

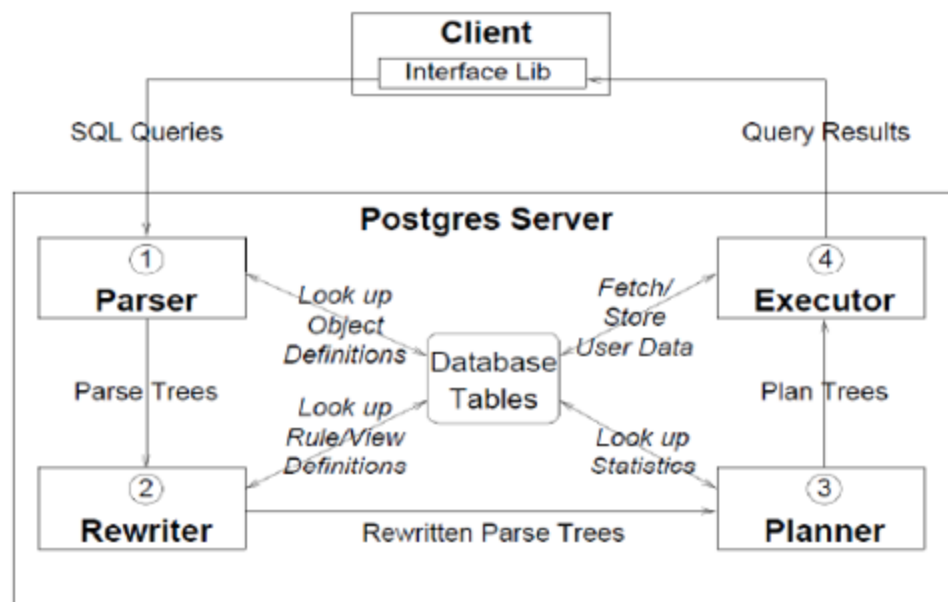
CSI 3130 Project Guidelines

Vikas Gogia (vgogi074@uottawa.ca)

Objective

- Implement a new **Symmetric Hash Join operator** replacing current hash join implementation
- Modifications to be done in **Optimizer** and **Executor** components

Postgres Backend



Relevant Files

- ☐ src/backend/executor
 - **nodeHashJoin.c**: actual hash join operator processing
 - **nodeHash.c**: creating and maintaining hash table

- ☐ src/backend/optimizer/plan/
 - **createplan.c**: contains code to create a hash join node in query plan
- ☐ src/include/nodes
 - **execnodes.h**: contains structure HashJoinState that maintains the state of hash join during execution

Deliverables

- All relevant files mentioned above are to be zipped and e-mailed to both of the TA's (**vgogi074@uottawa.ca**, **daniel.lobo@uottawa.ca**). Make sure the name of the zip file contains the enrollment numbers of all team members separated by an underscore. (e.g. 300291000_300291100.zip)
- Insert comments with explanations preceded by '**CSI3130:**'
 - A test query is available here ([link](#))

DON'T FORGET TO REFER TO THE SAMPLE SOLUTION FILES PROVIDED BY THE PROFESSOR ...

Changes in nodeHash.c

- Modify **ExecHash**
 - Disable batches (multiple batches of Hybrid Hash Join)
- ExecHashTableCreate**

```

hashtable = (HashJoinTable) palloc(sizeof(HashJoinTableData));
hashtable->nbuckets = nbuckets;
hashtable->buckets = NULL;
hashtable->nbatch = nbatch;
hashtable->curbatch = 0;
hashtable->nbatch_original = nbatch;
hashtable->nbatch_outstart = nbatch;
hashtable->growEnabled = true;
hashtable->totalTuples = 0;
hashtable->innerBatchFile = NULL;
hashtable->outerBatchFile = NULL;
hashtable->spaceUsed = 0;
hashtable->spaceAllowed = work_mem * 1024L;

```

- Implement **ExecScanHashBucket** for both Inner and Outer

Changes to ExecScanHashBucket

Present implementation includes 2 functions - one for probing inner and one for outer that returns a HeapTable

Change it to just 1 method for scanning both hash tables. Return HashTable instead of HeapTable. Make sure to check for the type of hash table in the method - is it inner or outer?

```
if(hjstate->probing_inner){
    hashtable = hjstate->inner_hj_HashTable;
    hashTuple = hjstate->inner_hj_CurTuple;
    hashvalue = hjstate->outer_hj_CurHashValue;
    bucketNo=hjstate->inner_hj_CurBucketNo;
    tupleSlot=hjstate->hj_InnerTupleSlot;
}else{
    hashtable = hjstate->outer_hj_HashTable;
    hashTuple = hjstate->outer_hj_CurTuple;
    hashvalue = hjstate->inner_hj_CurHashValue;
    bucketNo=hjstate->outer_hj_CurBucketNo;
    tupleSlot=hjstate->hj_OuterTupleSlot;
}
```

What next?

After incorporating required changes in the source code, try following to run your code:

1. clean

```
sudo make clean
```

2. make

```
sudo make
```

3. make install

```
sudo make install
```

4. run the server and run the test query.