

# 6.830: Lab 1 Review

# What is SimpleDB?

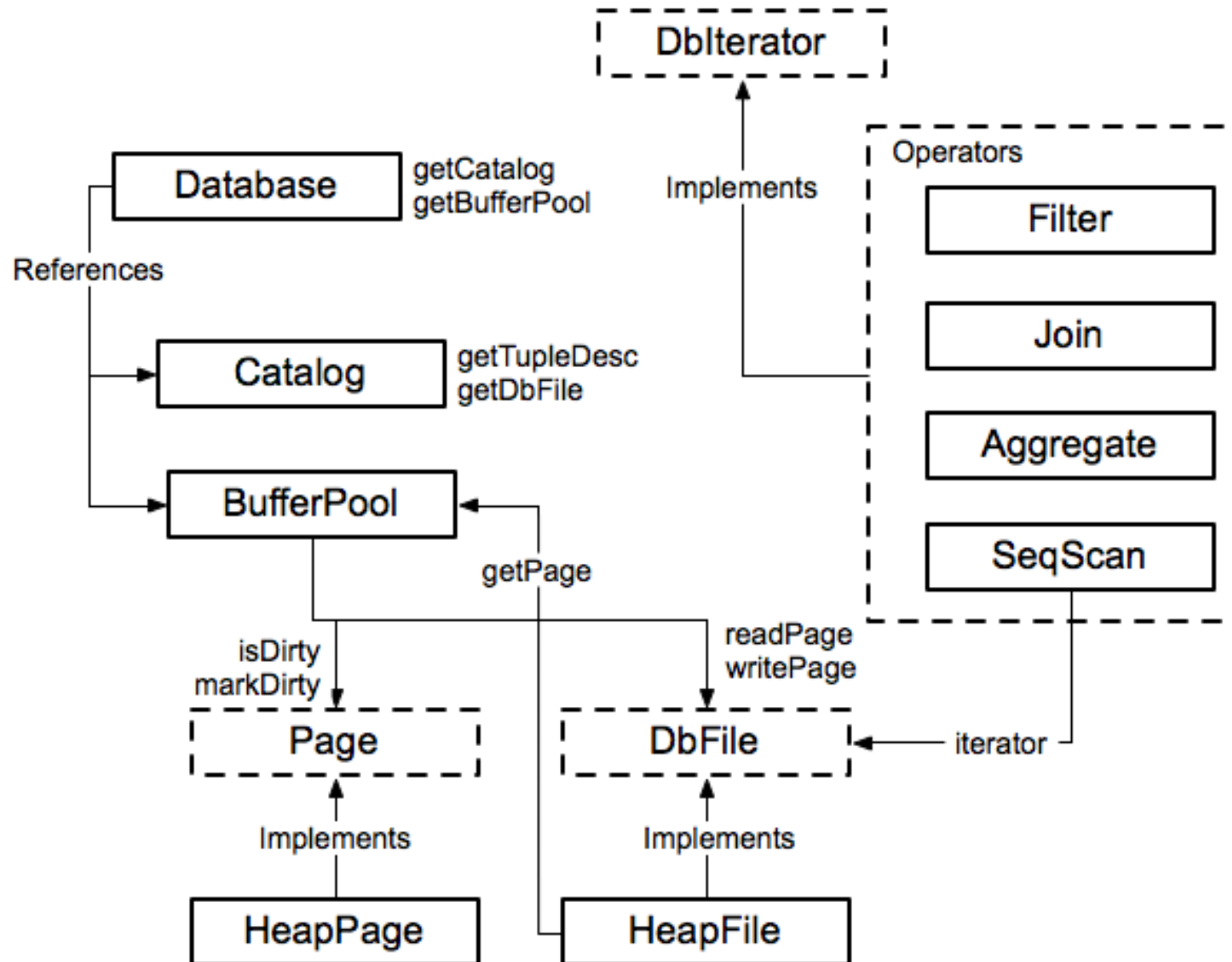
What is has?

- Heap Files
- Basic Operators
- Buffer Pool
- Transactions
- SQL Front-end

What is doesn't have?

- Fancy Operators
- Indices

# Module Diagram



# Catalog.java

- Stores a list of available tables and associated metadata like schema, name etc.
- Not persisted to disk!
- Interface:
  - `void addTable(DbFile f, String name, String pKey)`
  - `DbFile getDbFile(int tableid)`
  - `TupleDesc getTupleDesc(int tableid)`
  - ...

# DbFile.java

- Interface for physical representation of a table.
- Each tables is stored in a single DbFile.
- Can be in memory, on-disk, your call!
- Interface:
  - Page readPage(PageId pid)
  - void writePage(Page p)
  - DbFileIterator iterator(TransactionId tid)
  - TupleDesc getTupleDesc()
  - List<Page> insertTuple(TransactionId tid, Tuple t)
  - ...

# HeapFile.java

- Implementation of SimpleDB's physical representation
- Array of HeapPages on disk in a single file.
- Implement everything except addTuple(...) and removeTuple(...).

# HeapFileEncoder.java

- Because you haven't implemented insertTuple, you have no way to create data files!
- Converts CSV files to HeapFiles.
- Usage:  

```
java -jar dist/simpliedb.jar convert filename.txt numFields
```
- Produces a file filename.dat which can be passed to HeapFile constructor.

# Page.java

- Interface used to represent a single page resident in the Buffer Pool.
- Fixed size! `BufferPool.PAGE_SIZE`
- More in Lab 2.
- Interface:
  - `PageId getId()`
  - `byte[] getPageData()`
  - ...

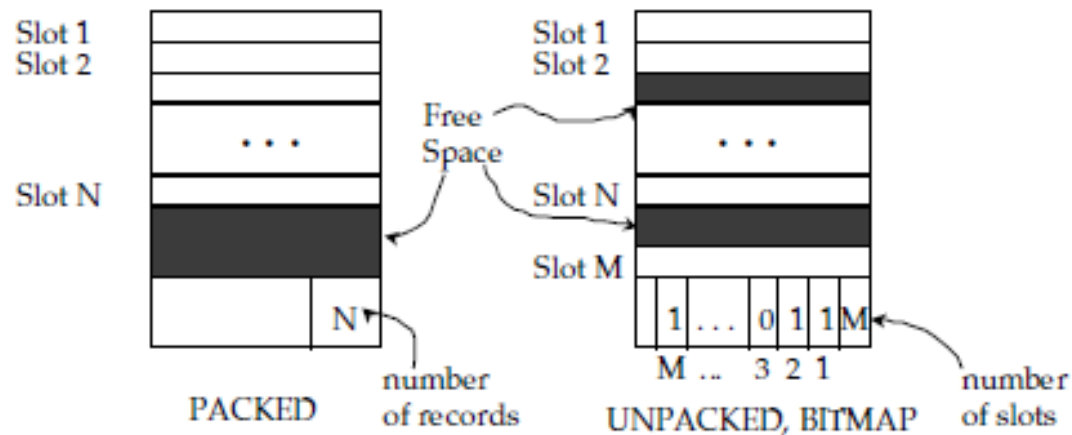


# HeapPage.java

- HeapFile's representation of a Page.
- Format:
  - Header which stores a bitmap
  - Array of fixed-length tuples
- Header size + size of all tuples  $\leq$  BufferPool.PAGE\_SIZE

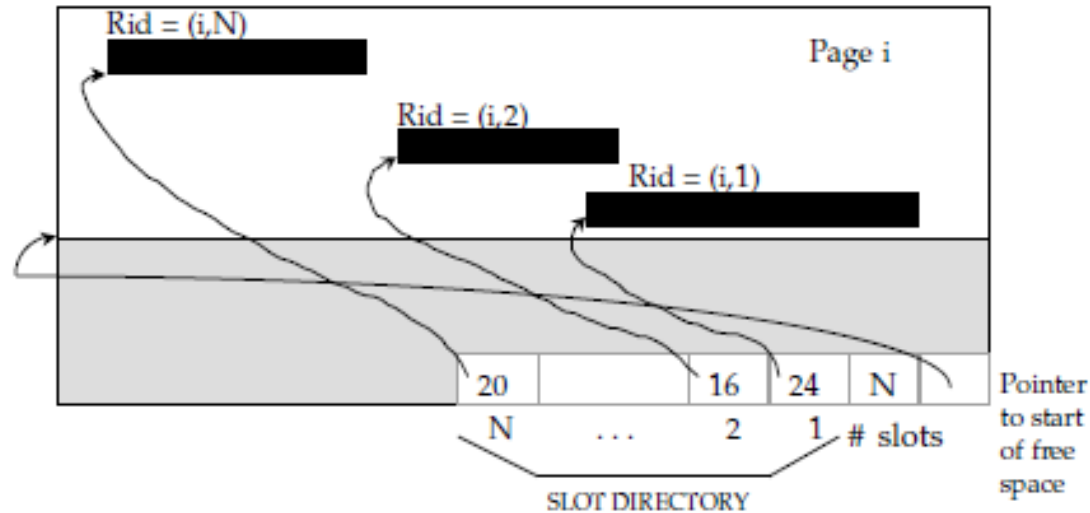
# Alternatives?

## Fixed-length records



Packed format can be a problem if there are external references to tuples.

# What about variable length record?



- Real systems: blob/shadow storage. Variable length fields stored there.
- PostgreSQL: char(n), varchar(n), text?

# BufferPool.java

- Cache for pages.
- Responsible for fetching pages from disk and implementing eviction policies (Lab 2).
- Always use `BufferPool.getPage(...)` even in `DbFile`!

# DbIterator.java

- Iterator class implemented by all operators.
- Interface:
  - boolean hasNext()
  - Tuple next()
  - void rewind()
  - ...
- Chaining iterators is awesome! (Lab 2)

# Compiling, Testing, Running!

- Use build tool called Apache Ant (similar to Make)
- Two types of tests:
  - Unit tests under test/simplydb/
  - System tests (end-to-end) under test/simplydb/systemtests
- Let's write some code!