

MySQL数据库基础

本节目标

- 数据库的操作：创建数据库、删除数据库
- 常用数据类型
- 表的操作：创建表、删除表

1. 数据库的操作

1.1 显示当前的数据库

```
SHOW DATABASES;
```

1.2 创建数据库

语法：

```
CREATE DATABASE [IF NOT EXISTS] db_name [create_specification [,  
create_specification] ...]
```

create_specification:

```
[DEFAULT] CHARACTER SET charset_name  
[DEFAULT] COLLATE collation_name
```

说明：

- 大写的表示关键字
- [] 是可选项
- CHARACTER SET: 指定数据库采用的字符集
- COLLATE: 指定数据库字符集的校验规则

示例：

- 创建名为 db_test1 的数据库

```
CREATE DATABASE db_test1;
```

说明：当我们创建数据库没有指定字符集和校验规则时，系统使用默认字符集：utf8，校验规则是：utf8_general_ci

- 如果系统没有 db_test2 的数据库，则创建一个名叫 db_test2 的数据库，如果有则不创建

```
CREATE DATABASE IF NOT EXISTS db_test2;
```

- 如果系统没有 db_test 的数据库，则创建一个使用utf8mb4字符集的 db_test 数据库，如果有则不创建

```
CREATE DATABASE IF NOT EXISTS db_test CHARACTER SET utf8mb4;
```

说明：MySQL的utf8编码不是真正的utf8，没有包含某些复杂的中文字符。MySQL真正的utf8是使用utf8mb4，建议大家使用utf8mb4

1.3 使用数据库

```
use 数据库名;
```

1.4 删除数据库

语法：

```
DROP DATABASE [IF EXISTS] db_name;
```

说明：

- 数据库删除以后，内部看不到对应的数据库，里边的表和数据全部被删除

```
drop database if exists db_test1;
drop database if exists db_test2;
```

2. 常用数据类型

2.1 数值类型：

分为整型和浮点型：

数据类型	大小	说明	对应java类型
BIT(M)	M指定位数，默认为1	二进制数，M范围从1到64，存储数值范围从0到2^M-1	常用Boolean对应BIT，此时默认是1位，即只能存0和1
TINYINT	1字节		Byte
SMALLINT	2字节		Short
INT	4字节		Integer
BIGINT	8字节		Long
FLOAT(M, D)	4字节	单精度，M指定长度，D指定小数位数。会发生精度丢失	Float
DOUBLE(M, D)	8字节		Double
DECIMAL(M, D)	M/D最大值+2	双精度，M指定长度，D表示小数点位数。精确数值	BigDecimal
NUMERIC(M, D)	M/D最大值+2	和DECIMAL一样	BigDecimal

扩展资料

数值类型可以指定为无符号（unsigned），表示不取负数。

1字节 (bytes) = 8bit。

对于整型类型的范围：

- 1. 有符号范围：-2[^] (类型字节数*8-1) 到2[^] (类型字节数*8-1) -1，如int是4字节，就是-2[^]31到2[^]31-1
- 2. 无符号范围：0到2[^] (类型字节数*8) -1，如int就是2[^]32-1

尽量不使用unsigned，对于int类型可能存放不下的数据，int unsigned同样可能存放不下，与其如此，还不如设计时，将int类型提升为bigint类型。

2.2 字符串类型

数据类型	大小	说明	对应java类型
VARCHAR (SIZE)	0-65,535字节	可变长度字符串	String
TEXT	0-65,535字节	长文本数据	String
MEDIUMTEXT	0-16 777 215字节	中等长度文本数据	String
BLOB	0-65,535字节	二进制形式的长文本数据	byte[]

2.3 日期类型

数据类型	大小	说明	对应java类型
DATETIME	8字节	范围从1000到9999年，不会进行时区的检索及转换。	java.util.Date、 java.sql.Timestamp
TIMESTAMP	4字节	范围从1970到2038年，自动检索当前时区并进行转换。	java.util.Date、 java.sql.Timestamp

3. 表的操作

需要操作数据库中的表时，需要先使用该数据库：

```
use db_test;
```

3.1 查看表结构

```
desc 表名;
```

示例:

mysql> desc users;

索引类型						
Field	Type	Null	Key	Default	Extra	
id	int(11)	YES		NULL		
name	varchar(20)	YES		NULL		
password	char(32)	YES		NULL		
birthday	date	YES		NULL		

↑ ↑ ↑ ↑ ↑

字段名字 字段类型 是否允许为空 默认值 扩充

3.2 创建表

语法:

```
CREATE TABLE table_name (  
    field1 datatype,  
    field2 datatype,  
    field3 datatype  
);
```

可以使用comment增加字段说明。

示例:

```
create table stu_test (  
    id int,  
    name varchar(20) comment '姓名',  
    password varchar(50) comment '密码',  
    age int,  
    sex varchar(1),  
    birthday timestamp,  
    amout decimal(13,2),  
    resume text  
);
```

3.4 删除表

语法格式:

```
DROP [TEMPORARY] TABLE [IF EXISTS] tbl_name [, tbl_name] ...
```

示例:

```
-- 删除 stu_test 表  
drop table stu_test;  
-- 如果存在 stu_test 表, 则删除 stu_test 表  
drop table if exists stu_test;
```

6. 内容重点总结

- 操作数据库：

```
-- 显示
show databases;
-- 创建
create database xxx;
-- 使用
use xxx;
-- 删除
drop database xxx;
```

- 常用数据类型：

INT：整型

DECIMAL(M, D)：浮点数类型

VARCHAR(SIZE)：字符串类型

TIMESTAMP：日期类型

- 操作表：

```
-- 查看
show 表;

-- 创建
create table 表名(
    字段1 类型1,
    字段2 类型2,
    ...
);

-- 删除
drop table 表名;
```

7. 课后作业

- 有一个商店的数据，记录客户及购物情况，有以下三个表组成：
 - 商品goods(商品编号goods_id, 商品名goods_name, 单价unitprice, 商品类别category, 供应商provider)
 - 客户customer(客户号customer_id, 姓名name, 住址address, 邮箱email, 性别sex, 身份证card_id)
 - 购买purchase(购买订单号order_id, 客户号customer_id, 商品号goods_id, 购买数量nums)
- SQL:

```
-- 创建数据库
create database if not exists bit32mall
default character set utf8 ;

-- 选择数据库
use bit32mall;
```

```
-- 创建数据库表
-- 商品
create table if not exists goods
(
    goods_id int comment '商品编号',
    goods_name varchar(32) comment '商品名称',
    unitprice int comment '单价, 单位分',
    category varchar(12) comment '商品分类',
    provider varchar(64) comment '供应商名称'
);

-- 客户
create table if not exists customer
(
    customer_id int comment '客户编号',
    name varchar(32) comment '客户姓名',
    address varchar(256) comment '客户地址',
    email varchar(64) comment '电子邮箱',
    sex bit comment '性别',
    card_id varchar(18) comment '身份证'
);

-- 购买
create table if not exists purchase
(
    order_id int comment '订单号',
    customer_id int comment '客户编号',
    goods_id int comment '商品编号',
    nums int comment '购买数量'
);
```