Practical 1:

Implementation of Data partitioning through Range and List partitioning

a. Range Partitioning

1.1 Create table sales with the following columns:

prod_id	Number
cust_id	Number
time_id	Date
channel_id	Char
promo_id	Number
quantity_sold	Number
amount_sold	Number

Partition this table into 4 using range partition and time_id as partitioning key. Give partition names as:

sales_q1_2006, sales_q2_2006, sales_q3_2006, sales_q4_2006.

Query:

```
CREATE TABLE sales (prod_id number,
cust_id number,
time_id date,
channel_id char,
promo_id number,
quantity_sold number,
amount_sold number)
PARTITION by range(time_id)(
PARTITION sales_q1_2006 VALUES less than (to_date('2006-mar-31','yyyy-MON-dd')),
PARTITION sales_q2_2006 VALUES less than (to_date('2006-jun-31','yyyy-MON-dd')),
PARTITION sales_q3_2006 VALUES less than (to_date('2006-sep-31','yyyy-MON-dd')),
PARTITION sales_q4_2006 VALUES less than (to_date('2006-dec-31','yyyy-MON-dd')));
```

Output:

```
SQL> CREATE TABLE sales(
  2   prod_id number,
  3   cust_id number,
  4   time_id date,
  5   channel_id char,
  6   promo_id number,
  7   quantity_sold number,
  8   amount_sold number
  9   )
 10   PARTITION by range(time_id) (
 11   PARTITION sales_q1_2006 VALUES less than (to_date('2006-mar-31','yyyy-MON-dd')),
 12   PARTITION sales_q2_2006 VALUES less than (to_date('2006-jun-30','yyyy-MON-dd')),
 13   PARTITION sales_q3_2006 VALUES less than (to_date('2006-sep-30','yyyy-MON-dd')),
 14   PARTITION sales_q4_2006 VALUES less than (to_date('2006-dec-31','yyyy-MON-dd')),
 15   PARTITION max_value VALUES less than (MAXVALUE)
 16   );

Table created.

SQL>
```

Store quarterly data into each partition. For example, partition sales_q1_2006 will store records for first quarter 01-jan-2006 to 01-mar-2006

sales_q1_2006 will store records for second quarter 01-apr-2006 to 01-jun-2006. And so on.

Query:

```
insert into sales values(46,1298, to_date('2006-jan-02','yyyy-MON-dd'),'A', 101, 23, 45032);
insert into sales values(5,1838, to_date('2006-feb-27','yyyy-MON-dd'),'X', 101, 7, 1432);
insert into sales values(1,1848, to_date('2006-mar-17','yyyy-MON-dd'),'Q', 101, 37, 35032);
insert into sales values(3,9566, to_date('2006-jun-11','yyyy-MON-dd'),'P', 101, 45, 65032);
insert into sales values(345,6355, to_date('2006-apr-21','yyyy-MON-dd'),'P', 101, 12, 5032);
insert into sales values(8,3365, to_date('2006-may-15','yyyy-MON-dd'),'O', 101, 19, 5932);
insert into sales values(180,0707, to_date('2006-jul-24','yyyy-MON-dd'),'O', 101, 82, 180000);
insert into sales values(6,1011, to_date('2006-aug-11','yyyy-MON-dd'),'B', 101, 29, 45132);
insert into sales values(44,9935, to_date('2006-oct-18','yyyy-MON-dd'),'M', 101, 34, 4232);
insert into sales values(12,5463, to_date('2006-oct-10','yyyy-MON-dd'),'M', 101, 3, 452);
insert into sales values(11,4378, to_date('2006-nov-02','yyyy-MON-dd'),'Y', 101, 8, 4526);
insert into sales values(14,5438, to_date('2006-dec-25','yyyy-MON-dd'),'Y', 101, 38, 11032);
```

Output:

```
SQL>  insert into sales values(1,1848, to_date('2006-mar-17','yyyy-MON-dd'),'Q', 101, 37, 35032);

1 row created.
```

Write a command to view records in each partition.

Query:

```
select * from sales PARTITION(sales_q1_2006);

select * from sales PARTITION(sales_q2_2006);

select * from sales PARTITION(sales_q3_2006);

select * from sales PARTITION(sales_q4_2006);
```

Output:

```
SQL> select * from sales PARTITION(sales_q1_2006);

   PROD_ID    CUST_ID TIME_ID   C   PROMO_ID QUANTITY_SOLD AMOUNT_SOLD
---------- ---------- --------- - ---------- ------------- -----------
        46       1298 02-JAN-06 A        101            23       45032
         5       1838 27-FEB-06 X        101             7        1432
         1       1848 17-MAR-06 Q        101            37       35032

SQL>

SQL> select * from sales PARTITION(sales_q2_2006);

   PROD_ID    CUST_ID TIME_ID   C   PROMO_ID QUANTITY_SOLD AMOUNT_SOLD
---------- ---------- --------- - ---------- ------------- -----------
         3       9566 11-JUN-06 P        101            45       65032
       345       6355 21-APR-06 P        101            12        5032
         8       3365 15-MAY-06 O        101            19        5932

SQL>  select * from sales PARTITION(sales_q3_2006);

   PROD_ID    CUST_ID TIME_ID   C   PROMO_ID QUANTITY_SOLD AMOUNT_SOLD
---------- ---------- --------- - ---------- ------------- -----------
       180        707 24-JUL-06 O        101            82      180000
         6       1011 11-AUG-06 B        101            29       45132

SQL>  select * from sales PARTITION(sales_q4_2006);

   PROD_ID    CUST_ID TIME_ID   C   PROMO_ID QUANTITY_SOLD AMOUNT_SOLD
---------- ---------- --------- - ---------- ------------- -----------
        44       9935 18-OCT-06 M        101            34        4232
        12       5463 10-OCT-06 M        101             3         452
        11       4378 02-NOV-06 Y        101             8        4526
        14       5438 25-DEC-06 Y        101            38       11032

SQL>
```

Write a command to display the partition structure.

Query:

```
select table_name,partition_name,partition_position,high_value,num_rows from ALL_TAB_PARTITIONS where table_name='SALES2';
```

Output:

```
SQL> select table_name,partition_name,partition_position,high_value,num_rows from ALL_TAB_PARTITIONS
 where table_name='SALES2';

TABLE_NAME                     PARTITION_NAME                 PARTITION_POSITION
------------------------------ ------------------------------ ------------------
HIGH_VALUE
--------------------------------------------------------------------------------
  NUM_ROWS
----------
SALES2                         SALES_Q2_2006                                   1
TO_DATE(' 2006-06-30 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIA
         3

SALES2                         SALES_Q3_2006                                   2
TO_DATE(' 2006-09-30 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIA
         2

TABLE_NAME                     PARTITION_NAME                 PARTITION_POSITION
------------------------------ ------------------------------ ------------------
HIGH_VALUE
--------------------------------------------------------------------------------
  NUM_ROWS
----------

SALES2                         SALES_Q4_2006                                   3
TO_DATE(' 2006-12-31 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIA
         4

SALES2                         SALES_Q1_2007                                   4
TO_DATE(' 2007-03-31 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIA

TABLE_NAME                     PARTITION_NAME                 PARTITION_POSITION
------------------------------ ------------------------------ ------------------
HIGH_VALUE
--------------------------------------------------------------------------------
  NUM_ROWS
----------
         0
```

Write a command to add a new partition called sales_q1_2007 for the next quarter value.

Query:

alter table sales add PARTITION sales_q1_2007 VALUES less than (to_date('2007-mar-31','yyyy-MON-dd'));

Output:

```
SQL> alter table sales add PARTITION sales_q1_2007 VALUES less than (to_date('2007-mar-31','yyyy-MON
-dd'));

Table altered.

SQL>
```

Write a command to delete all records from partition sales_q1_2006.

Query:

delete from sales partition(sales_q1_2006);

Output:

```
SQL> delete from sales partition(sales_q1_2006);

3 rows deleted.

SQL> |
```

Write a command to delete a partition.

Query:

alter table sales drop PARTITION sales_q1_2006;

Output:

```
SQL> alter table sales drop PARTITION sales_q1_2006;

Table altered.

SQL> |
```

List Partitioning

1.2 Create table Student with the following columns:

Student_id	Number
Student_name	Number
Student_dob	Date

Create list partition with student_name as partition key. Create following two partitions.

stu_divA with values 'a','b','c','d','e','f','g','h','i','j','k'

stu_divB with values 'n','o','p','q','r','s','t','u','v','w','x','y','z'

Query:

```
create table STUDENT (
        student_id number,
        student_name char,
        student_dob date
        )
        PARTITION by LIST(student_name) (
        PARTITION stu_divA values('a','b','c','d','e','f','g','h','i','j','k'),
        PARTITION stu_divB values('n','o','p','q','r','s','t','u','v','w','x','y','z')
        );
```

Output:

```
SQL> create table STUDENT (
  2   student_id number,
  3   student_name char,
  4   student_dob date
  5   )
  6   PARTITION by LIST(student_name) (
  7   PARTITION stu_divA values('a','b','c','d','e','f','g','h','i','j','k'),
  8   PARTITION stu_divB values('n','o','p','q','r','s','t','u','v','w','x','y','z')
  9   );

Table created.
```

Write a command to view records in each partition.

Query:

select * from STUDENT partition(stu_divA);

select * from STUDENT partition(stu_divB);

Output:

```
SQL> select * from STUDENT partition(stu_divA);

STUDENT_ID S STUDENT_D
---------- - ---------
       101 a 24-JUL-98
       103 b 11-AUG-98
       104 c 14-FEB-98
       105 d 20-JUN-98
       106 e 04-OCT-98

SQL>  select * from STUDENT partition(stu_divB);

STUDENT_ID S STUDENT_D
---------- - ---------
       102 n 02-FEB-99
       107 o 12-OCT-98
       108 p 10-OCT-98
       109 q 01-DEC-98
       110 r 19-SEP-98
```

Write a command to display the partition structure.

Query:

select table_name, partition_name, partition_position, high_value, num_rows from ALL_TAB_PARTITIONS where table_name='STUDENT';

Output:

```
SQL> select table_name,partition_name,partition_position,high_value,num_rows from ALL_TAB_PARTITIONS
 where table_name='STUDENT';

TABLE_NAME                     PARTITION_NAME                 PARTITION_POSITION
------------------------------ ------------------------------ ------------------
HIGH_VALUE
--------------------------------------------------------------------------------
  NUM_ROWS
----------
STUDENT                        STU_DIVA                                        1
'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k'
         5

STUDENT                        STU_DIVB                                        2
'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z'
         5

TABLE_NAME                     PARTITION_NAME                 PARTITION_POSITION
------------------------------ ------------------------------ ------------------
HIGH_VALUE
--------------------------------------------------------------------------------
  NUM_ROWS
----------


SQL>
```

Write a command to add a new partition called stu_null for the null values.

Query:

alter table STUDENT add PARTITION stu_null VALUES (NULL);

Output:

```
SQL> alter table STUDENT add PARTITION stu_null VALUES (NULL);

Table altered.
```

Write a command to display records from stu_null partition.

Query:

insert into STUDENT(student_id,student_dob) values(101,to_date('24-07-1998','dd-MM-yyyy')); //Inserting Values

select * from STUDENT partition(stu_null);

Output:

```
SQL> select * from STUDENT partition(stu_null);

STUDENT_ID S STUDENT_D
---------- - ---------
       101   24-JUL-98

SQL>
```

Write a command to add a new partition called stu_default for the default values.

Query:

alter table STUDENT add PARTITION stu_default VALUES (DEFAULT);

Output:

```
SQL> alter table STUDENT add PARTITION stu_default VALUES (DEFAULT);

Table altered.
```

Write a command to display records from stu_default partition.

Query:

select * from STUDENT partition(stu_default);

Output: Created a table with default value for student_name field as 'N'

```
SQL> insert into STUDENT(student_id,student_dob) values(101,to_date('24-07-1998','dd-MM-yyyy'));

1 row created.

SQL> select * from STUDENT partition(stu_default);

STUDENT_ID S STUDENT_D
---------- - ---------
       101 N 24-JUL-98

SQL>
```

Write a command to add values 'l' and 'm' in a partition stu_divA

Query:

alter table STUDENT modify PARTITION stu_divA ADD VALUES ('l', 'm');

Output:

```
SQL> alter table STUDENT modify PARTITION stu_divA ADD VALUES ('I', 'm');

Table altered.

SQL>
```

Write a command to display records from stu_divA partition.

Query:

select * from STUDENT partition(stu_divA);

Output:

```
SQL> select * from STUDENT partition(stu_divA);

STUDENT_ID S STUDENT_D
---------- - ---------
       101 a 24-JUL-98
       103 b 15-AUG-98
       104 c 14-FEB-98
       105 d 20-JUN-98
       106 e 04-OCT-98
       141 I 20-JUL-98
       120 m 24-NOV-98

7 rows selected.

SQL>
```