1. 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。

1.1. 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。

1.2. 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

1.3. 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 的优点如下：

- 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

1.4. 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。

1.4.1. 行

一条**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**。

1.5. SQL

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

所有的 DBMS 都支持 SQL。

2. 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。

3. 使用MySQL

3.1. 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

3.1.1. 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。

4. 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。

4.1. 检索单个列

```
SELECT prod_name
FROM products;
```

从 `products` 表中检索一个名为 `prod_name` 的列。

4.2. 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。

4.2.1. 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。

4.3. 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

4.3.1. 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
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。

4.4. 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。

4.5. 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;
```