

# DDL

## DDL功能

### 在数据库中创建表

### MySQL中的数据类型

#### 数值类型

#### 字符串类型

#### 日期和时间类型

### DDL语句对表结构的其他操作

### DDL修改表的结构

#### 新增列

#### 修改列名

#### 删除列

#### 修改表的名字

### DDL不涉及表中的数据

# DDL功能

功能	SQL
查看所有的数据库	<code>show databases;</code>
创建数据库 	<code>create database [if not exists] mydb1 [charset=utf8]</code>
切换 (选择要操作的) 数据库	<code>use mydb1;</code>
删除数据库	<code>drop database [if exists] mydb1;</code>
修改数据库编码	<code>alter database mydb1 character set utf8;</code>

```
con_Mysql  mysql 运行 停止
1  -- DDL操作之数据库操作
2  -- 查看当前数据库
3  show databases;
4  -- 创建数据库
5  create database mydb1;
6  create database if not exists mydb1;
7  -- 选择使用哪一个数据库
8  USE mydb1;
9  -- 删除数据库
10 DROP DATABASE mydb1;
```

## 在数据库中创建表

### □ 创建表格式

```
create table [if not exists] 表名(
    字段名1 类型[(宽度)] [约束条件] [comment '字段说明'],
    字段名2 类型[(宽度)] [约束条件] [comment '字段说明'],
    字段名3 类型[(宽度)] [约束条件] [comment '字段说明']
) [表的一些设置];
```

创建表是构建一张空表，指定这个表的名字，这个表有几列，每一列叫什么名字，以及每一列存储的数据类型。

```
use mydb1;
create table if not exists student(
    sid int,
    name varchar(20),
    gender varchar(20),
    age int,
    birth date,
    address varchar(20)
);
```

## MySQL中的数据类型

常用的有三种：

- 1.数值类型
- 2.日期和时间类型
- 3.字符串类型

# 数值类型

类型	大小	范围（有符号）	范围（无符号）	用途
TINYINT	1 byte	(-128, 127)	(0, 255)	小整数值
SMALLINT	2 bytes	(-32 768, 32 767)	(0, 65 535)	大整数值
MEDIUMINT	3 bytes	(-8 388 608, 8 388 607)	(0, 16 777 215)	大整数值
INT或INTEGER	4 bytes	(-2 147 483 648, 2 147 483 647)	(0, 4 294 967 295)	大整数值
BIGINT	8 bytes	(-9,223,372,036,854,775,808, 9 223 372 036 854 775 807)	(0, 18 446 744 073 709 551 615)	极大整数值
FLOAT	4 bytes	(-3.402 823 466 E+38, 3.402 823 466 351 E+38)	0, (1.175 494 351 E-38, 3.402 823 466 E+38)	单精度 浮点数值
DOUBLE	8 bytes	(-1.797 693 134 862 315 7 E+308, 1.797 693 134 862 315 7 E+308)	0, (2.225 073 858 507 201 4 E-308, 1.797 693 134 862 315 7 E+308)	双精度 浮点数值
DECIMAL		依赖于M和D的值	依赖于M和D的值	小数值

decimal(M,D)---decimal (5,2)保留5位有效数字， 2位小数

# 字符串类型

类型	大小	用途
CHAR	0-255 bytes	定长字符串
VARCHAR	0-65535 bytes	变长字符串
TINYBLOB	0-255 bytes	不超过 255 个字符的二进制字符串
TINYTEXT	0-255 bytes	短文本字符串
BLOB	0-65 535 bytes	二进制形式的长文本数据
TEXT	0-65 535 bytes	长文本数据
MEDIUMBLOB	0-16 777 215 bytes	二进制形式的中等长度文本数据
MEDIUMTEXT	0-16 777 215 bytes	中等长度文本数据
LOBLOB	0-4 294 967 295 bytes	二进制形式的极大文本数据
LONGTEXT	0-4 294 967 295 bytes	极大文本数据

# 日期和时间类型

类型	大小 ( bytes)	范围	格式	用途
DATE	3	1000-01-01/9999-12-31	YYYY-MM-DD	日期值
TIME	3	'-838:59:59'/'838:59:59'	HH:MM:SS	时间值或持续时间
YEAR	1	1901/2155	YYYY	年份值
DATETIME	8	1000-01-01 00:00:00/9999-12-31 23:59:59	YYYY-MM-DD HH:MM:SS	混合日期和时间值
TIMESTAMP	4	1970-01-01 00:00:00/2038 结束时间是第 2147483647 秒，北京时间 2038-1-19 11:14:07，格林尼治时间 2038年1月19日 凌晨 03:14:07	YYYYMMDD HHMMSS	混合日期和时间值，时间戳

## DDL语句对表结构的其他操作

功能	SQL
查看当前数据库的所有表名称	show tables;
查看指定某个表的创建语句	show create table 表名;
查看表结构	desc 表名
删除表	drop table 表名

## DDL修改表的结构

### 新增列

语法格式

```
alter table 表名 add 列名 类型(长度) [约束];
```

例子:

```
#为student表添加一个新的字段为: 系别 dept 类型为 varchar(20)
ALTER TABLE student ADD `dept` VARCHAR(20);
```

```
-- 添加列
ALTER TABLE student ADD 成绩 FLOAT;
```

## 修改列名

语法格式

```
alter table 表名 change 旧列名 新列名 类型(长度) 约束;
```

例子:

```
#为student表的dept字段更换为department varchar(30)
ALTER TABLE student change `dept` department VARCHAR(30);
```

```
ALTER TABLE student ADD 成绩 FLOAT;
-- 修改列成绩的名称
ALTER TABLE student CHANGE 成绩 score FLOAT;
```

## 删除列

◆ 修改表删除列.

语法格式:

```
alter table 表名 drop 列名;
```

例如:

```
#删除student表中department这列
ALTER TABLE student DROP department;
```

## 修改表的名字

语法格式:

```
rename table 表名 to 新表名;
```

例如:

```
#将表student改名为stu
rename table `student` to stu;
```