS_SUBP50
CUSTOMER                       CUST_RG_3                      SYS_SUBP51
TABLE_NAME                     PARTITION_NAME                 SUBPARTITION_NAME              NUM_ROWS
------------------------------ ------------------------------ ------------------------------ ----------
CUSTOMER                       CUST_RG_3                      SYS_SUBP52
CUSTOMER                       CUST_RG_4                      SYS_SUBP53
CUSTOMER                       CUST_RG_4                      SYS_SUBP54
CUSTOMER                       CUST_RG_4                      SYS_SUBP55
CUSTOMER                       CUST_RG_4                      SYS_SUBP56
*/



--    QUERY 13:
--    LIST(CUST_STATE) - HASH(CUST_ID) PARTITION

    DROP TABLE CUSTOMER;

    CREATE TABLE CUSTOMER(
```

```
        CUST_ID NUMBER(4) PRIMARY KEY,
        CUST_NAME VARCHAR2(20),
        CUST_STATE VARCHAR2(20),
        TIME_ID DATE
    )
    PARTITION BY LIST (CUST_STATE)
        SUBPARTITION BY HASH (CUST_ID)
            SUBPARTITIONS 4
        (
                PARTITION WEST VALUES ('MH','GJ'),
                PARTITION SOUTH VALUES ('TN','AP'),
                PARTITION NORTH VALUES ('UP','HP'),
                PARTITION UN_KNOWN VALUES (DEFAULT)
        )
    ;


    INSERT INTO CUSTOMER VALUES (123,'ABC','MH','01-JAN-2011');
    INSERT INTO CUSTOMER VALUES (124,'BCD','MH','01-FEB-2019');

    INSERT INTO CUSTOMER VALUES (125,'ABCD','AP','01-DEC-2001');
    INSERT INTO CUSTOMER VALUES (126,'AAAA','UP','01-DEC-2011');
    INSERT INTO CUSTOMER VALUES (127,'AAAB','UP','01-FEB-2011');
    INSERT INTO CUSTOMER VALUES (128,'AAC','CK','01-FEB-2015');
    INSERT INTO CUSTOMER VALUES (129,'POO','MH','04-NOV-2019');

    exec dbms_stats.gather_table_stats(USER,'CUSTOMER');

    SELECT TABLE_NAME,PARTITION_NAME, COMPOSITE, HIGH_VALUE,NUM_ROWS
        FROM USER_TAB_PARTITIONS WHERE TABLE_NAME='CUSTOMER';
/*
    TABLE_NAME                     PARTITION_NAME                 COM HIGH_VALUE                                                                      NUM_ROWS
------------------------------ ------------------------------ --- ---------------------------------------------------------------------------------------- ----------
CUSTOMER                       WEST                           YES 'MH', 'GJ'                                                                            3
CUSTOMER                       SOUTH                          YES 'TN', 'AP'                                                                            1
CUSTOMER                       NORTH                          YES 'UP', 'HP'                                                                            2
CUSTOMER                       UN_KNOWN                       YES DEFAULT                                                                                  1
*/


    exec dbms_stats.gather_table_stats(USER,'CUSTOMER');
    SELECT TABLE_NAME,PARTITION_NAME, SUBPARTITION_NAME, NUM_ROWS
        FROM USER_TAB_SUBPARTITIONS WHERE TABLE_NAME='CUSTOMER';
/*
TABLE_NAME                     PARTITION_NAME                 SUBPARTITION_NAME              NUM_ROWS
------------------------------ ------------------------------ ------------------------------ ----------
CUSTOMER                       WEST                           SYS_SUBP57
CUSTOMER                       WEST                           SYS_SUBP58
CUSTOMER                       WEST                           SYS_SUBP59
CUSTOMER                       WEST                           SYS_SUBP60
CUSTOMER                       SOUTH                          SYS_SUBP61
CUSTOMER                       SOUTH                          SYS_SUBP62
CUSTOMER                       SOUTH                          SYS_SUBP63
CUSTOMER                       SOUTH                          SYS_SUBP64
```

```
CUSTOMER                NORTH                   SYS_SUBP65
CUSTOMER                NORTH                   SYS_SUBP66
CUSTOMER                NORTH                   SYS_SUBP67
TABLE_NAME               PARTITION_NAME              SUBPARTITION_NAME             NUM_ROWS
------------------------------ ------------------------------ ------------------------------ ----------
CUSTOMER                NORTH                   SYS_SUBP68
CUSTOMER                UN_KNOWN                    SYS_SUBP69
CUSTOMER                UN_KNOWN                    SYS_SUBP70
CUSTOMER                UN_KNOWN                    SYS_SUBP71
CUSTOMER                UN_KNOWN                    SYS_SUBP72
*/


--      QUERY 14:
-- LIST(CUST_STATE) - LIST(CUST_ID)


DROP TABLE CUSTOMER;

      CREATE TABLE CUSTOMER(
            CUST_ID NUMBER(4) PRIMARY KEY,
            CUST_NAME VARCHAR2(20),
            CUST_STATE VARCHAR2(20),
            TIME_ID DATE
      )
      PARTITION BY LIST (CUST_STATE)
            SUBPARTITION BY LIST (CUST_ID)
                  SUBPARTITION TEMPLATE
            (
                  SUBPARTITION P1 VALUES (121,122,123),
                  SUBPARTITION P2 VALUES (124,125,126),
                  SUBPARTITION P3 VALUES (127,128),
                  SUBPARTITION P4 VALUES (DEFAULT)
            )
            (
                        PARTITION WEST VALUES ('MH','GJ'),
                        PARTITION SOUTH VALUES ('TN','AP'),
                        PARTITION NORTH VALUES ('UP','HP'),
                        PARTITION UN_KNOWN VALUES (DEFAULT)
            )
      ;


      INSERT INTO CUSTOMER VALUES (123,'ABC','MH','01-JAN-2011');
      INSERT INTO CUSTOMER VALUES (124,'BCD','MH','01-FEB-2019');

      INSERT INTO CUSTOMER VALUES (125,'ABCD','AP','01-DEC-2001');
      INSERT INTO CUSTOMER VALUES (126,'AAAA','UP','01-DEC-2011');
      INSERT INTO CUSTOMER VALUES (127,'AAAB','UP','01-FEB-2011');
      INSERT INTO CUSTOMER VALUES (128,'AAC','CK','01-FEB-2015');
      INSERT INTO CUSTOMER VALUES (129,'POO','MH','04-NOV-2019');

      exec dbms_stats.gather_table_stats(USER,'CUSTOMER');
```

```
    SELECT TABLE_NAME,PARTITION_NAME, COMPOSITE, HIGH_VALUE,NUM_ROWS
        FROM USER_TAB_PARTITIONS WHERE TABLE_NAME='CUSTOMER';
/*
    TABLE_NAME                PARTITION_NAME             COM HIGH_VALUE                                                                      NUM_ROWS
------------------------------ ------------------------------ --- ---------------------------------------------------------------------------------------------------- ----------
CUSTOMER                  WEST                       YES 'MH', 'GJ'                                                                          3
CUSTOMER                  SOUTH                       YES 'TN', 'AP'                                                                          1
CUSTOMER                  NORTH                       YES 'UP', 'HP'                                                                          2
CUSTOMER                  UN_KNOWN                        YES DEFAULT                                                                             1
*/

exec dbms_stats.gather_table_stats(USER,'CUSTOMER');
    SELECT TABLE_NAME,PARTITION_NAME, SUBPARTITION_NAME, NUM_ROWS
        FROM USER_TAB_SUBPARTITIONS WHERE TABLE_NAME='CUSTOMER';

/*
TABLE_NAME                PARTITION_NAME             SUBPARTITION_NAME              NUM_ROWS
------------------------------ ------------------------------ ------------------------------ ----------
CUSTOMER                  WEST                       WEST_P1                         1
CUSTOMER                  WEST                       WEST_P2                         1
CUSTOMER                  WEST                       WEST_P3                         0
CUSTOMER                  WEST                       WEST_P4                         1
CUSTOMER                  SOUTH                       SOUTH_P1                        0
CUSTOMER                  SOUTH                       SOUTH_P2                        1
CUSTOMER                  SOUTH                       SOUTH_P3                        0
CUSTOMER                  SOUTH                       SOUTH_P4                        0
CUSTOMER                  NORTH                       NORTH_P1                        0
CUSTOMER                  NORTH                       NORTH_P2                        1
CUSTOMER                  NORTH                       NORTH_P3                        1
TABLE_NAME                PARTITION_NAME             SUBPARTITION_NAME              NUM_ROWS
------------------------------ ------------------------------ ------------------------------ ----------
CUSTOMER                  NORTH                       NORTH_P4                        0
CUSTOMER                  UN_KNOWN                        UN_KNOWN_P1                     0
CUSTOMER                  UN_KNOWN                        UN_KNOWN_P2                     0
CUSTOMER                  UN_KNOWN                        UN_KNOWN_P3                     1
CUSTOMER                  UN_KNOWN                        UN_KNOWN_P4                     0
*/


-- QUERY 15:
-- LIST (CUST_STATE) - RANGE (CUST_ID)
DROP TABLE CUSTOMER;

    CREATE TABLE CUSTOMER(
        CUST_ID NUMBER(4) PRIMARY KEY,
        CUST_NAME VARCHAR2(20),
        CUST_STATE VARCHAR2(20),
        TIME_ID DATE
    )
    PARTITION BY LIST (CUST_STATE)
        SUBPARTITION BY RANGE (CUST_ID)
            SUBPARTITION TEMPLATE
        (
                SUBPARTITION CUST_SUB_ID_1 VALUES LESS THAN (124),
```

```
                SUBPARTITION CUST_SUB_ID_2 VALUES LESS THAN (126),
                SUBPARTITION CUST_SUB_ID_3 VALUES LESS THAN (128),
                SUBPARTITION CUST_SUB_ID_4 VALUES LESS THAN (MAXVALUE)
        )
        (
                PARTITION WEST VALUES ('MH','GJ'),
                PARTITION SOUTH VALUES ('TN','AP'),
                PARTITION NORTH VALUES ('UP','HP'),
                PARTITION UN_KNOWN VALUES (DEFAULT)
        )
    ;


    INSERT INTO CUSTOMER VALUES (123,'ABC','MH','01-JAN-2011');
    INSERT INTO CUSTOMER VALUES (124,'BCD','MH','01-FEB-2019');

    INSERT INTO CUSTOMER VALUES (125,'ABCD','AP','01-DEC-2001');
    INSERT INTO CUSTOMER VALUES (126,'AAAA','UP','01-DEC-2011');
    INSERT INTO CUSTOMER VALUES (127,'AAAB','UP','01-FEB-2011');
    INSERT INTO CUSTOMER VALUES (128,'AAC','CK','01-FEB-2015');
    INSERT INTO CUSTOMER VALUES (129,'POO','MH','04-NOV-2019');

    exec dbms_stats.gather_table_stats(USER,'CUSTOMER');

    SELECT TABLE_NAME,PARTITION_NAME, COMPOSITE, HIGH_VALUE,NUM_ROWS
        FROM USER_TAB_PARTITIONS WHERE TABLE_NAME='CUSTOMER';
/*
    TABLE_NAME                PARTITION_NAME             COM HIGH_VALUE                                                                      NUM_ROWS
------------------------------ ------------------------------ --- -------------------------------------------------------------------------------- ----------
CUSTOMER                WEST                    YES 'MH', 'GJ'                                                                        3
CUSTOMER                SOUTH                    YES 'TN', 'AP'                                                                        1
CUSTOMER                NORTH                    YES 'UP', 'HP'                                                                        2
CUSTOMER                UN_KNOWN                     YES DEFAULT                                                                           1
*/

exec dbms_stats.gather_table_stats(USER,'CUSTOMER');
    SELECT TABLE_NAME,PARTITION_NAME, SUBPARTITION_NAME, NUM_ROWS
        FROM USER_TAB_SUBPARTITIONS WHERE TABLE_NAME='CUSTOMER';

/*
TABLE_NAME                PARTITION_NAME             SUBPARTITION_NAME              NUM_ROWS
------------------------------ ------------------------------ ------------------------------ ----------
CUSTOMER                SOUTH                    SOUTH_CUST_SUB_ID_1
CUSTOMER                SOUTH                    SOUTH_CUST_SUB_ID_2
CUSTOMER                SOUTH                    SOUTH_CUST_SUB_ID_3
CUSTOMER                SOUTH                    SOUTH_CUST_SUB_ID_4
CUSTOMER                NORTH                    NORTH_CUST_SUB_ID_1
CUSTOMER                NORTH                    NORTH_CUST_SUB_ID_2
CUSTOMER                NORTH                    NORTH_CUST_SUB_ID_3
CUSTOMER                NORTH                    NORTH_CUST_SUB_ID_4
CUSTOMER                UN_KNOWN                     UN_KNOWN_CUST_SUB_ID_1
CUSTOMER                UN_KNOWN                     UN_KNOWN_CUST_SUB_ID_2
CUSTOMER                UN_KNOWN                     UN_KNOWN_CUST_SUB_ID_3
TABLE_NAME                PARTITION_NAME                 SUBPARTITION_NAME              NUM_ROWS
```

```
------------------------------ ------------------------------ ------------------------------ ----------
CUSTOMER                       UN_KNOWN                       UN_KNOWN_CUST_SUB_ID_4
CUSTOMER                       WEST                           WEST_CUST_SUB_ID_1
CUSTOMER                       WEST                           WEST_CUST_SUB_ID_2
CUSTOMER                       WEST                           WEST_CUST_SUB_ID_3
CUSTOMER                       WEST                           WEST_CUST_SUB_ID_4
16 rows selected.
*/
```