PRACTICAL 3

Aim:

To execute following data partitioning technique in data warehouse. Operations can be demonstrated on any schema.

a. Range Partitioning
b. List Partitioning
c. Hash Partitioning
d. Interval Partitioning
e. Reference Partitioning
f. Virtual Column based partitioning
g. Composite Partitioning

```
-- tablespace
CREATE TABLESPACE TSA1 DATAFILE 'C:\temp\tsa1.dbf' SIZE 10M;
CREATE TABLESPACE TSA2 DATAFILE 'C:\temp\tsa2.dbf' SIZE 10M;
CREATE TABLESPACE TSA3 DATAFILE 'C:\temp\tsa3.dbf' SIZE 10M;
CREATE TABLESPACE TSA4 DATAFILE 'C:\temp\tsa4.dbf' SIZE 10M;
```

QUERY 1: Write a query to create range portioned table:

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.

Each partition is given a name (sales_q1, sales_q2, ...), and each partition is contained in a separate tablespace (tsa, tsb, ...)

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('poonam_07','SALES');
        select partition_name,tablespace_name,high_value,num_rows from user_tab_partitions
where
                /*      no rows selected */

        select partition_name,tablespace_name,high_value,num_rows from user_tab_partitions
where
                /*
                PARTITION_NAME              TABLESPACE_NAME             HIGH_VALUE
                ---------------------------- ---------------------------- -------------------------------
                SALES_Q1                TSA1                    TO_DATE(' 2018-04-01 00:00:00',
                SALES_Q2                TSA2                    TO_DATE(' 2018-07-01 00:00:00',
                SALES_Q3                TSA3                    TO_DATE(' 2018-10-01 00:00:00',
                SALES_Q4                TSA4                    TO_DATE(' 2019-01-01 00:00:00',
                */

        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
                */
```

QUERY 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);

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

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 SALES1 PARTITION(SALES_Q1);
        /* no rows selected */

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
        */
```


QUERY 3: Create a table with list partition as follows:

Table having columns deptno, deptname, quarterly_sales and state.

Create partition on state: Northwest on OR and WA

Southwest on AZ, UT and NM ⍰ northeast on NY, VM and NJ
southeast on FL and GA ⍰ northcentral on SD and WI
southcentral on OK and TX

Add the following entries into the table and make conclusion to which partition the entry maps:

(10, 'accounting', 100, 'WA')
(20, 'R&D', 150, 'OR')
(30, 'sales', 100, 'FL')
(40, 'HR', 10, 'TX')
(50, 'systems engineering', 10, 'CA')

```
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
        */

ALTER TABLE SALES_BY_LIST ADD PARTITION Q1_NEW VALUES(DEFAULT);
INSERT INTO SALES_BY_LIST VALUES (50, 'SYSTEMS_ENG', 10, 'CA');

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
                */

        exec dbms_stats.gather_table_stats('poonam_07','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                   DEFAULT
1
                */
```

QUERY 4: Create a table with hash partition as follows: ⍰ Create table Emp with attributes empno, job, sal, deptno and perform hash partitioning on empno. Number of Partitions should be 5. Demonstarte using system defined and user defined partition concepts.

```
        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('poonam_07','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_P21
2
            EMPLOYEE_HASH           SYS_P22
2
            EMPLOYEE_HASH           SYS_P23
1
            EMPLOYEE_HASH           SYS_P24
0
            EMPLOYEE_HASH           SYS_P25
0
            */

        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('RAKSHIT_74','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
                */
```

QUERY 5: Create a multi-column range partitioned table as directed:

Create a table with the actual DATE information in three separate columns: year, month, and day. Also amount_ sold.

Create following partitions:

- o Before 2001: Less than jan 2001
- o Less than april 2001
- o Less than july 2001
- o Les than oct 2001
- o Less than jan 2002
- o Future with max incoming value

Insert values into table and show to which partition does the value belong.

- o (2001,3,17, 2000);
- o (2001,11,1, 5000);
- o (2002,1,1, 4000); Make conclusion for each result.

```
        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('poonam_07','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
                */
```

QUERY 6: Create a multicolumn partitioned table as directed:

Table supplier_parts, storing the information about which suppliers deliver which parts.

To distribute the data in equal-sized partitions, it is not sufficient to partition the table based on the supplier_id,because some suppliers might provide hundreds of thousands of parts, while others provide only a few specialty parts.
Instead, you partition the table on (supplier_id, partnum) to manually enforce equal-sized partitions.

Insert the following values(5,5, 1000);(5,150, 1000);(10,100, 1000);

```
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);

exec dbms_stats.gather_table_stats('poonam_07','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
        */
```


QUERY 7: Create interval partitioned table as directed:

Creates a table named- Sales consisting of four partitions,one for each quarter of sales.Each partition is given a name (sales_q1, sales_q2,..)

The columns for table must be prod_id, cust_id, promo_id, quantify sold,amount_sold – all in number format and month in number format

Perform interval partitioning on month and take interval of 01 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','02-JAN-2018','11',23,34.5);
INSERT INTO SALES_INT VALUES ('125','1234','22-DEC-2018','21',23,34.5);

select * from Sales_int;
    /*
    PROD_ID    CUST_ID TIME_ID     PROMO_ID   QTY_SOLD   AMT_SOLD
    ---------- ---------- --------- ---------- ---------- ----------
       123       1234 02-JAN-18        11         23       34.5
       125       1234 22-DEC-18        21         23       34.5
    */

INSERT INTO SALES_INT VALUES ('111','1234','25-Mar-2019','66',23,34.5);

select * from Sales_int;
    /*
    PROD_ID    CUST_ID TIME_ID     PROMO_ID   QTY_SOLD   AMT_SOLD
    ---------- ---------- --------- ---------- ---------- ----------
       123       1234 02-JAN-18        11         23       34.5
       125       1234 22-DEC-18        21         23       34.5
       111       1234 25-MAR-19        66         23       34.5
    */

exec dbms_stats.gather_table_stats('poonam_07','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             0
                SALES_INT                 SALES_Q3                  TO_DATE(' 2018-10-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIA             0
                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_P21                 TO_DATE(' 2019-04-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIA             1
                */

        INSERT INTO SALES_INT VALUES ('111','1234','25-Nov-2019','66',23,34.5);
        INSERT INTO SALES_INT VALUES ('111','1234','25-Oct-2019','66',23,34.5);
        INSERT INTO SALES_INT VALUES ('111','1234','25-Jan-2019','66',23,34.5);
        INSERT INTO SALES_INT VALUES ('111','1234','13-Jan-2019','66',23,34.5);

        exec dbms_stats.gather_table_stats('poonam_07','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                        0
                SALES_INT                 SALES_Q3             TO_DATE(' 2018-10-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIA                        0
                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_P24            TO_DATE(' 2019-02-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIA                        2
                SALES_INT                 SYS_P21            TO_DATE(' 2019-04-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIA                        1
                SALES_INT                 SYS_P23            TO_DATE(' 2019-11-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIA                        1
```

```
            SALES_INT                   SYS_P22               TO_DATE(' 2019-12-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIA                              1
            */
```

QUERY 8: Demonstrate reference partitioning as directed:
Create parent table Orders with the attributes order_id, order_date, customer_id, shipper_id.
Perform Range partitioning on Order Date. Take Range of 03 Months i.e. 01 quarter
Create child table order_items with attributes order_id, product_id, price and quantity.
Perform Reference partitioning on child table. ? Delete the created partitions.

```
        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-MAR-2018',34,89);
        INSERT INTO ORDERS VALUES (124,'15-NOV-2018',34,909);

        select * from orders;
            /*
             ORDER_ID ORDER_DAT    CUST_ID    SHIP_ID
            ---------- --------- ---------- ----------
                   123 12-MAR-18         34         89
                   124 15-NOV-18         34        909
            */
        INSERT INTO ORDER_ITEMS VALUES (111,123,456,78,90);
        INSERT INTO ORDER_ITEMS VALUES (112,124,456,78,90);

        select * from order_items;
                /*
                  ITEM_ID   ORDER_ID    PROD_ID      PRICE        QTY
                ---------- ---------- ---------- ---------- ----------
                       111        123        456         78         90
                       112        124        456         78         90
                */

        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_Q3;

        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_Q4
                ORDER_ITEMS                    ORDERS_Q1
                ORDER_ITEMS                    ORDERS_Q2
                ORDER_ITEMS                    ORDERS_Q4
                */

        ALTER TABLE ORDERS DROP PARTITION ORDERS_Q3;

        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_Q4
        ORDER_ITEMS                 ORDERS_Q1
        ORDER_ITEMS                 ORDERS_Q2
        ORDER_ITEMS                 ORDERS_Q4
        */
```

QUERY 9: Implement virtual column based partitioning as below:
Create table employee with attributes Emp_id, emp_name, fixed_salary, variable_salary. Generate Total salary as virtual colum.
Perform range partitioning on Total Salary with four partitions as below:

- o Partition P1 stores salary less than 25000
- o Partition P2 stores salary less than 50000
- o Partition P3 stores salary less than 75000
- o Partition P4 stores any salary above and equal to than 75000

```
    CREATE TABLE EMPLOYEE (
    EMP_ID NUMBER(4) PRIMARY KEY,
    EMP_NAME VARCHAR2(20),
    FIXED_SAL NUMBER(4),
    VARIABLE_SAL NUMBER(4),
    TOTAL_SAL NUMBER(6)
    GENERATED ALWAYS AS (
    FIXED_SAL + VARIABLE_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,FIXED_SAL,VARIABLE_SAL)
VALUES (124,'BBB',1000,1000);
    INSERT INTO EMPLOYEE (EMP_ID,EMP_NAME,FIXED_SAL,VARIABLE_SAL)
VALUES (123,'AAA',2000,9000);

    SELECT * FROM EMPLOYEE;
        /*
```

```
                    EMP_ID EMP_NAME             FIXED_SAL VARIABLE_SAL
TOTAL_SAL
          ---------- -------------------- ---------- ------------ ----------
                 124 BBB                  1000        1000      2000
                 123 AAA                  2000        9000     11000
          */
```

QUERY 10: Demonstrate Composite partitioning technique as directed
Implement range list partitioning for customer table having attributes cust_id, cust_name, cust_state, and time_id
o Perform range partitioning on time-id and list partitioning on state attributes. Also create maxvalue and default partition for
range and list partition respectively. o Partition definitions for range are as below:
Partition old should accept values less than 01-Jan-2005
Partition acquired should accept values less than 01-Jan-2010
Partition recent should accept values less than 01-Jan-2015
Partition unknown should accept values greater than 01-Jan-2015
o Partition definitions for list are as below:
Partition west should accept values (‘MH’, ‘GJ’)
Partition south should accept values (‘TN’, ‘AP’)
Partition north should accept values (‘UP’, ‘HP’)
Partition unknown should accept any other state.

```
-- RANGE LIST PARTITION
        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,'AAA','MH','01-JAN-2011');
        INSERT INTO CUSTOMER VALUES (124,'BBB','MH','01-FEB-2019');

        SELECT * FROM CUSTOMER;
                /*
                  CUST_ID CUST_NAME          CUST_STATE          TIME_ID
                ---------- -------------------- -------------------- ---------
                           123 AAA              MH               01-JAN-11
                           124 BBB              MH               01-FEB-19
                */

        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,'CCC' ,'MH','04-NOV-2019');

        SELECT * FROM CUSTOMER;
                /*
                CUST_ID CUST_NAME          CUST_STATE          TIME_ID
                ---------- -------------------- -------------------- ---------
                  125 ABCD             AP               01-DEC-01
                  123 AAA             MH               01-JAN-11
                  126 AAAA             UP               01-DEC-11
                  127 AAAB             UP               01-FEB-11
                  124 BBB             MH               01-FEB-19
                  129 CCC             MH               04-NOV-19
                  128 AAC             CK              01-FEB-15
                */

        exec dbms_stats.gather_table_stats('poonam_07','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 * from customer subpartition(CUST_RG_4_WEST);
        /*
        CUST_ID CUST_NAME          CUST_STATE         TIME_ID
        ---------- -------------------- -------------------- ---------
          124 BBB                MH                01-FEB-19
          129 CCC                MH                04-NOV-19
        */

    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
        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
        */

    select * from customer subpartition(CUST_RG_4_NORTH);
        /* no rows selected */

    select * from customer subpartition(CUST_RG_4_SOUTH);
        /* no rows selected */

    select * from customer subpartition(CUST_RG_4_WEST);
        /*
        CUST_ID CUST_NAME          CUST_STATE          TIME_ID
        ---------- -------------------- -------------------- ---------
          124 BBB                MH                  01-FEB-19
          129 CCC                MH                  04-NOV-19
        */

------------------------------------------------------------------------------------------------------------
-- QUERY 11:( RANGE on TIME_ID - RANGE on 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 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');

	exec dbms_stats.gather_table_stats('poonam_07','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
		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
		*/

----------------------------------------------------------------------------------------------------------------------
-- QUERY 12:(RANGE on TIME_ID - HASH on 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');

        exec dbms_stats.gather_table_stats('poonam_07','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
                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 on CUST_STATE - HASH on 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 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');

	exec dbms_stats.gather_table_stats('poonam_07','CUSTOMER');

	SELECT TABLE_NAME,PARTITION_NAME, COMPOSITE,
HIGH_VALUE,NUM_ROWSF ROM 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('poonam_07','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
                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 on CUST_STATE - LIST on 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');

        exec dbms_stats.gather_table_stats('poonam_07','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('poonam_07','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
            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 on CUST_STATE - RANGE on 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');

        exec dbms_stats.gather_table_stats('poonam_07','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('poonam_07','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
                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
                */
```