Practical - 2

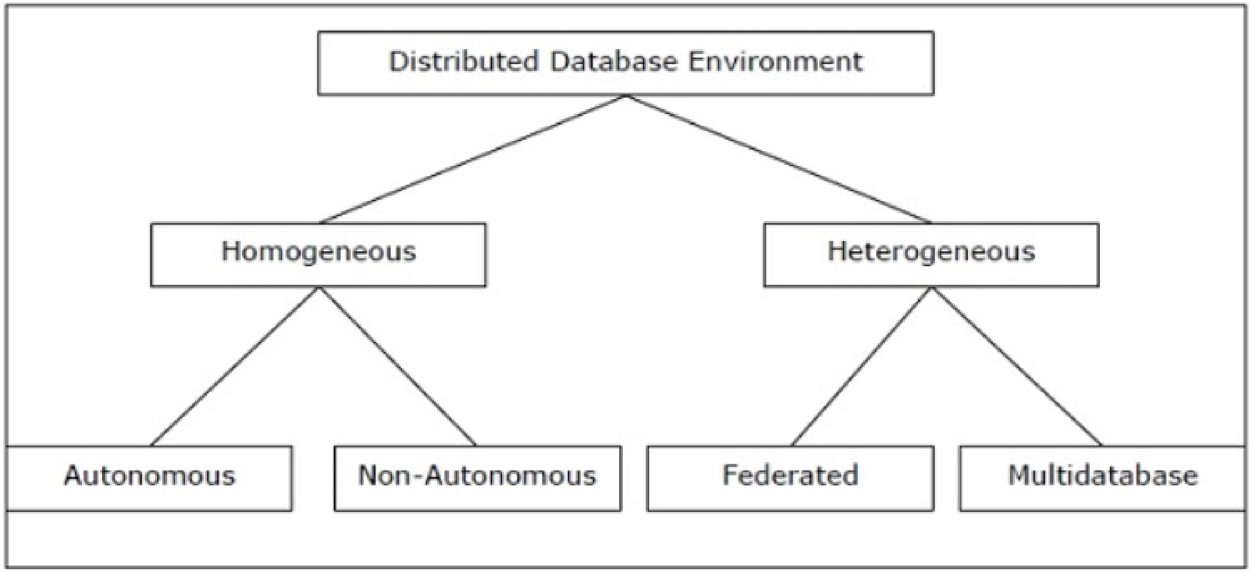
Aim: Implementation of Data Partitioning through Range and List Partitioning

Distributed Databases:

A distributed database (DDB) is an integrated collection of databases that is physically distributed across sites in a computer network. A distributed database management system (DDBMS) is the software system that manages a distributed database such that the distribution aspects are transparent to the users. To form a distributed database system (DDBS), the files must be structured, logically interrelated, and physically distributed across multiple sites. In addition, there must be a common interface to access the distributed data.

Types of Distributed Databases :

Distributed databases can be broadly classified into homogeneous and heterogeneous distributed database environments, each with further subdivisions, as shown in the following illustration.



Homogeneous Distributed Databases :

In a homogeneous distributed database, all the sites use identical DBMS and operating systems. Its properties are −

- The sites use very similar software.
- The sites use identical DBMS or DBMS from the same vendor.

- Each site is aware of all other sites and cooperates with other sites to process user requests.
- The database is accessed through a single interface as if it is a single database.

Types of Heterogeneous Distributed Databases :

- Autonomous − Each database is independent and functions on its own. They are integrated by a controlling application and use message passing to share data updates.
- Non-autonomous − Data is distributed across the homogeneous nodes and a central or master DBMS coordinates data updates across the sites.

Heterogeneous Distributed Databases :

In a heterogeneous distributed database, different sites have different operating systems, DBMS products and data models. Its properties are −

- Different sites use dissimilar schemas and software.
- The system may be composed of a variety of DBMSs like relational, network, hierarchical or object oriented.
- Query processing is complex due to dissimilar schemas.
- Transaction processing is complex due to dissimilar software.
- A site may not be aware of other sites and so there is limited co-operation in processing user requests.

Types of Heterogeneous Distributed Databases :

- Federated − The heterogeneous database systems are independent in nature and integrated together so that they function as a single database system.
- Un-federated − The database systems employ a central coordinating module through which the databases are accessed.

Data Partitioning :

Data partitioning is the technique of distributing data across multiple tables, disks, or sites in order to improve query processing performance and increase database manageability Query processing performance can be improved in one of two ways.

First, depending on how the data is partitioned, in some cases it can be determined a priori that a partition does not have to be accessed to process the query. Second, when data is partitioned across multiple disks or sites, I/O parallelism and in some cases query parallelism can be attained as different partitions can be accessed in parallel.

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 :

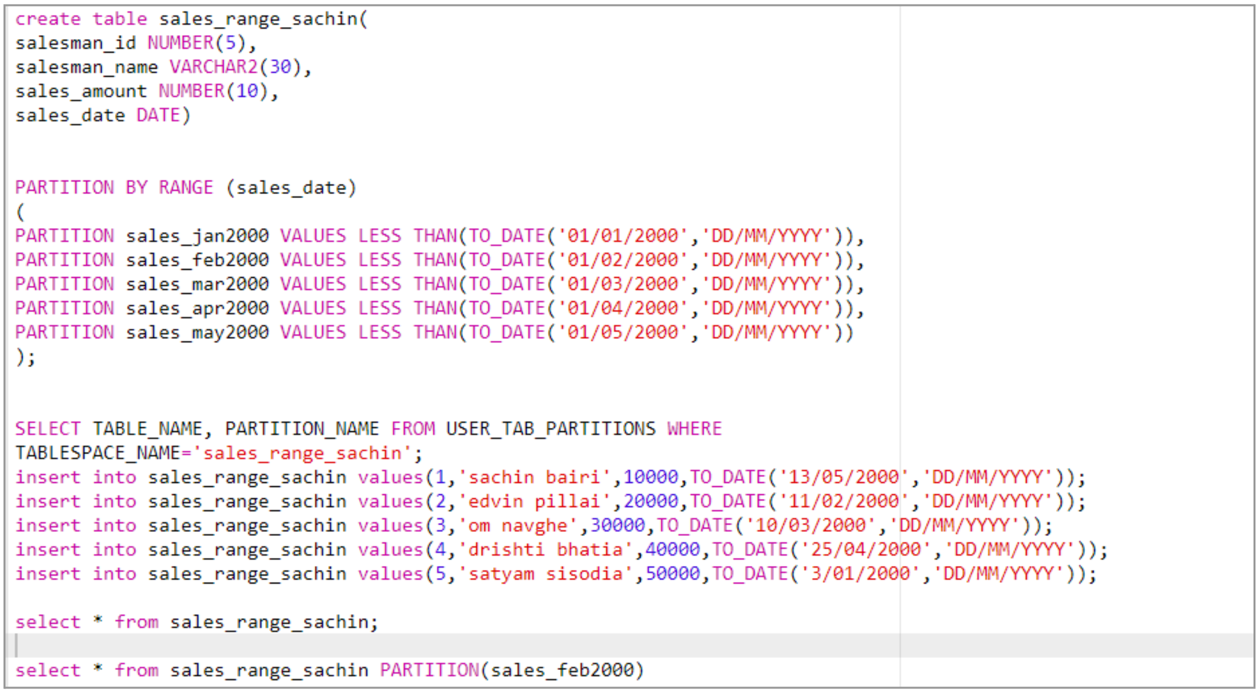
List partitioning enables you to explicitly control how rows map to partitions by specifying a list of discrete values for the partitioning key in the description for each partition. The advantage of list partitioning is that you can group and organize unordered and unrelated sets of data in a natural way.

1] Range Partitioning :

Code :

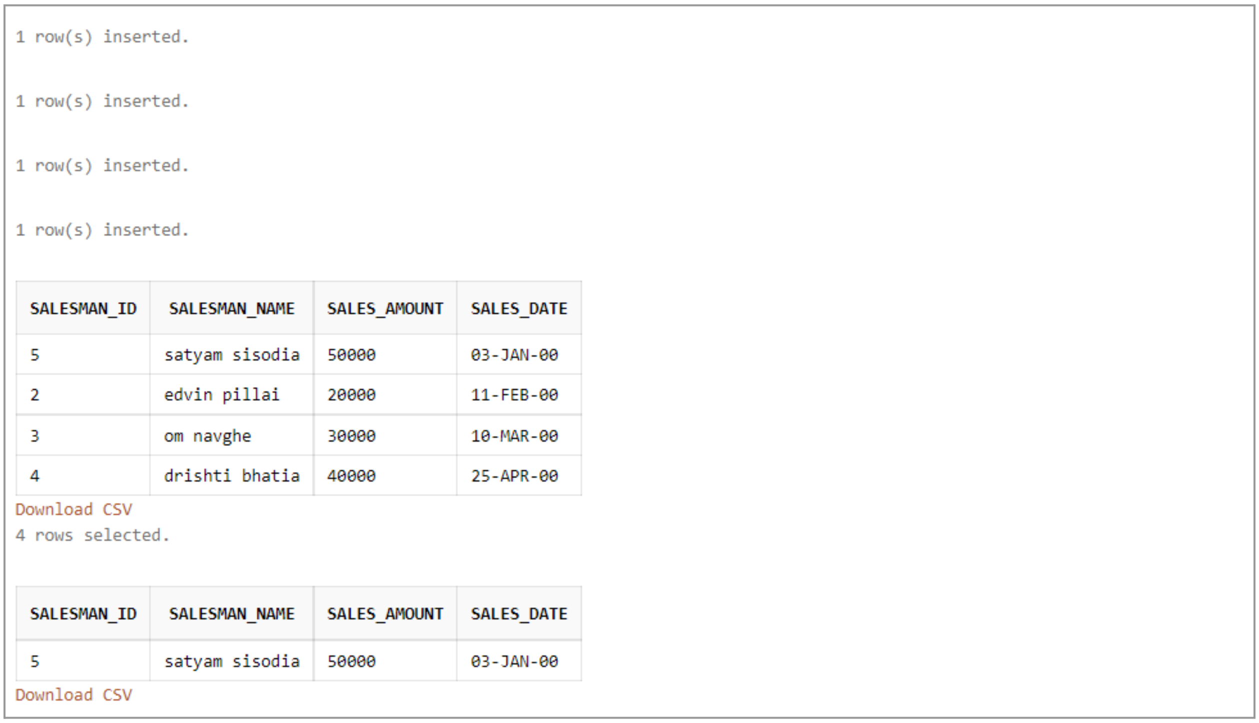
```
create table sales_range_sachin(
salesman_id NUMBER(5),
salesman_name VARCHAR2(30),
sales_amount NUMBER(10),
sales_date DATE)


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_sachin';
insert into sales_range_sachin values(1,'sachin
bairi',10000,TO_DATE('13/05/2000','DD/MM/YYYY'));
insert into sales_range_sachin values(2,'edvin
pillai',20000,TO_DATE('11/02/2000','DD/MM/YYYY'));
insert into sales_range_sachin values(3,'om
navghe',30000,TO_DATE('10/03/2000','DD/MM/YYYY'));
insert into sales_range_sachin values(4,'drishti
bhatia',40000,TO_DATE('25/04/2000','DD/MM/YYYY'));
insert into sales_range_sachin values(5,'satyam
sisodia',50000,TO_DATE('3/01/2000','DD/MM/YYYY'));

select * from sales_range_sachin;

select * from sales_range_sachin PARTITION(sales_feb2000)
```



```
create table sales_range_sachin(
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_sachin';
insert into sales_range_sachin values(1,'sachin bairi',10000,TO_DATE('13/05/2000','DD/MM/YYYY'));
insert into sales_range_sachin values(2,'edvin pillai',20000,TO_DATE('11/02/2000','DD/MM/YYYY'));
insert into sales_range_sachin values(3,'om navghe',30000,TO_DATE('10/03/2000','DD/MM/YYYY'));
insert into sales_range_sachin values(4,'drishti bhatia',40000,TO_DATE('25/04/2000','DD/MM/YYYY'));
insert into sales_range_sachin values(5,'satyam sisodia',50000,TO_DATE('3/01/2000','DD/MM/YYYY'));

select * from sales_range_sachin;

select * from sales_range_sachin PARTITION(sales_feb2000)
```

Output :



```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

SALESMAN_ID	SALESMAN_NAME	SALES_AMOUNT	SALES_DATE
5	satyam sisodia	50000	03-JAN-00
2	edvin pillai	20000	11-FEB-00
3	om navghe	30000	10-MAR-00
4	drishti bhatia	40000	25-APR-00

Download CSV
4 rows selected.

SALESMAN_ID	SALESMAN_NAME	SALES_AMOUNT	SALES_DATE
5	satyam sisodia	50000	03-JAN-00

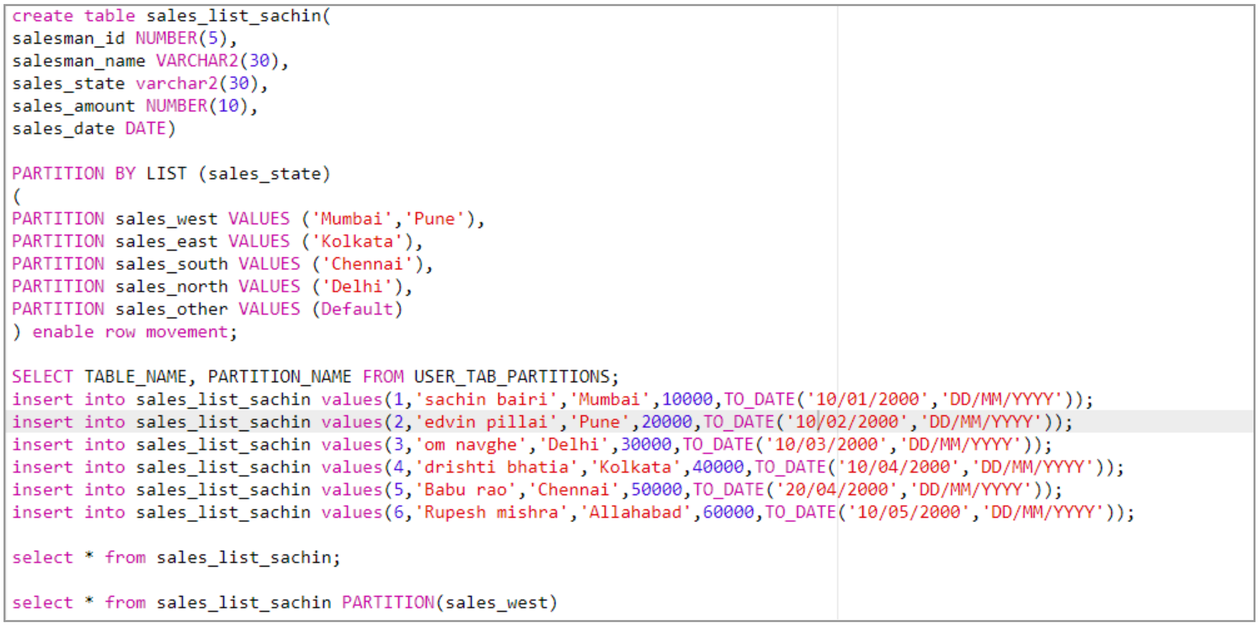
Download CSV

2] List Partitioning :

Code:

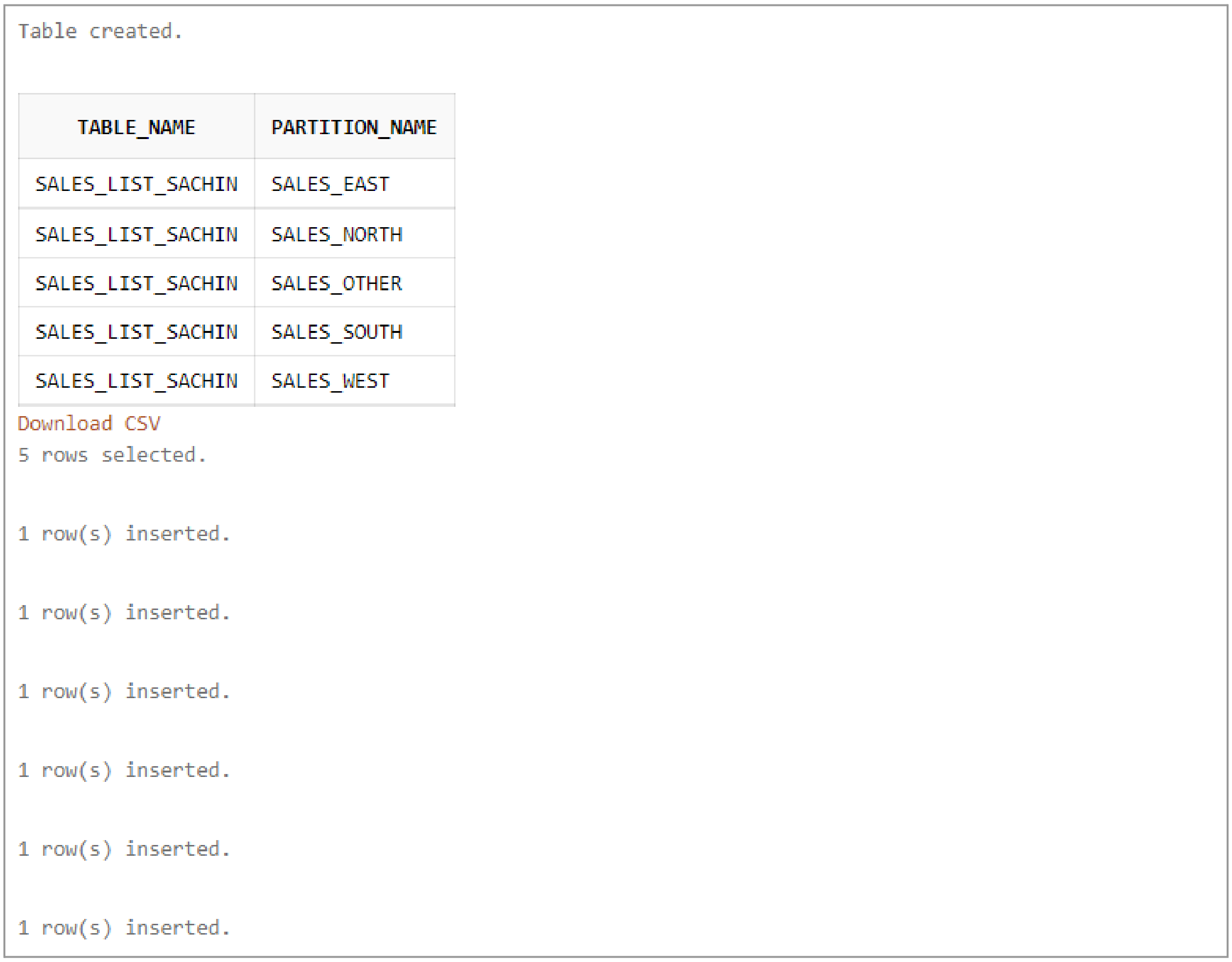
```
create table sales_list_sachin(
salesman_id NUMBER(5),
salesman_name VARCHAR2(30),
sales_state varchar2(30),
sales_amount NUMBER(10),
sales_date DATE)

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_sachin values(1,'sachin bairi','Mumbai',10000,TO_DATE('10/01/2000','DD/MM/YYYY'));
insert into sales_list_sachin values(2,'edvin pillai','Pune',20000,TO_DATE('10/02/2000','DD/MM/YYYY'));
insert into sales_list_sachin values(3,'om navghe','Delhi',30000,TO_DATE('10/03/2000','DD/MM/YYYY'));
insert into sales_list_sachin values(4,'drishti bhatia','Kolkata',40000,TO_DATE('10/04/2000','DD/MM/YYYY'));
insert into sales_list_sachin values(5,'Babu rao','Chennai',50000,TO_DATE('20/04/2000','DD/MM/YYYY'));
insert into sales_list_sachin values(6,'Rupesh mishra','Allahabad',60000,TO_DATE('10/05/2000','DD/MM/YYYY'));

select * from sales_list_sachin;

select * from sales_list_sachin PARTITION(sales_west)
```



```
create table sales_list_sachin(
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_sachin values(1,'sachin bairi','Mumbai',10000,TO_DATE('10/01/2000','DD/MM/YYYY'));
insert into sales_list_sachin values(2,'edvin pillai','Pune',20000,TO_DATE('10/02/2000','DD/MM/YYYY'));
insert into sales_list_sachin values(3,'om navghe','Delhi',30000,TO_DATE('10/03/2000','DD/MM/YYYY'));
insert into sales_list_sachin values(4,'drishti bhatia','Kolkata',40000,TO_DATE('10/04/2000','DD/MM/YYYY'));
insert into sales_list_sachin values(5,'Babu rao','Chennai',50000,TO_DATE('20/04/2000','DD/MM/YYYY'));
insert into sales_list_sachin values(6,'Rupesh mishra','Allahabad',60000,TO_DATE('10/05/2000','DD/MM/YYYY'));

select * from sales_list_sachin;

select * from sales_list_sachin PARTITION(sales_west)
```

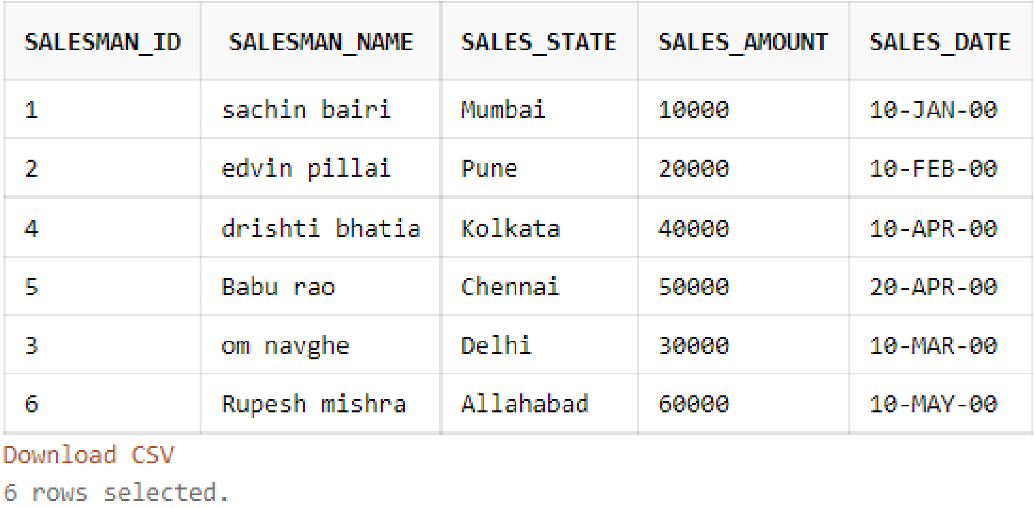
<u>Output :</u>



```
Table created.
```

TABLE_NAME	PARTITION_NAME
SALES_LIST_SACHIN	SALES_EAST
SALES_LIST_SACHIN	SALES_NORTH
SALES_LIST_SACHIN	SALES_OTHER
SALES_LIST_SACHIN	SALES_SOUTH
SALES_LIST_SACHIN	SALES_WEST

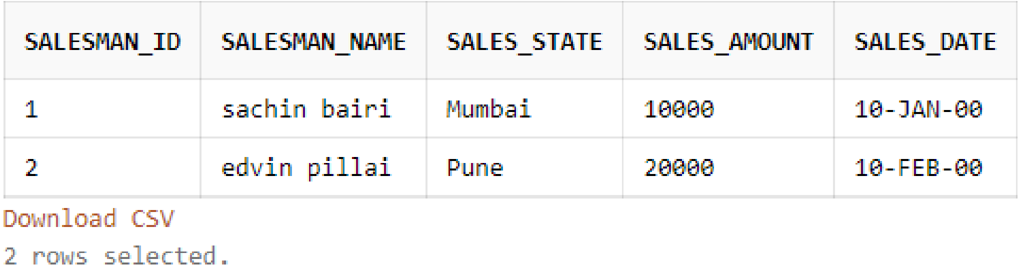
```
Download CSV
5 rows selected.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```



SALESMAN_ID	SALESMAN_NAME	SALES_STATE	SALES_AMOUNT	SALES_DATE
1	sachin bairi	Mumbai	10000	10-JAN-00
2	edvin pillai	Pune	20000	10-FEB-00
4	drishti bhatia	Kolkata	40000	10-APR-00
5	Babu rao	Chennai	50000	20-APR-00
3	om navghe	Delhi	30000	10-MAR-00
6	Rupesh mishra	Allahabad	60000	10-MAY-00

Download CSV

6 rows selected.



SALESMAN_ID	SALESMAN_NAME	SALES_STATE	SALES_AMOUNT	SALES_DATE
1	sachin bairi	Mumbai	10000	10-JAN-00
2	edvin pillai	Pune	20000	10-FEB-00

Download CSV

2 rows selected.