**PRACTICAL NO 9**

**AIM: To study of Parallel and Distributed Database.**

**Theory:**

**Partition:** 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. There are three types of partition, they are Range, List and Hash.

**I. Range Partition:** In MySQL, RANGE partitioning mode allows us to specify various ranges for which data is assigned. Ranges should be contiguous but not overlapping, and are defined using the VALUES LESS THAN operator.

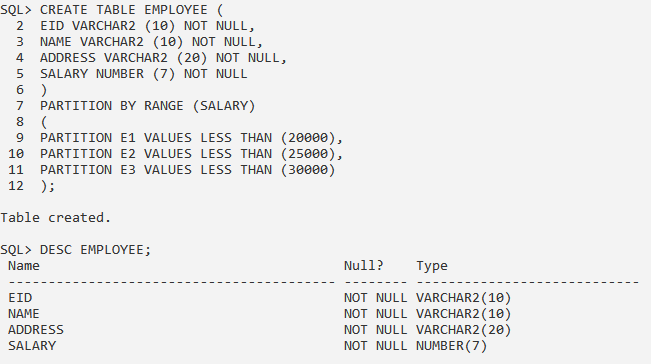
**II. List Partition:** List partition allows us to segment data based on a pre-defined set of values (e.g. 1, 2, 3). This is done by using PARTITION BY LIST (expr) where expr is a column value and then defining each partition by means of a VALUES IN (value\_list), where value\_list is a comma-separated list of integers.

**III. Hash Partition:** MySQL HASH partition is used to distribute data among a predefined number of partitions on a column value or expression based on a column value. This is done by using PARTITION BY HASH (expr) clause, adding in CREATE TABLE STATEMENT. In PARTITIONS num clause, num is a positive integer represents the number of partitions of the table.

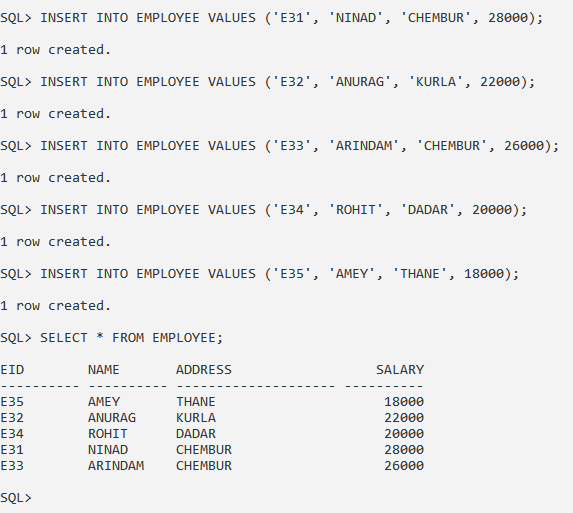
**Distributed Databases:** A distributed database is a collection of multiple interconnected databases and spread physically across various locations. All the interconnected databases communicate with each other over a network. A distributed Database management system manages the distributed database in a manner so that it looks like one single database to users. It is not a loosely connected file system. The main goal of the distributed database is to maximize the performance by distributing data on different network or location. It also helps to utilize IT resources in a cost-effective, reliable and in a transparent manner. It ensures fault tolerance also. This database system enables resource accessibility when any other components fail. This will happen with the help of data replication. Before this concept “distributed database”, data stored in one place that approach known as a centralized database. Firstly, We need to understand centralize database to know the features or the advantages of the distributed database over the centralized database. There are two types of distributed database architecture Homogeneous Distributed Database and Heterogeneous Distributed Database.

**RANGE PARTITION**

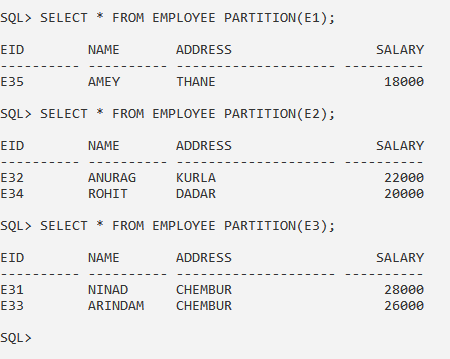
**1. Create a table Employee with attributes EId, Name, Address and Salary. Perform range partition on Salary column with at least three partitions.**

****

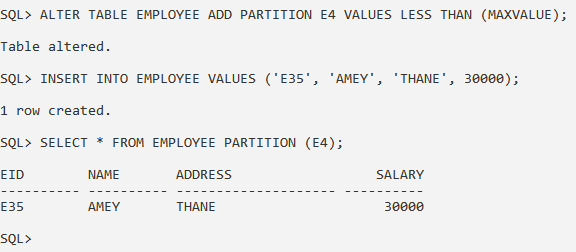
**2. Write a query to insert records into the table.**

****

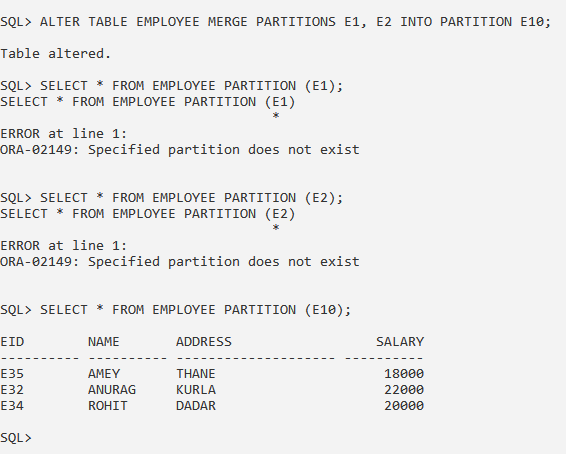
**3. Write a query to display records of different partitions.**

****

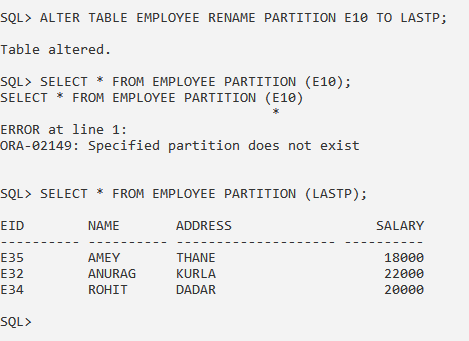
**4. Write a query to add a partition with value as MAXVALUE.**

****

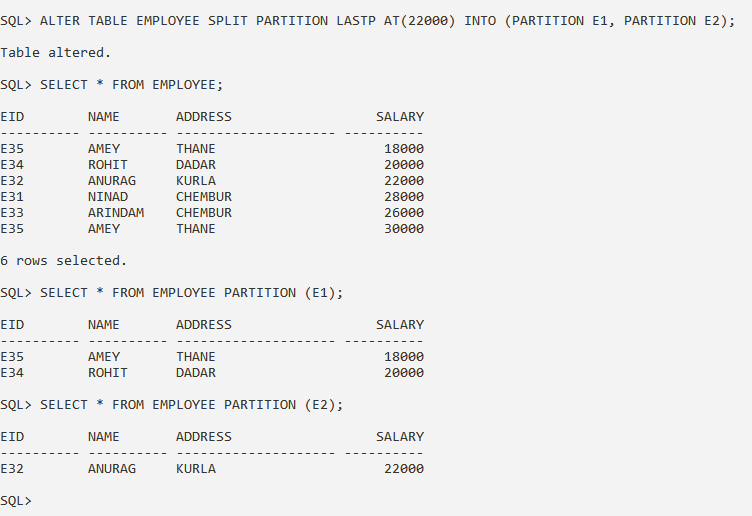
**5. Write a query to merge two partitions.**

****

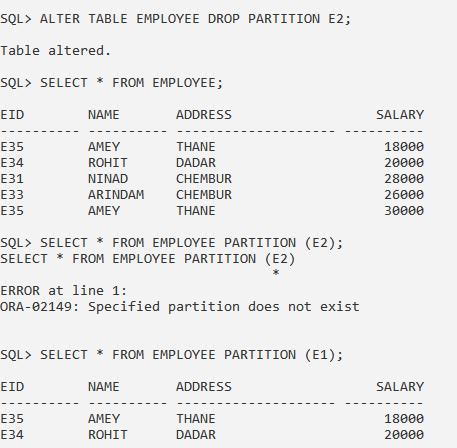
**6. Write a query to rename last partition.**

****

**7. Write a query to split a partition.**

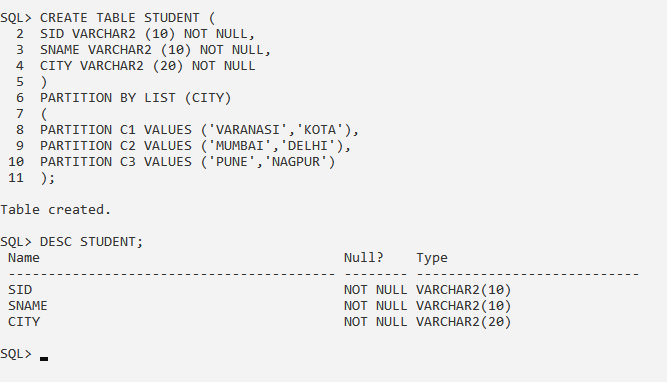
****

**8. Write a query to drop any one of the partitions.**

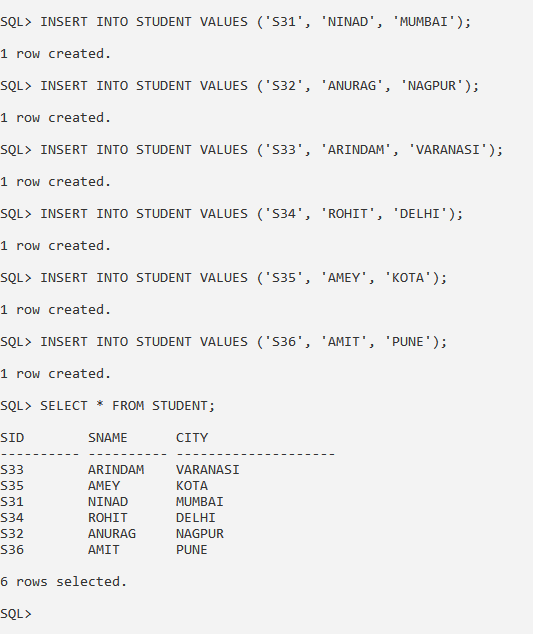
****

**LIST PARTITION**

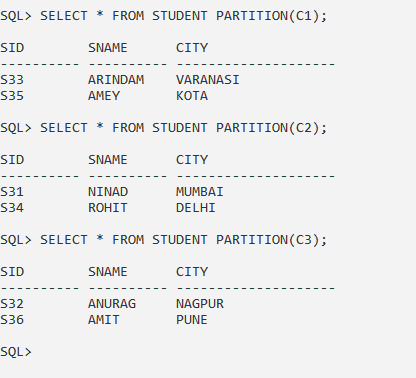
1. Create a table Student with attributes SId, SName and City. Perform list partition on City column with at least three partitions and two values in each partition.



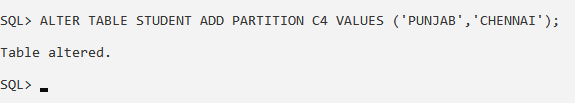
2. Write a query to insert at least two records in each partition.



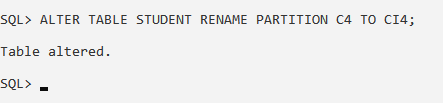
3. Write a query to display records of different partitions.



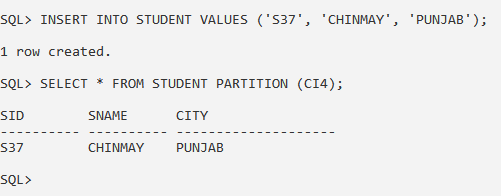
4. Write a query to add a partition.



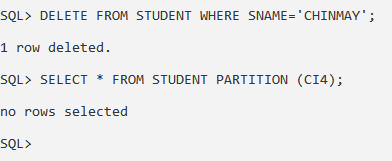
5. Write a query to rename a partition.



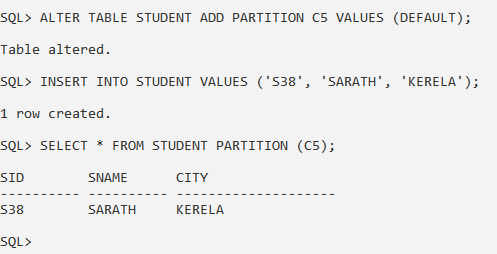
6. Write a query to add a value in a partition.



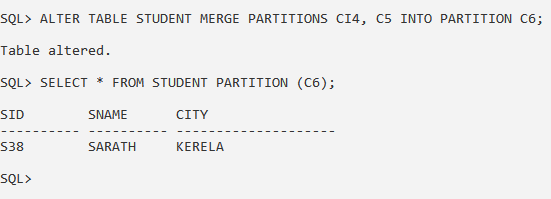
7. Write a query to delete a value from a partition.



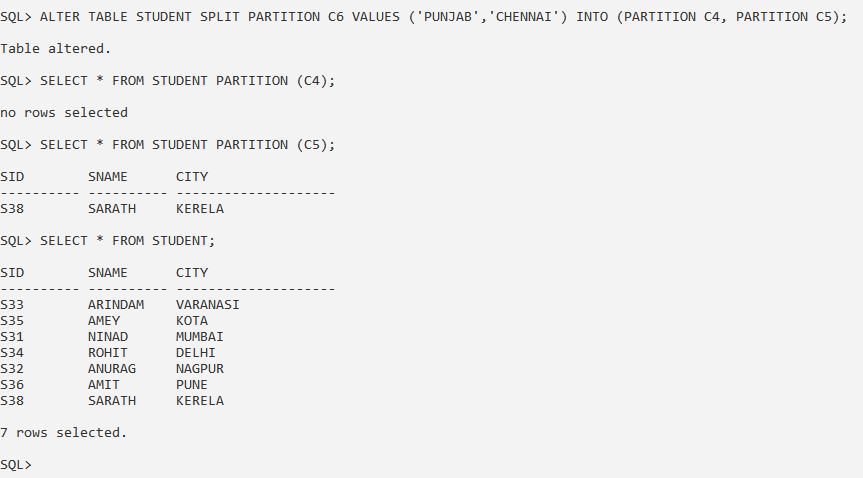
8. Write a query to add a partition with default value.



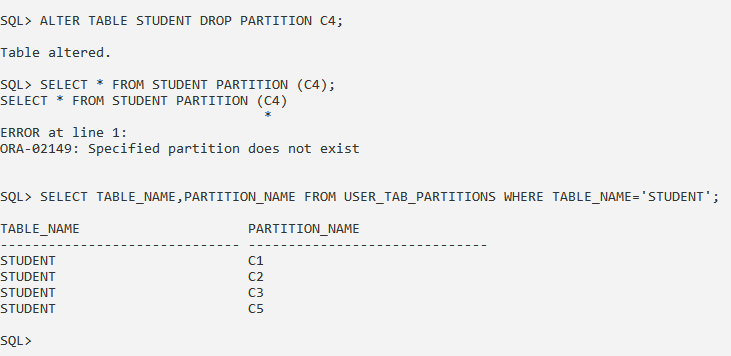
9. Write a query to merge two partitions.



10. Write a query to split a partition.

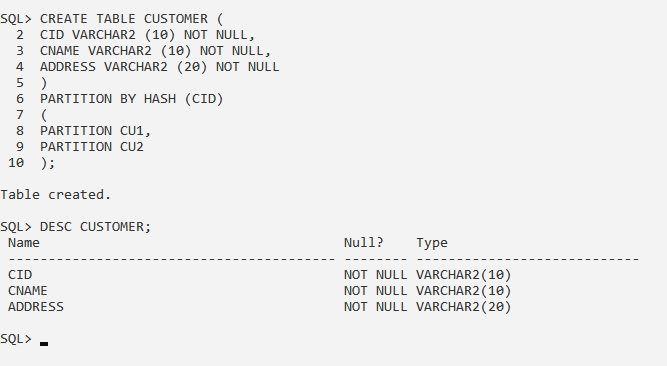


11. Write a query to drop any one of the partitions.

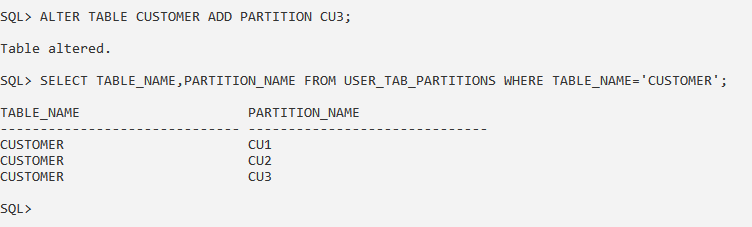


**HASH PARTITION**

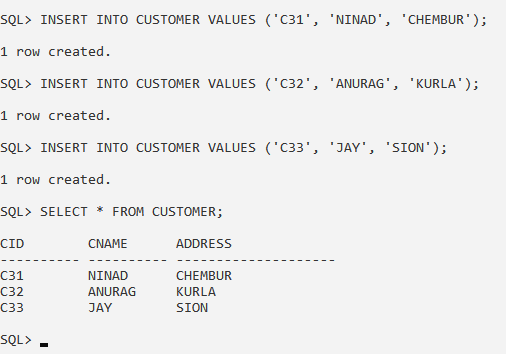
1. Create a table Customer with attributes CId, CName and Address. Perform hash partition on CId column.



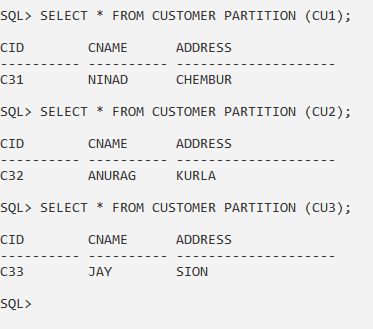
2. Write a query to add a partition.



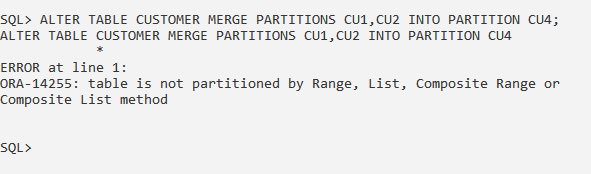
3. Write a query to insert records in the table.



4. Write a query to display records of different partitions.



5. Write a query to merge two partitions.

****

**CONCLUSION:** We have studied the Parallel and Distributed Database in details.