

事务模块

在事务模块的实现上，我们目前实现了 `read commit` 的隔离级别。我们实现了 `TransactionManager` 类，一个 `TransactionManager` 对象对应一个数据库，负责该数据库上所有事务的并发执行和记录。其内部维护多张列表，包括目前正在并发的会话列表，每个会话持有的写锁和读锁列表。在每一个 `Statement` 需要执行时，其对应的会话调用 `TransactionManager` 的 `exec` 接口，`TransactionManager` 根据 `Statement` 的类型，进行不同的加锁操作，并执行该操作，执行结束后将其写入日志。在需要回滚时，则根据目前 `Logger`（见下文介绍）的 `undo_list` 对操作逐一进行回滚操作。

WAL模块

在WAL模块上，我们为每个数据库设立一个 `Logger` 类对象，其负责将针对于该数据库的操作记录下来并读写对应的文件。在其内部，它分别维护 `undo_list` 和 `redo_list` 两个列表，分别记录可能需要 redo 和 undo 的 `Statement`。每当一个 `Statement` 在事务中被执行时，我们便将该 `Statement` 和撤销该操作需要的 `Statement` 记录下来。在需要保存时(`checkpoint` 或数据库 `shutdown` 时)，将其以JSON的格式写入磁盘。

在恢复时，数据库根据自身 `Logger` 对象对应的文件，读取结构化的JSON数据，进行数据的恢复。数据恢复过程大致为:对需要redo的操作(例如最后一个checkpoint之后的操作)进行redo，再对对应需要撤销的操作进行undo。

进阶要求

在目前阶段，我们只实现了事务和WAL的基本框架，之后计划继续完成多事务的并发恢复机制，多种不同的隔离级别，savepoint等功能。