

数据定义：创建与修改表

一、创建数据表

1. MySQL 基本数据类型

2. 创建表的语法

示例1：创建数据表 `tb_grade`

3. 查看表的数据定义

示例2：使用 `SHOW CREATE TABLE` 查看数据表 `tb_grade`

4. 查看表结构

示例3：使用 `DESCRIBE` 查看数据表 `tb_grade` 的结构

课堂练习2

二、数据表结构的修改

1. 修改表名

示例4：将数据表 `tb_grade` 名修改为 `grade`

2. 修改字段名称

示例5：将数据表 `grade` 中的 `name` 字段改为 `username`

3. 修改字段属性

示例6：将数据表 `grade` 中 `id` 字段的数据类型由 `INT(11)` 修改为 `INT(20)`

4. 添加字段

示例7：将数据表 `grade` 中的添加数据类型为 `DATE` 的字段 `birthday`

5. 删除字段

示例8：将数据表 `grade` 中 `birthday` 字段删除

6. 复制表

示例9：将数据表 `grade` 复制为 `grade_bak` 表

7. 字段排序

示例10：将数据表 `grade_bak` 中 `username` 字段修改至第一个字段

示例11：将数据表 `grade_bak` 中 `id` 字段插入到 `grade` 字段后面

8. 删除表

示例12：删除数据表 `grade_bak`

课堂练习3

数据定义：创建与修改表

一、创建数据表

1. MySQL 基本数据类型

- 数值类型

类型	大小	范围（由符号）	范围（无符号）	用途
<code>tinyint</code>	1字节	$(-2^8, 2^8 - 1)$	$(0, 2^8 - 1)$	整数值
<code>smallint</code>	2字节	$(-2^{15}, 2^{15} - 1)$	$(0, 2^{16} - 1)$	整数值
<code>mediumint</code>	3字节	$(-2^{23}, 2^{23} - 1)$	$(0, 2^{24} - 1)$	整数值
<code>int</code>	4字节	$(-2^{31}, 2^{31} - 1)$	$(0, 2^{32} - 1)$	整数值
<code>bigint</code>	8字节	$(-2^{63}, 2^{63} - 1)$	$(0, 2^{64} - 1)$	整数值
<code>float</code>	4字节	$(-3.402\ 823\ 466\ \text{E}+38,$ -1.175 494 351 E-38), 0, (1.175 494 351 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
<code>double</code>	8字节	$(-1.797\ 693\ 134\ 862\ 315\ 7\ \text{E}+308,$ -2.225 073 858 507 201 4 E-308), 0, (2.225 073 858 507 201 4 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

在这些数值类型前面加上 `unsigned`，则为无符号数值，即取值大于等于0。

- 字符类型

类型	大小	用途
char	0~255字节	定长字符串
varchar	0~65535字节	变长字符串
tinyblob	0~255字节	不超过 255 个字符的二进制字符串
tinytext	0~255字节	字符串
blob	0~65535字节	二进制形式的字符串
text	0~65535字节	字符串
mediumblob	0~16,777,215字节	二进制形式的字符串
mediumtext	0~16,777,215字节	字符串
longblob	0~4294,967,295字节	二进制形式的字符串
longtext	0~4294,967,295字节	字符串

- 时间/日期类型

类型	大小	范围	格式	用途
date	3 字节	'1000-01-01'/'9999-12-31'	YYYY-MM-DD	日期值
time	3 字节	'-838:59:59'/'838:59:59'	HH:MM:SS	时间值
year	1 字节	'1901'~'2155'	YYYY	年份值
datetime	8 字节	'1000-01-01 00:00:00' ~ '9999-12-31 23:59:59'	YYYY-MM-DD HH:MM:SS	混合时间日期
timestamp	4 字节	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	混合时间日期， 时间戳

- 枚举类型

`enum(A, B, C, ...)` 定义1个集合，最多可以存储65535个成员。

例如 性别 enum('男', '女')

2. 创建表的语法

```
1 CREATE [数据库名.]TABLE 表名
2 ( 字段名1 数据类型 [完整性约束条件],
3   字段名2 数据类型 [完整性约束条件],
4   . . . . .
5   字段名n 数据类型 [完整性约束条件],
6   完整性约束
7 ) [其它表级定义];
```

注意

- 对字段性质的不同说明用空格隔开
- 不同字段的定义用逗号 , 隔开
- 表定义结束后用 ; 终结

示例1：创建数据表 tb_grade

```
1 CREATE DATABASE stu_info; -- 准备工作
2 USE stu_info;
3 CREATE TABLE tb_grade(id INT(11),
4   name VARCHAR(20),
5   grade FLOAT);
```

注意

- 数据表必须定义在某一具体数据库当中，因此先创建了 stu_info 数据库，然后选定它并在其中创建表 tb_grade ；
- 也可不选定数据库，采用 <db_name>.<tb_name> 的方式指定在某一数据中创建表。

```
1 -- 指定在stu_info中创建表tb_grade_2
2 CREATE TABLE stu_info.tb_grade_2(id INT(11),
3   name VARCHAR(20),
4   grade FLOAT);
```

3. 查看表的数据定义

基本命令:

```
1 SHOW CREATE TABLE [<db_name>.<table_name>;
```

示例2：使用 SHOW CREATE TABLE 查看数据表 tb_grade

```
1 SHOW CREATE TABLE tb_grade\G
```

执行结果

```

1 ***** 1. row *****
2         Table: tb_grade
3         Create Table: CREATE TABLE `tb_grade` (
4             `id` int(11) DEFAULT NULL,
5             `name` varchar(20) DEFAULT NULL,
6             `grade` float DEFAULT NULL
7         ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
8         1 row in set (0.01 sec)

```

或者

```

1 SHOW CREATE TABLE tb_grade;

```

执行结果

```

1  +-----+-----+-----+-----+-----+-----+-----+-----+
2  | Table      | Create Table
3
4  | tb_grade | CREATE TABLE `tb_grade` (
5      `id` int(11) DEFAULT NULL,
6      `name` varchar(20) DEFAULT NULL,
7      `grade` float DEFAULT NULL
8  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
9  +-----+-----+-----+-----+-----+-----+-----+-----+
10 1 row in set (0.01 sec)

```

4. 查看表结构

基本命令:

```

1 DESC [<db_name>.<table_name>;
2 -- 或者写全
3 DESCRIBE [<db_name>.<table_name>;

```

注意: `DESC` 后不用加 `TABLE` 关键字

示例3: 使用 `DESCRIBE` 查看数据表 `tb_grade` 的结构

```

1 DESC tb_grade;

```

执行结果

```

1  +-----+-----+-----+-----+-----+
2  | Field | Type      | Null | Key | Default | Extra |
3  +-----+-----+-----+-----+-----+
4  | id    | int(11)   | YES  |     | NULL    |      |
5  | name  | varchar(20) | YES  |     | NULL    |      |
6  | grade | float     | YES  |     | NULL    |      |
7  +-----+-----+-----+-----+-----+
8  3 rows in set (0.02 sec)

```

课堂练习2

创建数据库Purchase，在该库中创建数据表Product，表结构和属性设置如下表所示：

字段	数据类型	说明
Product_ID	char(10)	商品编号
Product_Name	varchar(100)	商品名称
Product_Code	varchar(10)	商品编码
Price	float	商品价格
Place	char(10)	商品产地
Unit	varchar(20)	单位
Detail	varchar(20)	规格
SubSort_ID	varchar(10)	子类别编号
Sort_ID	varchar(10)	类别编号
Description	varchar(255)	商品说明

说明相当于编程中的注释，可以通过 `comment` 定义在表或者属性定义的末尾，例如：

```

1  CREATE TABLE test (t_id int(8) COMMENT '测验编号',
2                        t_name varchar(50) COMMENT '测验名称') COMMENT '测试';
3  SHOW CREATE TABLE test;

1  *****[ 1. row ]*****
2  Table          | test
3  Create Table | CREATE TABLE `test` (
4    `t_id` int(8) DEFAULT NULL COMMENT '测验编号',
5    `t_name` varchar(50) DEFAULT NULL COMMENT '测验名称'
6  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci COMMENT='测验'

```

二、数据表结构的修改

如果创建表之后发现有些字段、约束或者索引定义有误，则可以通过数据定义语言修改表结构。

1. 修改表名

基本命令：

```
1 ALTER TABLE 表名 RENAME 新表名；
```

首先创建 `tb_grade` 表：

```
1 CREATE TABLE tb_grade (id int(11), name varchar(20), grade float);
```

示例4：将数据表 `tb_grade` 名修改为 `grade`

```
1 ALTER TABLE tb_grade RENAME TO grade;
2 DESC grade;
```

2. 修改字段名称

基本命令：

```
1 ALTER TABLE 表名 CHANGE [COLUMN] 原字段名 数据类型 [约束条件][FIRST|AFTER 已存在字段名];
```

示例5：将数据表 `grade` 中的 `name` 字段改为 `username`

```
1 ALTER TABLE grade CHANGE `name` `username` VARCHAR(20);
2 DESC grade;
```

注意：更改后的属性的数据类型不能为空

3. 修改字段属性

基本命令：

```
1 ALTER TABLE 表名 MODIFY [COLUMN] 字段名 数据类型 [约束条件][FIRST|AFTER 已存在字段名];
```

示例6：将数据表 `grade` 中 `id` 字段的数据类型由 `INT(11)` 修改为 `INT(20)`

```
1 ALTER TABLE `grade` MODIFY `id` INT(20);
2 DESC `grade`;
3
4 -- 或者
5 ALTER TABLE `grade` CHANGE COLUMN `id` `id` INT(30);
6 DESC `grade`;
```

4. 添加字段

基本命令：

```
1 ALTER TABLE 表名 ADD [COLUMN] 新字段名 数据类型 [约束条件][FIRST|AFTER 已存在字段名];
```

注意：

- **FIRST** 是可选参数，用于将新添加的字段设置为表的第一个字段
- **AFTER** 是可选参数，用于将新添加的字段添加到指定的“已存在字段名”的后面。

示例7：将数据表 **grade** 中的添加数据类型为 **DATE** 的字段 **birthday**

```
1 ALTER TABLE `grade` ADD `birthday` DATE AFTER `username`;  
2 DESC `grade`;
```

5. 删除字段

基本命令：

```
1 ALTER TABLE 表名 DROP [COLUMN] 字段名;
```

示例8：将数据表 **grade** 中 **birthday** 字段删除

```
1 ALTER TABLE `grade` DROP `birthday`;  
2 DESC `grade`;
```

6. 复制表

基本命令：

```
1 CREATE TABLE 表1 SELECT * FROM 表2;
```

示例9：将数据表 **grade** 复制为 **grade_bak** 表

```
1 CREATE TABLE grade_bac select * from grade;  
2 DESC grade_bac;
```

7. 字段排序

基本命令：

```
1 ALTER TABLE 表名 MODIFY 字段名1 数据类型 [约束条件][FIRST|AFTER 已存在字段名];
```

示例10：将数据表 **grade_bak** 中 **username** 字段修改至第一个字段

```
1 ALTER TABLE grade_bac MODIFY username VARCHAR(20) FIRST;  
2 DESC grade_bac;
```

示例11：将数据表 **grade_bak** 中 **id** 字段插入到 **grade** 字段后面


```
1 ALTER TABLE grade_bac MODIFY id INT(20) AFTER grade;
2 DESC grade_bac;
```

8. 删除表

基本命令：

```
1 DROP TABLE 表名;
```

示例12：删除数据表 `grade_bak`

```
1 DROP TABLE grade_bac;
2 DESC grade;
```

课堂练习3

- 将数据表 `Product` 名修改为 `tb_product`。
- 修改数据表中字段 `Place` 名为 `Product_Place`，数据类型为 `varchar(50)`。
- 增加 `Product_Date` 字段，数据类型为 `Date`。
- 删除 `Description` 字段。
- 将 `Product_Place` 和 `Product_Date` 字段位置移动到 `Price` 字段之前。