```
--1. Write a query to create range partioned table:
--➢ Creates a table named- Sales consisting of four partitions, c
--sales. The column sale_date are the partitioning columns, while
--constitute the partitioning key of a specific row.
--➢ Each partition is given a name (sales_q1, sales_q2, ...), and
--contained in a separate tablespace (tsa, tsb, ...)
--➢ The columns for table must be prod_id, cust_id, promo_id, qua
--- all in number format and sale_date.-
create table Sales(
prod_id number(5),
cust_id number(5),
promo_id number(5),
quantiy_sold number(10),
amount_sold number(20),
sale_date date )
partition by range(sale_date)(
partition p1 values less than (to_date('01-04-2000','DD-MM-YYYY'))t
partition p2 values less than (to_date('01-07-2000','DD-MM-YYYY'))t
partition p3 values less than (to_date('01-10-2000','DD-MM-YYYY'))t
partition p4 values less than (to_date('01-01-2001','DD-MM-YYYY'))t
);

create tablespace tsa datafile
'C:\Users\acer\Desktop\tablespaces\tsa.dbf'size 10M;
create tablespace tsb datafile
'C:\Users\acer\Desktop\tablespaces\tsb.dbf'size 10M;
create tablespace tsc datafile
'C:\Users\acer\Desktop\tablespaces\tsc.dbf'size 10M;
create tablespace tsd datafile
'C:\Users\acer\Desktop\tablespaces\tsd.dbf'size 10M;



insert into Sales values(1,1,1,1,10,'01-01-2000');
```

```
insert into Sales values(3,3,3,3,30,'01-08-2000');
insert into Sales values(4,4,4,4,40,'01-11-2000');

select partition_name, tablespace_name
from user_tab_partitions
where table_name='SALES';

analyze table SALES compute STATISTICS;

--2. Create the same table as in Q1. With a different name with E
--MOVEMENT. Bring out the difference in these two tables.
create table Sales_Q2(
prod_id number(5),
cust_id number(5),
promo_id number(5),
quantiy_sold number(10),
amount_sold number(20),
sale_date date )
partition by range(sale_date)(
partition p1 values less than (to_date('01-04-2000','DD-MM-YYYY')),
partition p2 values less than (to_date('01-07-2000','DD-MM-YYYY')),
partition p3 values less than (to_date('01-10-2000','DD-MM-YYYY')),
partition p4 values less than (to_date('01-01-2001','DD-MM-YYYY'))
)
Enable Row Movement;

--3. Create a table with list partition as follows:
--➢ Table having columns deptno, deptname, quarterly_sales and st
--➢ 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
```

```
--➢ Add the following entries into the table and make conclusion
--entry maps:
--🕐 (10, 'accounting', 100, &#39;WA&#39;)
--🕐 (20, ';R&amp;D', 150, &#39;OR&#39;)
--🕐 (30, 'sales', 100, &#39;FL&#39;)
--🕐 (40, 'HR', 10, &#39;TX&#39;)
--🕐 (50, 'systems engineering', 10, &#39;CA&#39;)
create table Travel(
deptno number(5),
deptname varchar2(20),
quarterly_sales number(10),
state varchar2(5))
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')
);

insert into Travel values(10, 'accounting', 100, 'WA');
insert into Travel values(20, 'R and D', 150, 'OR');
insert into Travel values(30, 'sales', 100, 'FL');
insert into Travel values(40, 'HR', 10, 'TX');
insert into Travel values(50, 'System Engineering', 10, 'CA');

--4. Create a multi-column range partitioned table as directed:
--➢ Create a table with the actual DATE information in three sepa
--month, and day. Also amount_ sold.
--➢ Create following partitions:
--🕐 Before 2001: Less than jan 2001
--🕐 Less than april 2001
--🕐 Less than july 2001
```

```
--🕐 Less than jan 2002
--
--🕐 Future with max incoming value
--➢ Insert values into table and show to which partition does the
--🕐 (2001,3,17, 2000);
--🕐 (2001,11,1, 5000);
--🕐 (2002,1,1, 4000);
--
--Make conclusion for each result.
create table Date_info(
year number(4),
month number(2),
day number(2),
amount_sold number(10)
)
partition by range(year,month)(
partition d1 values less than ('2001','02'),
partition d2 values less than ('2001','05'),
partition d3 values less than ('2001','08'),
partition d4 values less than ('2001','11'),
partition d5 values less than ('2002','02'),
partition d6 values less than (MAXVALUE,MAXVALUE)
);

insert into date_info values('2001','03','17','2000');
insert into date_info values('2001','11','01','5000');
insert into date_info values('2002','01','01','4000');

analyze table date_info compute STATISTICS;

--5. Create a multicolumn partitioned table as directed:
--➢ Table supplier_parts, storing the information about which sup
--parts. To distribute the data in equal-sized partitions, it is
--the table based on the supplier_id, because some suppliers migh
```

```
--partition the table on (supplier_id, partnum) to manually enfor
--partitions.
--➢ Insert the following values
--(5,5, 1000); (5z,150, 1000); (10,100, 1000);
create table supplier_parts(
supplier_id number(5),
partnum number(5),
price number(10))
partition by range(supplier_id,partnum)(
partition p1 values less than (10,100),
partition p2 values less than (10,200),
partition p3 values less than (MAXVALUE,MAXVALUE)
);

INSERT INTO supplier_parts VALUES (5,5, 1000);
INSERT INTO supplier_parts VALUES (5,150, 1000);
INSERT INTO supplier_parts VALUES (10,100, 1000);

SELECT * FROM supplier_parts PARTITION (p1);

--SUPPLIER_ID    PARTNUM      PRICE
------------- ---------- ----------
--           5          5       1000
--           5        150       1000

SELECT * FROM supplier_parts PARTITION (p2);

--SUPPLIER_ID    PARTNUM      PRICE
------------- ---------- ----------
--          10        100       1000

--6. Create interval partitioned table as directed:
--➢ Creates a table named- Sales consisting of four partitions, c
--  sales. Each partition is given a name (sales_q1, sales_q2, ...
```

```
--amount_sold –all in number format and time_id in date format
--➢ Perform internal partitioning on time_id and take interval of
CREATE TABLE interval_sales
    ( prod_id        NUMBER(6)
    , cust_id        NUMBER
    , time_id        DATE
    , channel_id     CHAR(1)
    , promo_id       NUMBER(6)
    , quantity_sold  NUMBER(3)
    , amount_sold    NUMBER(10,2)
    )
  PARTITION BY RANGE (time_id)
  INTERVAL(NUMTOYMINTERVAL(1, 'MONTH'))
    ( PARTITION p0 VALUES LESS THAN (TO_DATE('1-1-2007', 'DD-MM-YYY
      PARTITION p1 VALUES LESS THAN (TO_DATE('1-1-2008', 'DD-MM-YYY
      PARTITION p2 VALUES LESS THAN (TO_DATE('1-7-2009', 'DD-MM-YYY
      PARTITION p3 VALUES LESS THAN (TO_DATE('1-1-2010', 'DD-MM-YYY

--7. Demonstrate Composite partitioning technique as directed
--➢ Implement range list partitioning for customer table having a
--cust_name, cust_state, and time_id
--➢ Perform range partitioning on time-id and list partitioning c
--Also create maxvalue and default partition for range and list p
--respectively.
--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-201
--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.
```

```
'C:\Users\acer\Desktop\tablespaces\ts7.dbf'size 10M;
CREATE TABLE quarterly_regional_sales
      (cust_id number,
       cust_name varchar2(50),
       cust_state varchar2(2),
       time_id DATE)
  TABLESPACE ts7
  PARTITION BY RANGE (time_id)
    SUBPARTITION BY LIST (state)
      (PARTITION old VALUES LESS THAN (TO_DATE('1-JAN-2005','DD-MON
         (SUBPARTITION west VALUES ('MH','GJ'),
          SUBPARTITION south VALUES ('TN','AP'),
          SUBPARTITION north VALUES ('UP','HP')
         ),
       PARTITION accquired VALUES LESS THAN (TO_DATE('1-JAN-2010','
         (SUBPARTITION west VALUES ('MH','GJ'),
          SUBPARTITION south VALUES ('TN','AP'),
          SUBPARTITION north VALUES ('UP','HP')
         ),
       PARTITION recent VALUES LESS THAN (TO_DATE('1-JAN-2015','DD-
         (SUBPARTITION west VALUES ('MH','GJ'),
          SUBPARTITION south VALUES ('TN','AP'),
          SUBPARTITION north VALUES ('UP','HP')
         ),
       PARTITION unknown VALUESLESS THAN (TO_DATE('1-JAN-2005','DD-
         (SUBPARTITION west VALUES ('MH','GJ'),
          SUBPARTITION south VALUES ('TN','AP'),
          SUBPARTITION north VALUES ('UP','HP')
         )
      );
```