

## CS660 PA1

### 1. Description

#### `Tuple.java`

- check index of 'Field' must be in the range  $[0, \text{fields.length}-1]$ ;

#### `TupleDesc.java`

- equals: case 1 check if 'o' equals to pointer 'this'; case 2 check if object 'o' is instance of 'TDItem'; case 3 check name equality, and type equality;
- In the constructor, 'TupleDesc()', `TDItem[]` array must be non-empty and must contain at least one item;

#### `Catalog.java`

- use a hash table to store the mapping between table id and table;
- use a hash table to store the mapping between table name and table id;
- table has the attributes (type): `DbFile`, table name, primary key;

#### `BufferPool.java`

- In 'getPage' method, when the page is in buffer pool, just return the page; otherwise we have to look up 'Catalog' and get corresponding page from 'DbFile'. If the page number exceed the size of buffer pool, call `evictPage()` method.

#### `HeapPageId.java` & `RecordID.java`

- for hash code, I choose  $\text{hash}(a,b) = \text{prime} * a + b$ ;

#### `HeapPage.java`

- for method 'isSlotUsed(int i)', check if  $(\text{header}[\text{byte\_num}] \gg \text{offset}) \% 2$  is 1, where  $\text{byte\_num} = i/8$ ;  $\text{offset} = i \% 8$ ;
- for method 'iterator()', just iterates each tuple in use;

#### `HeapFile.java`

- 'readPage()' use `RandomAccessFile` to read `PageSize` of data from disk, then create a new `HeapPage` with the data;
- in '`DbFileIterator iterator()`', 'open()' method return the iterator of the first heap page that can be used;

#### `SeqScan.java`

- use the 'tableid' to find corresponding heap file, and use the iterator of the heap file to iterate the table;
- the `TupleDesc` with field names from the underlying heap file should prefix with the `tableAlias` string from the constructor.

### 2. No change to the API.

### 3. Complete all requirements of Lab1.

### 4. Totally spend about 8 hours on this lab.