**Practical 1**

**Aim:**

Implementation of Data Partitioning through Range and List Partitioning.

**Objective:**

Implementation of various types of partitions: Range, Hash, List

1. **Range Partitioning**

This type of partitioning is useful when dealing with data that has logical ranges

into which it can be distributed;

for example, value of year.

Performance is best when the data evenly distributes across the range.

To partition the table’s records, use partition by range clause of create table

command.

When you partition a table or index using the range method, you specify a

maximum value for the partitioning key column(s) for each partition.

The ranges will determine the values stored in each partition.

* Creating table sales :

CREATE TABLE sales (year number(4),product varchar2(10),amt number(10,2))

PARTITION BY RANGE (year)

(PARTITION p1 VALUES LESS THAN (1990),

PARTITION p2 VALUES LESS THAN (1993),

PARTITION p3 VALUES LESS THAN (1996),

PARTITION p4 VALUES LESS THAN (1999));

* Inserting some values in it:

insert into sales values(1989,'Shoes',3000000);

insert into sales values(1985,'Formal Trousers',5000000);

insert into sales values(1991,'Belts',1000000);

insert into sales values(1992,'Pens',220000);

insert into sales values(1995,'Pencils',10000);

insert into sales values(1998,'Books',250000);

insert into sales values(1997,'notes',10000);

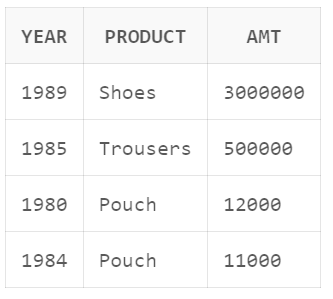
insert into sales values(1980,'Pouch',12000);

insert into sales values(1996,'Pen',12000);

insert into sales values(1984,'Pouch',11000);

* To retrieve the data from partition use following query:

select \* from sales partition(p1);



* Drop partition p4:

DELETE FROM sales WHERE year=1997;

DELETE FROM sales WHERE year=1998;

ALTER TABLE sales DROP PARTITION p4;

* After drop query if we try to select data from partition p4 , we would get the following error:

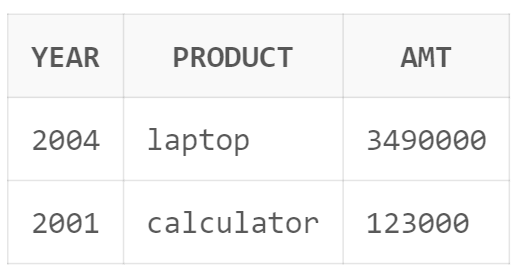
select \* from sales partition(p4);



* Add new partition p5:
* Note- this query will not work if last partition is defined as MAXVALUE.

ALTER TABLE sales ADD PARTITION p5 VALUES LESS THAN (2005);

select \* from sales partition(p5);



* To display the partition names using User\_tab\_partitions (data dictionary table)

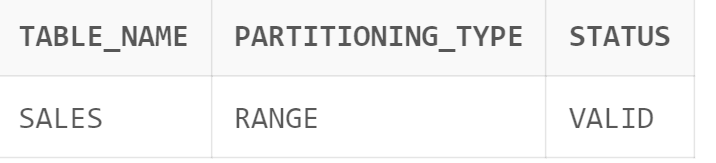
select PARTITION\_NAME from USER\_TAB\_PARTITIONS where TABLE\_NAME = 'SALES';



* To display the partition table details using user\_part\_tables (data dictionary table)

select TABLE\_NAME, PARTITIONING\_TYPE, STATUS from USER\_PART\_TABLES where

TABLE\_NAME = 'SALES';



1. **List Partitioning:**

Unlike range partitioning, with list partitioning, there is no apparent sense of

order between partitions.

You can also specify a default partition into which rows that do not map to any

other partition are mapped.

Use list partitioning when you require explicit control over how rows map to

partitions.

In List partitioning, you tell Oracle all of the possible values, and designate the

partitions into which the corresponding rows should be inserted.

List partitioning allows unordered and unrelated sets of data to be grouped and

organized together very naturally

* Creating table customer:

CREATE TABLE customers(custcode number(5),

cust\_name varchar2(20),

Address varchar2(10),

City varchar2(20),

Bal number(10))

partition by list(City)

(partition north values ('DELHI','CHANDIGARH'),

partition east values ('KOLKOTA','PATNA'),

partition south values ('HYDERABAD','BANGALORE','CHENNAI'),

partition west values ('BOMBAY','GOA'));

* Inserting some data in the table:

insert into customers values(1,'rupali','Ab Road','DELHI',450000);

insert into customers values(2,'hafsa','New road','DELHI',500000);

insert into customers values(3,'abdulla','sec 5','CHANDIGARH',450000);

insert into customers values(4,'rashmi','Z Road','CHANDIGARH',450000);

insert into customers values(5,'ayush','sec 2','KOLKOTA',100000;

insert into customers values(6,'maya','c Road','PATNA',230000);

insert into customers values(7,'rahul','sec 2','HYDERABAD',230000);

insert into customers values(8,'mayur','c Road','HYDERABAD',230000);

insert into customers values(9,'vishaha','ab house','BANGALORE',230000);

insert into customers values(10,'mina','c Road','CHENNAI',231000);

insert into customers values(11,'riya','sec 72','CHENNAI',340009;

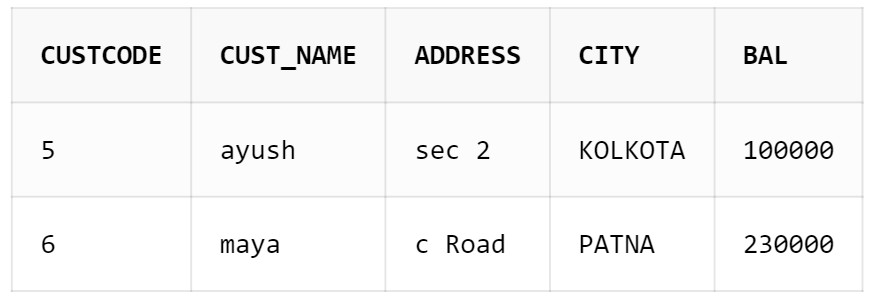
insert into customers values(12,'ramesh','sec 74','BOMBAY',450000);

insert into customers values(13,'aarti','sec 73','BOMBAY',340009);

insert into customers values(14,'nafia','V road','GOA',500000);

* Retrieving data from partition east:

select \* from customers partition(east);



* Adding partition central\_India consisting of values BHOPAL and Nagpur and inserting some data in it:

alter table customers add partition central\_India values('BHOPAL','Nagpur');

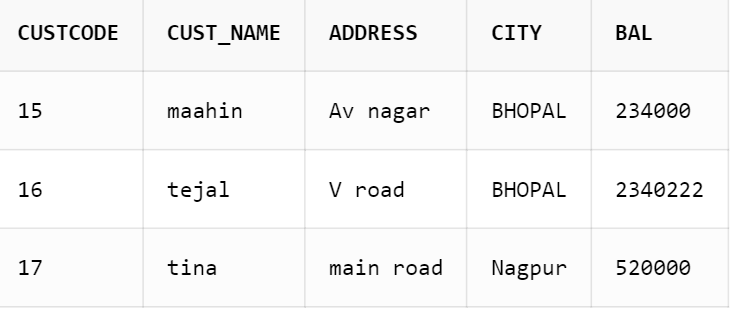
insert into customers values(17,'tina','main road','Nagpur',520000);

insert into customers values(16,'tejal','V road','BHOPAL',2340222);

insert into customers values(15,'maahin','Av nagar','BHOPAL',234000);

* Retrieving data from partition central\_India:

select \* from customers partition(central\_India)

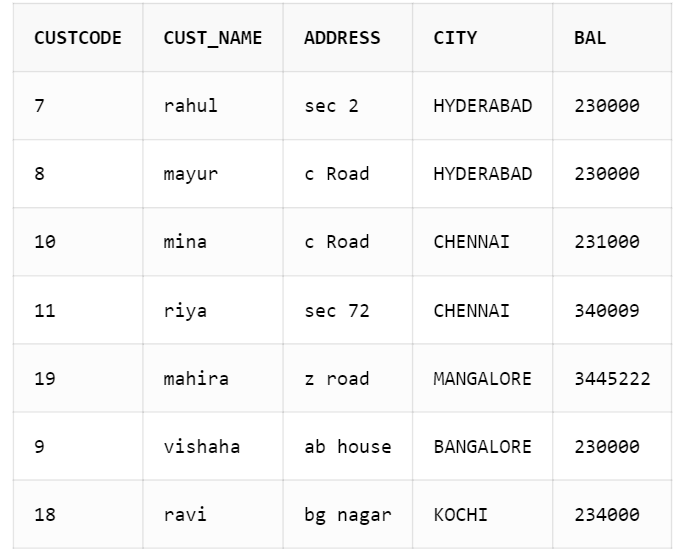


* The following query adds new set of cities (KOCHI,MANGALORE) to an existing partition list:

ALTER TABLE customers MODIFY PARTITION south ADD VALUES ('KOCHI', 'MANGALORE');

* After inserting data perform following query:

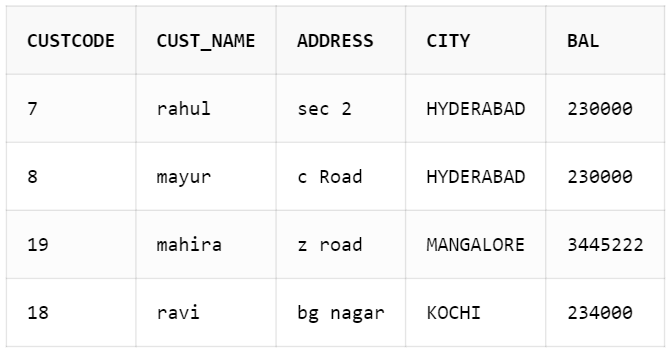
select \* from customers partition(south);



* Following query will drop a set of cities (‘CHENNAI’,’BANGALORE’) from an existing value list:

ALTER TABLE customers MODIFY PARTITION south DROP VALUES ('CHENNAI','BANGALORE');

select \* from customers partition(south);



“ CHENNAI and BANGALORE values are dropped from partition south. ”

* To split the partition west into two partitions give the following command:

ALTER TABLE customers SPLIT PARTITION west INTO

(PARTITION northwest VALUES ('GOA'),

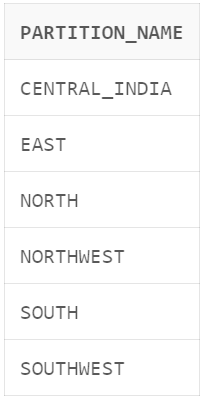
PARTITION southwest);

select \* from customers partition(northwest);



* To display the partition names using User\_tab\_partitions (data dictionary table):

select PARTITION\_NAME from USER\_TAB\_PARTITIONS where TABLE\_NAME = 'CUSTOMERS';



* To display the partition table details using user\_part\_tables (data dictionary table)

SELECT TABLE\_NAME, PARTITIONING\_TYPE, STATUS FROM USER\_PART\_TABLES WHERE

TABLE\_NAME = 'CUSTOMERS';



**Q.1 A. Create book table (book\_id(pk), title, author, price, book\_rating) with range Partition on rating with 1, 2 and 3 for three different partitions. Insert at least 10 record in the table.**

**B.Display all the books with rating 3 and price in the range 200 and 1000.**

**SQL Query:**

Create table book(book\_id varchar2(20) primary key,title varchar2(50),author varchar2(50), price number(10),book\_rating int)

partition by range(book\_rating)(

Partition p1 values less than (1),

Partition p2 values less than (2),

Partition p3 values less than (3),

Partition p4 values less than(MAXVALUE)

);

insert into book values('A01', 'Rich Dad Poor Dad', 'Robert KioSaki', 750, 5);

insert into book values('A02', 'Anna Karenina', 'Leo Tolstoy', 900, 4);

insert into book values('A03', 'Madame Bovary', 'Gustave Flaubert', 550, 5);

insert into book values('A04', 'War and Peace', 'Leo Tolstoy', 250, 2);

insert into book values('A05', 'The Great Gatsby', 'F. Scott Fitzgerald', 750, 2);

insert into book values('A06', 'Lolita', 'Vladimir Nabokov', 150, 1);

insert into book values('A07', 'Middlemarch', 'George Eliot', 850, 4);

insert into book values('A08', 'The Adventures of Huckleberry Finn', 'Mark Twain', 650, 1);

insert into book values('A09', 'In Search of Lost Time', 'Marcel Proust', 280, 2);

insert into book values('A10', 'Hamlet', 'William Shakespeare', 120, 1);

insert into book values('A11', 'Middlemarch', 'George Eliot', 850, 3);

insert into book values('A12', 'The Adventures of Huckleberry Finn', 'Mark Twain', 650, 3);

insert into book values('A13', 'In Search of Lost Time', 'Marcel Proust', 280, 3);

insert into book values('A14', 'Hamlet', 'William Shakespeare', 120, 3);

SELECT \* FROM book partition(p4)

WHERE price BETWEEN 200 AND 1000;

**Output:**



**Q.2 A. Create 3 Partition in table cabinet using range partition on cid Column of cabinet (cid, mem\_name, address, state\_rep, phone\_number, sal) P1 cid < 101; P2 cid < 501; P3 cid <1001.**

**Insert at least 10 records in the Table.**

1. **Display the Contents of third Partition.**

**SQL Query:**

Create table cabinet (

cid number(5),

mem\_name varchar2(20),

address varchar2(30),

state\_rep varchar2(30),

phone\_number number(12),

sal number (8)

)Partition by range(cid)(

Partition P1 values less than (101),

Partition P2 values less than (501),

Partition P3 values less than (1001)

);

insert into cabinet values(900,'Onkar','Canada','USA',7020767185,100000);

insert into cabinet values(95,'Rasika','Ratanagiri','Uk',8160124511,20000);

insert into cabinet values(200,'Atharv','Ramgad','SouthAmerica',8060124511,25000);

insert into cabinet values(800,'Abhi','Devagad','England',8067124511,200000);

insert into cabinet values(922,'Sonali','Canada','USA',8064021211,120000);

insert into cabinet values(611,'Vaishnavi','Europe','New Jeresy',8060664511,285400);

insert into cabinet values(422,'Rahul','Goa','New Jeresy',8770124511,20000);

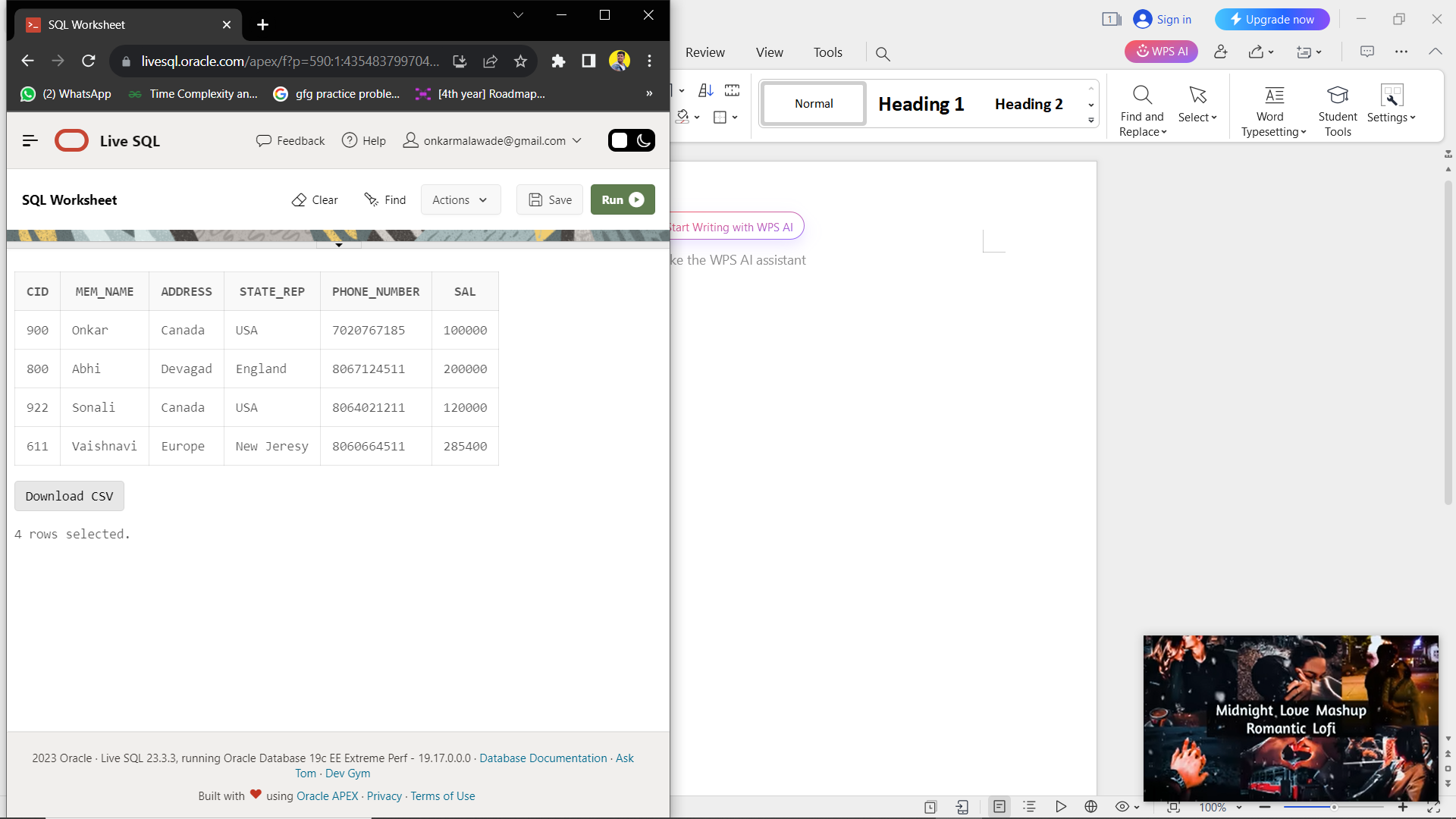
insert into cabinet values(80,'Aniket','Ratanagiri','New Jeresy',9060124511,206540);

insert into cabinet values(90,'Esha','America','New Jeresy',8055124511,200080);

insert into cabinet values(210,'Aditya','Phonda','New Jeresy',7060124511,207800);

select \* from cabinet partition (P3);

**Output:**



**Q.3 A.Create table Employee with attributes empid, name, age, salary and joining date by using hash partition based on employee salary with minimum 3 partitons. Insert at least 10 records in the table.**

**B.Display the contents of the third partition.**

**SQL Query:**

CREATE TABLE employee

(empid NUMBER,

empname VARCHAR2 (100),

age NUMBER,

joiningDate date,

salary NUMBER

)

PARTITION BY HASH (salary)

(PARTITION s1, PARTITION s2, PARTITION s3);

insert into employee values(1,'Rupali',22,'12-jan-2013',450000);

insert into employee values(2,'Vishal',23,'21-feb-2015',529999);

insert into employee values(3,'Mahek',24,'10-jan-2011',480000);

insert into employee values(4,'Hafsa',22,'1-march-2013',600000);

insert into employee values(5,'Nafia',21,'18-aug-2017',740000);

insert into employee values(6,'Mahesh',25,'29-sep-2013',370000);

insert into employee values(7,'Mayur',24,'25-oct-2014',620000);

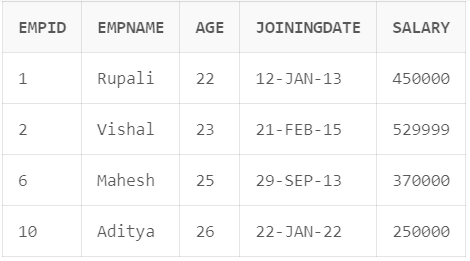
insert into employee values(8,'Rahul',21,'16-nov-2015',762000);

insert into employee values(9,'Onkar',23,'18-dec-2020',400000);

insert into employee values(10,'Aditya',26,'22-jan-2022',250000);

select \* from employee partition (s3) order by empid;

**Output:**



**Q.4 A.Create test\_record(test\_id, test\_type, patient\_name, employee\_no, labno, result) with list partition on result field as below:**

**P1 = (positive)**

**P2 = (negative)**

**Insert at least 10 records in the table.**

**Display the test\_records which have negative result.**

**SQL Query:**

Create table test\_record(

test\_id varchar2(20),

test\_type varchar2(50),

patient\_name varchar2(50),

employee\_no number(10),

labno varchar2(20),

res varchar2(10))

partition by list(res)(

partition res1 values('positive'),

partition res2 values('negative')

);

insert into test\_record values('A101', 'Haemoglobin Test','Raju',7020304010,'LAB01','positive');

insert into test\_record values('A102', 'Haemoglobin Test','kaju',7020304011,'LAB02','positive');

insert into test\_record values('A103', 'Haemoglobin Test','Paju',7020304012,'LAB01','positive');

insert into test\_record values('A104', 'Haemoglobin Test','Naju',7020304013,'LAB02','positive');

insert into test\_record values('A105', 'Haemoglobin Test','Haju',7020304013,'LAB01','negative');

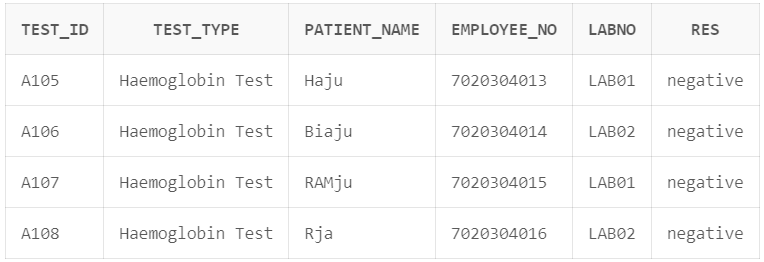
insert into test\_record values('A106', 'Haemoglobin Test','Biaju',7020304014,'LAB02','negative');

insert into test\_record values('A107', 'Haemoglobin Test','RAMju',7020304015,'LAB01','negative');

insert into test\_record values('A108', 'Haemoglobin Test','Rja',7020304016,'LAB02','negative');

select \* from test\_record partition(res2);

**Output:**



1. **5 A.Create table Bank with fields BankId, BName, Location. Partition the**

**Bank table based on Location as per following.**

**BK1 = (Mumbai,Pune,Nashik), BK2 = (Lucknow, Kanpur, Varanasi),**

**BK3 = (Chandigarh, Mohali, Amritsar), BK4 = (GandhiNagar,Ahmedabad,Surat)**

**Insert 10 records in Bank Table.**

**B.Add values “Ratnagiri” in BK1 partition.**

**SQL Query:**

CREATE TABLE Bank(BankId number(5),

BName varchar2(20),

Location varchar2(30))

partition by list(Location)

(partition BK1 values ('Mumbai','Pune','Nashik'),

partition BK2 values ('Lucknow','Kanpur','Varanasi'),

partition BK3 values ('Chandigarh','Mohali','Amritsar'),

partition BK4 values ('GandhiNagar','Ahmedabad','Surat'));

insert into Bank values(101,'Talere','Mumbai');

insert into Bank values(102,'Talere','Pune');

insert into Bank values(103,'Talere','GandhiNagar');

insert into Bank values(104,'Talere','Mohali');

insert into Bank values(105,'Talere','Ahmedabad');

insert into Bank values(106,'Talere','Kanpur');

insert into Bank values(107,'Talere','Chandigarh');

insert into Bank values(108,'Talere','Varanasi');

insert into Bank values(109,'Talere','Lucknow');

insert into Bank values(110,'Talere','Pune');

ALTER TABLE Bank

MODIFY PARTITION BK1

ADD VALUES ('Ratanagiri');

insert into Bank values(111,'Talere','Ratanagiri');

insert into Bank values(112,'Talere','Ratanagiri');

insert into Bank values(113,'Talere','Ratanagiri');4

insert into Bank values(114,'Talere','Ratanagiri');

select \* from Bank partition(BK1);

**Output:**

