# MPI CODE README FILE

## PARTITION.C

This code runs on rudimentary SELECT and JOIN queries on a partitioned database on a minimum of 2 up to a maximum of 8 nodes.

**Pre Requisites**

1. This code has been developed and tested on Ubuntu
2. To run this code, all nodes should have MPICH2 & NFS. One node should be dedicated as master
3. Master – Slave password less login is also required.

**Set up Guidelines**

1. The database (comma separated files with metadata) should be correctly copied into all nodes. For quick setup they can just be copied where the executable is present.
2. For database, by default, the program searches for PAR_DB_CLUST[VALUE] folder where [VALUE] is the number of nodes in the cluster.
3. A 2-Node sample dataset is provided with this documentation.

**Running the SELECT Query**

1. The program supports the following query syntax, which can be given through stdin.
   SELECT [**COLS**] FROM [**TABLENAME**] WHERE [**CONDITION**]
2. [COLS]: Can have any column names in the form of *TABLENAME.COLUMNNAME* or a simple * to get all columns
3. [TABLENAME]: Name of the table present in the database
4. [CONDITION]: Conditions supported are =, !=, >, <, <=, >= in the form of *TABLENAME.COLUMNNAME=Rachit*
5. Program can be compiled using the code *mpicc partition.c*
6. The Program can be executed in the following manner:
   *mpirun –np [no-of-hosts] –hosts [ip-addresses] ./a.out –console –size 10K*
7. –console: OPTIONAL, This will print the output to the console
8. –size 10K: REQUIRED, This will let the program know which table to get from the database (10K, 100K, 1000K etc)
9. If –console is not used the output is given in a file named OUTPUT

Rachit Magon (2014H112175P)
Rishi Raj Sahu (2014H112178P)

**Running the JOIN Query**

1. The JOIN Query can be executed as follows:
   SELECT [**COLS**] FROM [**TABLE1**],[**TABLE2**] WHERE **TABLE1.COLUMN=TABLE2.COLUMN**
2. [COLS]: Can have any column names in the form of *TABLENAME.COLUMNNAME* or a simple * to get all columns
3. [TABLE1]: Names of the table with **primary key**
4. [TABLE2]: Name of the table with foreign key
5. TABLE1.COLUMN=TABLE2.COLUMN: The join clause
6. This, like select can be run using the command:
   *mpirun –np [no-of-hosts] –hosts [ip-addresses] ./a.out –console –size 10K*
7. All meaning are the same as select.

## REPLICATED.C

This code runs on rudimentary SELECT and JOIN queries on a replicated database on a minimum of 2 up to a maximum of 8 nodes, where the Replication Factor is given by a file named REPLICATED_MAPPING.

**Pre Requisites**

1. Same as PARTITION.C

**Set up Guidelines**

1. The DataBase folder which the program searches for is [NODE]_REP_DB.
2. A 2-Node 10K sample dataset is provided with this documentation.
3. All or few nodes can have a copy of the complete table replicated in their respective directories.
4. The Mapping file should be in the same directory as the code.

**Running the SELECT Query**

1. Running queries is a little different in Replicated.c than what they are in Partition.c, the format of the SELECT QUERY is below:
   SELECT [**COLS**] FROM [**TABLENAME-SIZE**] WHERE [**CONDITION**]
2. Here the main difference is in the **SIZE** field, for example if the Table Name is Person and the required dataset is of 10K values, the query should be .. FROM Person-10K WHERE..
3. Program can be compiled using the code *mpicc partition.c*
4. The Program can be executed in the following manner:
   *mpirun –np [no-of-hosts] –hosts [ip-addresses] ./a.out*

Rachit Magon (2014H112175P)
Rishi Raj Sahu (2014H112178P)

5. Note that the –size parameter is no longer needed

**Running the JOIN Query**

1. The JOIN Query can be executed as follows:
   SELECT [**COLS**] FROM [**TABLE1-SIZE**],[**TABLE2**] WHERE
   **TABLE1.COLUMN=TABLE2.COLUMN**
2. All the fields are the same as in the Join query of the partitioned model.
3. It can be run using the command:
   *mpirun –np [no-of-hosts] –hosts [ip-addresses] ./a.out*
4. The code automatically detects whether it is a SELECT command or a JOIN command.