第一部分：DBMS实现概述

韩丽萍
计算机学院
联系邮件：liping@njupt.edu.com

内容提要

1、Megatron2000原型概况

2、 Megatron2000原型的不足

3、DBMS提供的能力

4、DBMS结构

Megatron2000原型

1、Stanford数据库研究组的一个内部原型系统

2、采用文件系统来存储关系

3、支持简单的SQL语句

Megatron2000原型实现概述

1 模式如何存储

2 实例如何存储

3 查询如何响应

4 多表连接查询如何响应

模式存储实现

- 模式存储在特定文件/usr/db/schema中
- 对于每一个关系（表），schema中有一个以该关系的名字起始的行
- 行中包含多个属性名|数据类型
- 用特定符号作为分割符，比如“#”

模式存储实现

模式存储示意(两张表students和depts)

students#name|string#id|integer#dept|string#

depts#name|string#office|string#

注：只是逻辑示意，实际实现会有出入，比如进一步将表模式和各个字段的模式信息分在不同的块或文件中

模式存储实现

实际实现时，schema文件分成两部分：

<

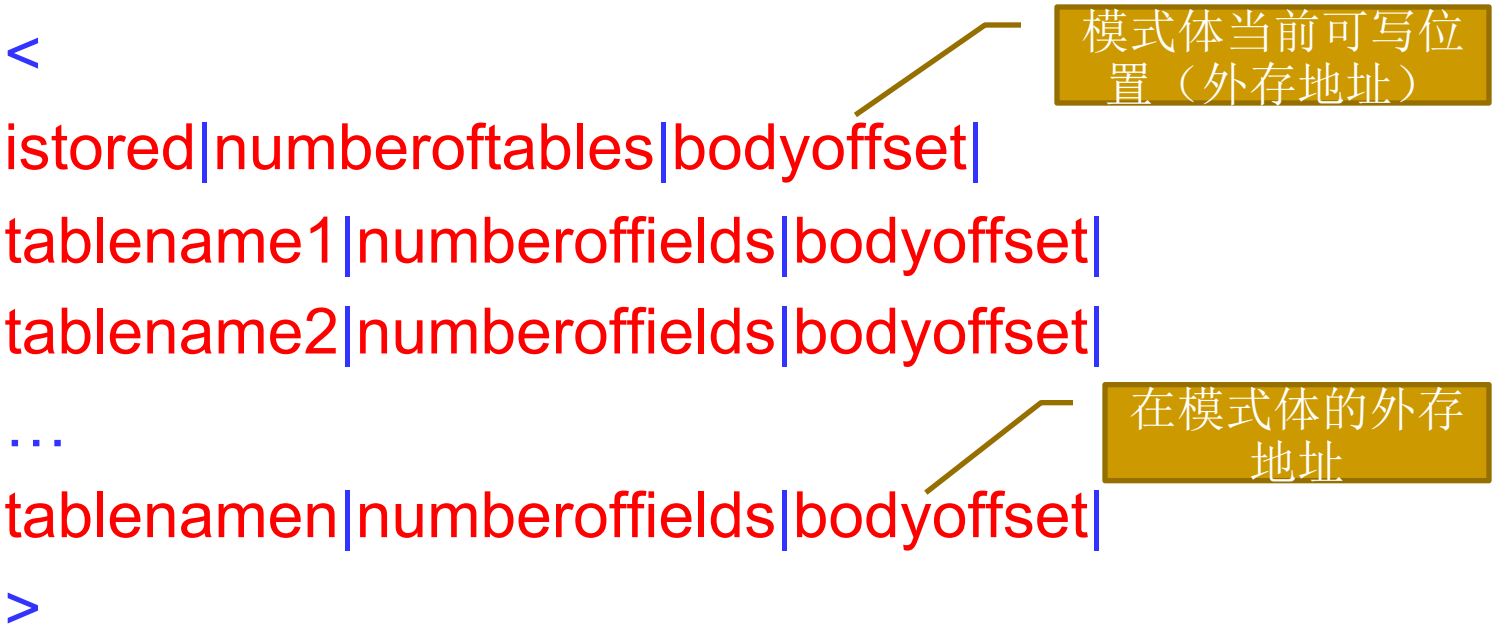
文件头——记录元信息，放在若干个块（block）中，块大小取决于系统设定，一般4K或8K

文件体——记录每个表的模式信息，放在若干个块（4K或8K）中

>

模式存储实现

模式文件头的结构示意：



模式存储实现

模式文件体的结构:（实现方式1——每个域的长度固定）

<

f1datatype1f2datatype2

…

fndatatypen

>

文件体结构（实现方式2－采用分隔符）

<

f1|datatype1#f2|datatype2#

…

fn|datatypen#

>

实例存储实现

如何存储students(name,id,dept)的实例数据?

1 数据存储在一个单独的文件/usr/db/students中

2 每个元组对应文件一行（行与行用回车或特定符号分割）

3 元组的各个字段值存储成用#号分割的字符串

实例存储实现

Smith	123	CS
Johnson	522	EE
Tom	445	CS

实例存储实现

实例文件的结构:（实现方式1——控制每个域的长度）

<Smith###123#####CS#######

Johnson#522#####EE#######

Tom#####445#####CS#######>

实例文件的结构（实现方式2－采用分隔符）

<Smith|123|CS||

Johnson|522|EE||

Tom|445|CS||>

查询实现（简单查询）

简单SQL查询

Select f1 from students where <condition>

原始的物理执行计划

（1）读文件/usr/db/schema的对应行，确定有哪些属性和对应的数据类型

（2）检查condition对于关系R的语义合法性

（3）读取/usr/db/students的每一行，

（a）检查每一行是否符合条件

（b）若符合条件，则显示该行为一个元祖

查询实现（简单查询生成临时表）

SQL语句

Select * from student where <condition> |T

含义：
查询的结果生成一个临时表T(类似oracle中的select * into T from student where <condition>)

查询实现（简单查询生成临时表）

原始的物理执行计划

（1）同前文

（2）每条符合条件的结果写入到一个新的文件/usr/db/T中

（3）往文件/usr/db/schema中添加新的一行

T#name#string#id#integer#dept#string

查询实现（多表连接）

SQL语句

```
select office
from students,depts
where students.name='smith' and
students.dept=depts.name
```

查询实现（多表连接）

原始的物理执行计划

```
For (each tuple s in students)
        for(each tuple d in depts.)
                if s and d satisfy the condition
                        display the office value
```

（1）算法复杂度是N*N

（2）考虑缓冲控制，即在给定的内存大小限制，实现元组的比对和连接

备注：自己实现

Megatron 2000的部分问题

（1）元组在磁盘上的排列不够好，修改不灵活，比如students元组中将EE修改为COM,文件如何重写？

（2）中途失败，数据会丢失，不满足ACID准则

（3）没有考虑缓存数据，数据都是从硬盘直接读取

（4）查询的响应缺少优化方法

…

DBMS提供的能力

（1）持久存储

—独立于程序，效率和灵活性更高

（2）编程接口

—SQL语句

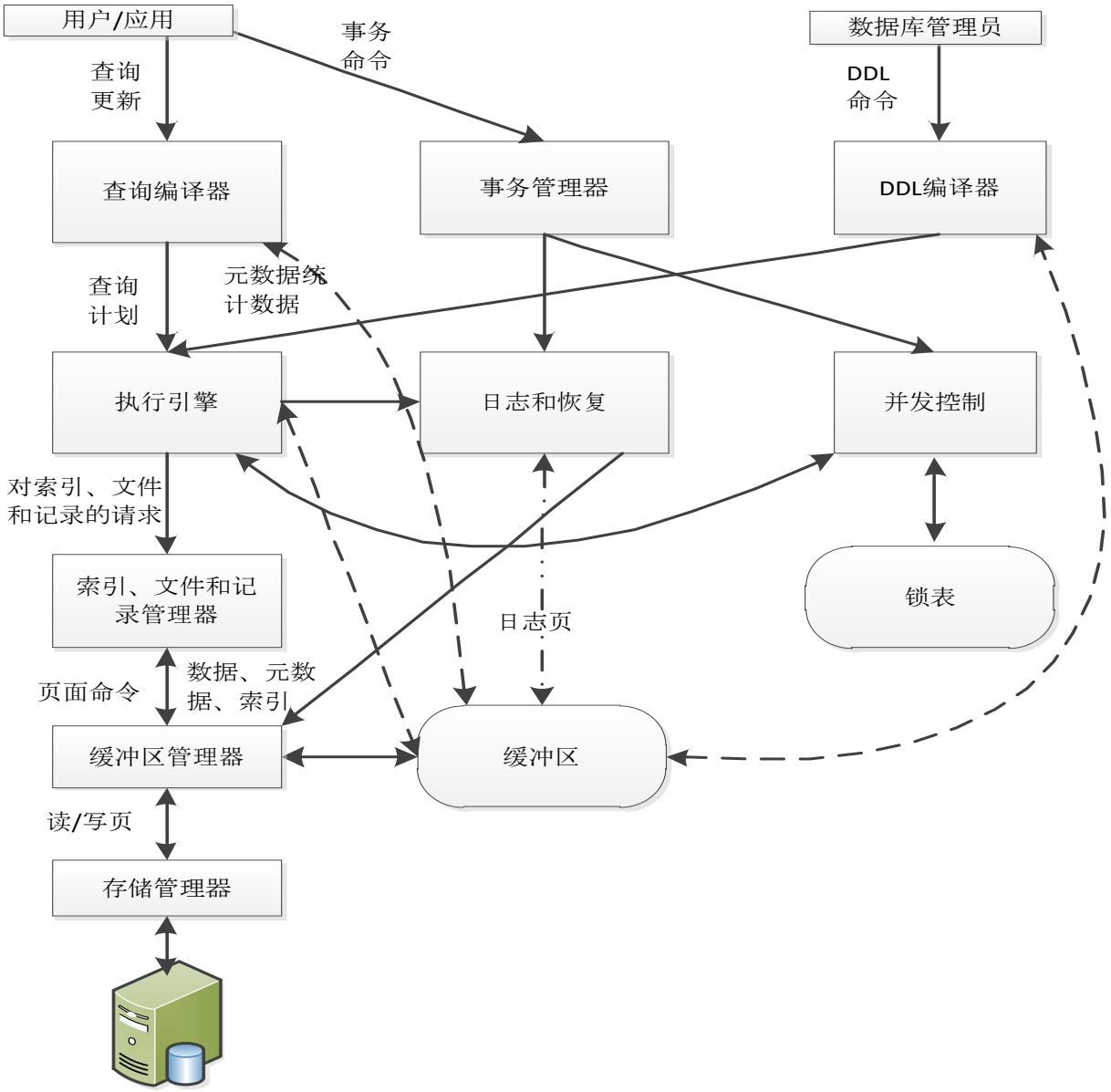
（3）查询优化

—索引如何用

（4）事务管理

—ACID准则

DBMS的成分


用户/应用
事务
命令
数据库管理员
查询
更新
DDL
命令
查询编译器
事务管理器
DDL编译器
查询
计划
元数据统
计数据
执行引擎
日志和恢复
并发控制
对索引、文件
和记录的请求
索引、文件和记
录管理器
锁表
日志页
数据、元数
据、索引
页面命令
缓冲区管理器
缓冲区
读/写页
存储管理器

DBMS概述

DBMS接受两类命令来源

（1）DBA建立数据库模式的命令
(DDL, Data Definition Language)

（2）普通用户和应用程序对数据的查询命令和修改命令
(DML, Data Manipulation Language）。

数据定义语言命令

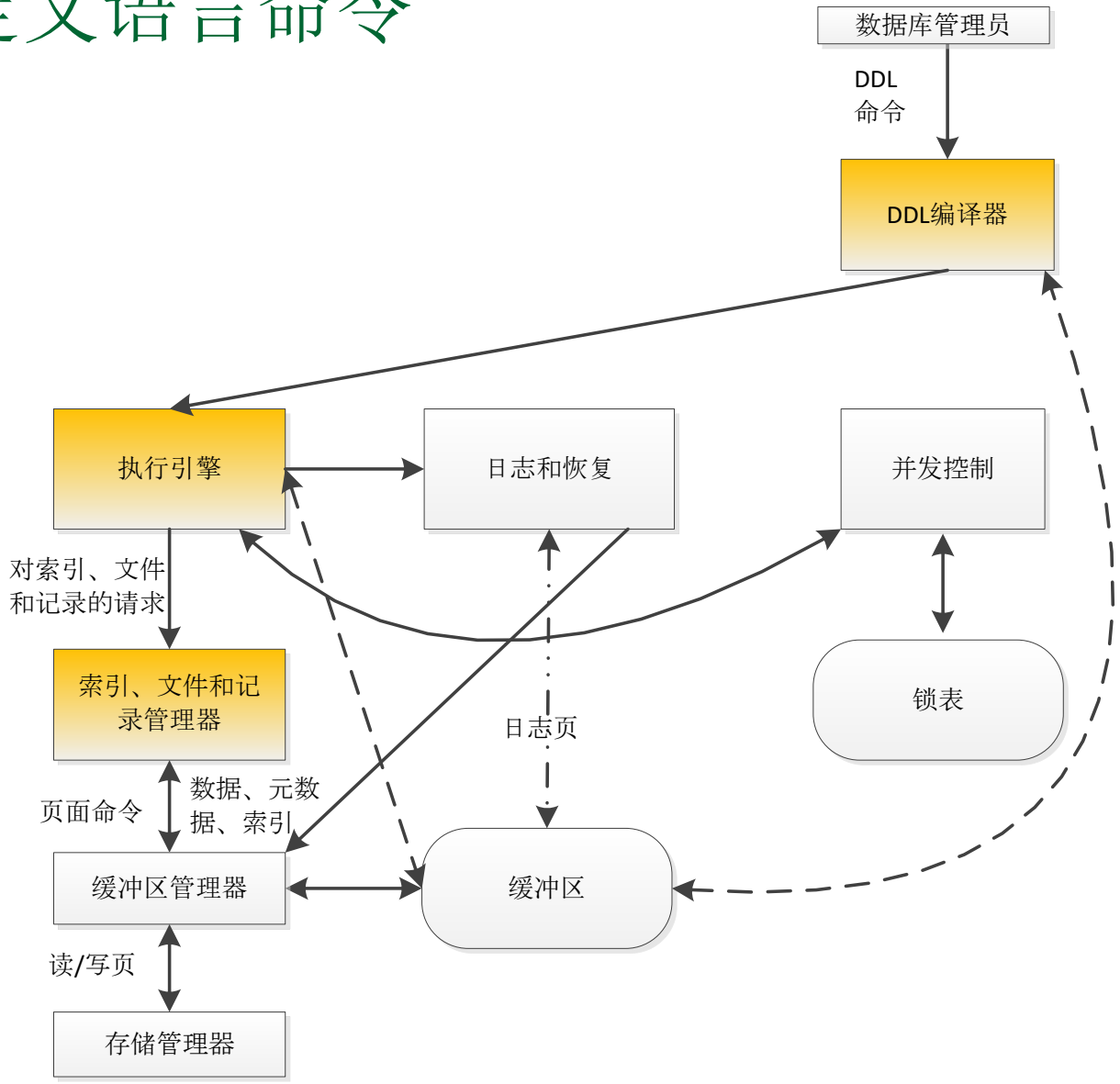
DDL(Create, Alter, Drop)的执行过程

（1）由DDL编译器进行分析

（2）分析结果传给执行引擎

（3）由执行引擎经过索引/记录管理器改变数据库模式

数据定义语言命令


数据库管理员
DDL
命令
DDL编译器
执行引擎
日志和恢复
并发控制
对索引、文件
和记录的请求
索引、文件和记
录管理器
锁表
日志页
页面命令
数据、元数
据、索引
缓冲区管理器
缓冲区
读/写页
存储管理器

查询（DML部分）处理概述——查询响应

（1）由查询编译器对查询进行分析和优化，得到查询计划

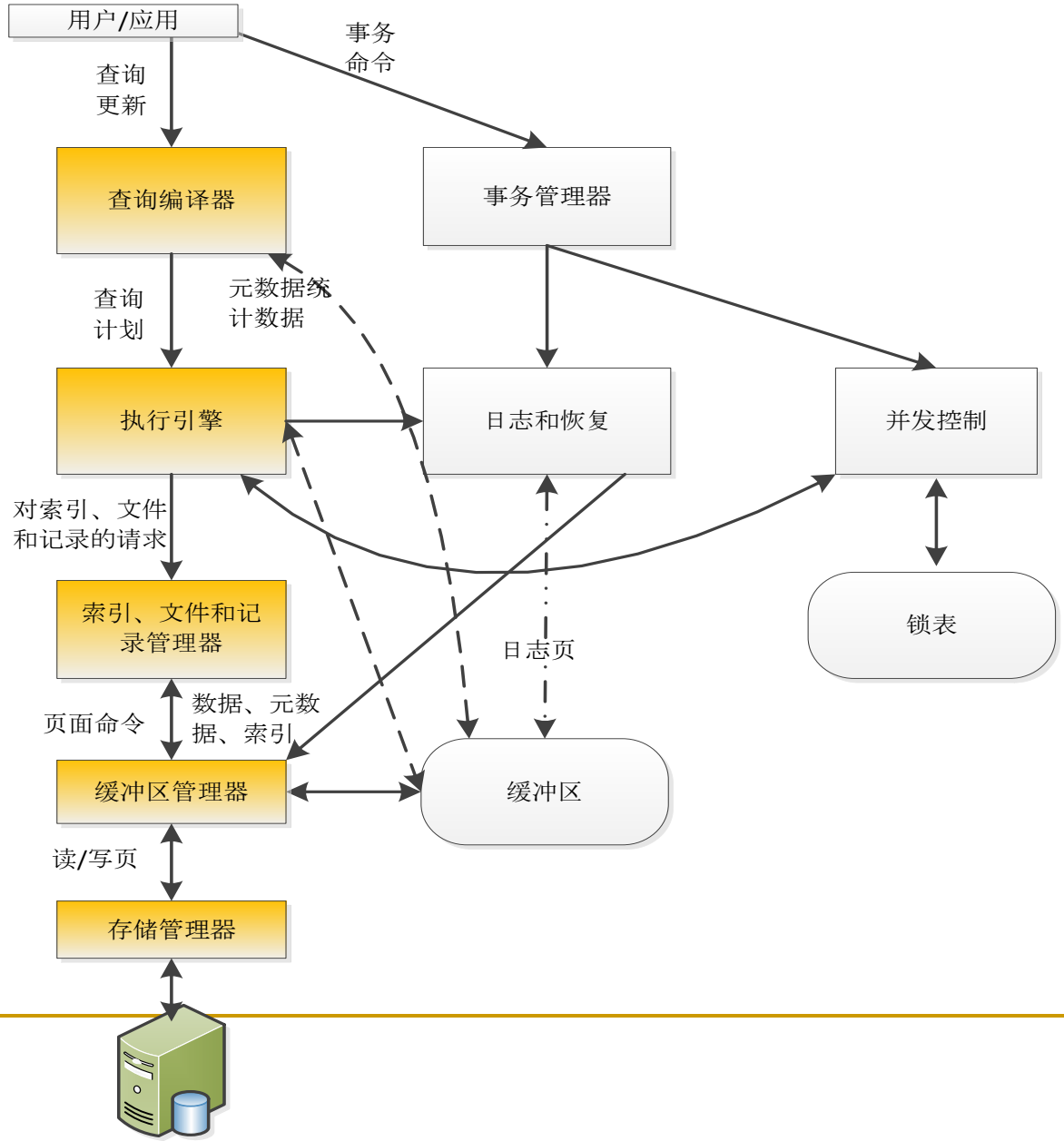
（2）查询计划由执行引擎执行，向索引/记录管理器发出数据请求

（3）数据请求被翻译成对页面的请求，传给缓冲区管理器

（4）缓冲区管理和存储管理器进行通信

备注：存储管理器可以使用操作系统命令，也可以直接发命令给磁盘控制器

查询（DML部分）处理概述



查询处理——两个关键部件

（1）查询编译器

（a）查询分析器，由文本形式的查询出发，建立语法分析树

（b）查询预处理器，对查询进行语义检查，并将语法分析树转换成关系代数表达式树

（c）查询优化器，一般采用代数优化和代价优化策略选择好的执行路径（要访问元数据和统计数据）

（2）执行引擎

它要和索引/记录管理器、日志管理器和并发恢复控制器进行交互。

查询处理概述——事务处理

每一个查询（select）或修改(update)动作就是一个事务(transaction)，事务处理器主要分成两个部分

（1）并发控制管理器（或调度器），保证原子性（Atomicity）、隔离性(Isolation)

（2）日志和恢复管理器，保证一致性(Consistency)、持久性(Durability)

缓冲区管理器

缓冲区管理器负责将主存分割成缓冲区，所有外存数据都要在缓冲区处理，这些数据包括

（1）元数据，比如表模式，日志等

（2）统计量，比如列的数据分布直方图

（3）索引，是提高查询响应速度的“利剑”

（4）用户数据

后续内容安排

三块任务（存储和索引、查询处理、事务管理）

（1）存储管理（教材的第3章，自学第2章）

（2）索引管理（教材的第4、5两章）

（3）查询分析器（教材的第6、7两章）

（4）日志管理（见PPT，结合教材第8章）

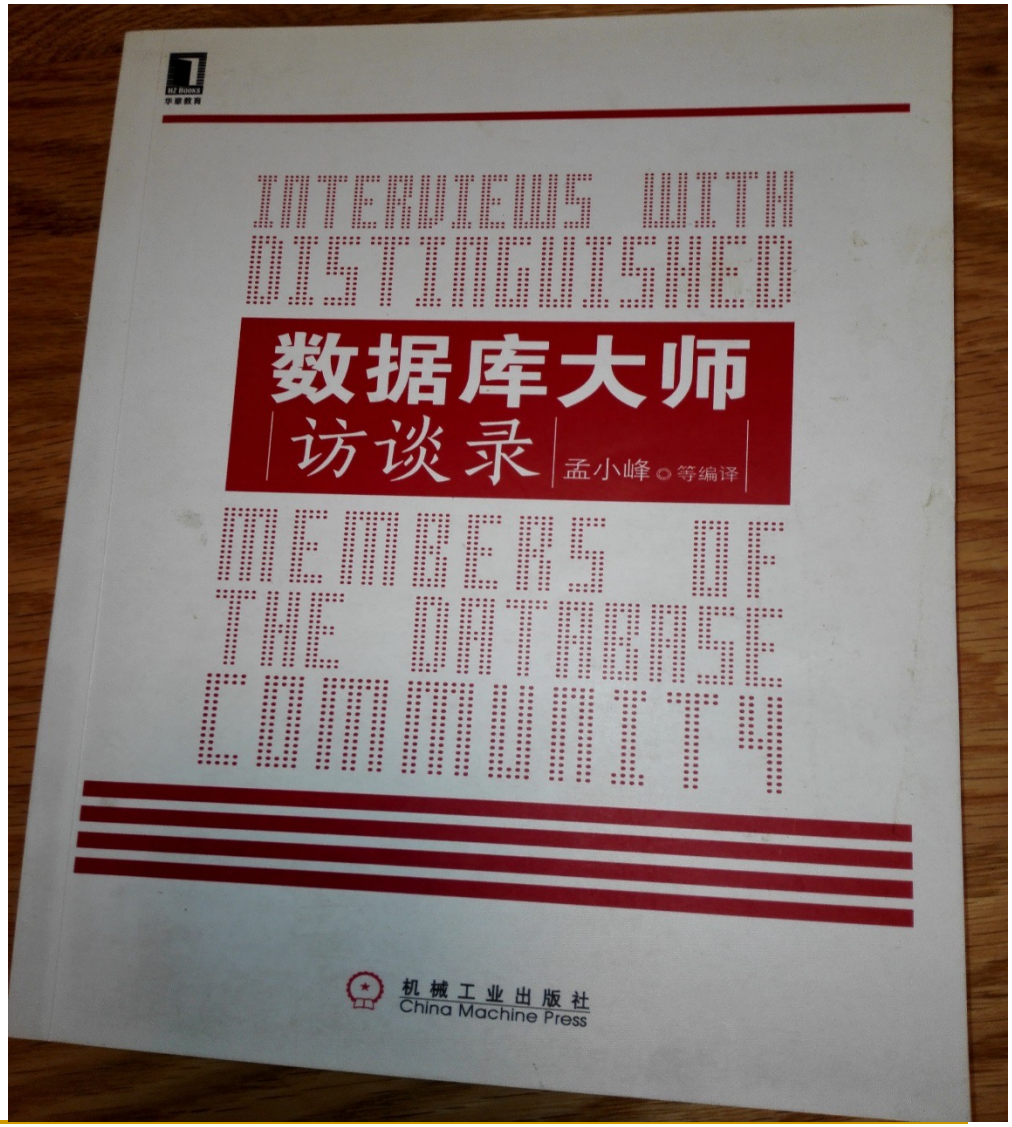
编程作业

结合PPT和演示代码，填充示例系统（mini_base_blank）中的空白代码

参考资料

《数据库大师访谈录》
（孟小峰等翻译）

http://idke.ruc.edu.cn/publications/books/Database%20Distinguished%20Profile.htm



参考资料

国内数据库大牛的工作（孟小峰）

http://idke.ruc.edu.cn/index_cn.htm

两个华人大牛：

陈品山（Peter Chen）——ER模型

韩家炜 （Jiawei Han）——数据挖掘