

数据库

数据库基本概念

数据库

DataBase(数据库), 存储数据的仓库。

之前我们项目中将用户信息写入一个文件, 最终用一个目录users保存所有用户信息。这个users目录就可以称为

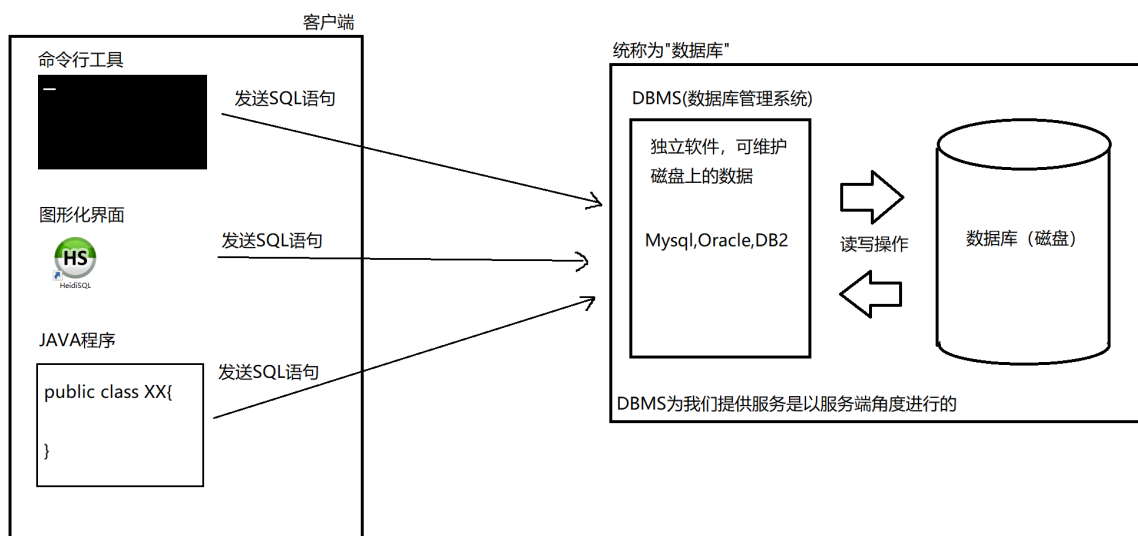
是一个数据库。利用文件系统进行操作, 经常已手动或半自动形式维护数据的操作, 缺点:低效。

数据库管理系统

DBMS(数据库管理系统)。可独立运行的软件, 用于维护数据的操作。

常见的DBMS有:

- mysql
- oracle
- db2
- sqlserver



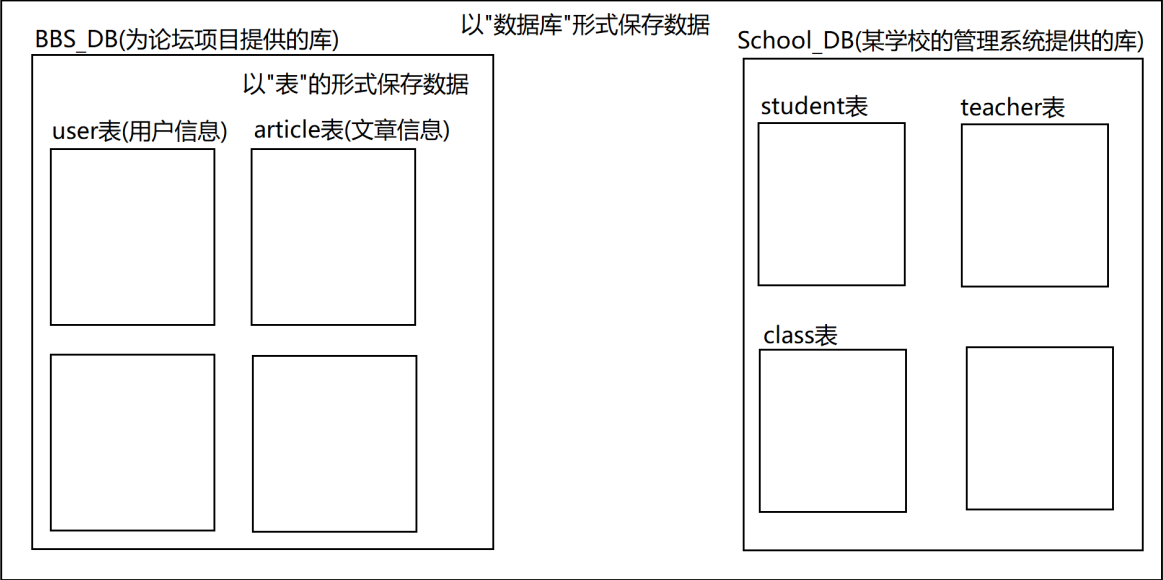
数据库与表的概念

以Mysql为例:

我们可以在Mysql中创建若干个数据库, 每个数据库用于一个项目。

每个数据库中又可以创建多个表, 表是用于保存一组数据的。

示意图:



数据表是由行和列构成

其中列被称为字段，就是一组数据中各部分信息。

其中行被称为记录，就是各部分信息组成的一条数据。

注:用面向对象的角度思考:

表相当于是一个java类。比如User类

字段相当于类中的属性。比如User类中有username,password,nickname,age四个属性

记录相当于类的一个实例。比如一个User实例就可以表示一个用户信息

记录 ➡

username	password	nickname	age
张三	123456	阿三	22
李四	556677	老四	33
...

↓ 字段

如何操作DBMS

所有的DBMS都支持通过SQL语句进行操作。我们向数据库发送特定的SQL语句来表达要进行某项操作。

SQL有标准：SQL92

所有的DBMS都支持SQL92标准。

注:Structured Query Language (SQL) 结构化查询语言

连接数据库的方式

1. 命令行形式
2. 第一方或第三方提供的图形化界面的客户端
3. 在集成开发环境中(IDEA,Eclipse)
4. JDBC(java 数据库连接), java程序中若需要使用数据库则这样连接(IDEA,Eclipse都采取这种方式)

SQL语句分类

- DDL 数据定义语言
CREATE,DROP,ALTER
对数据库对象进行操作的语言。数据库对象有:数据库, 表, 视图, 索引等。
- DML 数据操作语言
INSERT,UPDATE,DELETE
对表中的数据进行操作的语言。
- DQL 数据查询语言
SELECT
对表中的数据进行查询的语言。
- DCL 数据控制语言
DBA用于管理数据库的语言。
- TCL 事物控制语言
COMMIT,ROLLBACK
对DML数据操作保证具有原子性, 一致性。

DDL 数据定义语言

查看DBMS中已有的数据库

```
SHOW DATABASES
```

数据库相关操作

新建一个数据库

基本语法:

```
CREATE DATABASE 数据库名 [CHARSET=字符集名称]
```

例如:

```
新建一个名为mydb的数据库  
CREATE DATABASE mydb
```

创建数据库时可以指定字符集

```
CREATE DATABASE mydb1 CHARSET=UTF8  
CREATE DATABASE mydb2 CHARSET=GBK
```

查看数据库信息

```
SHOW CREATE DATABASE 数据库名
```

例:

```
SHOW CREATE DATABASE mydb1
```

删除数据库

```
DROP DATABASE 数据库名
```

例:

```
DROP DATABASE mydb1
```

使用一个数据库(切换一个数据库)

若希望保存数据，则数据必须保存在某张表上。而表必须保存在某个数据库上，因此后期为了对某个数据库的表进行操作，要先切换到该数据库上，才能进行操作。

```
USE 数据库名
```

例:切换到mydb数据库

```
USE mydb
```

练习:

1. 创建 mydb1和mydb2 数据库 字符集分别为utf8和gbk

```
CREATE DATABASE mydb1 CHARSET=utf8  
CREATE DATABASE mydb2 CHARSET=gbk
```
2. 查询所有数据库检查是否创建成功

```
SHOW DATABASES
```
3. 检查两个数据库的字符集是否正确

```
SHOW CREATE DATABASE mydb1  
SHOW CREATE DATABASE mydb2
```
4. 先使用mydb2 再使用 mydb1

```
USE mydb2  
USE mydb1
```
5. 删除两个数据库

```
DROP DATABASE mydb1  
DROP DATABASE mydb2
```

表相关操作

创建表

语法:

```
CREATE TABLE 表名(  
    字段名 类型,  
    字段名 类型,  
    ...  
)
```

例:

```
CREATE TABLE user(  
    id INT,  
    username VARCHAR(30),  
    password VARCHAR(30),  
    nickname VARCHAR(30),  
    age INT(3)  
)
```

INT在mysql中为整数类型。
VARCHAR在mysql中为字符串类型，长度为字节，
VARCHAR(30)则字符转换的字节最多30个，
若UTF-8编码则中文最多10个字(每个中文3字节)。
INT指定长度为位数。这里INT(3)为最多3位整数。

查看已创建的表的表结构

语法:

```
DESC 表名
```

例:

查看user表的表结构

```
DESC user
```

查看已创建表的详细信息

```
SHOW CREATE TABLE 表名
```

例如:

```
SHOW CREATE TABLE user
```

删除表

语法:

```
DROP TABLE 表名
```

例:

删除user表

```
DROP TABLE user
```

查看数据库中所有的表

```
SHOW TABLES
```

修改表

- 修改表名

语法:

```
RENAME TABLE 原表名 TO 新表名
```

例如:

将表user改名为userinfo

```
RENAME TABLE user TO userinfo
```

- 修改表结构

准备一张表测试：

```
CREATE TABLE hero(  
    name VARCHAR(30),  
    age INT(3)  
)
```

Field	Type
name	varchar(30)
age	int(3)

○ 添加字段

- 向表末尾追加新的字段

```
ALTER TABLE 表名 ADD 字段名 类型
```

例:

```
ALTER TABLE hero ADD gender VARCHAR(10)
```

Field	Type
name	varchar(30)
age	int(3)
gender	varchar(10)

- 将字段添加到表的第一个字段位置上

```
ALTER TABLE 表名 ADD 字段名 类型 FIRST
```

例:

在hero表第一个字段上添加id

```
ALTER TABLE hero ADD id INT FIRST
```

Field	Type
id	int(11)
name	varchar(30)
age	int(3)
gender	varchar(10)

- 将字段插入到表中某个位置

将字段插入到表中某字段的后面

ALTER TABLE 表名 **ADD** 字段名 类型 **AFTER** 表中某字段

例:

将密码(pwd)字段插入到name和age之间

ALTER TABLE hero **ADD** pwd **VARCHAR**(30) **AFTER** name

	Field	Type
1	id	int(11)
2	name	varchar(30)
3	pwd	varchar(30)
4	age	int(3)
5	gender	varchar(10)

○ 删除字段

ALTER TABLE 表名 **DROP** 字段名

例:

将pwd字段从hero表中删除

ALTER TABLE hero **DROP** pwd

○ 修改现有字段

ALTER TABLE 表名 **CHANGE** 原字段名 新字段名 类型

例:

- 将hero表中age字段的长度改为5

ALTER TABLE hero **CHANGE** age age **INT**(5)

	Field	Type
1	id	int(11)
2	name	varchar(30)
3	age	int(3)
4	gender	varchar(10)




	Field	Type
1	id	int(11)
2	name	varchar(30)
3	age	int(5)
4	gender	varchar(10)

- 将hero表中age字段的类型改为VARCHAR(10)

ALTER TABLE hero **CHANGE** age age **VHARCHAR**(10)


Field	Type
1 id	int(11)
2 name	varchar(30)
3 age	int(5)
4 gender	varchar(10)



Field	Type
1 id	int(11)
2 name	varchar(30)
3 age	varchar(10)
4 gender	varchar(10)

- 将hero表中gender字段名改为pwd
`ALTER TABLE hero CHANGE gender pwd VARCHAR(10)`

Field	Type
1 id	int(11)
2 name	varchar(30)
3 age	varchar(10)
4 gender	varchar(10)



Field	Type
1 id	int(11)
2 name	varchar(30)
3 age	varchar(10)
4 pwd	varchar(10)

- 注意事项:
 修改表字段应当在表中没有数据时进行。如果表中已经存在数据, 则修改字段可能不成功。
 比如:
 hero表中有10条记录, 每条记录gender字段已经有值(每个人的性别)
 若此时将表字段gender修改类型为INT。此时字符串转换int("男"怎么转成整数???)
 比如:
 hero表中pwd记录了每个人的密码。原长度为10。此时若修改字段长度为5, 会导致原密码保存不下则修改失败。

练习:

1. 创建数据库mydb3 字符集gbk 并使用
`CREATE DATABASE mydb3 CHARSET=gbk`
`USE mydb3`
2. 创建t_hero英雄表, 有名字和年龄字段
`CREATE TABLE t_hero(
 name VARCHAR(30),
 age INT(3)
)`
3. 修改表名为hero
`RENAME TABLE t_hero TO hero`
4. 查看表hero的信息
`SHOW CREATE TABLE hero`
5. 查询表hero结构
`DESC hero`
6. 删除表hero
`DROP TABLE hero`
7. 删除数据库mydb3
`DROP DATABASE mydb3`

作业:

1. 创建数据库mydb4 字符集utf8并使用
2. 创建teacher表 有名字(name)字段
3. 添加表字段: 最后添加age 最前面添加id(int型) , age前面添加salary工资(int型)
4. 删除age字段
5. 修改表名为t
6. 删除表t
7. 删除数据库mydb4