

Pr. 8 Database

2017.6

数据库是什么？

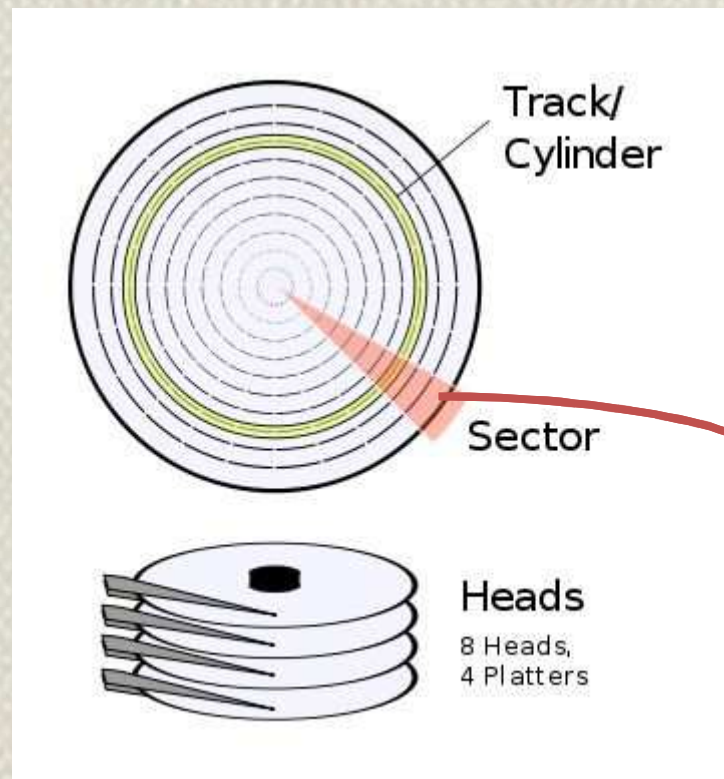
- 是对数据的文件化保存
- 是对数据文件的组织和管理
- 是对数据进行方便操作的系统

数据库主要有哪些功能？

- 从硬盘读取数据
- 增/删/改/查数据
- 数据写入硬盘
- 日志

为什么要用到数据库

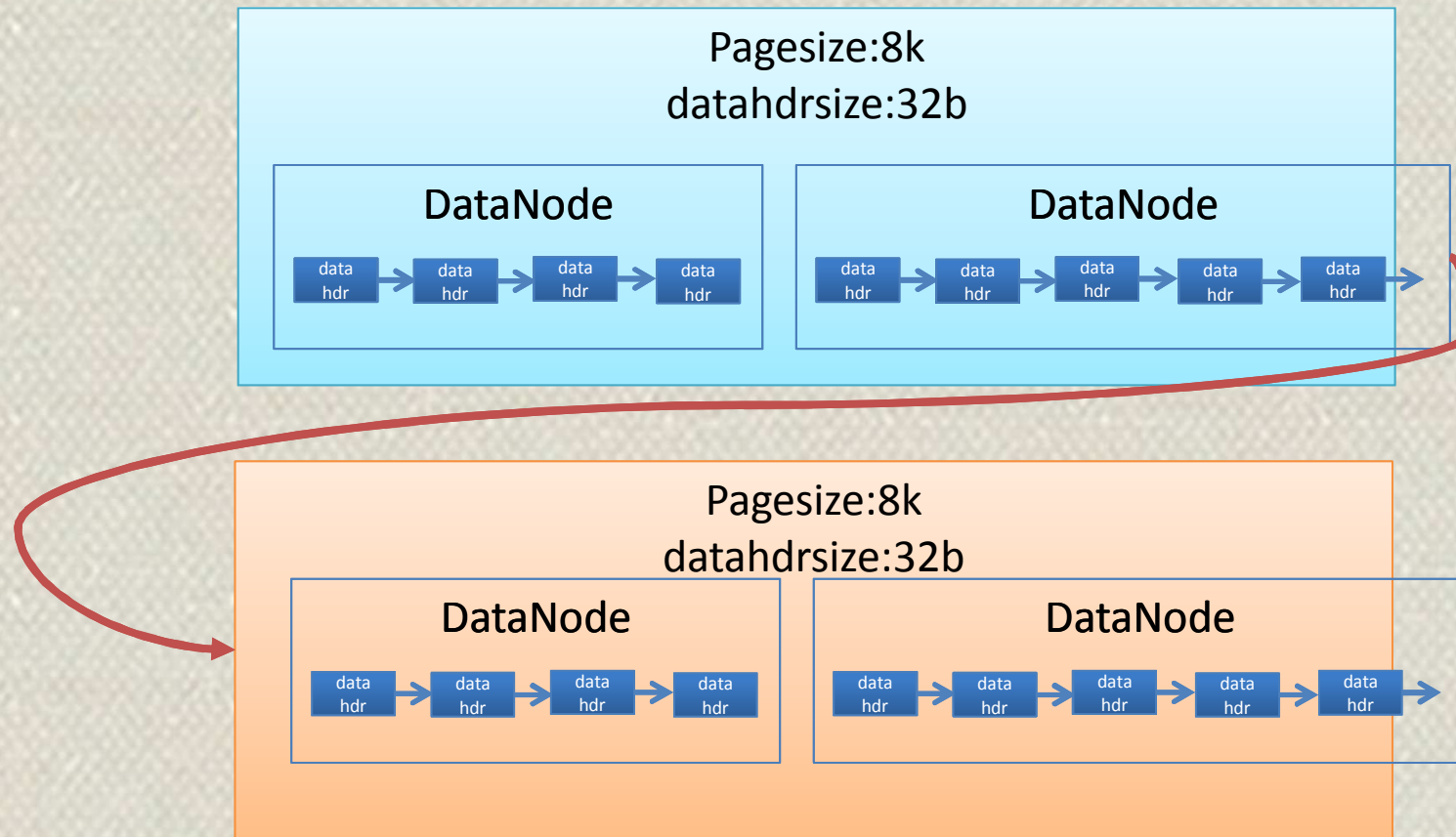
- 文件在硬盘上是怎么存储的？
- 怎样访问硬盘上的文件更快更有效？
- 数据访问更快的方法



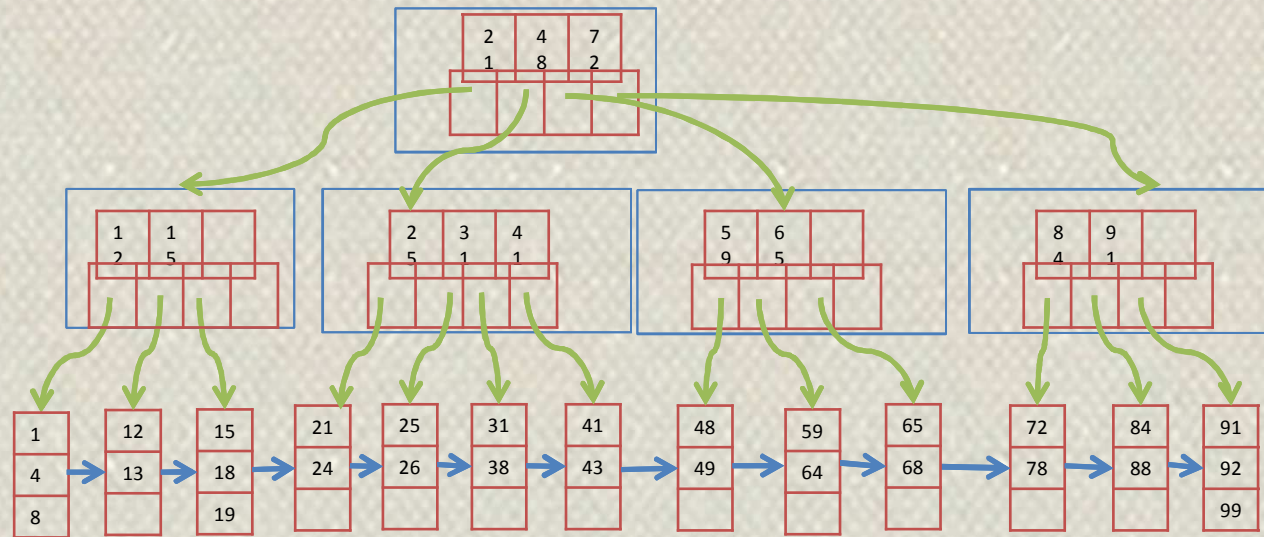
文件的存储方式

- 每个sector上存储 512 字节
- 每个文件存储在[1, n]个sector上
- 文件块之间有索引
- 怎么能快速读出一个文件的数据？

算法基础及数据结构(DataPage)



算法基础及数据结构(index和B+树)

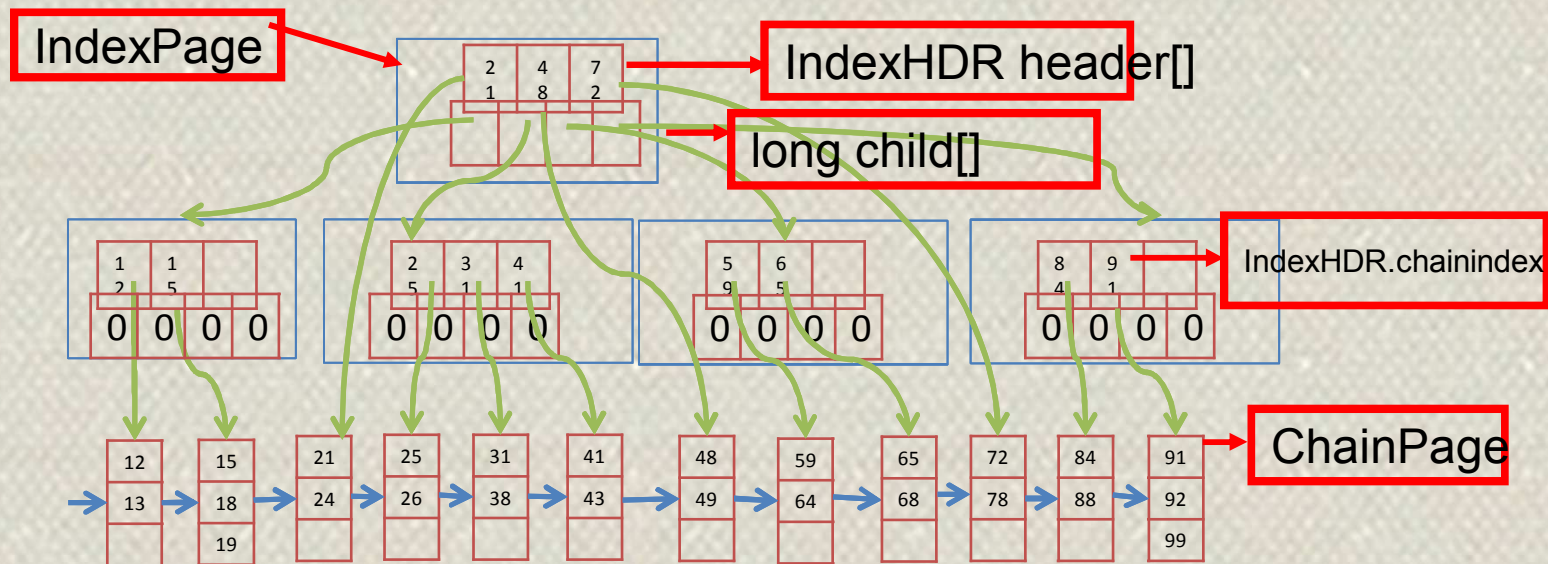


1. 定义任意非叶子结点最多只有 M 个儿子($M > 2$)
2. 根结点的儿子数为 $[2, M]$
3. 除根结点以外的非叶子结点的儿子数为 $[M/2, M]$
4. 每个结点存放至少 $M/2 - 1$ (取上整) 和至多 $M - 1$ 个关键字
5. 非叶子结点的关键字个数 = 指向儿子的指针个数 - 1
6. 非叶子结点的关键字 $K[1], \dots, K[M-1]$; 且 $K[i] < K[i+1]$
7. 非叶子结点的指针: $P[1], \dots, P[M]$; 其中 $P[i]$ 指向关键字属于 $[K[i-1], K[i])$ 的子树
8. 所有叶子结点位于同一层, 所有叶子节点构成一条链

B/B+树的优势

- 平衡二叉树，快速访问到叶子
- 扁平，读文件的话，不用多次读取文件

算法基础及数据结构(index和B+树)



1. Header数组存放IndexHDR, IndexHDR里面有chainindex和chainpage的firstkey
2. Child数组存放child indexpage的addr
3. Leaf IndexPage的child数组全是0

数据库的一些基本操作

- 请玩一玩mysql
- Access也可以用
- 用户/授权/增/删/改/查

创建用户

- `mysql -u root -padminpwd`
- `use mysql`
- `create user 'username'@'localhost' identified by '123456'`
- `mysql - username test -p123456`

创建数据库和数据表

- 以管理员身份登录
- `mysql > create database mydbname;`
- `mysql > use mydbname;`
- `mysql> create table MyClass(
 > id int(4) not null primary key auto_increment,
 > name char(20) not null,
 > sex int(4) not null default '0',
 > degree double(16,2));`

Grant

- **mysql> GRANT ALL PRIVILEGES ON *.* TO 'monty'@'localhost' IDENTIFIED BY '123456'**

增/删/改/查

- insert into (where)
- del from/drop table/drop database
- update (where)
- select from (where)

程序连接

- 每个数据系统都有相关的程序API
- 一般在安装目录下的include目录里
- ODBC/ADO
- JDBC

连接示例

- MySQL数据库安装之后在/MySQL Server 5.0/lib/opt目录下
 - libmysql.lib
 - libmysql.dll
- include目录下的几个文
 - mysql_version.h/my_list.h/mysql_com.h/mysql_time.h
 - mysql.h/my_alloc.h/typelib.h

数据库连接

```
req.c
1  int VspdCToMySQL::ConnMySQL(char *host, char * port, char * Db, char * user, char* passwd, char * charset, char * Msg)
2  {
3      if (mysql_init(&mysql) == NULL)
4      {
5          Msg = "inital mysql handle error";
6          return 1;
7      }
8
9      if (mysql_real_connect(&mysql, host, user, passwd, Db, 0, NULL, 0) == NULL)
10     {
11         Msg = "Failed to connect to database: Error";
12         return 1;
13     }
14
15     if (mysql_set_character_set(&mysql, "GBK") != 0)
16     {
17         Msg = "mysql_set_character_set Error";
18         return 1;
19     }
20     return 0;
21 }
```


Select

```
23 string VspdCToMySQL::SelectData(char * SQL, int Cnum, char * Msg)
24 {
25     MYSQL_ROW m_row; MYSQL_RES *m_res; char sql[2048];
26     sprintf(sql, SQL);
27     int rnum = 0; char rg = 0x06; char cg = { 0x05 }; //字段隔开
28     if (mysql_query(&mysql, sql) != 0)
29     {
30         Msg = "select ps_info Error";
31         return "";
32     }
33     m_res = mysql_store_result(&mysql);
34
35     if (m_res == NULL)
36     {
37         Msg = "select username Error";
38         return "";
39     }
40     string str("");
41     while (m_row = mysql_fetch_row(m_res))
42     {
43         for (int i = 0; i < Cnum; i++)
44         {
45             str += m_row[i];
46             str += rg;
47         }
48         str += cg;
49         rnum++;
50     }
51     mysql_free_result(m_res);
52     return str;
```

JDBC—连接

limax ▸ src ▸ (default package) ▸ Test ▸ getConn() : Connection

```
1 import java.sql.DriverManager;
2 import java.sql.SQLException;
3
4 import com.mysql.jdbc.Connection;
5
6 public class Test {
7     private static Connection getConn() {
8         String driver = "com.mysql.jdbc.Driver";
9         String url = "jdbc:mysql://localhost:3306/samp_db";
10        String username = "root";
11        String password = "";
12        Connection conn = null;
13        try {
14            Class.forName(driver); //classLoader, 加载对应驱动
15            conn = (Connection) DriverManager.getConnection(url, username, password);
16        } catch (ClassNotFoundException e) {
17            e.printStackTrace();
18        } catch (SQLException e) {
19            e.printStackTrace();
20        }
21        return conn;
22    }
}
```


JDBC—insert

```
limax ▸ src ▸ (default package) ▸ Test ▸ insert(String, String, String) : int
4 import com.mysql.jdbc.Connection;
5 import com.mysql.jdbc.PreparedStatement;
6
7 public class Test {
8     private static int insert(String name, String id, String age) {
9         Connection conn = getConn();
10        int i = 0;
11        String sql = "insert into students (Name,Sex,Age) values(?,?,?)";
12        PreparedStatement pstmt;
13        try {
14            pstmt = (PreparedStatement) conn.prepareStatement(sql);
15            pstmt.setString(1, name);
16            pstmt.setString(2, id);
17            pstmt.setString(3, age);
18            i = pstmt.executeUpdate();
19            pstmt.close();
20            conn.close();
21        } catch (SQLException e) {
22            e.printStackTrace();
23        }
24        return i;
25    }
26}
```