**Partitioning Technique:-**

**package** part;

**import** java.util.Scanner;

**import** com.datastax.driver.core.Cluster;

**import** com.datastax.driver.core.ResultSet;

**import** com.datastax.driver.core.Row;

**import** com.datastax.driver.core.Session;

**public** **class** Partitioning

{

**static** Cluster *cluster*;

**static** Session *session*;

**static** ResultSet *results*;

**static** Row *rows*;

**public** **static** **void** main(String[] args)

{

**try** {

*cluster* = Cluster.*builder*().addContactPoint("localhost").build();

*session* = *cluster*.connect();

*session*.execute("CREATE KEYSPACE IF NOT EXISTS Partition WITH replication " + "= {'class':'SimpleStrategy','replication\_factor':1}; ");

*session*.execute("USE Partition");

*session*.execute("CREATE TABLE if not exists student (rno int PRIMARY KEY, name text, address text, email text, mark int);");

*session*.execute("insert into student (rno, name, address, email, mark) values (101, 'AMIT', 'kamshet', 'amit@gmail.com', 50);");

*session*.execute("insert into student (rno, name, address, email, mark) values (102, 'KAILAS', 'talegaon', 'kailas@gmail.com', 55);");

*session*.execute("insert into student (rno, name, address, email, mark) values (103, 'SUNIL', 'talegaon', 'sunil@gmail.com', 25);");

*session*.execute("insert into student (rno, name, address, email, mark) values (104, 'SUCHIT', 'lonavala', 'suchit@gmail.com', 35);");

*session*.execute("insert into student (rno, name, address, email, mark) values (105, 'SACHIN', 'pune', 'sachin@gmail.com', 45);");

*session*.execute("insert into student (rno, name, address, email, mark) values (106, 'VISHAL', 'pimpri', 'vishal@gmail.com', 55);");

*session*.execute("insert into student (rno, name, address, email, mark) values (107, 'RAHUL', 'kamshet', 'rahul@gmail.com', 65);");

// partitioing

//Round Robin

*results* = *session*.execute("SELECT \* FROM student ");

**int** i = 0, disk\_no = 0, total\_disks = 4;

System.***out***.println("\nRound Robin Partitioning");

**for** (Row row : *results*)

{

disk\_no = i++ % total\_disks;

System.***out***.println("Disk: "+ disk\_no+"\t\t "+row.getInt("rno")+" "+row.getString("name")+" "+row.getString("address")+" "+row.getString("email")+" "+ row.getInt("mark"));

}

//Hash

**int** age;

*results* = *session*.execute("SELECT \* FROM student ");

System.***out***.println("\nHash Partitioning");

**for** (Row row : *results*)

{

age = row.getInt("mark"); //partitioning attribute

disk\_no = age % total\_disks;

System.***out***.println("Disk: "+ disk\_no+"\t\t "+row.getInt("rno")+" "+row.getString("name")+" "+row.getString("address")+" "+row.getString("email")+" "+ row.getInt("mark"));

}

//Range Partitioning

**int** v, n;

*results* = *session*.execute("SELECT \* FROM student ");

n = total\_disks;

**int**[] p\_vector = **new** **int**[n - 1];

**int** flag;

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("\nRange Partitioning");

System.***out***.println("\nPlease enter Partitioning vector values (mark range):- ");

**for** (i = 0; i <= n - 2; i++)

{

System.***out***.println("Enter v" + i + " : ");

p\_vector[i] = Integer.*parseInt*(sc.next());

}

System.***out***.println("\nRange Partitioning - output \n");

**for** (Row row : *results*)

{

v = row.getInt("mark");

flag = 0;

**for** (i = 0; i < n - 2; i++)

{

**if** (p\_vector[i] <= v && v < p\_vector[i + 1])

{

disk\_no = i + 1;

flag = 1;

**break**;

}

}

**if** (flag == 0)

{

**if** (v < p\_vector[0])

{

disk\_no = 0;

} **else** **if** (v >= p\_vector[n - 2])

{

disk\_no = n - 1;

}

}

System.***out***.println("Disk: "+ disk\_no+"\t\t "+row.getInt("rno")+" "+row.getString("name")+" "+row.getString("address")+" "+row.getString("email")+" "+ row.getInt("mark"));

}

sc.close();

} **catch** (Exception e) {

System.***out***.println("Error: " + e.getMessage());}

*cluster*.close();

}

}

**Output:-**

Round Robin Partitioning

Disk: 0 105 SACHIN pune sachin@gmail.com 45

Disk: 1 107 RAHUL kamshet rahul@gmail.com 65

Disk: 2 104 SUCHIT lonavala suchit@gmail.com 35

Disk: 3 102 KAILAS talegaon kailas@gmail.com 55

Disk: 0 106 VISHAL pimpri vishal@gmail.com 55

Disk: 1 101 AMIT kamshet amit@gmail.com 50

Disk: 2 103 SUNIL talegaon sunil@gmail.com 25

Hash Partitioning

Disk: 1 105 SACHIN pune sachin@gmail.com 45

Disk: 1 107 RAHUL kamshet rahul@gmail.com 65

Disk: 3 104 SUCHIT lonavala suchit@gmail.com 35

Disk: 3 102 KAILAS talegaon kailas@gmail.com 55

Disk: 3 106 VISHAL pimpri vishal@gmail.com 55

Disk: 2 101 AMIT kamshet amit@gmail.com 50

Disk: 1 103 SUNIL talegaon sunil@gmail.com 25

Range Partitioning

Please enter Partitioning vector values (mark range):-

Enter v0 :

40

Enter v1 :

50

Enter v2 :

60

Range Partitioning - output

Disk: 1 105 SACHIN pune sachin@gmail.com 45

Disk: 3 107 RAHUL kamshet rahul@gmail.com 65

Disk: 0 104 SUCHIT lonavala suchit@gmail.com 35

Disk: 2 102 KAILAS talegaon kailas@gmail.com 55

Disk: 2 106 VISHAL pimpri vishal@gmail.com 55

Disk: 2 101 AMIT kamshet amit@gmail.com 50

Disk: 0 103 SUNIL talegaon sunil@gmail.com 25