....prac3.................................

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 TABLESPACE ABC1 DATAFILE 'F:/APP/DELL/ORADATA/ORCL/abc1.dbf' SIZE 10M;
Tablespace created.
CREATE TABLESPACE ABC2 DATAFILE 'F:/APP/DELL/ORADATA/ORCL/abc2.dbf' SIZE 10M;
Tablespace created.
CREATE TABLESPACE ABC3 DATAFILE 'F:/APP/DELL/ORADATA/ORCL/abc3.dbf' SIZE 10M;
Tablespace created.
CREATE TABLESPACE ABC4 DATAFILE 'F:/APP/DELL/ORADATA/ORCL/abc4.dbf' SIZE 10M;
Tablespace created.

CREATE TABLE SALES(
PROD_ID NUMBER(5) NOT NULL,
CUST_ID NUMBER NOT NULL,
PROMO_ID NUMBER(5) NOT NULL,
QUANTITY_SOLD NUMBER(5) NOT NULL,
AMOUNT_SOLD NUMBER(5) NOT NULL,
TIME_ID DATE NOT NULL)
PARTITION BY RANGE(TIME_ID)
(PARTITION SALES_Q1 VALUES LESS THAN('01-APR-2017') TABLESPACE ABC1,
PARTITION SALES_Q2 VALUES LESS THAN('01-JUL-2017') TABLESPACE ABC2,
PARTITION SALES_Q3 VALUES LESS THAN('01-OCT-2017') TABLESPACE ABC3,
PARTITION SALES_Q4 VALUES LESS THAN('01-JAN-2018') TABLESPACE ABC4
);

Table created.

INSERT INTO SALES VALUES(10001,10001,10001,25,30000,'20-JAN-2017');
1 row created.
INSERT INTO SALES VALUES(10002,10002,10002,55,40000,'20-MAY-2017');
1 row created.
INSERT INTO SALES VALUES(10003,10003,10003,50,80000,'20-JUL-2017');
1 row created.
INSERT INTO SALES VALUES(10004,10004,10004,100,90000,'20-DEC-2017');
1 row created.

EXEC dbms_stats.gather_table_stats('RCOEM','SALES');

PL/SQL procedure successfully completed.

SELECT TABLESPACE_NAME,PARTITION_NAME,NUM_ROWS FROM USER_TAB_PARTITIONS WHERE
TABLE_NAME='SALES';

TABLESPACE_NAME                PARTITION_NAME                   NUM_ROWS
------------------------------ ------------------------------ ----------
ABC1                           SALES_Q1                                1
ABC2                           SALES_Q2                                1
ABC3                           SALES_Q3                                1
ABC4                           SALES_Q4                                1
```

QUERY 2:

Create the same table as in Q1. With a different name with ENABLE ROW MOVEMENT.

```
CREATE TABLE SALES_2(
PROD_ID NUMBER(5) NOT NULL,
CUST_ID NUMBER NOT NULL,
PROMO_ID NUMBER(5) NOT NULL,
QUANTITY_SOLD NUMBER(5) NOT NULL,
AMOUNT_SOLD NUMBER(5) NOT NULL,
TIME_ID DATE NOT NULL)
PARTITION BY RANGE(TIME_ID)
(PARTITION SALES_Q1 VALUES LESS THAN('01-APR-2017') TABLESPACE ABC1,
PARTITION SALES_Q2 VALUES LESS THAN('01-JUL-2017') TABLESPACE ABC2,
PARTITION SALES_Q3 VALUES LESS THAN('01-OCT-2017') TABLESPACE ABC3,
PARTITION SALES_Q4 VALUES LESS THAN('01-JAN-2018') TABLESPACE ABC4
)
ENABLE ROW MOVEMENT;

Table created.

INSERT INTO SALES_2 VALUES(10001,10001,10001,25,30000,'20-JAN-2017');
1 row created.
INSERT INTO SALES_2 VALUES(10002,10002,10002,55,40000,'20-MAY-2017');
1 row created.
INSERT INTO SALES_2 VALUES(10003,10003,10003,50,80000,'20-JUL-2017');
1 row created.
INSERT INTO SALES_2 VALUES(10004,10004,10004,100,90000,'20-DEC-2017');
1 row created.

UPDATE SALES_2 SET TIME_ID='01-FEB-2017' WHERE PROD_ID=10003;
1 row updated.

EXEC dbms_stats.gather_table_stats('RCOEM','SALES_2');
PL/SQL procedure successfully completed.
SELECT TABLESPACE_NAME,PARTITION_NAME,NUM_ROWS FROM USER_TAB_PARTITIONS WHERE
TABLE_NAME='SALES_2';

TABLESPACE_NAME                PARTITION_NAME                   NUM_ROWS
------------------------------ ------------------------------ ----------
ABC1                           SALES_Q1                                2
ABC2                           SALES_Q2                                1
ABC3                           SALES_Q3                                0
ABC4                           SALES_Q4                                1
```

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 DEPT(
DEPTNO NUMBER(3) NOT NULL,
DEPTNAME VARCHAR2(20) NOT NULL,
QUATERLY_SALES NUMBER(5) NOT NULL,
STATE VARCHAR2(2) NOT NULL)
PARTITION BY LIST(STATE)
(
PARTITION NORTHWEST VALUES('OR','WA'),
PARTITION SOUTHWEST VALUES('AZ','UT','NM'),
PARTITION NORTHEAST VALUES('NY','VM','NJ'),
PARTITION SOUTHEAST VALUES('FL','GA'),
PARTITION NORTHCENTRAL VALUES('SD','WI'),
PARTITION SOUTHCENTRAL VALUES('OK','TX')
);

Table created.

INSERT INTO DEPT VALUES(10,'ACCOUNTING',100,'WA');
1 row created.
INSERT INTO DEPT VALUES(20,'R&D',150,'OR');
1 row created.
INSERT INTO DEPT VALUES(30,'SALES',100,'FL');
1 row created.
INSERT INTO DEPT VALUES(40,'HR',10,'TX');
1 row created.
INSERT INTO DEPT VALUES(50,'SYSTEM ENGINEERING',10,'CA');
INSERT INTO DEPT VALUES(50,'SYSTEM ENGINEERING',10,'CA')
            *
ERROR at line 1:
ORA-14400: inserted partition key does not map to any partition

ALTER TABLE DEPT ADD PARTITION DEF_STATE VALUES(DEFAULT);
Table altered.

INSERT INTO DEPT VALUES(50,'SYSTEM ENGINEERING',10,'CA');
1 row created.

SELECT * FROM DEPT;

    DEPTNO DEPTNAME             QUATERLY_SALES ST
---------- -------------------- -------------- --
        10 ACCOUNTING                      100 WA
        20 R                               150 OR
        30 SALES                           100 FL
        40 HR                               10 TX
        50 SYSTEM ENGINEERING               10 CA


EXEC dbms_stats.gather_table_stats('RCOEM','DEPT');
PL/SQL procedure successfully completed.

SELECT TABLESPACE_NAME,PARTITION_NAME,NUM_ROWS FROM USER_TAB_PARTITIONS WHERE
TABLE_NAME='DEPT';
TABLESPACE_NAME        PARTITION_NAME               NUM_ROWS
------------------------------  ------------------------------           ----------
USERS                              NORTHWEST                                   2
USERS                              SOUTHWEST                                   0
USERS                              NORTHEAST                                    0
USERS                              SOUTHEAST                                    1
USERS                              NORTHCENTRAL                            0
USERS                              SOUTHCENTRAL                             1
USERS                              DEF_STATE                                         1

7 rows selected.
```

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 EMP(
EMPNO NUMBER(5) NOT NULL,
JOB VARCHAR2(10) NOT NULL,
SAL NUMBER(5) NOT NULL,
DEPTNO NUMBER(5) NOT NULL)
PARTITION BY HASH(EMPNO)
PARTITIONS 5;
Table created.

INSERT INTO EMP VALUES(110,'QA',35000,3);
```

```
1 row created.
INSERT INTO EMP VALUES(210,'DEV',35000,4);
1 row created.
INSERT INTO EMP VALUES(250,'QA',45000,3);
1 row created.
INSERT INTO EMP VALUES(310,'CEO',90000,1);
1 row created.

EXEC dbms_stats.gather_table_stats('RCOEM','EMP');
PL/SQL procedure successfully completed.

 SELECT TABLESPACE_NAME,PARTITION_NAME,NUM_ROWS FROM USER_TAB_PARTITIONS WHERE
 TABLE_NAME='EMP';

TABLESPACE_NAME                PARTITION_NAME                   NUM_ROWS
------------------------------ ------------------------------ ----------
USERS                          SYS_P26                                 1
USERS                          SYS_P27                                 2
USERS                          SYS_P28                                 1
USERS                          SYS_P29                                 0
USERS                          SYS_P30                                 0


CREATE TABLE EMP_2(
EMPNO NUMBER(5) NOT NULL,
JOB VARCHAR2(10) NOT NULL,
SAL NUMBER(5) NOT NULL,
DEPTNO NUMBER(5) NOT NULL)
PARTITION BY HASH(EMPNO)
(PARTITION H1,
PARTITION H2,
PARTITION H3,
PARTITION H4,
PARTITION H5);
Table created.

INSERT INTO EMP_2 VALUES(110,'QA',35000,3);
1 row created.
INSERT INTO EMP_2 VALUES(210,'DEV',35000,4);
1 row created.
INSERT INTO EMP_2 VALUES(250,'QA',45000,3);
1 row created.
INSERT INTO EMP_2 VALUES(310,'CEO',90000,1);
1 row created.

EXEC dbms_stats.gather_table_stats('RCOEM','EMP_2');
PL/SQL procedure successfully completed.

SELECT TABLESPACE_NAME,PARTITION_NAME,NUM_ROWS FROM USER_TAB_PARTITIONS WHERE
TABLE_NAME='EMP_2';

TABLESPACE_NAME     PARTITION_NAME       NUM_ROWS
------------------------------ ------------------------------ ----------
USERS                          H1                                            1
USERS                          H2                                      2
USERS                          H3                                            1
USERS                          H4                                            0
USERS                          H5                                            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:
oBefore 2001: Less than jan 2001
oLess than april 2001
oLess than july 2001
oLes than oct 2001
oLess than jan 2002

oFuture 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.

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_PARTS
(
SUPPLIER_ID NUMBER(5) ,
PARTNUM NUMBER(5) ,
PRICE NUMBER(5) )
PARTITION BY RANGE(SUPPLIER_ID,PARTNUM)(
PARTITION P1 VALUES LESS THAN(5,100),
PARTITION P2 VALUES LESS THAN(5,150),
PARTITION P3 VALUES LESS THAN(5,200),
PARTITION P4 VALUES LESS THAN(10,100),
PARTITION P5 VALUES LESS THAN(10,150),
PARTITION P6 VALUES LESS THAN(MAXVALUE,MAXVALUE)
);
Table created.
INSERT INTO SUPPLIER_PARTS VALUES(5,5,1000);
1 row created.
INSERT INTO SUPPLIER_PARTS VALUES(5,150,1000);
1 row created.
INSERT INTO SUPPLIER_PARTS VALUES(10,100,1000);
1 row created.
EXEC dbms_stats.gather_table_stats('MANJARI','SUPPLIER_PARTS');
PL/SQL procedure successfully completed.

SELECT TABLESPACE_NAME,PARTITION_NAME,NUM_ROWS FROM USER_TAB_PARTITIONS WHERE
TABLE_NAME='SUPPLIER_PARTS';

TABLESPACE_NAME        PARTITION_NAME      NUM_ROWS
------------------------------ ------------------------------   ----------
USERS                            P1                                  2
USERS                            P2                                  0
USERS                            P3                                  2
USERS                            P4                                  0
USERS                            P5                                  2
USERS                            P6                                  0

6 rows selected.
```

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(
  PROD_ID NUMBER(3) NOT NULL,
  CUST_ID NUMBER(3) NOT NULL,
  PROMO_ID NUMBER(3) NOT NULL,
  QUANTITY_SOLD NUMBER(3) NOT NULL,
  AMOUNT_SOLD NUMBER(3) NOT NULL,
  MONTH NUMBER(2) NOT NULL)
  PARTITION BY RANGE(MONTH)
  INTERVAL(1)(
  PARTITION SALES_Q1 VALUES LESS THAN(4),
  PARTITION SALES_Q2 VALUES LESS THAN(7),
  PARTITION SALES_Q3 VALUES LESS THAN(10)
  );
Table created.
  INSERT INTO SALES VALUES(101,101,101,3,300,3);
1 row created.
  INSERT INTO SALES VALUES(102,102,102,4,400,4);
1 row created.
  INSERT INTO SALES  VALUES(103,103,103,7,700,6);
1 row created.
  INSERT INTO SALES  VALUES(104,104,104,1,100,11);
1 row created.

EXEC dbms_stats.gather_table_stats('MANJARI','SALES');
PL/SQL procedure successfully completed.
SELECT TABLESPACE_NAME,PARTITION_NAME,NUM_ROWS FROM USER_TAB_PARTITIONS WHERE
TABLE_NAME='SALES';
TABLESPACE_NAME     PARTITION_NAME       NUM_ROWS
------------------------------ ------------------------------   ----------
USERS                            SALES_Q1                           1
USERS                            SALES_Q2                           2
USERS                            SALES_Q3                           0
USERS                            SYS_P41                              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(3) PRIMARY KEY,
ORDER_DATE DATE NOT NULL,
CUSTOMER_ID NUMBER(3) NOT NULL,
SHIPPER_ID NUMBER(3) NOT NULL)
PARTITION BY RANGE(ORDER_DATE)
(PARTITION ORDERS_Q1 VALUES LESS THAN('01-APR-2017'),
PARTITION ORDERS_Q2 VALUES LESS THAN('01-JUL-2017') ,
PARTITION ORDERS_Q3 VALUES LESS THAN('01-OCT-2017'),
PARTITION ORDERS_Q4 VALUES LESS THAN('01-JAN-2018')
);
Table created.

CREATE TABLE ORDER_ITEMS
( ORDER_ID NUMBER(3) NOT NULL,
PRODUCT_ID NUMBER(3) NOT NULL,
PRICE NUMBER(5) NOT NULL,
QUANTITY NUMBER(3) NOT NULL,
CONSTRAINT ORDERS_ORDER_ID_FK FOREIGN KEY(ORDER_ID) REFERENCES ORDERS)
PARTITION BY REFERENCE(ORDERS_ORDER_ID_FK);
Table created.

INSERT INTO ORDERS VALUES(111,'01-FEB-2017',101,101);
1 row created.
INSERT INTO ORDERS VALUES(222,'01-JUN-2017',102,102);
1 row created.
```

```
INSERT INTO ORDERS VALUES(333,'01-AUG-2017',103,103);
1 row created.
INSERT INTO ORDERS VALUES(444,'01-NOV-2017',104,104);
1 row created.

INSERT INTO ORDER_ITEMS VALUES(111,202,10000,3);
1 row created.
INSERT INTO ORDER_ITEMS VALUES(222,204,20000,6);
1 row created.
INSERT INTO ORDER_ITEMS VALUES(333,206,15000,2);
1 row created.
INSERT INTO ORDER_ITEMS VALUES(444,208,45000,1);
1 row created.

SELECT * FROM ORDERS PARTITION(ORDERS_Q1);

  ORDER_ID ORDER_DAT CUSTOMER_ID SHIPPER_ID
    ---------- --------- ----------- ----------
         111 01-FEB-17         101        101

SELECT * FROM ORDER_ITEMS PARTITION(ORDERS_Q1);

  ORDER_ID PRODUCT_ID  PRICE   QUANTITY
      ---------- ---------- ---------- ----------
          111        202      10000          3

ALTER TABLE ORDERS DROP PARTITION(ORDERS_Q1);
Table altered.
```

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

```
CREATE TABLE EMPLOYEE(
EMP_ID NUMBER(3) NOT NULL,
EMP_NAME VARCHAR2(20) NOT NULL,
VARIABLE_SALARY NUMBER(5) NOT NULL,
FIXED_SALARY NUMBER(5) NOT NULL,
TOTAL NUMBER(7)
GENERATED ALWAYS AS
( FIXED_SALARY+VARIABLE_SALARY) VIRTUAL
) PARTITION BY RANGE(TOTAL)(
PARTITION P1 VALUES LESS THAN (25000),
PARTITION P2 VALUES LESS THAN (50000),
PARTITION P3 VALUES LESS THAN (75000),
PARTITION P4 VALUES LESS THAN (MAXVALUE)
);

INSERT INTO EMPLOYEE(EMP_ID,EMP_NAME,VARIABLE_SALARY,FIXED_SALARY) VALUES(111,'Jiawei Han',30000,25000);
INSERT INTO EMPLOYEE(EMP_ID,EMP_NAME,VARIABLE_SALARY,FIXED_SALARY) VALUES(222,'Will Smith',40000,10000);
INSERT INTO EMPLOYEE(EMP_ID,EMP_NAME,VARIABLE_SALARY,FIXED_SALARY) VALUES(333,'Jiawei Han',60000,45000);

SELECT * FROM EMPLOYEE;

    EMP_ID EMP_NAME   VARIABLE_SALARY FIXED_SALARY  TOTAL
     ----------  --------------------   ---------------   ------------    ----------
     111         Jiawei Han                  30000           25000           55000
     222        Will Smith                   40000           10000           50000
     333        Jiawei Han                   60000           45000           105000
```

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
oPerform range partitioning on time-id and list partitioning on state attributes. Also create maxvalue and default partition for range and list partition respectively.
oPartition 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
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.

```
CREATE TABLE CUSTOMER_RANGE_LIST(
CUST_ID NUMBER(3) NOT NULL,
CUST_NAME VARCHAR2(20) NOT NULL,
CUST_STATE VARCHAR2(20) NOT NULL,
TIME_ID DATE NOT NULL)
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 UNKNOWN VALUES(DEFAULT))
(PARTITION OLD VALUES LESS THAN('01-JAN-2005'),
PARTITION ACQUIRED VALUES LESS THAN('01-JAN-2010'),
PARTITION RECENT VALUES LESS THAN('01-JAN-2015'),
PARTITION UNKOWN VALUES LESS THAN(MAXVALUE)
);
Table created.

INSERT INTO CUSTOMER_RANGE_LIST VALUES(111,'WILL SMITH','MH','01-AUG-2005');
1 row created.
INSERT INTO CUSTOMER_RANGE_LIST VALUES(222,'SRK','TN','01-AUG-2010');
1 row created.
INSERT INTO CUSTOMER_RANGE_LIST VALUES(333,'SALMAN','HP','01-AUG-2015');
1 row created.
INSERT INTO CUSTOMER_RANGE_LIST VALUES(444,'AAMIR','MP','01-AUG-2018');
1 row created.

EXEC dbms_stats.gather_table_stats('MANJARI','CUSTOMER_RANGE_LIST');
PL/SQL procedure successfully completed.

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

PARTITION_NAME                 SUBPARTITION_NAME                NUM_ROWS
------------------------------ ------------------------------ ----------
OLD                            OLD_WEST                                0
OLD                            OLD_SOUTH                               0
OLD                            OLD_NORTH                               0
OLD                            OLD_UNKNOWN                             0
ACQUIRED                       ACQUIRED_WEST                           1
ACQUIRED                       ACQUIRED_SOUTH                          0
ACQUIRED                       ACQUIRED_NORTH                          0
ACQUIRED                       ACQUIRED_UNKNOWN                        0
RECENT                         RECENT_WEST                             0
RECENT                         RECENT_SOUTH                            1
RECENT                         RECENT_NORTH                            0

PARTITION_NAME                 SUBPARTITION_NAME                NUM_ROWS
------------------------------ ------------------------------ ----------
RECENT                         RECENT_UNKNOWN                          0
UNKOWN                         UNKOWN_WEST                             0
UNKOWN                          UNKOWN_SOUTH                             0
UNKOWN                          UNKOWN_NORTH                             1
UNKOWN                          UNKOWN_UNKNOWN                           1

16 rows selected.

SELECT * FROM CUSTOMER_RANGE_LIST SUBPARTITION(ACQUIRED_WEST);
CUST_ID CUST_NAME            CUST_STATE           TIME_ID
---------- -------------------- -------------------- ---------
       111 WILL SMITH           MH                   01-AUG-05


....................................prac3_+5 problem
statment................................
11)RANGE-HASH
 SQL> CREATE TABLE composite_rng_hash(
 2 cust_id NUMBER(10),
 3 cust_name VARCHAR2(25),
 4 cust_state VARCHAR2(2),
 5 amt_sold VARCHAR2(2),
 6 time_id DATE)
 7 PARTITION BY RANGE(time_id)
 8 SUBPARTITION BY HASH(cust_id)
 9 SUBPARTITION TEMPLATE(
 10 SUBPARTITION h1,
 11 SUBPARTITION h2,
 12 SUBPARTITION h3)
 13 (PARTITION YEAR_2006 VALUES LESS THAN (TO_DATE('01-APR-2006','DD-MON-YYYY')),
 14 PARTITION YEAR_2007 VALUES LESS THAN(TO_DATE('01-APR-2007','DD-MON-YYYY')),
 15 PARTITION YEAR_2008 VALUES LESS THAN(TO_DATE('01-APR-2008','DD-MON-YYYY'))
 16 );

 Table created.

 SQL> DESC composite_rng_hash;
 Name Null? Type
 ----------------------------------------- -------- ----------------------------
 CUST_ID NUMBER(10)
 CUST_NAME VARCHAR2(25)
 CUST_STATE VARCHAR2(2)
 AMT_SOLD VARCHAR2(2)
 TIME_ID DATE


 SQL> insert into composite_rng_hash values(11,'cse','lp',21,'11-feb-2008');

 1 row created.

 SQL> SELECT partition_name, subpartition_name, num_rows
 2 FROM user_tab_subpartitions where table_name='COMPOSITE_RNG_HASH';

 PARTITION_NAME SUBPARTITION_NAME NUM_ROWS
 ------------------------------ ------------------------------ ----------
 YEAR_2006 YEAR_2006_H1
 YEAR_2006 YEAR_2006_H2
 YEAR_2006 YEAR_2006_H3
 YEAR_2007 YEAR_2007_H1
 YEAR_2007 YEAR_2007_H2
 YEAR_2007 YEAR_2007_H3
 YEAR_2008 YEAR_2008_H1
 YEAR_2008 YEAR_2008_H2
 YEAR_2008 YEAR_2008_H3

 9 rows selected.

 select * from composite_rng_hash subpartition(YEAR_2008_h1);
 CUST_ID CUST_NAME CU AM TIME_ID
 ---------- ------------------------- -- -- ---------
 11 cse lp 21 11-FEB-08
```

```
SQL> select * from composite_rng_hash subpartition(YEAR_2008_h2);

no rows selected

SQL> select * from composite_rng_hash subpartition(YEAR_2008_h3);

no rows selected


12)RANGE-RANGE
CREATE TABLE composite_rng_rng (
cust_id NUMBER(10),
cust_name VARCHAR2(25),
cust_state VARCHAR2(2),
amt_sold VARCHAR2(2),
time_id DATE)
PARTITION BY RANGE(time_id)
SUBPARTITION BY RANGE (cust_id)
SUBPARTITION TEMPLATE(
SUBPARTITION original VALUES LESS THAN (1001),
SUBPARTITION acquired VALUES LESS THAN (8001),
SUBPARTITION recent VALUES LESS THAN (MAXVALUE))
(PARTITION YEAR_2006 VALUES LESS THAN (TO_DATE('01-APR-2006','DD-MON-YYYY')),
PARTITION YEAR_2007 VALUES LESS THAN(TO_DATE('01-APR-2007','DD-MON-YYYY')),
PARTITION YEAR_2008 VALUES LESS THAN(TO_DATE('01-APR-2008','DD-MON-YYYY'))
);
SQL> desc composite_rng_rng;
Name Null? Type
----------------------------------------- -------- ----------------------------
CUST_ID NUMBER(10)
CUST_NAME VARCHAR2(25)
CUST_STATE VARCHAR2(2)
AMT_SOLD VARCHAR2(2)
TIME_ID DATE

SQL> insert into composite_rng_rng values(11,'cse','OR',21,'11-feb-2007');

1 row created.

SQL> insert into composite_rng_rng values(11,'cse','OR',21,'11-feb-2008');

1 row created.

SQL> SELECT partition_name, subpartition_name, num_rows
2 FROM user_tab_subpartitions where table_name='COMPOSITE_RNG_RNG';

PARTITION_NAME SUBPARTITION_NAME NUM_ROWS
------------------------------ ------------------------------ ----------
YEAR_2006 YEAR_2006_ORIGINAL
YEAR_2006 YEAR_2006_ACQUIRED
YEAR_2006 YEAR_2006_RECENT
YEAR_2007 YEAR_2007_ORIGINAL
YEAR_2007 YEAR_2007_ACQUIRED
YEAR_2007 YEAR_2007_RECENT
YEAR_2008 YEAR_2008_ORIGINAL
YEAR_2008 YEAR_2008_ACQUIRED
YEAR_2008 YEAR_2008_RECENT

9 rows selected.

select * from composite_rng_rng subpartition(YEAR_2008_original);
CUST_ID CUST_NAME CU AM TIME_ID
---------- ------------------------- -- -- ---------
11 cse OR 21 11-FEB-08

13)LIST-HASH
SQL> CREATE TABLE composite_list_hash (
2 cust_id NUMBER(10),
3 cust_name VARCHAR2(25),
4 cust_state VARCHAR2(2),
5 amt_sold VARCHAR2(2),
6 time_id DATE)
7 PARTITION BY LIST(cust_state)
8 SUBPARTITION BY HASH (cust_id)
9 SUBPARTITION TEMPLATE(
10 SUBPARTITION h1,
11 SUBPARTITION h2,
12 SUBPARTITION h3)
13 (PARTITION west VALUES ('OR', 'WA'),
14 PARTITION east VALUES ('NY', 'CT'),
15 PARTITION cent VALUES ('IL', 'MN')
16 );

Table created.

SQL> desc composite_list_hash;
Name Null? Type
----------------------------------------- -------- ----------------------------
CUST_ID NUMBER(10)
CUST_NAME VARCHAR2(25)
CUST_STATE VARCHAR2(2)
AMT_SOLD VARCHAR2(2)
TIME_ID DATE

SQL> SELECT partition_name, subpartition_name, num_rows
2 FROM user_tab_subpartitions where table_name='COMPOSITE_LIST_HASH';

PARTITION_NAME SUBPARTITION_NAME NUM_ROWS
------------------------------ ------------------------------ ----------
WEST WEST_H1
WEST WEST_H2
WEST WEST_H3
EAST EAST_H1
EAST EAST_H2
EAST EAST_H3
CENT CENT_H1
CENT CENT_H2
CENT CENT_H3

9 rows selected.
SQL> insert into composite_list_hash values(2,'MEC','NY',22,'10-feb-2018');

1 row created.
SQL> insert into composite_list_hash values(2,'CSE','IL',22,'10-JAN-2018');

1 row created.

select * from composite_list_hash subpartition(west_h1);

SQL> select * from composite_list_hash subpartition(east_h1);

no rows selected

SQL> select * from composite_list_hash subpartition(east_h2);

no rows selected

SQL> select * from composite_list_hash subpartition(east_h3);

CUST_ID CUST_NAME CU AM TIME_ID
---------- ------------------------- -- -- ---------
2 MEC NY 22 10-FEB-18

14)LIST-LIST
SQL> CREATE TABLE composite_list_list(
2 cust_id NUMBER(10),
3 cust_name VARCHAR2(25),
4 cust_state VARCHAR2(2),
5 amt_sold VARCHAR2(2),
6 time_id DATE)
7 PARTITION BY LIST(cust_state)
8 SUBPARTITION BY LIST(cust_id)
9 SUBPARTITION TEMPLATE
10 (SUBPARTITION original VALUES(1001),
11 SUBPARTITION acquired VALUES (8001),
12 SUBPARTITION recent VALUES (default))
13 (PARTITION west VALUES ('OR', 'WA'),
14 PARTITION east VALUES ('NY', 'CT'),
15 PARTITION cent VALUES ('IL', 'MN')
16 );

Table created.

SQL> desc composite_list_list;
Name Null? Type
----------------------------------------- -------- ----------------------------
CUST_ID NUMBER(10)
CUST_NAME VARCHAR2(25)
CUST_STATE VARCHAR2(2)
AMT_SOLD VARCHAR2(2)
TIME_ID DATE
SQL> SELECT partition_name, subpartition_name, num_rows
2 FROM user_tab_subpartitions where table_name='COMPOSITE_LIST_LIST';

PARTITION_NAME SUBPARTITION_NAME NUM_ROWS
------------------------------ ------------------------------ ----------
WEST WEST_ORIGINAL
WEST WEST_ACQUIRED
WEST WEST_RECENT
EAST EAST_ORIGINAL
EAST EAST_ACQUIRED
EAST EAST_RECENT
CENT CENT_ORIGINAL
CENT CENT_ACQUIRED
CENT CENT_RECENT

9 rows selected.
SQL> insert into composite_list_list values (21,'IND','IL',22,'10-feb-2019');

1 row created.

SQL> insert into composite_list_list values(21,'IND','IL',32,'10-APR-2020');

1 row created.

SQL> select * from composite_list_list subpartition(cent_recent);

CUST_ID CUST_NAME CU AM TIME_ID
---------- ------------------------- -- -- ---------
21 IND IL 22 10-FEB-19
21 IND IL 32 10-APR-20

15)LIST-RANGE
SQL> CREATE TABLE composite_list_rng (
2 cust_id NUMBER(10),
3 cust_name VARCHAR2(25),
4 cust_state VARCHAR2(2),
5 amt_sold VARCHAR2(2),
6 time_id DATE)
7 PARTITION BY LIST(cust_state)
8 SUBPARTITION BY RANGE (cust_id)
9 SUBPARTITION TEMPLATE(
10 SUBPARTITION original VALUES LESS THAN (1001),
11 SUBPARTITION acquired VALUES LESS THAN (8001),
12 SUBPARTITION recent VALUES LESS THAN (MAXVALUE))
13 (PARTITION west VALUES ('OR', 'WA'),
 14 PARTITION east VALUES ('NY', 'CT'),
 15 PARTITION cent VALUES ('IL', 'MN')
 16 );

 Table created.

 SQL> desc composite_list_rng;
 Name Null? Type
 ----------------------------------------- -------- ----------------------------
 CUST_ID NUMBER(10)
 CUST_NAME VARCHAR2(25)
 CUST_STATE VARCHAR2(2)
 AMT_SOLD VARCHAR2(2)
 TIME_ID DATE

 SQL> SELECT partition_name, subpartition_name, num_rows
 FROM user_tab_subpartitions where table_name='COMPOSITE_LIST_RNG';

 PARTITION_NAME SUBPARTITION_NAME NUM_ROWS
 ------------------------------ ------------------------------ ----------
 CENT CENT_ORIGINAL
 CENT CENT_ACQUIRED
 CENT CENT_RECENT
 EAST EAST_ORIGINAL
 EAST EAST_ACQUIRED
 EAST EAST_RECENT
 WEST WEST_ORIGINAL
 WEST WEST_ACQUIRED
 WEST WEST_RECENT

 9 rows selected.
SQL> insert into composite_list_rng values(1,'cse','OR',2,'10-feb-2018');
 1 row created.

 SQL> insert into composite_list_rng values(2,'MEC','OR',22,'10-feb-2018');
 1 row created.

 SQL> select * from composite_list_rng partition(west);
 CUST_ID CUST_NAME CU AM TIME_ID
 ---------- ------------------------- -- -- ---------
 1 cse OR 2 10-FEB-18
 2 MEC OR 22 10-FEB-18

 SQL> select * from composite_list_rng subpartition(west_original);

 CUST_ID CUST_NAME CU AM TIME_ID
 ---------- ------------------------- -- -- ---------
 1 cse OR 2 10-FEB-18
 2 MEC OR 22 10-FEB-18
```