a very simple code in mysql to Data Partitioning in a Database , show right from creation of db, table, and trasaction step by step, a table with just 2 to 3 fields

**EX.NO:1 :Write a Program to Perform Transaction Processing for a Database**

mysql> CREATE DATABASE IF NOT EXISTS example\_db;

Query OK, 1 row affected (0.01 sec)

mysql> USE example\_db;

Database changed

mysql> CREATE TABLE IF NOT EXISTS transactions (transaction\_id INT AUTO\_INCREMENT PRIMARY KEY,description VARCHAR(255),amount DECIMAL(10, 2));

Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> INSERT INTO transactions (description, amount) VALUES('Initial deposit', 1000.00),('Purchase of goods', -500.50),('Salary credit', 1500.00);

Query OK, 3 rows affected (0.01 sec)

Records: 3 Duplicates: 0 Warnings: 0

mysql> SELECT \* FROM transactions;

+----------------+-------------------+---------+

| transaction\_id | description | amount |

+----------------+-------------------+---------+

| 1 | Initial deposit | 1000.00 |

| 2 | Purchase of goods | -500.50 |

| 3 | Salary credit | 1500.00 |

+----------------+-------------------+---------+

3 rows in set (0.00 sec)

mysql> START TRANSACTION;

Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE transactions SET amount = amount - 50.00 WHERE transaction\_id = 2;

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE transactions SET amount = amount - 50.00 WHERE transaction\_id = 2;

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> SELECT \* FROM transactions;

+----------------+-------------------+---------+

| transaction\_id | description | amount |

+----------------+-------------------+---------+

| 1 | Initial deposit | 1000.00 |

| 2 | Purchase of goods | -600.50 |

| 3 | Salary credit | 1500.00 |

+----------------+-------------------+---------+

3 rows in set (0.00 sec)

mysql> COMMIT;

Query OK, 0 rows affected (0.00 sec)

mysql> select \* from transactions

-> ;

+----------------+-------------------+---------+

| transaction\_id | description | amount |

+----------------+-------------------+---------+

| 1 | Initial deposit | 1000.00 |

| 2 | Purchase of goods | -600.50 |

| 3 | Salary credit | 1500.00 |

+----------------+-------------------+---------+

3 rows in set (0.00 sec)

mysql>

**Ex No. 2Write a Program to do Data Partitioning in a Database**

CREATE DATABASE IF NOT EXISTS partitioning\_example;

Query OK, 1 row affected, 1 warning (0.00 sec)

mysql> USE partitioning\_example;

Database changed

mysql>

mysql> CREATE TABLE IF NOT EXISTS partitioned\_data (

id INT AUTO\_INCREMENT PRIMARY KEY,

description VARCHAR(255),

amount DECIMAL(10, 2),

transaction\_date DATE)

-> PARTITION BY RANGE (YEAR(transaction\_date)) (

-> PARTITION p0 VALUES LESS THAN (2020),

-> PARTITION p1 VALUES LESS THAN (2021),

-> PARTITION p2 VALUES LESS THAN (2022),

-> PARTITION p3 VALUES LESS THAN MAXVALUE

-> );

Query OK, 0 rows affected, 1 warning (0.01 sec)

mysql> INSERT INTO partitioned\_data (description, amount, transaction\_date) VALUES

-> ('Initial deposit', 1000.00, '2020-01-01'),

-> ('Purchase of goods', -500.50, '2021-03-15'),

-> ('Salary credit', 1500.00, '2022-05-20');

Query OK, 3 rows affected (0.00 sec)

Records: 3 Duplicates: 0 Warnings: 0

mysql> SELECT \* FROM partitioned\_data;

+----+-------------------+---------+------------------+

| id | description | amount | transaction\_date |

+----+-------------------+---------+------------------+

| 1 | Initial deposit | 1000.00 | 2020-01-01 |

| 2 | Purchase of goods | -500.50 | 2021-03-15 |

| 3 | Salary credit | 1500.00 | 2022-05-20 |

+----+-------------------+---------+------------------+

3 rows in set (0.00 sec)

SELECT \* FROM partitioned\_data WHERE YEAR(transaction\_date) < 2020;

Empty set (0.00 sec)

mysql> SELECT \* FROM partitioned\_data WHERE YEAR(transaction\_date) > 2020;

+----+-------------------+---------+------------------+

| id | description | amount | transaction\_date |

+----+-------------------+---------+------------------+

| 2 | Purchase of goods | -500.50 | 2021-03-15 |

| 3 | Salary credit | 1500.00 | 2022-05-20 |

+----+-------------------+---------+------------------+

1. rows in set (0.00 sec)

**EX.NO:3 Write a Program to Perform Parallel Indexing for a Database**

CREATE DATABASE IF NOT EXISTS parallel\_indexing\_example;

Query OK, 1 row affected (0.01 sec)

mysql> USE parallel\_indexing\_example;

Database changed

mysql> CREATE TABLE IF NOT EXISTS my\_table (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255),

value INT);

Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO my\_table (name, value) VALUES

-> ('John', 100),

-> ('Jane', 150),

-> ('Bob', 200),

-> ('Alice', 120),

-> ('Charlie', 180);

Query OK, 5 rows affected (0.01 sec)

Records: 5 Duplicates: 0 Warnings: 0

mysql> set session transaction isolation level serializable;

Query OK, 0 rows affected (0.00 sec)

mysql> START TRANSACTION;

Query OK, 0 rows affected (0.00 sec)

mysql> CREATE INDEX idx\_value\_parallel ON my\_table (value)

->ALGORITHM=INPLACE, LOCK=NONE;

mysql> COMMIT;

Query OK, 0 rows affected (0.00 sec)

mysql> SHOW INDEX FROM my\_table;

+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+

| Table | Non\_unique | Key\_name | Seq\_in\_index | Column\_name | Collation | Cardinality | Sub\_part | Packed | Null | Index\_type | Comment | Index\_comment |

+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+

| my\_table | 0 | PRIMARY | 1 | id | A | 5 | NULL | NULL | | BTREE | | |

+----------+------------+----------+--------------+-------------+-----------+-------------+----------+--------+------+------------+---------+---------------+

1. row in set (0.00 sec)

**EX.NO:4 Write a Program to do Parallel Sort in a Database**

CREATE DATABASE IF NOT EXISTS mydatabase;

Query OK, 1 row affected (0.00 sec)

mysql> USE mydatabase;

Database changed

mysql> CREATE TABLE mytable1 (

-> id INT AUTO\_INCREMENT PRIMARY KEY,

-> name VARCHAR(50),

-> age INT

-> );

Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO mytable (name, age) VALUES

-> ('Alice', 25),

-> ('Bob', 30),

-> ('Charlie', 22),

-> ('David', 28);

Query OK, 4 rows affected (0.01 sec)

Records: 4 Duplicates: 0 Warnings: 0

select \* from mytable;

+----+---------+------+

| id | name | age |

+----+---------+------+

| 1 | Alice | 25 |

| 2 | Bob | 30 |

| 3 | Charlie | 22 |

| 4 | David | 28 |

+----+---------+------+

4 rows in set (0.00 sec)

mysql> SELECT \* FROM mytable ORDER BY age;

+----+---------+------+

| id | name | age |

+----+---------+------+

| 3 | Charlie | 22 |

| 1 | Alice | 25 |

| 4 | David | 28 |

| 2 | Bob | 30 |

+----+---------+------+

1. rows in set (0.00 sec)

**EX.NO: 5 Write a Program to do Parallel Join in a Database**

USE mydatabase;

Database changed

mysql> CREATE TABLE users (

-> user\_id INT AUTO\_INCREMENT PRIMARY KEY,

-> username VARCHAR(50)

-> );

Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE orders (

-> order\_id INT AUTO\_INCREMENT PRIMARY KEY,

-> user\_id INT,

-> product\_name VARCHAR(50),

-> );

Query OK, 0 rows affected (0.02 sec)

mysql> INSERT INTO users (username) VALUES

-> ('Alice'),

-> ('Bob'),

-> ('Charlie');

Query OK, 3 rows affected (0.01 sec)

Records: 3 Duplicates: 0 Warnings: 0

mysql> INSERT INTO orders (user\_id, product\_name, amount) VALUES

-> (1, 'Laptop', 1200.50),

-> (2, 'Phone', 599.99),

-> (1, 'Headphones', 89.99),

-> (3, 'Tablet', 299.99);

Query OK, 4 rows affected (0.01 sec)

Records: 4 Duplicates: 0 Warnings: 0

mysql> select \* from orders;

+----------+---------+--------------+---------+

| order\_id | user\_id | product\_name | amount |

+----------+---------+--------------+---------+

| 1 | 1 | Laptop | 1200.50 |

| 2 | 2 | Phone | 599.99 |

| 3 | 1 | Headphones | 89.99 |

| 4 | 3 | Tablet | 299.99 |

+----------+---------+--------------+---------+

4 rows in set (0.00 sec)

mysql> select \* from users;

+---------+----------+

| user\_id | username |

+---------+----------+

| 1 | Alice |

| 2 | Bob |

| 3 | Charlie |

+---------+----------+

3 rows in set (0.00 sec)

mysql> SELECT users.user\_id, users.username, orders.order\_id, orders.product\_name, orders.amount FROM users JOIN orders ON users.user\_id = orders.user\_id;

+---------+----------+----------+--------------+---------+

| user\_id | username | order\_id | product\_name | amount |

+---------+----------+----------+--------------+---------+

| 1 | Alice | 1 | Laptop | 1200.50 |

| 2 | Bob | 2 | Phone | 599.99 |

| 1 | Alice | 3 | Headphones | 89.99 |

| 3 | Charlie | 4 | Tablet | 299.99 |

+---------+----------+----------+--------------+---------+

1. rows in set (0.00 sec)

**EX.NO:6 Write a Program to Perform Distributed Transactions for a Database**

CREATE DATABASE IF NOT EXISTS distributed\_transactions;

Query OK, 1 row affected (0.01 sec)

mysql> USE distributed\_transactions;

Database changed

mysql> CREATE TABLE IF NOT EXISTS orders (

-> order\_id INT PRIMARY KEY,

-> product\_name VARCHAR(255),

-> quantity INT

-> );

Query OK, 0 rows affected (0.02 sec)

mysql> INSERT INTO orders (order\_id, product\_name, quantity) VALUES

-> (1, 'Product A', 10),

-> (4, 'Product B', 5);

Query OK, 2 rows affected (0.01 sec)

Records: 2 Duplicates: 0 Warnings: 0

mysql> **START TRANSACTION;**

Query OK, 0 rows affected (0.00 sec)

mysql> select \* from orders;

+----------+--------------+----------+

| order\_id | product\_name | quantity |

+----------+--------------+----------+

| 1 | Product A | 10 |

| 2 | Product B | 5 |

+----------+--------------+----------+

2 rows in set (0.00 sec)

mysql> UPDATE orders SET quantity = 8 WHERE order\_id = 1;

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE orders SET quantity = 3 WHERE order\_id = 2;

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> select \* from orders;

+----------+--------------+----------+

| order\_id | product\_name | quantity |

+----------+--------------+----------+

| 1 | Product A | 8 |

| 2 | Product B | 3 |

+----------+--------------+----------+

2 rows in set (0.00 sec)

mysql> commit;

Query OK, 0 rows affected (0.00 sec)

**EX.NO:7 Write a Program to Perform Concurrency Control for a Database**

CREATE DATABASE IF NOT EXISTS concurrency\_example;

Query OK, 1 row affected (0.01 sec)

mysql> USE concurrency\_example;

Database changed

mysql> CREATE TABLE IF NOT EXISTS inventory (

-> product\_id INT PRIMARY KEY,

-> product\_name VARCHAR(255),

-> quantity INT

-> );

Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO inventory (product\_id, product\_name, quantity) VALUES

-> (3, 'Product A', 10),

-> (4, 'Product B', 5);

Query OK, 2 rows affected (0.01 sec)

Records: 2 Duplicates: 0 Warnings: 0

mysql> START TRANSACTION;

Query OK, 0 rows affected (0.00 sec)

mysql> select \* from inventory;

+------------+--------------+----------+

| product\_id | product\_name | quantity |

+------------+--------------+----------+

| 1 | Product A | 10 |

| 2 | Product B | 5 |

+------------+--------------+----------+

2 rows in set (0.00 sec)

mysql> DO SLEEP(5);

Query OK, 0 rows affected (5.01 sec)

mysql> UPDATE inventory SET quantity = quantity - 2 WHERE product\_id = 1;

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> UPDATE inventory SET quantity = quantity + 2 WHERE product\_id = 2;

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> COMMIT;

Query OK, 0 rows affected (0.00 sec)

mysql> select \* from inventory;

+------------+--------------+----------+

| product\_id | product\_name | quantity |

+------------+--------------+----------+

| 1 | Product A | 8 |

| 2 | Product B | 7 |

+------------+--------------+----------+

2 rows in set (0.00 sec)

**8.Write a Program to do Replication in a Database**

CREATE DATABASE IF NOT EXISTS my\_database;

mysql> USE my\_database;

Database changed

mysql> CREATE TABLE IF NOT EXISTS my\_table (

-> id INT AUTO\_INCREMENT PRIMARY KEY,

-> name VARCHAR(50),

-> age INT

-> );

mysql> INSERT INTO my\_table (name, age) VALUES ('Alice', 25);

mysql> INSERT INTO my\_table (name, age) VALUES ('Bob', 30);

mysql> select \* from my\_table;

+----+-------+------+

| id | name | age |

+----+-------+------+

| 2 | Alice | 25 |

| 3 | Bob | 30 |

+----+-------+------+

mysql> UPDATE my\_table SET age = 35 WHERE name = 'Bob';

Rows matched: 1 Changed: 1 Warnings: 0

mysql> select \* from my\_table;

+----+-------+------+

| id | name | age |

+----+-------+------+

| 2 | Alice | 25 |

| 3 | Bob | 35 |

+----+-------+------+

mysql> DELETE FROM my\_table WHERE name = 'Alice';

mysql> select \* from my\_table;

+----+------+------+

| id | name | age |

+----+------+------+

| 3 | Bob | 35 |

+----+------+------+

mysql> SHOW MASTER STATUS;

Empty set (0.00 sec)

mysql> select \* from my\_table;

+----+------+------+

| id | name | age |

+----+------+------+

| 3 | Bob | 35 |

+----+------+------+

1 row in set (0.00 sec)

mysql> SHOW SLAVE STATUS;

+----------------------+-----------------------+------------------+-------------+---------------+------------------+---------------------+----------------------------------+---------------+-----------------------+------------------+-------------------+-----------------+---------------------+--------------------+------------------------+-------------------------+-----------------------------+------------+------------+--------------+---------------------+-----------------+-----------------+----------------+---------------+--------------------+--------------------+--------------------+-----------------+-------------------+----------------+-----------------------+-------------------------------+---------------+--------------------------------------------------------------------------------------------------------+----------------+----------------+-----------------------------+------------------+-------------+--------------------------------------------------------+-----------+---------------------+--------------------------------------------------------+--------------------+-------------+-------------------------+--------------------------+----------------+--------------------+--------------------+-------------------+---------------+----------------------+--------------+--------------------+

| Slave\_IO\_State | Master\_Host | Master\_User | Master\_Port | Connect\_Retry | Master\_Log\_File | Read\_Master\_Log\_Pos | Relay\_Log\_File | Relay\_Log\_Pos | Relay\_Master\_Log\_File | Slave\_IO\_Running | Slave\_SQL\_Running | Replicate\_Do\_DB | Replicate\_Ignore\_DB | Replicate\_Do\_Table | Replicate\_Ignore\_Table | Replicate\_Wild\_Do\_Table | Replicate\_Wild\_Ignore\_Table | Last\_Errno | Last\_Error | Skip\_Counter | Exec\_Master\_Log\_Pos | Relay\_Log\_Space | Until\_Condition | Until\_Log\_File | Until\_Log\_Pos | Master\_SSL\_Allowed | Master\_SSL\_CA\_File | Master\_SSL\_CA\_Path | Master\_SSL\_Cert | Master\_SSL\_Cipher | Master\_SSL\_Key | Seconds\_Behind\_Master | Master\_SSL\_Verify\_Server\_Cert | Last\_IO\_Errno | Last\_IO\_Error | Last\_SQL\_Errno | Last\_SQL\_Error | Replicate\_Ignore\_Server\_Ids | Master\_Server\_Id | Master\_UUID | Master\_Info\_File | SQL\_Delay | SQL\_Remaining\_Delay | Slave\_SQL\_Running\_State | Master\_Retry\_Count | Master\_Bind | Last\_IO\_Error\_Timestamp | Last\_SQL\_Error\_Timestamp | Master\_SSL\_Crl | Master\_SSL\_Crlpath | Retrieved\_Gtid\_Set | Executed\_Gtid\_Set | Auto\_Position | Replicate\_Rewrite\_DB | Channel\_Name | Master\_TLS\_Version |

+----------------------+-----------------------+------------------+-------------+---------------+------------------+---------------------+----------------------------------+---------------+-----------------------+------------------+-------------------+-----------------+---------------------+--------------------+------------------------+-------------------------+-----------------------------+------------+------------+--------------+---------------------+-----------------+-----------------+----------------+---------------+--------------------+--------------------+--------------------------+----------------------+--------------+--------------------+

| Connecting to master | your\_master\_server\_ip | replication\_user | 3306 | 60 | mysql-bin.000001 | 123456 | DESKTOP-TJ8B6AI-relay-bin.000062 | 4 | mysql-bin.000001 | Connecting | Yes | | | | | | | 0 | | 0 | 123456 | 154 | None | | 0 | No | | | | | | NULL | No | 2005 | error connecting to master 'replication\_user@your\_master\_server\_ip:3306' - retry-time: 60 retries: 40 | 0 | | | 0 | | C:\ProgramData\MySQL\MySQL Server 5.7\Data\master.info | 0 | NULL | Slave has read all relay log; waiting for more updates | 86400 | | 240325 11:42:15 | | | | | | 0 | | | |

+----------------------+-----------------------+------------------+-------------+---------------+------------------+---------------------+----------------------------------+---------------+-----------------------+------------------+-----

1 row in set (0.00 sec)

mysql>