数据库物理结构设计

引言

设计数据库物理结构前的准备工作

- 了解所用的DBMS的功能、性能和特点，包括提供的物理环境、存储结构、存取方法和可利用的工具；
- 了解应用需求，包括对数据库的操作方式和处理频率、时间响应方面的要求；
- 了解数据库存储设备的特性，如磁盘存储区的划分原则，磁盘块的大小以及I/O特性等。

引言

数据库的物理结构设计解决的问题

- **数据库存储结构设计**

  解决数据文件中记录的存储问题，使应用所要访问的记录尽量存储在同一磁盘块上，对磁盘的I/O操作次数最少。

- **数据库存取方法设计**

  解决如何尽快找到所需记录，找到记录所在磁盘块的磁盘I/O操作次数最少。

讲授内容

1. 定义数据库的模式结构
2. 创建数据库

文件组织结构
文件存储结构
数据存取方法

定义数据库模式结构

- 用SQL定义关系数据库三级模式结构
  - 库名称、库文件名称
  - 库中表的名称，表中各属性的名称及其数据类型
  - 视图名称及其属性名称
  - 索引名称

使用英文名称

从DBMS中选择

案例系统

班级（班级号，所在系，专业，人数）

教研室（名称，所在系，室主任，人数）

课程（课程号，课程名，学分）

学生（学号，姓名，性别，所在班级号）

教师（教师编号，姓名，职称，所在教研室）

班级选课（班级号，课程号，教师编号，上课年度，上课学期）

选课成绩（学号，课程号，成绩）

研究生（学号，所在学科，研究方向，指导教师）

导师（教师编号，所在学科，导师类别）

定义概念模式

学生（学号，姓名，性别，所在班级号）

```
CREATE TABLE student
( sno char(10) primary key,
  sname char(10),
  gender char (2),
  cno char (10),
  foreign key(cno) references class(cno),
  check  (gender  in ('男','女'))
)
```

定长记录

定义概念模式

学生（学号，姓名，性别，所在班级号）

```
CREATE TABLE student
( sno char(10) primary key,
  sname varchar(20),
  gender char (2),
  cno char (10),
  foreign key(cno) references class(cno),
  check  (gender  in ('男','女'))
)
```

变长记录

定义概念模式

学生（学号，姓名，性别，所在班级号）

```
CREATE TABLE student
( sno char(10) primary key,
  sname varchar(20),
  gender char (2),
  cno char (10),
  foreign key(cno) references class(cno),
  check  (gender  in ('男','女'))
)
```

文件组织结构
- 定长记录
- 变长记录

定义概念模式

课程（课程号，课程名，学分）

```
CREATE TABLE lesson
( lno char(10) primary key,
  lname char(20),
  credit int
)
```

定义概念模式

班级（班级号，所在系，专业，人数）

```
CREATE TABLE class
( cno char(10) primary key,
  department char(20),
  speciality char(20),
  number int,
)
```

定义概念模式

教师（教师编号，姓名，职称，所在教研室）

```
CREATE TABLE teacher
( tID char(10) primary key,
  tname char(10),
  title char(10),
  oname char(20),
  foreign key(oname) references office (oname)
)
```

定义概念模式

研究生（学号，所在学科，研究方向，指导教师）

```
CREATE TABLE postgraduate
( sno char(10) primary key,
  discipline char(20),
  specialization char (20),
  mentorID char (10),
  foreign key(sno) references student(sno)
        ON DELETE CASCADE
        ON UPDATE CASCADE ,
  foreign key(mentorID) references teacher(tID)
)
```

定义用户外模式

学生（学号，姓名，性别，所在班级号）

研究生（学号，所在学科，研究方向，指导教师）

```
CREATE VIEW  s_postgraduate(sno, sname, gender, cno, discipline,
specialization, mentorID)
   AS  SELECT   student.sno, sname, gender, cno, discipline,
specialization, mentorID
          FROM student, postgraduate
          WHERE student.sno=postgraduate.sno
```

定义内模式

- 索引是DBMS所采用的主要存取方法，DBMS可基于索引制定优化查询计划，提高数据查询的效率。
- 索引的创建可确定数据文件的存储结构。

定义内模式

选课成绩（学号，课程号，成绩）

```
CREATE  TABLE  grade
( sno   char(10),
  lno   char(10),
  score  dec(4,1),
  primary  key  CLUSTERED  (sno, lno),
  foreign  key(sno) references  student(sno),
  foreign  key(lno) references  lesson(lno)
)
```

生成顺序存储文件

高效地完成按主键进行查询

定义内模式

选课成绩（学号，课程号，成绩）

```
CREATE  TABLE  grade
( sno   char(10),
  lno    char(10),
  score  dec(4,1),
  primary  key  NONCLUSTERED  (sno, lno),
  foreign  key(sno) references  student(sno),
  foreign  key(lno) references  lesson(lno)
)
```

生成堆存储文件

利用索引进行快速访问

定义内模式

选课成绩（学号，课程号，成绩）

```
CREATE  TABLE  grade
( sno   char(10),
  lno    char(10),
  score  dec(4,1),
  primary  key  NONCLUSTERED  (sno, lno),
  foreign  key(sno) references  student(sno),
  foreign  key(lno) references  lesson(lno)
)
```

文件存储结构

- 顺序存储文件
- 堆存储文件

案例系统

- 查询2018年度给两个或两个以上班级讲授过同一门课程的教师编号和所教授的课程号。
- 统计“计算机系”所有教师的教师编号，教师姓名，2018年度教授的总课程数和总学分数，按总学分数从低到高排列。
- 查询选修了“数据库”但没有选修“软件工程”的班级号，所属专业和该班学生人数。
- 统计“计算机系”学生中“数据库原理与应用”课程分数最高的学生学号，姓名和所得分数。

定义内模式

课程（课程号，课程名，学分）

```
CREATE TABLE lesson
( lno char(10) primary key,
  lname char(20),
  credit int
)
```

```
CREATE  INDEX  l1
   ON lesson(lname)
```

定义数据库

创建数据库的语句格式:

- CREATE DATABASE <数据库名>
  [ON
  <filespec>[,<filespec>,...]
  [LOG ON <filespec>[,<filespec>,...]]

[PRIMARY] [（[NAME=<逻辑数据文件名>，]
FILENAME= ‘<操作系统数据文件路径和文件名>’
[，SIZE=<文件长度>]
[，MAXSIZE=<最大长度>]
[，FILEGROWTH=<文件增长率或文件增长大小>]）[，...n]]

定义数据库

```
CREATE DATABASE TMS
  ON
   ( NAME='tms_dat',
      FILENAME='d:\tmsdat.mdf',
      SIZE=10MB,
      MAXSIZE=50MB,
      FILEGROWTH=5%)
  LOG ON
   ( NAME='tms_log',
      FILENAME='e:\tmslog.ldf',
      SIZE=5MB,
      MAXSIZE=25MB,
      FILEGROWTH=5MB)
```

数据文件

日志文件

定义数据库

```
CREATE DATABASE TMS
 ON PRIMARY
  ( NAME='tms_dat',
     FILENAME='d:\tmsdat.mdf',
     SIZE=10MB,
     MAXSIZE=50MB,
     FILEGROWTH=5%),
 FILEGROUP TMS_FG1
  ( NAME='TMS_FG1_dat1',
    FILENAME='d:\TMSFG1dat1.ndf'),
 FILEGROUP TMS_FG2
  ( NAME='TMS_FG2_dat1',
     FILENAME=  'e:\TMSFG2dat1.ndf'),
  ( NAME= 'TMS_FG2_dat2',
     FILENAME=  'f:\TMSFG2dat2.ndf')
 LOG ON
     (.....)
```

主文件组

用户定义文件组

次要文件

定义数据库

```
CREATE DATABASE TMS
 ON PRIMARY
  ( NAME='tms_dat',
     FILENAME='d:\tmsdat.mdf',
     SIZE=10MB,
     MAXSIZE=50MB,
     FILEGROWTH=5%),
 FILEGROUP TMS_FG1
  ( NAME='TMS_FG1_dat1',
    FILENAME='d:\TMSFG1dat1.ndf'),
 FILEGROUP TMS_FG2
  ( NAME='TMS_FG2_dat1',
     FILENAME=  'e:\TMSFG2dat1.ndf'),
  ( NAME= 'TMS_FG2_dat2',
     FILENAME=  'f:\TMSFG2dat2.ndf')
 LOG ON
     (……)
```

```
CREATE TABLE student
( sno char(10) primary key,
  sname char(10),
  gender char (2),
  cno char (10),
  foreign key(cno) references class(cno),
  check  (gender  in ('男','女'))
)
ON TMS_FG1

CREATE   TABLE  grade
( sno   char(10),
  lno    char(10),
  score  dec(4,1),
  primary  key (sno, lno),
  foreign  key(sno) references  student(sno),
  foreign  key(lno) references  lesson(lno)
)
ON TMS_FG1
```

定义数据库

```
CREATE DATABASE TMS
 ON PRIMARY
  ( NAME='tms_dat',
     FILENAME='d:\tmsdat.mdf',
     SIZE=10MB,
     MAXSIZE=50MB,
     FILEGROWTH=5%),
 FILEGROUP TMS_FG1
  ( NAME='TMS_FG1_dat1',
    FILENAME='d:\TMSFG1dat1.ndf'),
 FILEGROUP TMS_FG2
  ( NAME='TMS_FG2_dat1',
     FILENAME=  'e:\TMSFG2dat1.ndf'),
  ( NAME= 'TMS_FG2_dat2',
     FILENAME=  'f:\TMSFG2dat2.ndf')
 LOG ON
     (……)
```

```
CREATE TABLE election
(cno char(10) ,
 lno char(10),
 tID char(10),
 year int ,
 semester char(4),
 primary key(cno,lno),
 foreign  key (cno) references  class(cno),
 foreign  key (lno) references  lesson(lno),
 foreign  key (tID) references  teacher(tID)
)
ON TMS_FG2
```

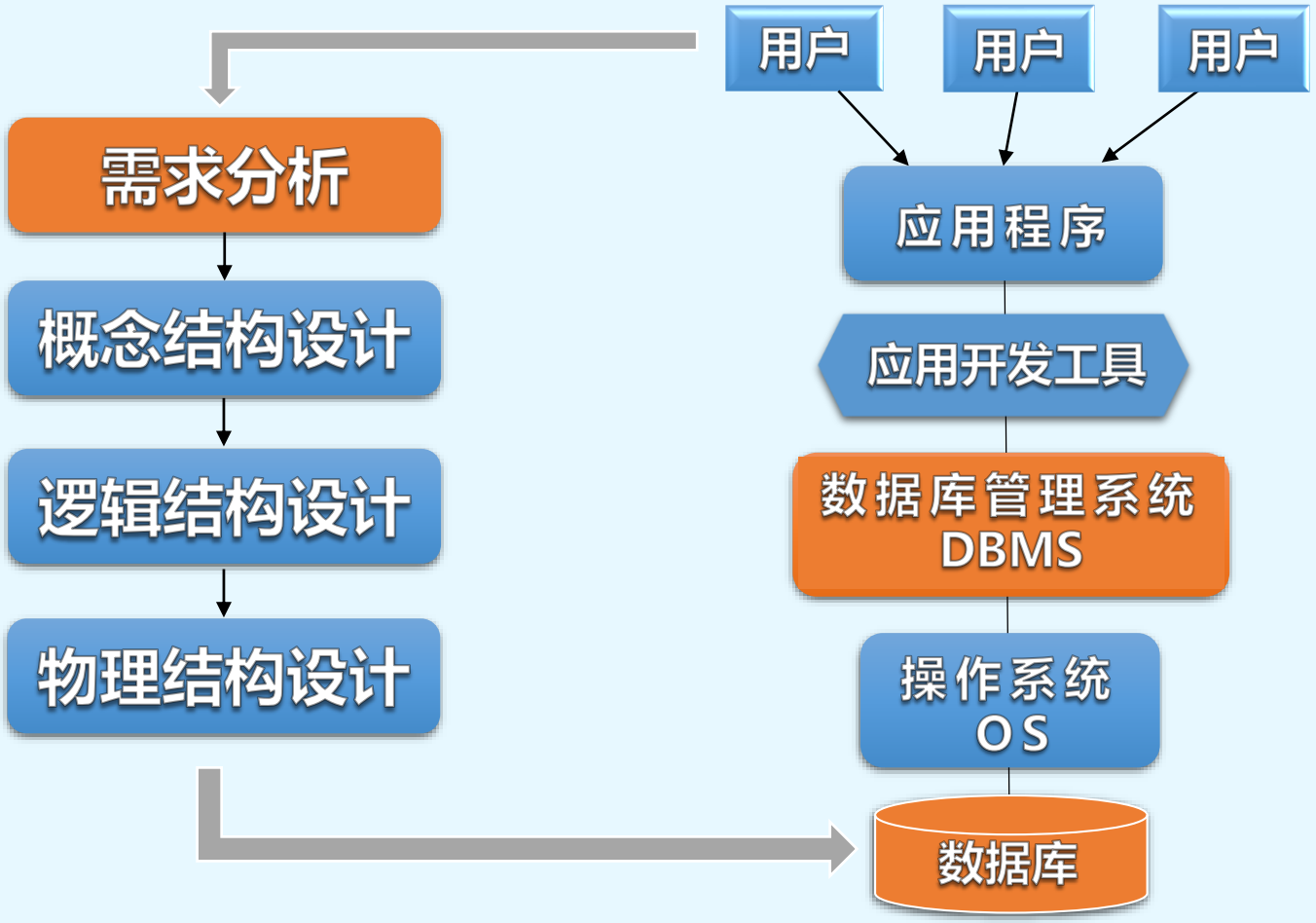
物理结构设计策略

- 通过改变存储结构，提高存取效率。
- 通过完善数据布局，减少访问磁盘的I/O操作的次数。
- 有效利用多磁盘驱动器的并发存取能力，加快存取数据的速度。

小结

- 设计人员可能还需要优化设置DBMS产品的系统配置变量、存储分配参数等。
- 数据库的三级模式的建立，应先创建数据库、库中基本表，再创建视图和索引等。

总结


用户
用户
用户
需求分析
概念结构设计
逻辑结构设计
物理结构设计
应用程序
应用开发工具
数据库管理系统
DBMS
操作系统
OS
数据库

总结

概念模型

DBMS支持的数据模型

物理模型

抽象与模型