# 1 Benchmark Function (main function)

**run\_benchmark**()

save the current time as the start of the 'benchmark' action

**READ** the configuration parameters into the memory (config params `file.configuration.name ...`)

**READ** the bulk file data into the partitioned collection bulk\_data\_partitions (config param 'file.bulk.name')

partition key = modulo (ASCII value of 1st byte of key \* 256 + ASCII value of 2nd byte of key,

number partitions (config param 'benchmark.number.partitions'))

Create a separate database connection (without auto commit behaviour) for each partition

trial\_no = 0

**WHILE** trial\_no < config\_param 'benchmark.trials'

**DO** **run\_trial**(database connections, trial\_no, bulk\_data\_partitions)

**ENDWHILE**

partition\_no = 0

**WHILE** partition\_no < config\_param 'benchmark.number.partitions'

close the database connection

**ENDWHILE**

**WRITE** an entry for the action 'benchmark' in the result file (config param 'file.result.name')

# 2 Trial Function

**run\_trial**(database connections, trial\_no, bulk\_data\_partitions)

**INPUT**: the database connections

the current trial number

the partitioned bulk data

save the current time as the start of the 'trial' action

create the database table (config param 'sql.create')

**IF** error

drop the database table (config param 'sql.drop')

create the database table (config param 'sql.create')

**ENDIF**

**DO** **run\_insert**(database connections, trial\_no, bulk\_data\_partitions)

**DO** **run\_select**(database connections, trial\_no, bulk\_data\_partitions)

drop the database table (config param 'sql.drop')

**WRITE** an entry for the action 'trial' in the result file (config param 'file.result.name')

# 3 Insert Control Function

**run\_insert**(database connections, trial\_no, bulk\_data\_partitions)

**INPUT**: the database connections

the current trial number

the partitioned bulk data

save the current time as the start of the 'query' action

partition\_no = 0

**WHILE** partition\_no < config\_param 'benchmark.number.partitions'

**IF** config\_param 'benchmark.core.multiplier' = 0

**DO** **insert**(database connections(partition\_no), bulk\_data\_partitions(partition\_no))

**ELSE**

**DO** **insert**(database connections(partition\_no), bulk\_data\_partitions(partition\_no)) as a thread

**ENDIF**

**ENDWHILE**

**WRITE** an entry for the action 'query' in the result file (config param 'file.result.name')

# 4 Insert Function

insert(database connection, bulk\_data\_partition)

**INPUT**: the database connection

the bulk data partition

count = 0

collection batch\_collection = empty

**WHILE** iterating through the collection bulk\_data\_partition

count + 1

add the SQL statement in config param 'sql.insert' with the current bulk\_data entry to the collection batch\_collection

**IF** config\_param 'benchmark.batch.size' > 0

**IF** count modulo config param 'benchmark.batch.size' = 0

execute the SQL statements in the collection batch\_collection

batch\_collection = empty

**ENDIF**

**ENDIF**

**IF** config param 'benchmark.transaction.size' > 0 AND count modulo config param 'benchmark.transaction.size' = 0

commit

**ENDIF**

**ENDWHILE**

**IF** collection batch\_collection is not empty

execute the SQL statements in the collection batch\_collection

**ENDIF**

commit

# 5 Select Control Function

**run\_select**(database connections, trial\_no, bulk\_data\_partitions)

**INPUT**: the database connections

the current trial number

the partitioned bulk data

save the current time as the start of the 'query' action

partition\_no = 0

**WHILE** partition\_no < config\_param 'benchmark.number.partitions'

**IF** config\_param 'benchmark.core.multiplier' = 0

DO Select(database connections(partition\_no), bulk\_data\_partitions(partition\_no, partition\_no)

**ELSE**

**DO** Select(database connections(partition\_no), bulk\_data\_partitions(partition\_no, partition\_no) as a thread

**ENDIF**

**ENDWHILE**

**WRITE** an entry for the action 'query' in the result file (config param 'file.result.name')

# 6 Select Function

**run\_select**(database connection, bulk\_data\_partition, partition\_no)

**INPUT**: the database connection

the bulk data partition

the current partition number

save the current time as the start of the 'query' action

count = 0

execute the SQL statement in config param 'sql.select'

**WHILE** iterating through the result set

count + 1

**ENDWHILE**

**IF NOT** count = size(bulk\_data\_partition)

display an error message

**ENDIF**