# 50.043 Database Systems Project
# Group 32 Lab 1

Group Members: Aditya Vishwanath (1004281), Visshal Natrajan (1005254), Mohammed Fauzaan(1005404)

## 1. Design Decisions

---

- In Exercise 1, in Tuple.java, fieldsList is implemented as a HashMap, allowing O(1) amortized access time.
- In Exercise 2, Catalog.java mainly consists of a mapping between fileID (which is the tableID), and a Table object. The table object contains necessary information that is called by the get methods present in the rest of the document.
  The catalog variable is implemented as a concurrentHashMap, as it allows multiple processes to access it in a thread safe manner, without suffering from race conditions. Catalog.java is central to many implementations, and changes caused to the catalog from one file must not affect the access to catalog from another file.
- In Exercise 3, the `Concurrent Hash Map` data type is used for cache as opposed to a normal hash map. As it supports full concurrency of retrievals which would be useful in future lab implementations
- In Exercise 5, the iterator for a HeapMap was implemented as a nested class within the heapfile. This would allow us to logically group classes that are only used in one place, increase the use of encapsulation, and create more readable and maintainable code.

## 2. Non-Trivial Aspects

---

- In Exercise 5, the implementation of the HeapMapIterator utilizes the logic of utilizing page numbers to call the pages from the bufferpool and then use the HeapPage iterator to iterate through the tuples in the given Page. Once all the tuples in a page are completed, it increments the page number. This continues until the end of the file is reached.

## 3. Changes Made to API

---

- The associated functions for Lab-1 were completed
- HeapMapIterator class added (refer to 1.2 & 2.1)

## 4. Incomplete Elements of Code

---

- All the elements of the code not required in Lab-1 are incomplete.