

基本概念

DB是数据库, DBMS是数据库管理系统。现在的DBMS一般都可以进行创建数据库, 管理数据库等操作。两者不是割裂开来的概念。

简单来看, 数据库用来存储数据, 由表构成。如EXCEL一样。

数据库实际上就是一个文件的集合, 按照特定的格式把数据组织存储起来。

关系型数据库OR非关系型数据库

这里只考虑关系型数据库。

数据库系统

一般来说, 数据库系统包括:

- 数据库: 用来存储数据
- 数据库管理系统: 用来管理数据库
- 数据库应用程序: 软件补充, 比如和用户直接打交道的系统程序

为什么需要DB

因为小型的数据存储组织不适用大型系统。也不可使用多人同时访问、操作等。数据量很大的时候, DB往往性能更好。

如 (同样数据量的EXCEL文件) 进行管理操作, 将会很麻烦。DB的有点如下:

- 可以较好的组织海量数据, 方便用户进行检索和访问。
- 数据库可以保持数据的一致性、完整性
- 数据库可以更好的多人共享

表

是一种抽象化的表。就是数据库的一种文件组织结构 (逻辑结构)。一般可以将DB理解为table的集合, 但是DB也包含了各类针对table的操作。

table_name:

id	name	age
01	Jack	40
02	Pony	43

可以看出基本上就和EXCEL中的二维表格一样。

行

一行就是一条数据记录。一个列称为字段 (field) , 具有相同数据类型的集合。

每个表有一个主键 (primary key) , 用来唯一标识自己。上面中的id就是primary key。主键primary key, 可以是一列或一组列。主键需要唯一标识一行, 因此只要满足这一条件都可成为主键。并且主键不可以为NULL值。

SQL

SQL (structured Query Language) , 结构化查询语言。

所有的DBMS都支持SQL。

MySQL

MySQL就是一个DBMS。可以建立数据库, 管理数据库。

- 基于C/S模式的DBMS
- 基于共享文件系统的DBMS, 如Microsoft access, 用于桌面系统

基于C/S模式, 服务器部分负责所有数据访问和处理的一个软件。运行在服务器上。用户部分是与用户打交道的软件。客户端通过网络提交请求给服务器端, 然后服务器处理这个请求, 根据需要过滤, 丢弃和排序数据, 返回到客户端。

无论服务端和用户端在不在同一个机器上, 都需要进行通信。

请求可以为数据添加、删除、和数据更新。服务器负责这些请求。

使用MySQL

连接

为了连接MySQL, 需要以下信息:

- 主机名 (计算机名) --如果连接到本地服务器, 则为localhost;
- 端口 (默认使用3306)
- username
- password

简单的mysql命令

! 任何mysql命令都需要;or \g结尾。

假设存在名为mysql的数据库, 则可以使用

```
use mysql;
```

打开数据库, 然后进行读取等操作。

```
show databases; // 显示当前可用的数据库列表
show tables from db_name; // 展示名为db_name的数据库的所有table

// 如果选定了某一个数据库, 即先使用
use mysql;
show tables; // 此时, 打开名为mysql的数据库中的所有tables
show columns from table1; // 打开mysql数据库中名为table1的表的所有列
```

注: 这些命令不需要记忆, 因为后续使用navicate管理工具即可。

SQL语句

这部分是重点内容。SQL语句中的关键字是不区分大小写的。

检索单个列

```
SELECT prod_name
FROM products;
```

从products表中检索一个名为prod_name的列。

检索多个列

很容易想到, 检索多个列, 用, 隔开即可:

```
SELECT prod_id, prod_name, prod_price
FROM products;
```

检索结果按照相应的列的检索顺序显示。

检索所有列

更一般的检索所有列, 使用通配符*即可:

```
SELECT *
FROM products;
```

返回结果, 一般和table中一致, 也可能不一致(显示的数据有可能经过排序)。

通配符的检索, 效率一般较低。

检索distinct的行

```
SELECT ven_id
FROM products;
```

可以检索products表中的ven_id一列, 可以通过distinct关键字筛选掉结果中的重复记录(行)。

```
SELECT DISTINCT ven_id
FROM products;
```

假设有如下table, 名为user:

ID	name
01	jack
02	jack
03	tony
04	pony
05	pony

```
SELECT DISTINCT name
FROM user;
```

检索结果如下(示意图):

name
jack
tony
pony

3 rows in set

一个有趣的问题

```
SELECT DISTINCT name, ID
FROM user;
```

会是什么结果呢? 答案是会显示出所有唯一的(name, ID)记录。即

name	ID
jack	01

name	ID
jack	02
tony	03
pony	04
pony	05

! DISTINCT关键字是作用于之后的所有列的,而不仅仅是name。

```
select distinct name, distinct id from user;是错误的语法。
```

限制结果

```
SELECT prod_name  
FROM products  
LIMIT 5;
```

LIMIT关键字,将会限制检索结果的记录(行)数。如上LIMIT 5表明MySQL返回不多于5行。

LIMIT beg,end;其中beg默认为0,即从第1行开始。

使用完全限定

完全限定即指明检索的列为哪个表的哪个库的。

```
SELECT 库名称.表名称.列名  
FROM 库名称.表名;
```