**Aim :- Implementation of Data partitioning through Range and List partitioning.**

**Distributed Databases: -**

A distributed database is basically a database that is not limited to one system, it is spread over different sites, i.e, on multiple computers or over a network of computers. A distributed database system is located on various sited that don’t share physical components. This maybe required when a particular database needs to be accessed by various users globally. It needs to be managed such that for the users it looks like one single database.

**Types:**

**1. Homogeneous Database:** In a homogeneous database, all different sites store database identically. The operating system, database management system and the data structures used – all are same at all sites. Hence, they’re easy to manage.

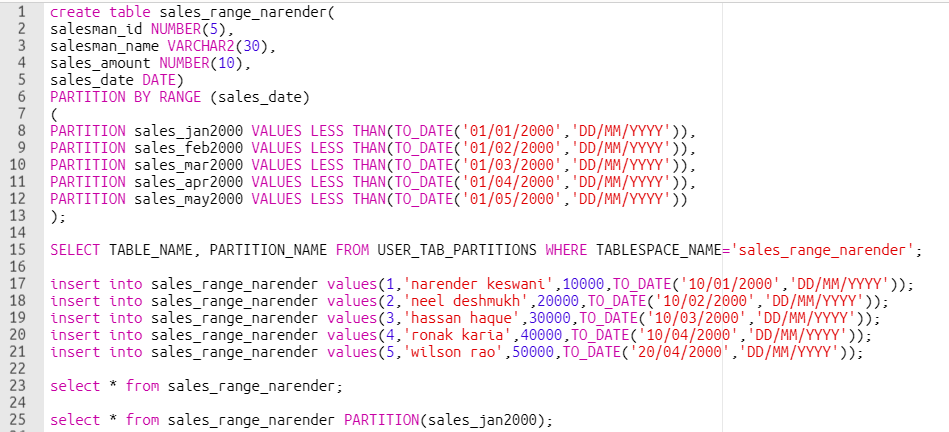
**2. Heterogeneous Database:** In a heterogeneous distributed database, different sites can use different schema and software that can lead to problems in query processing and transactions. Also, a particular site might be completely unaware of the other sites. Different computers may use a different operating system, different database application. They may even use different data models for the database. Hence, translations are required for different sites to communicate.

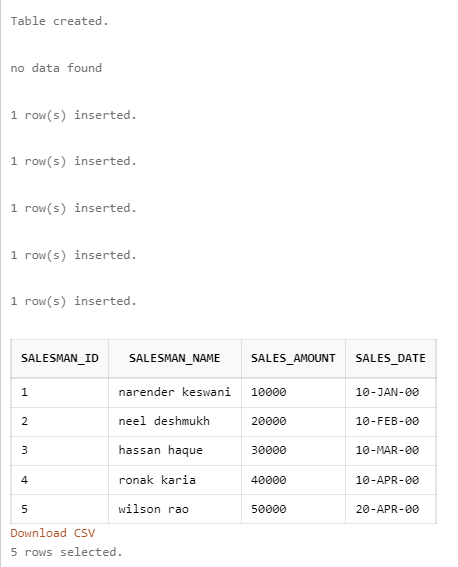
**Data Partitioning: -** Partitioning is the database process where very large tables are divided into multiple smaller parts. By splitting a large table into smaller, individual tables, queries that access only a fraction of the data can run faster because there is less data to scan. The main of goal of partitioning is to aid in maintenance of large tables and to reduce the overall response time to read and load data for particular SQL operations.

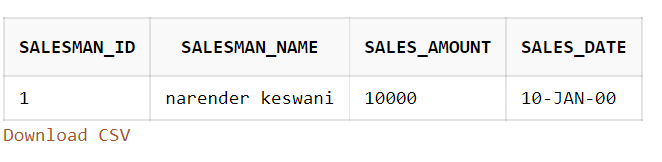
**Range Partitioning: -** Range partitioning is a type of relational database partitioning wherein the partition is based on a predefined range for a specific data field such as uniquely numbered IDs, dates or simple values like currency. A partitioning key column is assigned with a specific range, and when a data entry fits this range, it is assigned to this partition; otherwise it is placed in another partition where it fits.

**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.

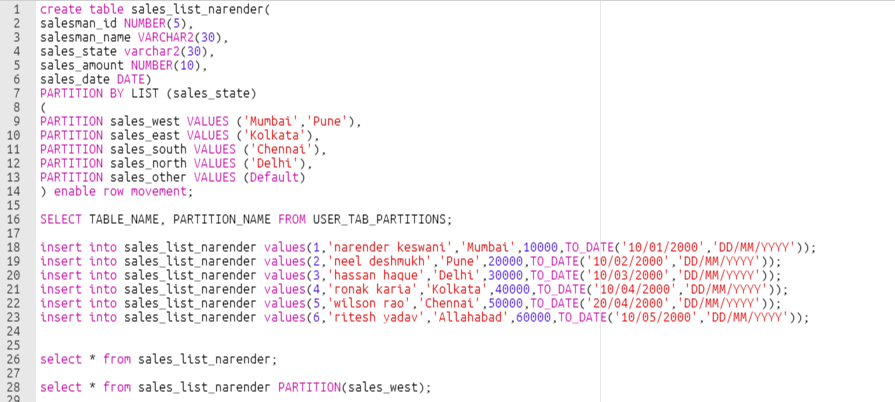
1. **Range Partitioning:**

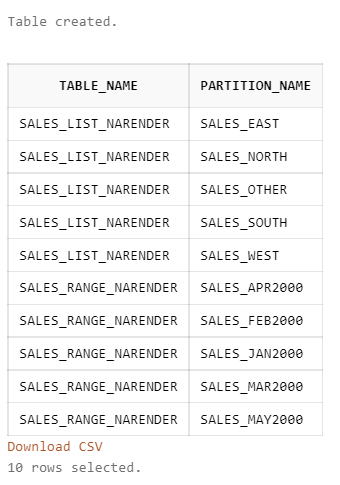


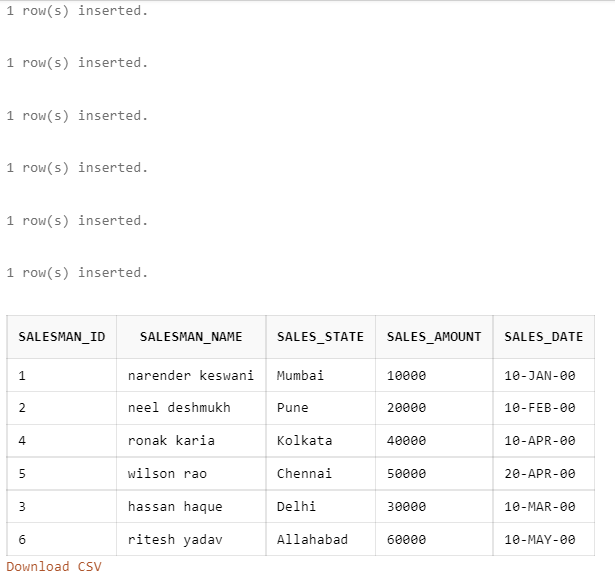


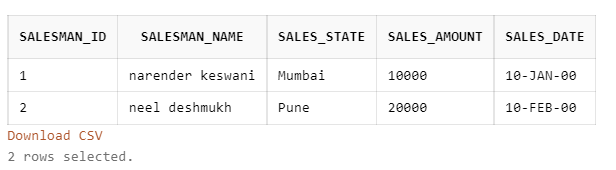


1. **List Partitioning:**









**CONCLUSION:**

I have learned the basics of distributed database management system with techniques such as: range and list in oracle.

**SOURCE CODE:**

create table sales\_range\_narender(

salesman\_id NUMBER(5),

salesman\_name VARCHAR2(30),

sales\_amount NUMBER(10),

sales\_date DATE)

PARTITION BY RANGE (sales\_date)

(

PARTITION sales\_jan2000 VALUES LESS THAN(TO\_DATE('01/01/2000','DD/MM/YYYY')),

PARTITION sales\_feb2000 VALUES LESS THAN(TO\_DATE('01/02/2000','DD/MM/YYYY')),

PARTITION sales\_mar2000 VALUES LESS THAN(TO\_DATE('01/03/2000','DD/MM/YYYY')),

PARTITION sales\_apr2000 VALUES LESS THAN(TO\_DATE('01/04/2000','DD/MM/YYYY')),

PARTITION sales\_may2000 VALUES LESS THAN(TO\_DATE('01/05/2000','DD/MM/YYYY'))

);

SELECT TABLE\_NAME, PARTITION\_NAME FROM USER\_TAB\_PARTITIONS WHERE TABLESPACE\_NAME='sales\_range\_narender';

insert into sales\_range\_narender values(1,'narender keswani',10000,TO\_DATE('10/01/2000','DD/MM/YYYY'));

insert into sales\_range\_narender values(2,'neel deshmukh',20000,TO\_DATE('10/02/2000','DD/MM/YYYY'));

insert into sales\_range\_narender values(3,'hassan haque',30000,TO\_DATE('10/03/2000','DD/MM/YYYY'));

insert into sales\_range\_narender values(4,'ronak karia',40000,TO\_DATE('10/04/2000','DD/MM/YYYY'));

insert into sales\_range\_narender values(5,'wilson rao',50000,TO\_DATE('20/04/2000','DD/MM/YYYY'));

select \* from sales\_range\_narender;

select \* from sales\_range\_narender PARTITION(sales\_feb2000);

------------------------------------------------------------------------------------------------------------------

create table sales\_list\_narender(

salesman\_id NUMBER(5),

salesman\_name VARCHAR2(30),

sales\_state varchar2(30),

sales\_amount NUMBER(10),

sales\_date DATE)

PARTITION BY LIST (sales\_state)

(

PARTITION sales\_west VALUES ('Mumbai','Pune'),

PARTITION sales\_east VALUES ('Kolkata'),

PARTITION sales\_south VALUES ('Chennai'),

PARTITION sales\_north VALUES ('Delhi'),

PARTITION sales\_other VALUES (Default)

) enable row movement;

SELECT TABLE\_NAME, PARTITION\_NAME FROM USER\_TAB\_PARTITIONS;

insert into sales\_list\_narender values(1,'narender keswani','Mumbai',10000,TO\_DATE('10/01/2000','DD/MM/YYYY'));

insert into sales\_list\_narender values(2,'neel deshmukh','Pune',20000,TO\_DATE('10/02/2000','DD/MM/YYYY'));

insert into sales\_list\_narender values(3,'hassan haque','Delhi',30000,TO\_DATE('10/03/2000','DD/MM/YYYY'));

insert into sales\_list\_narender values(4,'ronak karia','Kolkata',40000,TO\_DATE('10/04/2000','DD/MM/YYYY'));

insert into sales\_list\_narender values(5,'wilson rao','Chennai',50000,TO\_DATE('20/04/2000','DD/MM/YYYY'));

insert into sales\_list\_narender values(6,'ritesh yadav','Allahabad',60000,TO\_DATE('10/05/2000','DD/MM/YYYY'));

select \* from sales\_list\_narender;

select \* from sales\_list\_narender PARTITION(sales\_west);