

创建数据库

`CREATE DATABASE` 数据库名;

删除数据库

`drop database` [数据库名];

选择数据库

`use` [数据库名]

创建表

```
CREATE TABLE IF NOT EXISTS [表名](  
[属性名1] [数据类型] [是否允许空值],  
[属性名2] [数据类型] [是否允许空值],  
.....  
PRIMARY KEY ( [属性名] )  
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

- [属性名]: 自定义, 区分大小写, 不可为中文。
- [数据类型]: int , varchar 等等

如果数据是整数类型, 那么还有以下几个设定:

- **自动增加** `AUTO_INCREMENT` : 每一次数据录入都可以不用输入此项的值, 默认增加一。

通常用于主键中

- **无符号** `UNSIGNED` : 整数类型通常是负多少到正多少的范围, 设置了无符号后, 就不能出现负数了, 但是正数的数值上限多了一倍 (把负数的范围补到了正数上)
- `PRIMARY KEY ([属性名])` : 设置哪个属性为主键。

主键: 在表中具有唯一性, 不可重复。

- 是否允许空值: 允许空值就填 `null` , 不允许就填 `not null` 。
- `ENGINE=InnoDB` : 设置数据库引擎为 InnoDB 。
- `CHARSET=utf` : 设置字符编码类型为 `utf8` 。

删除数据表

```
DROP TABLE [表名1],[表名2] ;
```

插入数据

- ```
INSERT INTO [表名] ([数据名1],[数据名2],[.....])
VALUES
([数据11], [数据12],[.....]);
([数据21], [数据22],[.....]);
.....
```

- 数据名必须与数据值一一对应
- 如果数据本身就与表中的数据名对应，那么数据名那一段就不需要，如：

```
INSERT INTO [表名] VALUES
([数据11], [数据12],[.....]);
([数据21], [数据22],[.....]);
.....
```

- 注意数据一定要与表中设置的格式相同，int对应数字，varchar对应混合，穿错会报错。
- mysql也支持将一个表的内容插入到另一个表中，代码如下

```
INSERT INTO [表名] ([数据名1],[数据名2],[.....]) SELECT [数据名1],[数据名2],[.....] FROM
```

注意，插入指令在列表中已有数据的情况下是无法进行的，有些时候哪怕数据是null也不行。

## 数据查询条件语句

```
SELECT [数据名1],[数据名2],[.....] FROM [表名1],[表名2],[.....]
WHERE [条件1] AND/OR [条件2] AND/OR [条件3].....
```

- WHERE 语句通常是 数据名为条件的，如

```
where id>100
where name="李华"
```

- WHERE 的判定符有以下几个（符合判定条件会返回 true，不符合返回 false）

- = : 等号, 检测两个值是否相等
- <>, != : 不等于, 检测两个值是否相等
- > : 大于号, 检测左边的值是否大于右边的值
- < : 小于号, 检测左边的值是否小于右边的值
- >= : 大于等于号, 检测左边的值是否大于或等于右边的值
- <= : 小于等于号, 检测左边的值是否小于或等于右边的值

## 数据更新条件语句

- `UPDATE [表名] SET [属性名1]=[属性值], [属性名2]=[属性值]  
WHERE [条件1] AND/OR [条件2] AND/OR [条件3].....`
- 数据更新一样支持将其他表的数据复制到本表中

`UPDATE [表名] set [表名].[数据名] = [源表名].[源数据名] , set [表名].[数据名] =  
where [条件1],[条件2],.....`