

Assignment No. 6

Group ID: 2018BCGRP10

Batch: T2

PRN No: 2018BTECS00072 & 2018BTECS00086

- **Title: Parallel Databases**

- **Aim:** To build a generic parallel sort and parallel join algorithm.

1. Implement a Python function ParallelSort() that takes as input: (1) InputTable stored in a Oracle/MySQL database, (2) SortingColumnName the name of the column used to order the tuples by. ParallelSort() then sorts all tuples (using five parallelized threads) and stores the sorted tuples in a table named OutputTable (the output table name is passed to the function). The OutputTable contains all the tuples present in InputTable sorted in ascending Order.

Function Interface: -

ParallelSort (InputTable, SortingColumnName, OutputTable, openconnection)

InputTable – Name of the table on which sorting needs to be done.

SortingColumnName – Name of the column on which sorting needs to be done, would be either of type integer or real or float. Basically Numeric format. Will be Sorted in Ascending order.

OutputTable – Name of the table where the output needs to be stored.

openconnection – connection to the database.

2. Implement a Python function ParallelJoin() that takes as input: (1) InputTable1 and InputTable2 table stored in a Oracle/MySQL database,

(2) Table1JoinColumn and Table2JoinColumn that represent the join key in each input table respectively. ParallelJoin() then joins both InputTable1 and InputTable2 (using five parallelized threads) and stores the resulting joined tuples in a table named OutputTable (the output table name is passed to the function). The schema of OutputTable should be similar to the schema of both InputTable1 and InputTable2 combined.

Function Interface: -

ParallelJoin (InputTable1, InputTable2, Table1JoinColumn, Table2JoinColumn, OutputTable, openconnection)

InputTable1 – Name of the first table on which you need to perform a join.

InputTable2 – Name of the second table on which you need to perform join.

Table1JoinColumn – Name of the column from first table i.e. join key for first table.

Table2JoinColumn – Name of the column from second table i.e. join key for second table.

OutputTable - Name of the table where the output needs to be stored.

openconnection – connection to the database.

- **Introduction:**

Companies need to handle huge amounts of data with high data transfer rate. The client server and centralized system is not very efficient. The need to improve efficiency gave birth to the concept of Parallel Databases.

Parallel database system improves performance of data processing using multiple resources in parallel, like multiple CPU and disks are used parallelly. It also performs many parallelization operations like data loading and query processing.

Parallel Database:

A **parallel database** system seeks to improve performance through parallelization of various operations, such as loading data, building indexes and evaluating queries. Although data may be stored in a distributed fashion, the distribution is governed solely by performance considerations. Parallel databases improve processing and input/output speeds by using multiple CPUs and disks in parallel.

Goals Of Parallel Databases:

The concept of Parallel Database was built with a goal to:

- 1. Improve performance:**

The performance of the system can be improved by connecting multiple CPUs and disks in parallel. Many small processors can also be connected in parallel.

- 2. Improve availability of data:**

Data can be copied to multiple locations to improve the availability of data.

For example: if a module contains a relation (table in database) which is unavailable then it is important to make it available from another module.

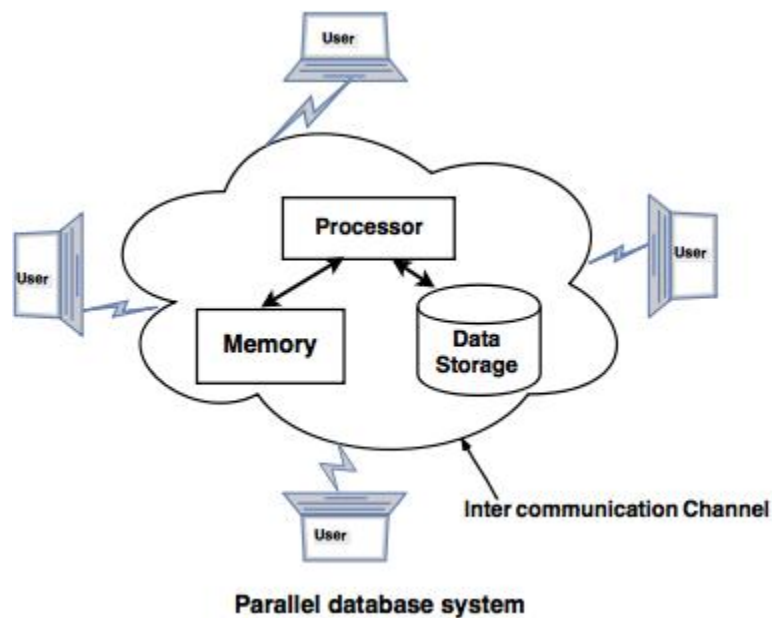
- 3. Improve reliability:**

Reliability of the system is improved with completeness, accuracy and availability of data.

- 4. Provide distributed access of data:**

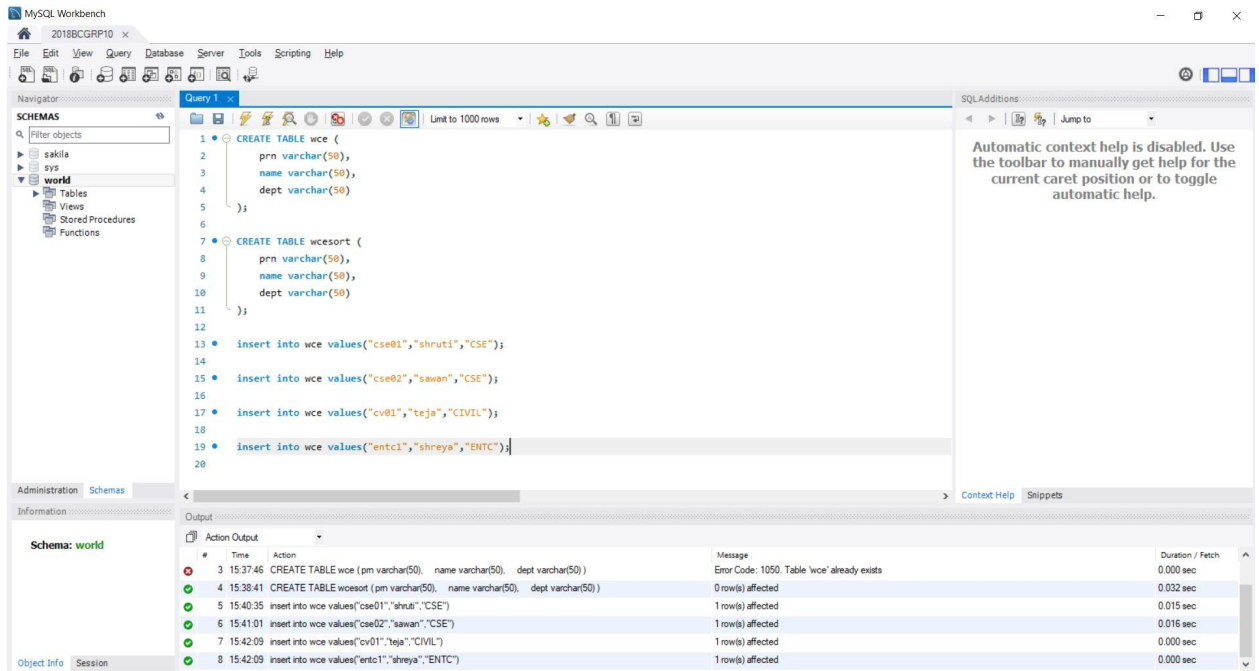
Companies having many branches in multiple cities can access data with the help of a parallel database system.

- **Functional Block Diagram:**

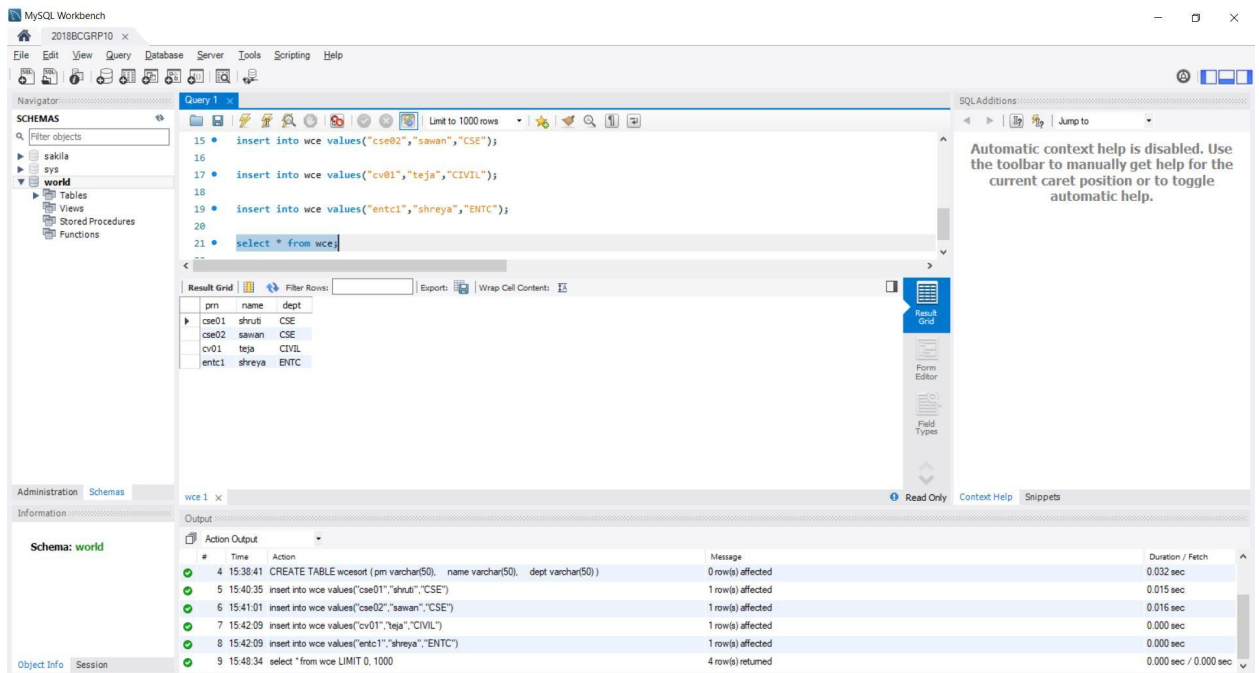


- **Screenshots Of Experiment:**

Created tables 'wce' and 'wcejoin' and inserted data into the 'wce' table:



‘wce’ table:



Created tables ‘first’, ‘second’ and ‘output’ and inserted data in ‘first’ and ‘second’ the tables:

MySQL Workbench

2018BCGRP10

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

sakila
 sys
 world

Tables
 Views
 Stored Procedures
 Functions

Administration Schemas

Information

Schema: world

Object Info Session

Query 1

```

23 CREATE TABLE first (
24     prn varchar(50),
25     name varchar(50)
26 );
27
28 CREATE TABLE second (
29     prn varchar(50),
30     dept varchar(50)
31 );
32
33 insert into first values("cse01","shruti");
34 insert into first values("cse02","sawan");
35 insert into first values("cv01","teja");
36 insert into first values("entc1","shreya");
37
38 insert into second values("cse01","CSE");
39 insert into second values("cse02","CSE");
40 insert into second values("cv01","CIVIL");
41 insert into second values("entc1","ENTC");
42
43

```

SQL Additions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Output

Action Output

| # | Time | Action | Message | Duration / Fetch |
|----|----------|--|-------------------|------------------|
| 14 | 15:53:38 | insert into first values("cv01","teja") | 1 row(s) affected | 0.000 sec |
| 15 | 15:53:38 | insert into first values("entc1","shreya") | 1 row(s) affected | 0.000 sec |
| 16 | 15:54:49 | insert into second values("cse01","CSE") | 1 row(s) affected | 0.000 sec |
| 17 | 15:54:49 | insert into second values("cse02","CSE") | 1 row(s) affected | 0.000 sec |
| 18 | 15:54:49 | insert into second values("cv01","CIVIL") | 1 row(s) affected | 0.000 sec |
| 19 | 15:54:49 | insert into second values("entc1","ENTC") | 1 row(s) affected | 0.016 sec |

'first' table:

MySQL Workbench

2018BCGRP10

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

sakila
 sys
 world

Tables
 Views
 Stored Procedures
 Functions

Administration Schemas

Information

Schema: world

Object Info Session

Query 1

```

35 insert into first values("cv01","teja");
36 insert into first values("entc1","shreya");
37
38 insert into second values("cse01","CSE");
39 insert into second values("cse02","CSE");
40 insert into second values("cv01","CIVIL");
41 insert into second values("entc1","ENTC");
42
43 select * from first;
44
45

```

SQL Additions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid

| prn | name |
|-------|--------|
| cse01 | shruti |
| cse02 | sawan |
| cv01 | teja |
| entc1 | shreya |

first 2

Output

Action Output

| # | Time | Action | Message | Duration / Fetch |
|----|----------|--|-------------------|-----------------------|
| 15 | 15:53:38 | insert into first values("entc1","shreya") | 1 row(s) affected | 0.000 sec |
| 16 | 15:54:49 | insert into second values("cse01","CSE") | 1 row(s) affected | 0.000 sec |
| 17 | 15:54:49 | insert into second values("cse02","CSE") | 1 row(s) affected | 0.000 sec |
| 18 | 15:54:49 | insert into second values("cv01","CIVIL") | 1 row(s) affected | 0.000 sec |
| 19 | 15:54:49 | insert into second values("entc1","ENTC") | 1 row(s) affected | 0.016 sec |
| 20 | 15:55:35 | select * from first LIMIT 0, 1000 | 4 row(s) returned | 0.000 sec / 0.000 sec |

'second' table:

- **Conclusion:**

Parallel sort and parallel join algorithm has been implemented using MySQL.

‘wcesort’ table:

The screenshot shows the MySQL Workbench interface. The 'SCHEMAS' pane on the left shows the 'world' database selected. The 'Query' editor in the center contains the following SQL code:

```
41 insert into second values("entc1","ENTC");
42
43
44 select * from wcesort;
45
46 DROP TABLE wcesort;
47
48 CREATE TABLE wcesort (
49   prn varchar(50),
50   name varchar(50),
51   dept varchar(50)
52 );
53
54 select * from wcesort;
```

The 'Result Grid' below the query editor shows the following data:

| prn | name | dept |
|-------|--------|-------|
| cse01 | shruti | CSE |
| cse02 | saman | CSE |
| cv01 | teja | CIVIL |
| entc1 | shreya | ENTC |

The 'Output' pane at the bottom shows the execution log:

| # | Time | Action | Message | Duration / Fetch |
|---|----------|--|-------------------|-----------------------|
| 2 | 17:49:36 | CREATE TABLE wcesort (prn varchar(50), name varchar(50), dept varchar(50)) | 0 row(s) affected | 0.047 sec |
| 3 | 17:50:21 | select * from wcesort LIMIT 0, 1000 | 4 row(s) returned | 0.016 sec / 0.000 sec |

‘output’ table:

The screenshot shows the MySQL Workbench interface. The 'SCHEMAS' pane on the left shows the 'world' database selected. The 'Query' editor in the center contains the following SQL code:

```
31
32
33 insert into first values("cse01","shruti");
34 insert into first values("cse02","saman");
35 insert into first values("cv01","teja");
36 insert into first values("entc1","shreya");
37
38 insert into second values("cse01","CSE");
39 insert into second values("cse02","CSE");
40 insert into second values("cv01","CIVIL");
41 insert into second values("entc1","ENTC");
42
43
44 select * from output;
```

The 'Result Grid' below the query editor shows the following data:

| prn | name | dept |
|-------|--------|-------|
| cse01 | shruti | CSE |
| cse02 | saman | CSE |
| cv01 | teja | CIVIL |
| entc1 | shreya | ENTC |

The 'Output' pane at the bottom shows the execution log:

| # | Time | Action | Message | Duration / Fetch |
|----|----------|-------------------------------------|--------------------|-----------------------|
| 26 | 17:26:55 | select * from wcesort LIMIT 0, 1000 | 12 row(s) returned | 0.000 sec / 0.000 sec |
| 27 | 17:29:48 | select * from output LIMIT 0, 1000 | 4 row(s) returned | 0.000 sec / 0.000 sec |

- **References:**

1. <https://www.tutorialride.com/parallel-databases/parallel-databases-tutorial.htm>
2. https://en.wikipedia.org/wiki/Parallel_database#:~:text=A%20parallel%20database%20system%20seeks,building%20indexes%20and%20evaluating%20queries.&text=Parallel%20databases%20improve%20processing%20and,CPU%20and%20disks%20in%20parallel.
3. <https://www.youtube.com/watch?v=UsEZUQKgO5A>