

**Milestone 4**  
**CS165, Fall 2016**  
**Kevin Zhang**

## **1 Goals**

The goal of this milestone is to add join functionality to our database. Joins will be implemented in two ways: hash joins, and nested-loop joins. A comparison will be done between the relative performance of both methods.

## **2 Technical Description**

To implement hash joins, we need to make a hashtable data structure that supports keys and values. This was accomplished as part of project 0. Each node in the hashtable consists of a linked list, with multiple key value pairs at each node. Experiments can be done to determine optimal node sizes, if time allows. Nested loops joins are even simpler; we simply check every possible pair of indices and store the matched indices.

To implement hash joins, we start by inserting every value-index pair into the hashtable from one of the fetch-select pairs. Once that's done, we retrieve all value-index pairs for every value from the second fetch-select pair. We then store all of the matched indices in a results array. We store all index pairs for each value in the second pair first.

Nested-loops are similar; the only change we make is to iterate first across all value-index pairs of the smaller fetch-select pair. Since we have to check every possible pair of values from the two sets of results anyways, we reduce data movement by choosing to iterate first across the smaller set. By this, we mean that we check every value from the larger pair against a single value from the smaller set, and do this sequentially for each value in the smaller set.

## **3 Evaluation**

Not implemented yet!

## **4 Challenges & Open Items**

Haven't begun implementation yet.