

## 存储接口设计

存储接口主要体现在对 `Table` 类的记录插入、删除、修改和查询上

- `void insert(Row row)`  
插入记录 `row`
- `void delete(Row row)`  
删除记录 `row`
- `void update(Row newRow, Row oldRow)`  
更新记录 `oldRow` 至 `newRow`，会首先检查原记录和新纪录主键是否相同，若相同，则直接更新即可；  
若不同，首先删除 `oldRow`，接着插入 `newRow`
- `Row get(Entry entry)`  
根据主键获取记录
- `void persist()`  
持久化所有记录
- `void recover()`  
根据持久化的文件恢复所有的记录

## 记录持久化

记录持久化主要使用 `java` 的 `serialize` 和 `deserialize`，在持久化时，`serialize` 所有记录保存到文件中，在恢复时，对文件进行 `deserialize` 获取所有记录。

## 测试

在 `TableTest` 文件中对上述功能进行了测试。