**Installation instruction**

1. Unzip the file
2. Import the directory into Eclipse as an existing project
3. Build the project
4. Run test.project2.CreateTestTables.java. This runnable function would clear the directory and initialize SimpleDB automatically, then create 5 new tables with 3 different indices, insert values into tables. Also it would run SelectTestTables.java and JoinTestTables.java. These two test would process select and join with tables in different indices, and print out the time cost of each.

**Design description**

**Task1**

Task 1 is relatively easy. Basically we just follow the instruction and implement it.

**Task2**

For task2, we build a two-level index to implement it. The external index (ExtHashIndex.java) stores the index of the directory, including its corresponding key of the bucket and its local depth. The internal index (SecHashIndex.java) stores the all the records inside the bucket.

The insertion and deletion would be passed to internal index to implement. However, if one bucket is full, the internal index would tell the external index to split the bucket by deleting all elements in the bucket, creating a new bucket and reinsert deleted elements. Also, it would increase the global index by adding new indices as needed.

During task2, we struggled for many times. For example the generation of local depth and global depth is pretty tricky. Sometimes we need the global depth dynamically and sometimes we need a static one. This task is kind of complicated and indicated us the importance of testing

**Task 3**

Task3 is about testing. Nothing special need to be designed.