Name : Kreena Shah
Sapid : 60004210243
Batch : Comps C’32
Subject : Advance Database Management System (ADBMS)

Experiment No 5 - Distributed Database Design

Aim : Perform Fragmentation (Range, List, Hash and Key) in DDBS design.

Write Ups :

Koreena Shah - 60004210244
Date of Submission - 26/10/2023
Date of Performance - 02/11/2023

Page No. (1)
Date

26/10/2023 ADMS - Exp 5 5/15

Aim : To implement Fragmentation (Range, List, Key, Hash)

Theory :

Fragmentation

(1) Fragmentation refers to the division of data & database objects into smaller, more manageable pieces
(2) Fragmentation must be done in such a way that the database can be reconstructed from the fragments
(3) There are two types of fragmentation :
- Horizontal Fragmentation
- Vertical Fragmentation

Horizontal Fragmentation

It involves dividing a table into subsets of rows based on specific condition. It is useful for distributing data in distributed database. The fragments can be then reconstructed using simple union operation.

Vertical Fragmentation

This type divides a table into subsets of columns or attributes. Each subset contains a different set of attributes It is used to improve query performance & manage data access. The fragments can then be reconstructed using JOIN operation

Mixed / Hybrid Fragmentation

Mixed Fragmentation combines both horizontal & vertical

Page No.
Date

fragmentation allowing for more granular control over data distribution in complex distributed database systems. It is used when data needs to be distributed with different structures in various locations.

Partitioning

(1) It is a MySQL database feature that can be used to improve the performance & manageability of your database. It enables you to define groups of rows or index keys within a table according to some algorithm or scheme. You can store each group/fragment/partition in a separate space associated with a specific physical disk. One can use SQL statements to create the fragments & assign them to space.

Types :
⇒ Range Partitioning
⇒ List Partitioning
⇒ Key Partitioning
⇒ Hash Partitioning

Range Partitioning
Here data is partitioned based on a specific range of values for a chosen value/column. It is suitable for data that can be logically divided into non-overlapping ranges

List Partitioning
Here, data is classified into partiti partitions based on

Page No.
Date

specific values for a selected column. It defines a list of discrete values for this column & data rows are placed into partitions according to matching values

Key Partitioning

Here, partition of data is done using a combination of columns. The data is divided based on a function of multiple columns & partitions are defined by the columns i.e. identified by the result of this function.

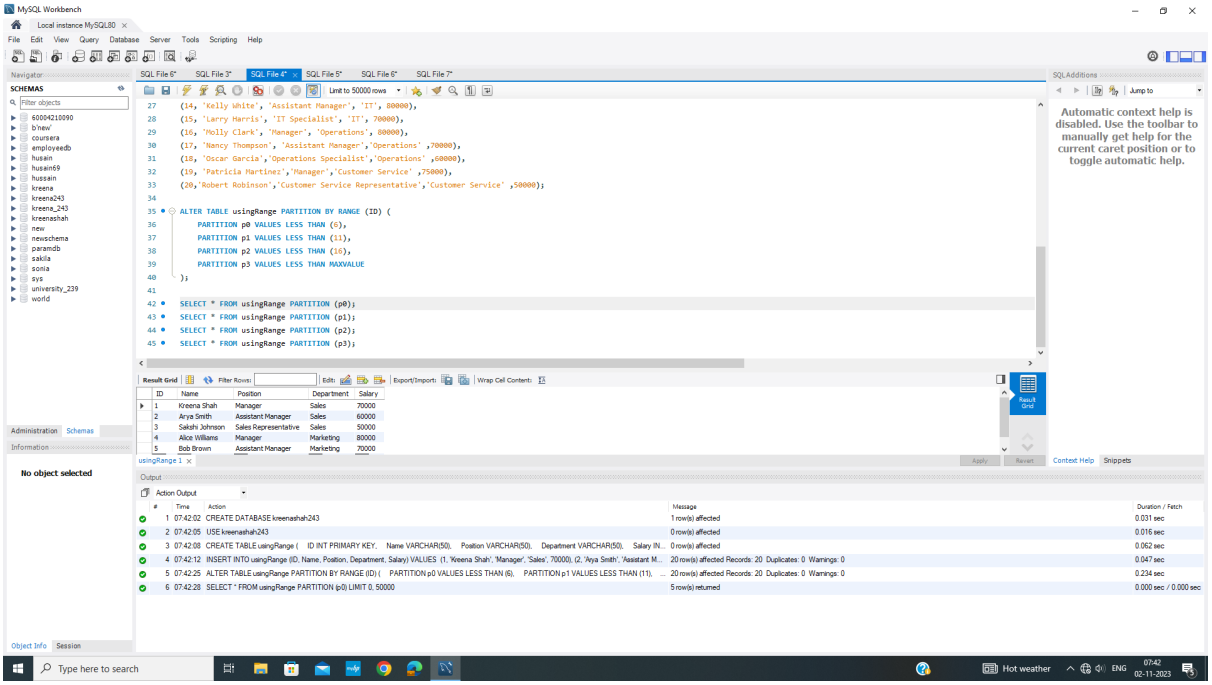
Hash Partitioning

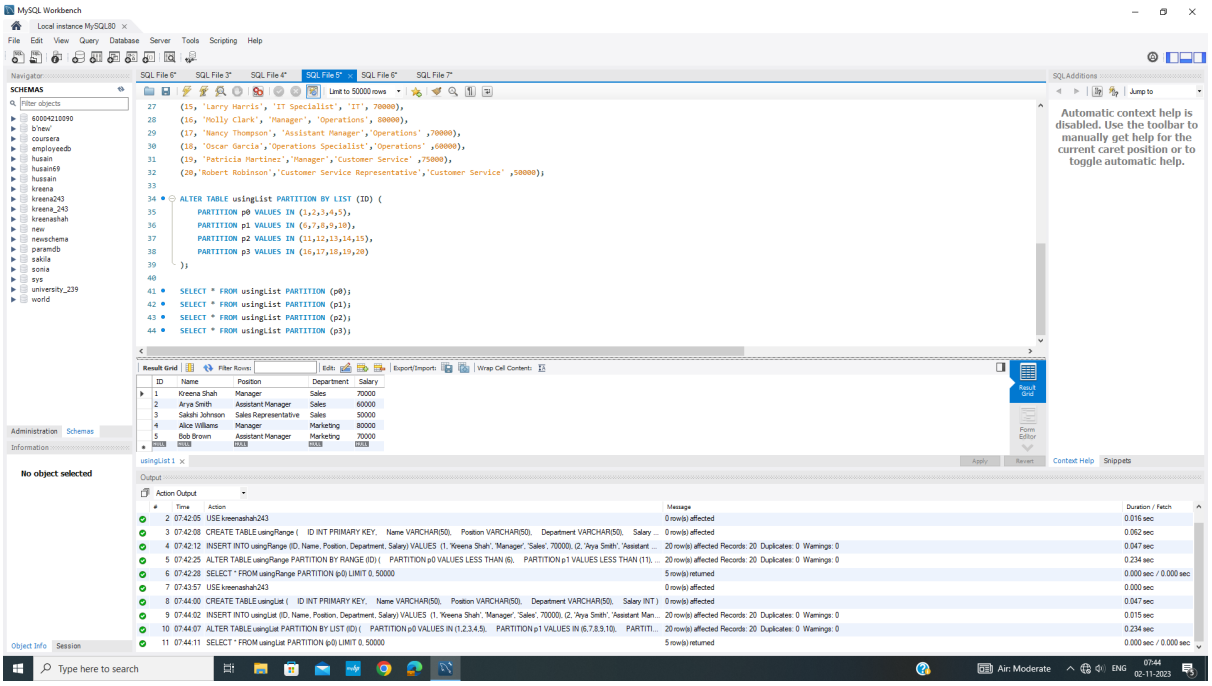
This involves partitioning data based on a hash function applied to a chosen columns.

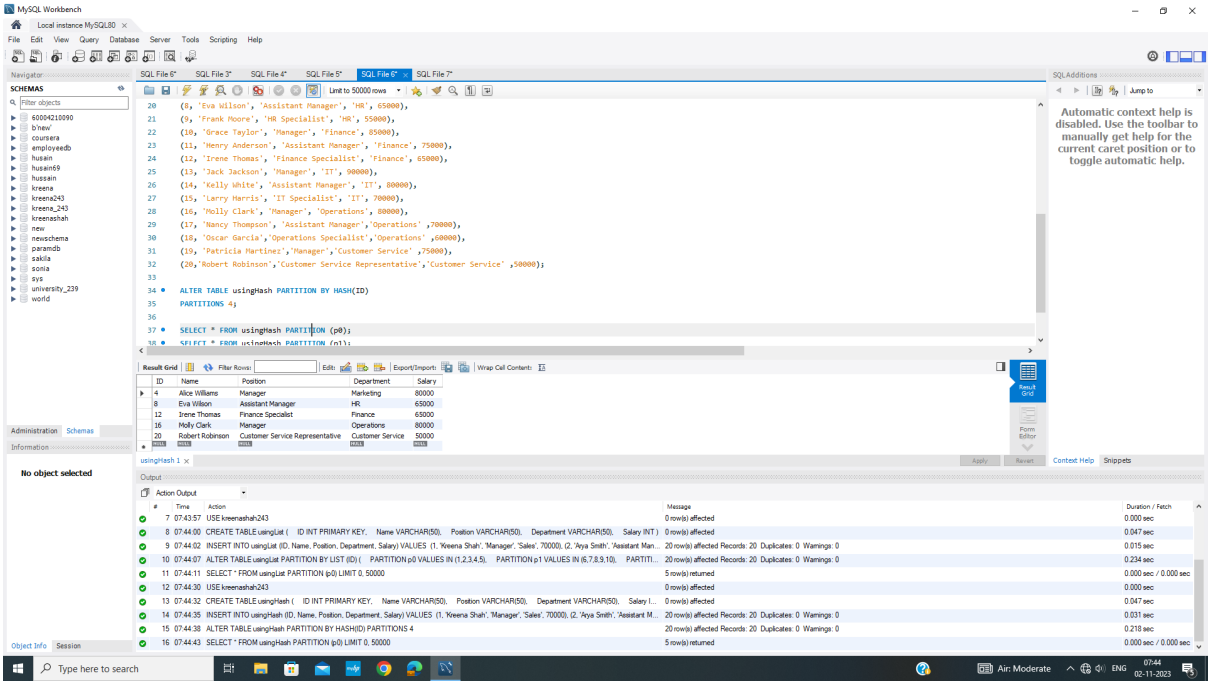
Conclusion :

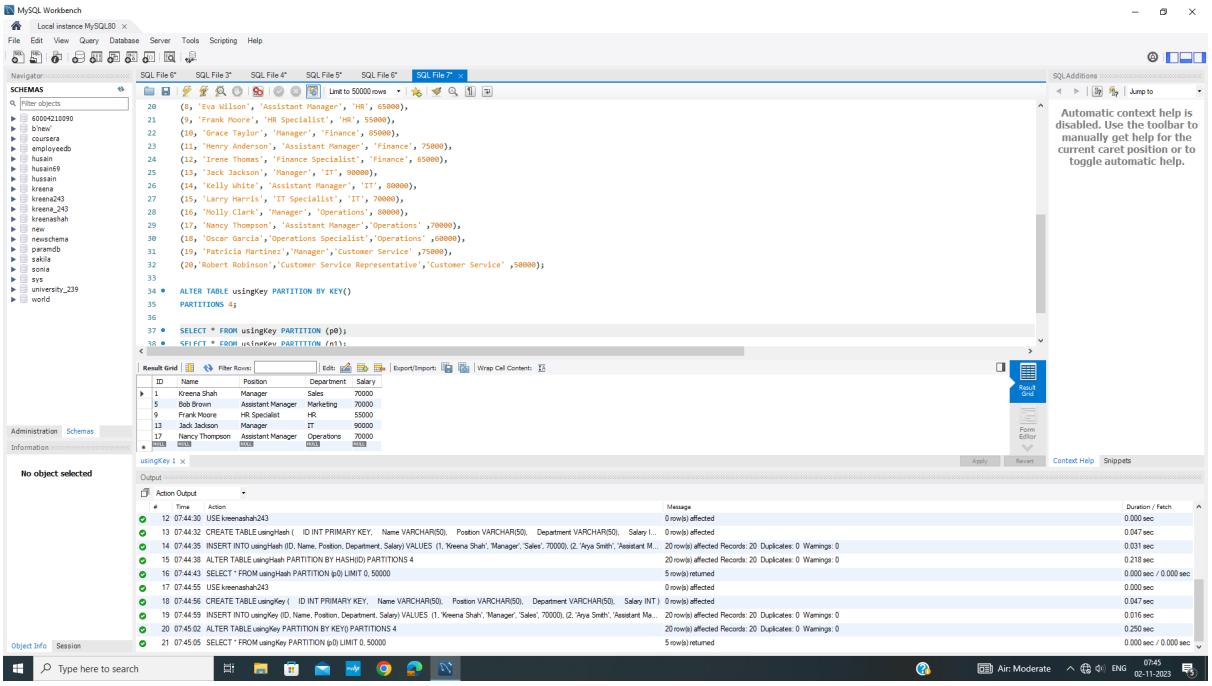
Implementing fragmentation using various techniques like range, list, key, hash partitioning is vital for optimizing database performance & data distribution. These methods enhances data organization & retrieval, benefiting data availability, scalability & performance

Screenshots :


MySQL Workbench
(14, 'Kelly White', 'Assistant Manager', 'IT', 80000),
(15, 'Larry Harris', 'IT Specialist', 'IT', 70000),
(16, 'Molly Clark', 'Manager', 'Operations', 80000),
(17, 'Nancy Thompson', 'Assistant Manager','Operations' ,70000),
(18, 'Oscar Garcia','Operations Specialist','Operations' ,60000),
(19, 'Patricia Martinez','Manager','Customer Service' ,75000),
(20,'Robert Robinson','Customer Service Representative','Customer Service' ,50000);
ALTER TABLE usingRange PARTITION BY RANGE (ID) (
PARTITION p0 VALUES LESS THAN (6),
PARTITION p1 VALUES LESS THAN (11),
PARTITION p2 VALUES LESS THAN (16),
PARTITION p3 VALUES LESS THAN MAXVALUE
);
SELECT * FROM usingRange PARTITION (p0);
SELECT * FROM usingRange PARTITION (p1);
SELECT * FROM usingRange PARTITION (p2);
SELECT * FROM usingRange PARTITION (p3);
No object selected


MySQL Workbench
(15, 'Larry Harris', 'IT Specialist', 'IT', 70000),
(16, 'Molly Clark', 'Manager', 'Operations', 80000),
(17, 'Nancy Thompson', 'Assistant Manager','Operations' ,70000),
(18, 'Oscar Garcia','Operations Specialist','Operations' ,60000),
(19, 'Patricia Martinez','Manager','Customer Service' ,75000),
(20,'Robert Robinson','Customer Service Representative','Customer Service' ,50000);
ALTER TABLE usingList PARTITION BY LIST (ID) (
PARTITION p0 VALUES IN (1,2,3,4,5),
PARTITION p1 VALUES IN (6,7,8,9,10),
PARTITION p2 VALUES IN (11,12,13,14,15),
PARTITION p3 VALUES IN (16,17,18,19,20)
);
SELECT * FROM usingList PARTITION (p0);
SELECT * FROM usingList PARTITION (p1);
SELECT * FROM usingList PARTITION (p2);
SELECT * FROM usingList PARTITION (p3);
No object selected


MySQL Workbench
(8, 'Eva Wilson', 'Assistant Manager', 'HR', 65000),
(9, 'Frank Moore', 'HR Specialist', 'HR', 55000),
(10, 'Grace Taylor', 'Manager', 'Finance', 85000),
(11, 'Henry Anderson', 'Assistant Manager', 'Finance', 75000),
(12, 'Irene Thomas', 'Finance Specialist', 'Finance', 65000),
(13, 'Jack Jackson', 'Manager', 'IT', 90000),
(14, 'Kelly White', 'Assistant Manager', 'IT', 80000),
(15, 'Larry Harris', 'IT Specialist', 'IT', 70000),
(16, 'Molly Clark', 'Manager', 'Operations', 80000),
(17, 'Nancy Thompson', 'Assistant Manager','Operations' ,70000),
(18, 'Oscar Garcia','Operations Specialist','Operations' ,60000),
(19, 'Patricia Martinez','Manager','Customer Service' ,75000),
(20,'Robert Robinson','Customer Service Representative','Customer Service' ,50000);
ALTER TABLE usingHash PARTITION BY HASH(ID)
PARTITIONS 4;
SELECT * FROM usingHash PARTITION (p0);
No object selected


MySQL Workbench
(8, 'Eva Wilson', 'Assistant Manager', 'HR', 65000),
(9, 'Frank Moore', 'HR Specialist', 'HR', 55000),
(10, 'Grace Taylor', 'Manager', 'Finance', 85000),
(11, 'Henry Anderson', 'Assistant Manager', 'Finance', 75000),
(12, 'Irene Thomas', 'Finance Specialist', 'Finance', 65000),
(13, 'Jack Jackson', 'Manager', 'IT', 90000),
(14, 'Kelly White', 'Assistant Manager', 'IT', 80000),
(15, 'Larry Harris', 'IT Specialist', 'IT', 70000),
(16, 'Molly Clark', 'Manager', 'Operations', 80000),
(17, 'Nancy Thompson', 'Assistant Manager','Operations' ,70000),
(18, 'Oscar Garcia','Operations Specialist','Operations' ,60000),
(19, 'Patricia Martinez','Manager','Customer Service' ,75000),
(20,'Robert Robinson','Customer Service Representative','Customer Service' ,50000);
ALTER TABLE usingKey PARTITION BY KEY()
PARTITIONS 4;
SELECT * FROM usingKey PARTITION (p0);
No object selected