项目5 MySQL的常用数据类型和函数

学习目标

(1) 熟悉 MySQL 支持的数据类型。

(2) 熟练区分不同的数据类型的使用规范。

(3) 了解 MySQL 的常用函数。

(4) 熟悉利用 MySQL 的常用函数解决生活中的实际问题。

匠人匠心

(1) 了解数据类型的基本概念,理解数据类型就是对数据的行为规范,引导学生懂得“没有规矩不成方圆”。

(2) 熟悉 MySQL 支持的各种数据类型,懂得相同大小的数据选择不同的类型代表不同的精度或意义,引得学生学会不同环境的角色转变。

(3) 在评估使用哪种类型时,引导学生考虑数据的存储空间和可靠性的平衡问题,使其学会取舍。

(4) 熟悉 MySQL 的常用函数并灵活运用,引导学生做事遇到困难求教他人,借助外界力量达到事半功倍的效果。

视频讲解

任务 1　MySQL 的数据类型

【任务描述】

数据类型是指系统中所允许的数据的类型。MySQL 数据类型定义了数据列中可以存储的数据以及该数据怎样存储的规则。

【任务要求】

在 MySQL 客户端通过命令查看 MySQL 支持的数据类型,学习常用数据类型及其灵活应用。

具体操作要求如下:

(1) 查看 MySQL 数据库支持的所有的数据类型。

(2) 查看不同数据类型的取值范围及基本规则。

(3) 创建一个 student 表,包含 xh、xm 和 banji 三个字段,分别设置为 CHAR、CHAR(6)和

VARCHAR(9)类型并区分存储的不同方式。

(4) 创建一个 stu_info 表,包含 xm 和 zzmm 两个字段,其中,zzmm(政治面貌)可选值有“党员”“团员”“群众”。

(5) 创建一个 xuesheng 表,包含 xm 和 zhiwu 两个字段,其中设置 zhiwu(职务)可选值有“班长”“团支书”“纪律委员”“生活委员”“学生会干事”“组织部部长”“文体部部长”“学生会主席”。

【相关知识】

1. MySQL 数据类型

数据类型用于指定列所包含数据的规则,它决定了数据保存在列中的方式,包括分配给列的宽度,以及值是否可以是字母、数字、日期和时间等。

数据库中的每列都应该有适当的数据类型,用于限制或允许该列存储的数据。MySQL 支持的数据类型及具体类别如表 5-1 所示。

表 5-1　MySQL 支持的数据类型及具体类别

类 型 名 称	具 体 类 别
整数类型	TINYINT、SMALLINT、MEDIUMINT、INT(INTEGER)、BIGINT
浮点数类型	FLOAT、DOUBLE
定点数类型	DECIMAL
位类型	BIT
日期时间类型	YEAR、TIME、DATE、DATETIME、TIMESTAMP
字符串类型	CHAR、VARCHAR、TINYTEXT、TEXT、MEDIUMTEXT、LONGTEXT 等
枚举类型	ENUM
SET 类型	SET
二进制字符串类型	BINARY、VARBINARY、TINYBLOB、BLOB、MEDIUMBLOB、LONGBLOB
JSON 类型	JSON 对象、JSON 数组
空间数据类型	单值类型：GEOMETRY、POINT、LINESTRING、POLYGON； 集合类型：MULTIPOINT、MULTILINESTRING、MULTIPOLYGON、 GEOMETRYCOLLECTION

2. MySQL 整数类型

MySQL 支持的整数类型有 5 种,分别是微整型(TINYINT)、小整型(SMALLINT)、中等大小的整型(MEDIUMINT)、普通整型(INT 或 INTEGER)和大整型(BIGINT)。

不同类型的整数存储所需的字节数不同。MySQL 整数类型所占存储空间大小及取值范围如表 5-2 所示,占用的字节越多的类型所能表示的数值范围越大。

表 5-2　MySQL 整数类型所占存储空间大小及取值范围

整数类型名称	存储空间	取值范围(有符号)	取值范围(无符号)
微整型(TINYINT)	1 字节	−128～127	0 ～255
小整型(SMALLINT)	2 字节	−32 768～32 767	0～65 535
中等大小的整型 (MEDIUMINT)	3 字节	−8 388 608～8 388 607	0～16 777 215
普通整型(INT|INTEGER)	4 字节	−2 147 483 648～ 2 147 483 647	0～4 294 967 295

续表

整数类型名称	存储空间	取值范围(有符号)	取值范围(无符号)
大整型(BIGINT)	8 字节	−9 223 372 036 854 775 808～9 223 372 036 854 775 807	0～18 446 744 073 709 551 615

微课课堂

MySQL 整数类型：

(1) INT 和 INTEGER 是同一种数据类型。

(2) 每种数据类型的取值范围可以根据所占字节数计算得出。

3. MySQL 浮点数类型

MySQL 需要在数据库中存储小数的类型时，可以借助浮点数类型。浮点数类型有两种，分别是单精度浮点数(FLOAT)和双精度浮点数(DOUBLE)。MySQL 浮点数类型所占存储空间大小及取值范围如表 5-3 所示。

表 5-3 MySQL 浮点数类型所占存储空间大小及取值范围

浮点数类型名称	存储空间	取值范围(有符号)	取值范围(无符号)
单精度浮点数(FLOAT)	4 字节	(−3.402 823 466E＋38 ，−1.175 494 351E−38)，0，(1.175 494 351E−38 ，3.402 823 466E＋38)	0，(1.175 494 351E−38～3.402 823 466E＋38)
双精度浮点数(DOUBLE)	8 字节	(−1.7 976 931 348 623 157E＋308 ，−2.2 250 738 585 072 014E−308)，0，(2.2 250 738 585 072 014E−308 ，1.7 976 931 348 623 157E＋308)	0 ，(2.2 250 738 585 072 014E−308～1.7 976 931 348 623 157E＋308)

微课课堂

查看数据类型的取值范围：

浮点数类型的取值范围很大，不需要手动计算录入，可以在 MySQL 客户端借助命令查看。例如，查看双精度类型的取值范围，则输入 HELP DOUBLE。

4. MySQL 定点数类型

MySQL 定点数类型(DECIMAL)是精确值存储。定点数类型在数据库中是以字符串形式存储的，该数据类型用于精度要求非常高的计算中。MySQL 定点数类型所占存储空间大小及取值范围如表 5-4 所示。

表 5-4 MySQL 定点数类型所占存储空间大小及取值范围

类 型 名 称	存储空间	取值范围(有符号)
定点数类型(DECIMAL(M，D))	M＋2	DECIMAL 的存储空间并不是固定的，有效的数据范围是由 M 和 D 决定的。其中，M 表示数据的精度，D 表示小数位数，并且该数据共占用的存储空间为 M＋2 字节

微课课堂

DECIMAL 类型：

(1) 当 DECIMAL 类型不指定 M 和 D 时，其默认为 DECIMAL(10，0)。当数据的精度超出定点数类型的精度范围时，MySQL 同样会进行四舍五入处理。

（2）M 的取值范围：1～65，取 0 时会被设置为默认值 10，超出范围会报错。

（3）D 的取值范围：1～30，同时必须满足 $D \leqslant M$，否则会报错。

5. 位类型

位类型(BIT)中存储的是二进制值。BIT 类型使用 b'value'的形式存储二进制数据，其中 value 指的是一个由 0 和 1 组成的二进制数据。如果 value 值的位数小于指定的 M，则会在 value 值的左侧补 0。MySQL 位类型所占存储空间大小及取值范围如表 5-5 所示。

表 5-5　MySQL 位类型所占存储空间大小及取值范围

类 型 名 称	存 储 空 间	取 值 范 围
位类型(BIT(M))	1～8 字节	BIT(1)～BIT(8)

6. 日期时间类型

为了方便在数据库中处理日期时间类型的数据，MySQL 提供了 5 种不同的日期和时间数据类型。MySQL 日期时间类型所占存储空间大小及取值范围如表 5-6 所示。

表 5-6　MySQL 日期时间类型所占存储空间大小及取值范围

日期时间类型名称	存储空间	日 期 格式	最小值	最大值
年(YEAR)	1 字节	YYYY/YY	1000	9999
时间(TIME)	3 字节	HH：MM：SS	－838：59：59.000000	838：59：59.000000
日期(DATE)	3 字节	YYYY-MM-DD	1000-01-01	9999-12-31
日 期 时 间 (DATETIME)	8 字节	YYYY-MM-DD HH：MM：SS	1000-01-01 00：00：00.000000	9999-12-31 23：59：59.999999
时 区 时 间 (TIMESTAMP)	4 字节	YYYY-MM-DD HH：MM：SS	1970-01-01 00：00：01.000000	2038-01-19 03：14：07.999999

微课课堂

日期时间类型：

（1）在 MySQL 8.0 默认的情况下，日期时间类型的数据处于严格模式，即 STRICT_TRANS_TABLES。

（2）如果在 my.ini 配置文件中删除 STRICT_TRANS_TABLES，则日期时间类型的数据更改为非严格模式。

（3）严格模式下，如果插入的数据不合法或超出范围均会提示错误，则插入的数据不会被系统接收。

（4）非严格模式下，如果插入的数据不合法或合法但超出范围，只会给出警告，但会插入成功，插入的数据为边界值。

7. 字符串类型

字符串类型是在数据库中存储字符串的数据类型。使用不同的字符串类型可以实现从一个简单的字符到超大的文本块或二进制字符串数据的存储。MySQL 字符串类型所占存储空间大小及取值范围如表 5-7 所示。

表 5-7　MySQL 字符串类型所占存储空间大小及取值范围

字符串类型名称	存 储 空 间	描　　述
CHAR(M)	0～255 字节	允许长度为 0～M 字节的定长字符串

续表

字符串类型名称	存 储 空 间	描　　述
VARCHAR(*M*)	0～65 535 字节	允许长度为 0～*M* 字节的变长字符串
BINARY(*M*)	0～255 字节	允许长度为 0～*M* 字节的定长二进制字符串
VARBINARY(*M*)	0～65 535 字节	允许长度为 0～*M* 字节的变长二进制字符串
TINYBLOB	0～255 字节	二进制形式的短文本数据
TINYTEXT	0～255 字节	短文本数据
BLOB	0～65 535 字节	二进制形式的长文本数据
TEXT	0～65 535 字节	长文本数据
MEDIUMBLOB	0～16 777 215 字节	二进制形式的中等长度文本数据
MEDIUMTEXT	0～16 777 215 字节	中等长度文本数据
LONGBLOB	0～4 294 967 295 字节	二进制形式的极大文本数据
LONGTEXT	0～4 294 967 295 字节	极大文本数据

8. 枚举类型

枚举类型(ENUM)是一个字符串对象，其值通常选自一个允许值列表，该列表在创建表时会被明确地设定。

枚举类型在使用时，其具体的语法格式如下：

```
ENUM('value1','value2','value3','value4','value5',…)
```

每一个字符串成员都会对应一个索引值，索引值依次为 1,2,3,4,5,…存储在数据库中的就是字符串成员所对应的索引值，而不是字符串本身。

9. SET 类型

SET 类型是一个字符串对象，和 ENUM 类型类似但又不完全相同。SET 类型可以从允许值列表中选择一个元素或多个元素的组合。当取多个元素时，不同元素之间用逗号隔开，但成员个数的上限为 64。

SET 类型在使用时，其具体的语法格式如下：

```
SET('value1','value2','value3','value4','value5'…)
```

有关 SET 类型的使用，具体可以参见本项目的知识拓展。

【任务实现】

【例 5-1】 查看 MySQL 数据库支持的所有的数据类型。

具体操作步骤如下：

(1) 打开 MySQL Command Line Client，在光标闪动的位置输入安装时设置的密码“123456”。

(2) 系统进入 MySQL 的命令行客户端(Command Line Client)工作界面。

(3) 在图 4-6 所示的光标闪动的位置输入命令。

```
HELP DATA TYPES;
```

实现显示 MySQL 数据库支持的所有的数据类型，执行结果如图 5-1 所示。

图 5-1　MySQL 数据库支持的所有的数据类型

【例 5-2】 查看不同数据类型的取值范围及基本规则。

具体操作步骤如下：

在如图 4-6 所示的界面中，输入命令 help＋数据类型即可查看不同数据类型的取值范围及基本规则。

（1）输入命令。

```
HELP INT;
```

实现查看普通整型的有符号和无符号数据的取值范围，执行结果如图 5-2 所示。

图 5-2　查看普通整型的有符号和无符号数据的取值范围

（2）输入命令。

```
HELP DOUBLE;
```

实现查看双精度数据的取值范围及基本规则，执行结果如图 5-3 所示。

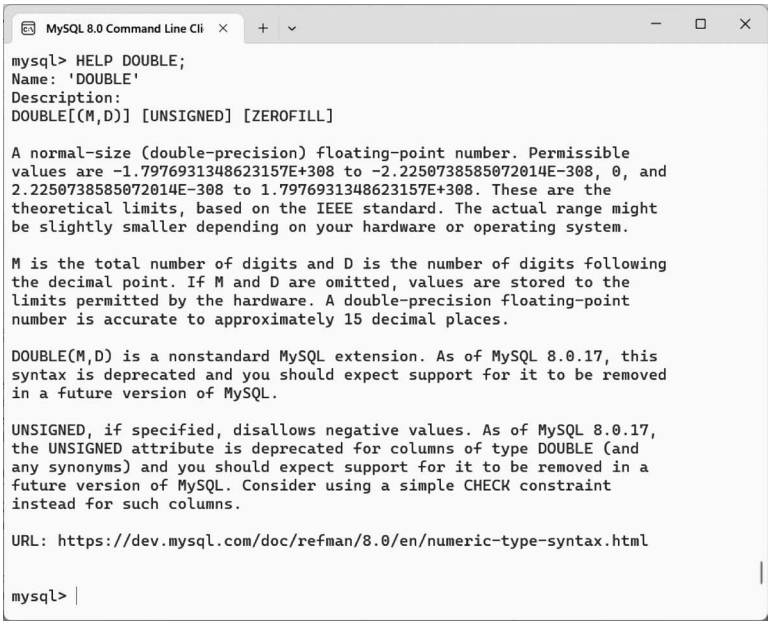

图 5-3　查看双精度数据的取值范围及基本规则

（3）输入命令。

```
HELP DATETIME;
```

实现查看日期时间类型数据的取值范围及基本规则，执行结果如图 5-4 所示。

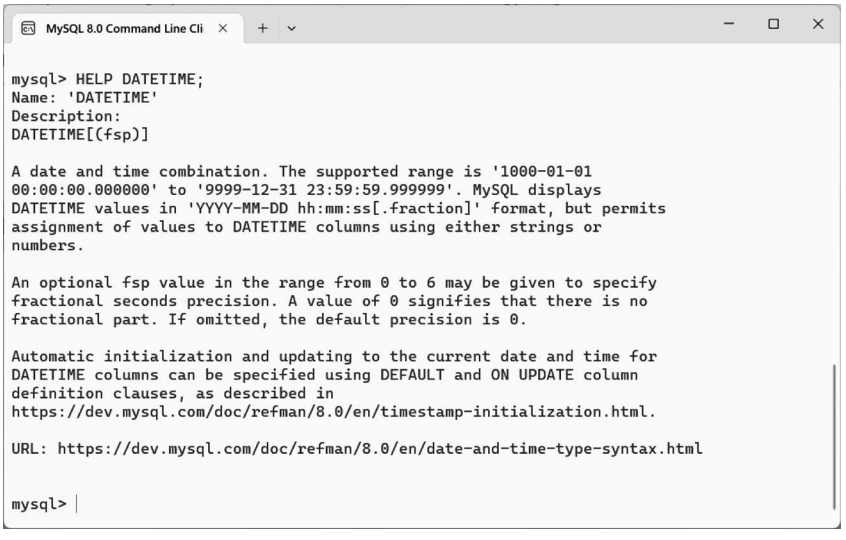

图 5-4　查看日期时间类型数据的取值范围及基本规则

【例 5-3】　创建一个 student 表，包含 xh、xm 和 banji 三个字段，分别设置为 CHAR、CHAR(6)和 VARCHAR(9)类型并区分存储的不同方式。

具体操作步骤如下：

(1) 在图 4-6 的界面中，输入命令。

```
CREATE TABLE student(
   xh CHAR,
   xm CHAR(6),
   banji VARCHAR(9)
   );
```

实现创建一个 student 表，包含 xh、xm 和 banji 三个字段，分别设置为 CHAR、CHAR(6)类型和 VARCHAR(9)类型，执行结果如图 5-5 所示。

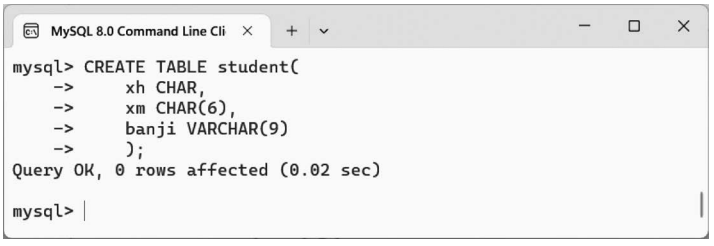

图 5-5　创建 student 表

(2) 输入命令。

```
DESC student;
```

实现查看 student 表的结构信息，执行结果如图 5-6 所示。

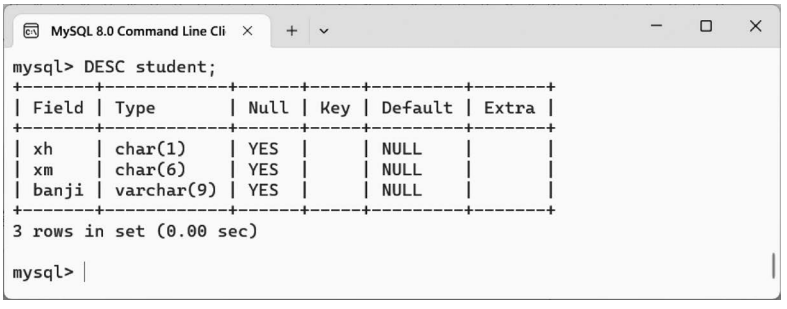

图 5-6　查看 student 表的结构信息

(3) 依次输入命令。

```
INSERT INTO student VALUES('1','张小','人智 2201 班');
INSERT INTO student VALUES('2','张小小','人工智能 2201 班');
```

实现向 student 表中插入 2 条记录，如图 5-7 所示。

(4) 输入命令。

```
SELECT CHAR_LENGTH(xh),CHAR_LENGTH(xm),CHAR_LENGTH(banji) FROM student;
```

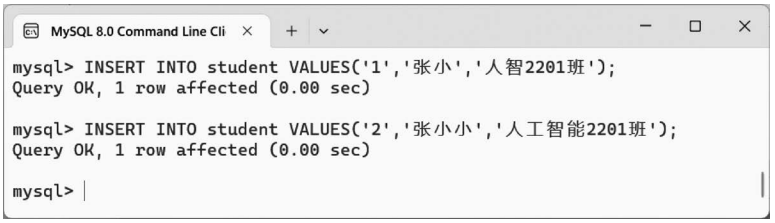

图 5-7　向 student 表添加 2 条记录

实现查看 student 表中各个字段所占长度，执行结果如图 5-8 所示。

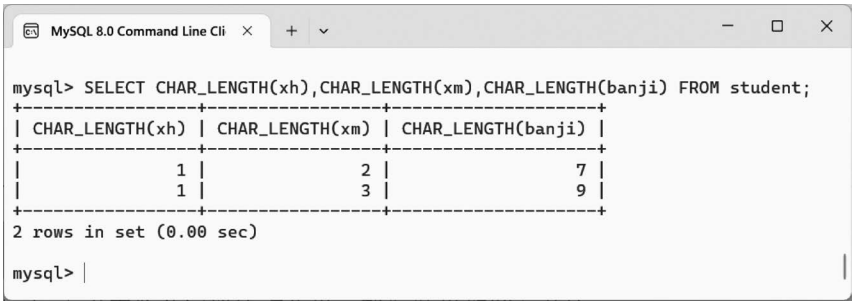

图 5-8　查看 student 表中各个字段所占长度

（5）输入命令。

```
INSERT INTO student(xm,banji) VALUES(' ',' ');
```

实现向 student 表中插入一条带空格的记录，执行结果如图 5-9 所示。

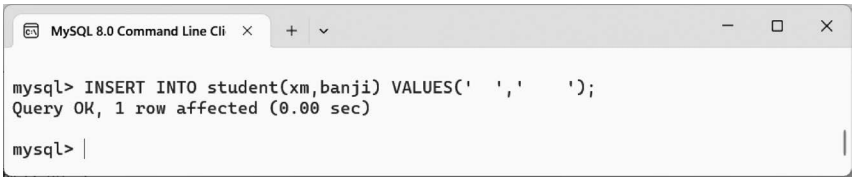

图 5-9　向 student 表中插入一条带空格的记录

（6）输入命令。

```
SELECT CHAR_LENGTH(xh),CHAR_LENGTH(xm),CHAR_LENGTH(banji) FROM student;
```

实现查看 student 表现有记录中各个字段所占长度，执行结果如图 5-10 所示。

微课课堂

CHAR 和 VARCHAR 类型说明：

（1）CHAR(*M*)类型如果不指定 *M*，则表示长度默认是 1 个字符。

（2）当 MySQL 检索 CHAR 类型的数据，CHAR 类型的字段会去除尾部的空格而检索 VARCHAR 类型的字段数据时，会保留数据尾部的空格。

【例 5-4】 创建一个 stu_info 表，包含 xm 和 zzmm 两个字段，其中，zzmm(政治面貌)可选值有“党员”“团员”“群众”。

具体操作步骤如下：

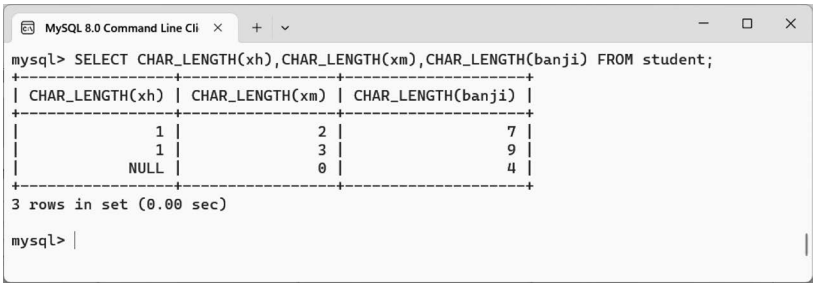

图 5-10　查看 student 表现有记录中各个字段所占长度

(1) 在图 4-6 所示的窗口中,输入命令。

```
CREATE TABLE stu_info(
    xm CHAR(6),
    zzmm ENUM('党员','团员','群众')
    );
```

实现创建 stu_info 表,包含 xm 和 zzmm 两个字段,其中 zzmm(政治面貌)设定为枚举类型,其可选值有“党员”“团员”“群众”,执行结果如图 5-11 所示。

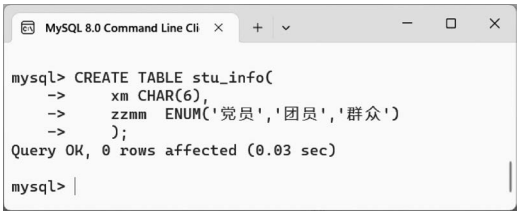

图 5-11　创建 stu_info 表

(2) 输入命令。

```
DESC stu_info;
```

实现查看 stu_info 表的结构信息,执行结果如图 5-12 所示。

图 5-12　查看 stu_info 表的结构信息

(3) 依次输入命令。

```
INSERT INTO stu_info VALUES('王明明',1);
INSERT INTO stu_info VALUES('张小小',2);
INSERT INTO stu_info VALUES('赵媛媛',3);
```

实现向 stu_info 表插入以枚举类型的索引值形式的 3 条记录，执行结果如图 5-13 所示。

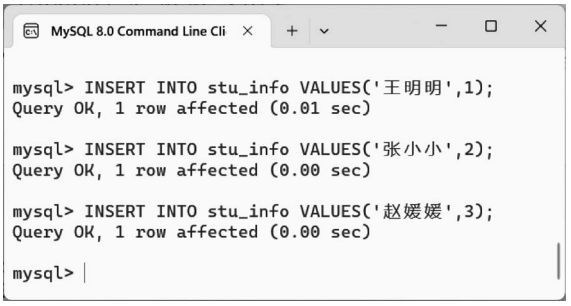

图 5-13　向 stu_info 表插入以枚举类型的索引值形式的 3 条记录

（4）依次输入命令。

```
INSERT INTO stu_info VALUES('李天','党员');
INSERT INTO stu_info VALUES('陈新','群众');
```

实现向 stu_info 表插入以枚举类型具体枚举值的 2 条记录，执行结果如图 5-14 所示。

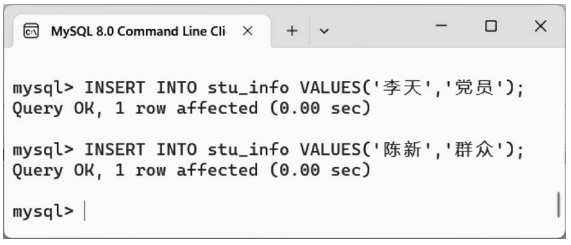

图 5-14　向 stu_info 表插入以枚举类型具体枚举值的 2 条记录

（5）输入命令。

```
SELECT * FROM stu_info;
```

实现查询 stu_info 表的记录信息，执行结果如图 5-15 所示。

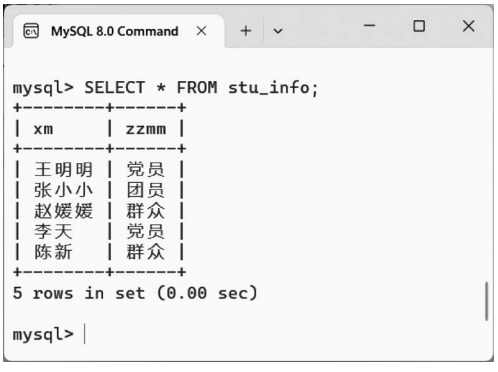

图 5-15　查询 stu_info 表的记录信息

【例 5-5】 创建一个 xuesheng 表，包含 xm 和 zhiwu 两个字段，其中设置 zhiwu(职务)可选值有“班长”“团支书”“纪律委员”“生活委员”“学生会干事”“组织部部长”“文体部部长”

“学生会主席”。

具体操作步骤如下：

（1）输入命令。

```
CREATE TABLE xuesheng(xh char(3),
    xm CHAR(6),
    zhiwu SET('班长','团支书','纪律委员','生活委员','学生会干事','组织部部长','文体部部长',
'学生会主席')
    );
```

实现创建 xuesheng 表，包含 xm 和 zhiwu 两个字段，其中设定 zhiwu(职务)为 SET 类型，其可选值有“班长”“团支书”“纪律委员”“生活委员”“学生会干事”“组织部部长”“文体部部长”“学生会主席”，执行结果如图 5-16 所示。

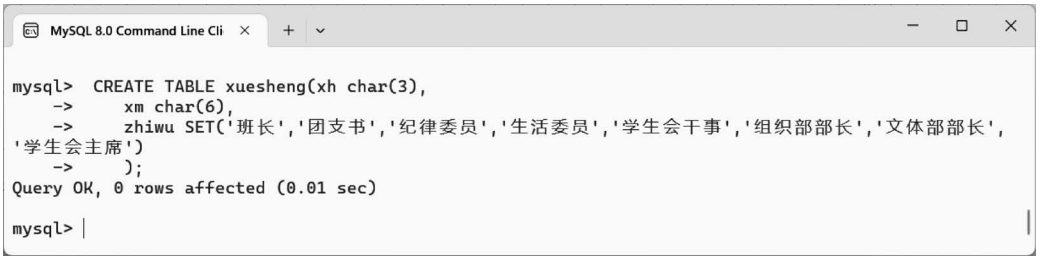

图 5-16　创建 xuesheng 表

（2）输入命令。

```
DESC xuesheng;
```

实现查看 xuesheng 表的结构信息，执行结果如图 5-17 所示。

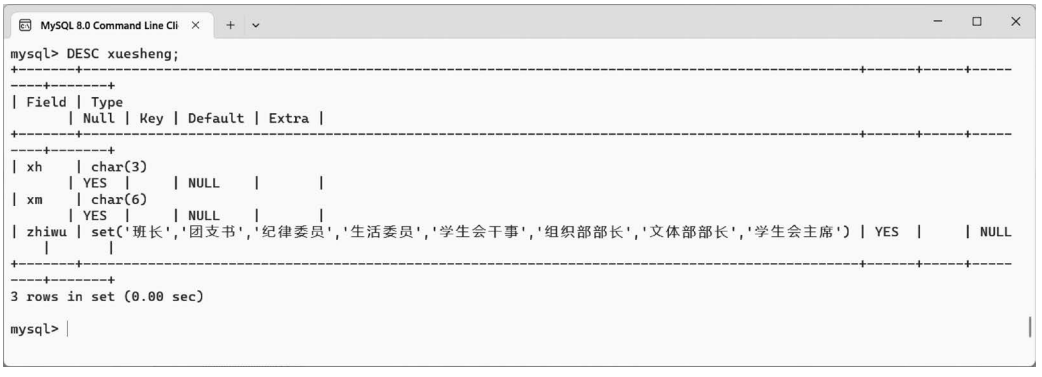

图 5-17　查看 xuesheng 表的结构信息

（3）依次输入命令。

```
INSERT INTO xuesheng VALUES('001','徐乐','团支书');
INSERT INTO xuesheng VALUES('002','王想','班长,学生会干事');
INSERT INTO xuesheng VALUES('003','张礼峰','班长,团支书,学生会主席');
INSERT INTO xuesheng VALUES('004','陈悦','生活委员,学生会干事,班长');
```

实现向 xuesheng 表中插入 4 条记录，实现不同学生可以有多种不同职务，执行结果如图 5-18 所示。

图 5-18　向 xuesheng 表中插入 4 条不同职务的记录

（4）依次输入命令。

```
INSERT INTO xuesheng VALUES('005','钱亦','4');
INSERT INTO xuesheng VALUES('006','张芳芳','5');
INSERT INTO xuesheng VALUES('007','韩冬','7');
```

实现向 xuesheng 表中插入 3 条记录，实现利用索引值录入多条不同职务的学生记录，执行结果如图 5-19 所示。

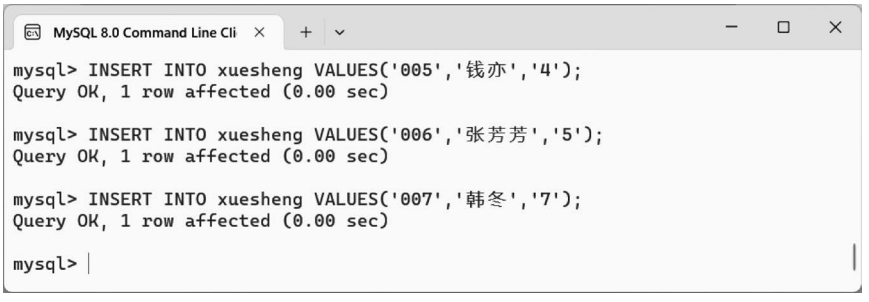

图 5-19　向 xuesheng 表中利用索引值实现录入多条不同职务的学生记录

（5）输入命令。

```
SELECT * FROM xuesheng;
```

实现查询 xuesheng 表的记录信息，执行结果如图 5-20 所示。

微课课堂

SET 类型：

（1）值中的每个元素都只会出现一次。

（2）值忽略大小写，在存储时将它们都转换为创建表时定义的大小写。

（3）与插入时元素的顺序无关，会按照表创建时指定的顺序列出。

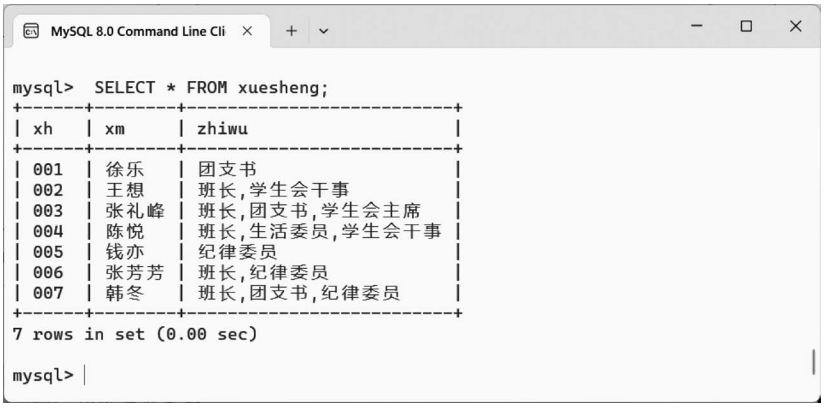

图 5-20　查询 xuesheng 表的记录信息

任务 2　常用函数


视频讲解

【任务描述】

MySQL 包含了大量并且丰富的函数，具体包含有聚合函数、数值型函数、字符串型函数、日期时间函数和流程控制函数等。

【任务要求】

在 MySQL 客户端通过命令创建 score 表，包含 xm、xb、csrq、math、mysql、english、chinese 和 jtdz 字段，输入若干记录后，进行相应操作。

具体操作要求如下：

(1) 对 score 表统计 math 科目的总和、mysql 科目的最高分、english 科目的平均分和 chinese 科目的最低分。

(2) 对 score 表插入带 NULL、* 值的记录，并统计男、女生人数。

(3) 先对 score 表统计男、女生人数及具体人名信息，然后对姓名按照拼音字母排列，从大到小的顺序输出，再对 xm 字段实现去重管理。

(4) 显示当前系统的日期和时间，并以各种不同的格式显示输出。

(5) 对 score 表实现将 xm、xb、jtdz 字段连接成新的字符串输出，再对其按照空 2 格的方式输出并求其对应字符串的长度，最后对 jtdz 字段截取对应的省份。

(6) 先对 score 表增加 age 和 sum_score 字段，然后根据 csrq 字段求每人的 age 值，根据每人的各科成绩求其总和 sum_score，再对 sum_score 进行判断，350 分以上显示优秀，其余为合格。

【相关知识】

1. 聚合函数

MySQL 聚合函数可以实现根据一组数据求出一个值，聚合函数的结果值只根据选定数据行中非 NULL 值进行计算，NULL 值被忽略。MySQL 聚合函数及作用如表 5-8 所示。

表 5-8 MySQL 聚合函数及作用

函数名称	作用
AVG	计算表中某个字段取值的平均值
COUNT	对于除 * 以外的任何参数，返回所选择集合中非 NULL 值的行的数目 对于参数 * ，则返回所选择集合中所有行的数目，包含 NULL 值的行
GROUP_CONCAT	返回由属于一组的列值连接组合而成的结果
MAX	求出表中某个字段值的最大值
MIN	求出表中某个字段值的最小值
SUM	计算表中某个字段取值的总和

2. 数值型函数

数值型函数主要用于处理包括整型、浮点数等数字类型的数据。MySQL 常用的数值型函数及作用如表 5-9 所示。

表 5-9 MySQL 常用的数值型函数及作用

函数名称	作用
ABS	绝对值
ACOS	反余弦值，和函数 COS 互为反函数
ASIN	反正弦值，和函数 SIN 互为反函数
ATAN	反正切值，和函数 TAN 互为反函数
CEIL|CEILING	两个函数功能相同，都是返回不小于参数的最小整数，即向上取整
COS	余弦值
COT	余切值
FLOOR	向下取整，返回值转换为一个 BIGINT
POW|POWER	两个函数的功能相同，都是所传参数的次方的结果值
MOD	余数
RAND	生成一个 0～1 的随机数，传入的参数是整数值，则会产生重复序列
ROUND	对所传参数进行四舍五入
SIGN	返回参数的符号
SIN	正弦值
SQRT	二次方根
TAN	正切值

3. 日期时间函数

日期时间函数主要用于对日期和时间数据进行处理。MySQL 常用函数及作用如表 5-10 所示。

表 5-10 MySQL 常用函数及作用

函数名称	作用
ADDTIME	时间加法运算，在原始时间上添加指定的时间
CURDATE|CURRENT_DATE	两个函数作用相同，返回当前系统的日期值
CURTIME|CURRENT_TIME	两个函数作用相同，返回当前系统的时间值
FROM_UNIXTIME	将 UNIX 时间戳转换为时间格式，和 UNIX_TIMESTAMP 互为反函数
DATEDIFF	获取两个日期之间间隔，返回参数 1 减去参数 2 的值
DATE_ADD|ADDDATE	两个函数功能相同，都是向日期添加指定的时间间隔
DATE_SUB|SUBDATE	两个函数功能相同，都是向日期减去指定的时间间隔

续表

函 数 名 称	作　　用
DAYOFYEAR	获取指定日期是一年中的第几天，返回值范围是 1～366
DATE_FORMAT	格式化指定的日期，根据参数返回指定格式的值
DAYOFWEEK	获取指定日期对应的一周的索引位置值
MONTH	获取指定日期中的月份
MONTHNAME	获取指定日期中的月份英文名称
NOW|SYSDATE	两个函数作用相同，返回当前系统的日期和时间值
SEC_TO_TIME	将秒数转换为时间，和 TIME_TO_SEC 互为反函数
SUBTIME	时间减法运算，在原始时间上减去指定的时间
TIME_TO_SEC	将时间参数转换为秒数
UNIX_TIMESTAMP	获取 UNIX 时间戳函数，返回一个以 UNIX 时间戳为基础的无符号整数
WEEK	获取指定日期是一年中的第几周，返回值的范围为 0～52 或 1～53
WEEKDAY	获取指定日期在一周内的对应的工作日索引
YEAR	获取年份，返回值范围是 1970～2069

日期时间函数在使用时，除了需要借助日期时间函数外，还需要对输出的格式进行区分。MySQL 日期时间函数输出格式及作用如表 5-11 所示。

表 5-11　MySQL 日期时间函数输出格式及作用

格　　式	作　　用
%c	月份(1,2,3,…,12)
%d	日(01,02,03,…,31)
%e	日(1,2,3,…,31)
%h	小时(十二进制)
%j	一年中的天数(001,002,003,…,366)
%i	分钟(00,01,02,…,59)
%m	月份(01,02,03,…,12)
%Y	年，4 位
%y	年，2 位
%s	秒(00,01,02,…,59)
%T	时间，24 小时(hh：mm：ss)
%u	一年中的周数(1,2,3,…,53)
%w	一个星期中的天数(0＝sunday,…,6＝saturday)

4. 字符串类型函数

字符串函数主要用来处理数据表中的字符类型的数据，实现拼接、去空格等。MySQL 常用字符串类型函数及作用如表 5-12 所示。

表 5-12　MySQL 常用字符串类型函数及作用

函 数 名 称	作　　用
CONCAT(s1,s2,…,sn)	合并 s1,s2,…,sn 字符串函数，返回结果为连接参数产生的字符串，参数可以是一个或多个
concat_ws(sep,s1,s2,…,sn)	将 s1,s2,…,sn 连接成字符串，并用 sep 字符间隔
LEFT(str,x)	返回字符串 str 中最左边的 x 个字符
LENGTH(str)	计算字符串长度函数，返回字符串的字节数

续表

函 数 名 称	作　　用
LOWER(str)	将字符串中的字母转换为小写字母
REPLACE(str1,str2)	字符串替换函数,返回替换后的新字符串
REVERSE(str)	字符串反转(逆序)函数,返回和原始字符串顺序相反的字符串
RIGHT(str,x)	返回字符串 str 中最右边的 x 个字符
SUBSTRING(str,start,len)	截取字符串,返回从指定位置开始的指定长度的字符串
strcmp(s1,s2)	比较字符串 s1 和 s2 是否相同,若相同则返回 0,否则返回－1
TRIM(str)	删除字符串左右两侧的空格
UPPER(str)	将字符串中的字母转换为大写

5. 流程控制函数

MySQL 流程控制函数及作用如表 5-13 所示。

表 5-13　MySQL 流程控制函数及作用

函 数 名 称	作　　用
IF(test,t,f)	如果 test 是真,则返回 t,否则返回 f
IFNULL(arg1,arg2)	如果 arg1 不是空,则返回 arg1,否则返回 arg2
NULLIF(arg1,arg2)	如果 arg1＝arg2 则返回 null,否则返回 arg1
(1) 搜索 CASE WHEN 格式。 CASE WHEN <求值表达式 1> THEN <表达式 1> WHEN <求值表达式 2> THEN <表达式 2> ELSE <表达式> END (2) 简单 CASE 表达式格式。 CASE <表达式> WHEN <表达式 1> THEN <表达式 2> WHEN <表达式 3> THEN <表达式 4> ELSE <表达式 5> END	两种格式使用功能相同,根据求值表达式找到对应的入口,然后执行 THEN 后面的表达式

【任务实现】

【例 5-6】 在 MySQL 客户端通过命令创建 score 表,包含 xm、xb、csrq、math、mysql、english、chinese 和 jtdz 字段。输入记录后,对 score 表统计 math 科目的总和、mysql 科目的最高分、english 科目的平均分和 chinese 科目的最低分。

具体操作步骤如下:

(1) 输入命令。

```
 CREATE TABLE score(xm CHAR(6),
xb CHAR(2),csrq DATE,
math FLOAT,mysql FLOAT,
english FLOAT,chinese FLOAT,
jtdz VARCHAR(30)
);
```

实现创建 score 表，包含 xm、xb、csrq、math、mysql、english、chinese 和 jtdz 字段，执行结果如图 5-21 所示。

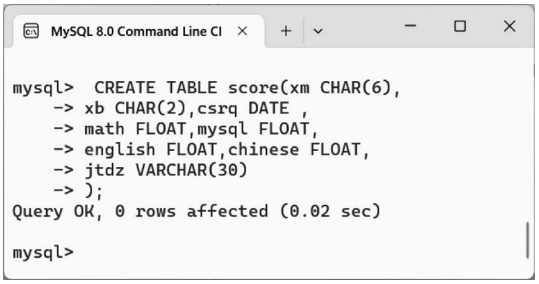

图 5-21　创建 score 表

（2）依次输入命令。

```
INSERT INTO score VALUES('王明明','男','2002-1-5',60,77,83,90,'重庆市九龙坡区');
INSERT INTO score VALUES('张小小','女','2002-11-6',84,87,73,93,'四川省成都市');
INSERT INTO score VALUES('赵媛媛','女','2001-12-5',86,78,83,86,'重庆市万州区');
INSERT INTO score VALUES('赵媛媛','女','2002-6-2',83,85,88,86,'重庆市沙坪坝区');
INSERT INTO score VALUES('李天','男','2002-8-8',76,68,83,76,'重庆市北碚区');
INSERT INTO score VALUES('陈新','女','2002-6-9',86,88,89,96,'陕西省延安市');
```

实现对 score 表输入多条记录，执行结果如图 5-22 所示。

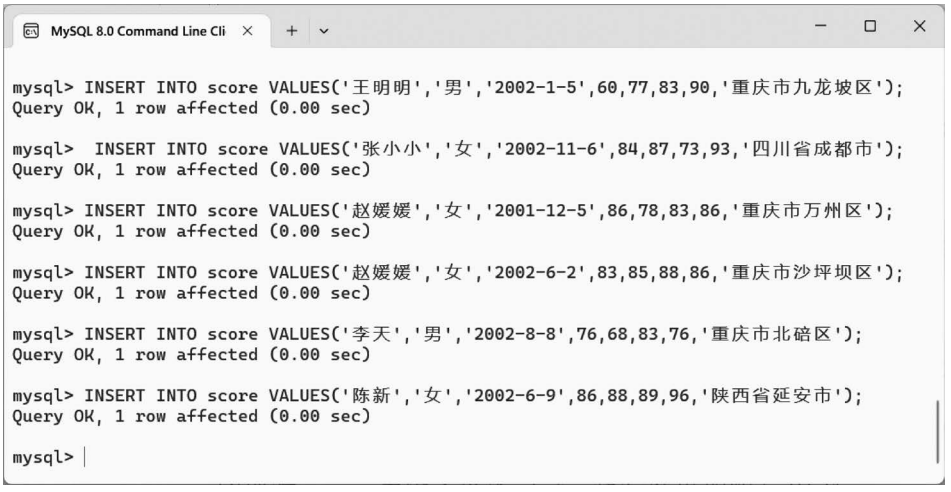

图 5-22　对 score 表输入多条记录

（3）输入命令。

```
SELECT COUNT(*),SUM(math),MAX(mysql),AVG(english),MIN(chinese) FROM score;
```

实现对 score 表统计 math 科目的总和、mysql 科目的最高分、english 科目的平均分和 chinese 科目的最低分，执行结果如图 5-23 所示。

【例 5-7】 对 score 表插入带 NULL、* 值的记录，并统计男、女生人数。

具体操作步骤如下：

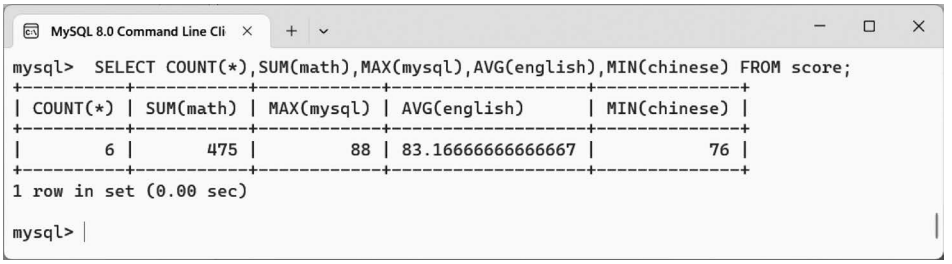

图 5-23　对 score 表进行统计计算

（1）依次输入命令。

```
INSERT INTO score VALUES(NULL,'男','2002-4-4',66,77,88,99,'重庆市长寿区');
INSERT INTO score VALUES('*','女','2002-6-6',99,88,77,66,'重庆市长寿区');
```

实现对 score 表的 xm 字段插入带 NULL、* 的记录，如图 5-24 所示。

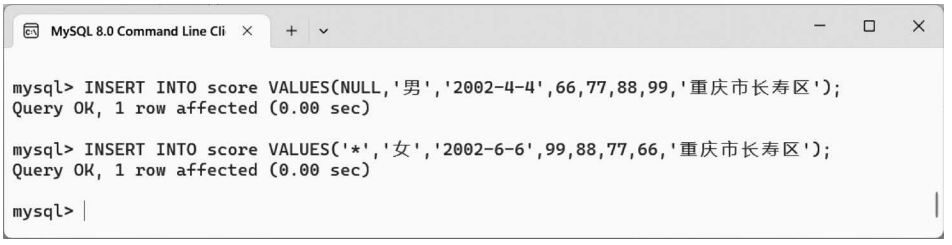

图 5-24　对 score 表的 xm 字段插入带 NULL、* 的记录

（2）输入命令。

```
SELECT xb,COUNT(XM) FROM SCORE GROUP BY xb;
```

实现对 score 表统计男、女生人数，执行结果如图 5-25 所示。

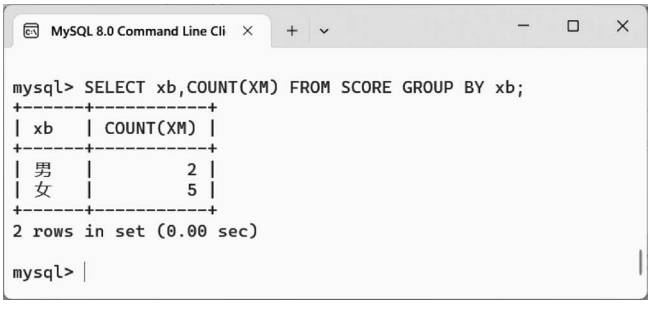

图 5-25　对 score 表统计男、女生人数

微课课堂

聚合函数 COUNT()：

（1）对于 * 值的行，COUNT()会统计；

（2）对于 NULL 值的行，COUNT()不统计。

【例 5-8】 先对 score 表统计男、女生人数及具体人名信息，然后对姓名按照拼音字母

排列，从大到小的顺序输出，再对 xm 字段实现去重管理。

具体操作步骤如下：

（1）依次输入命令。

```
SELECT xb,GROUP_CONCAT(xm) FROM score GROUP BY xb;
```

实现对 score 表统计男、女生人数及具体人名，执行结果如图 5-26 所示。

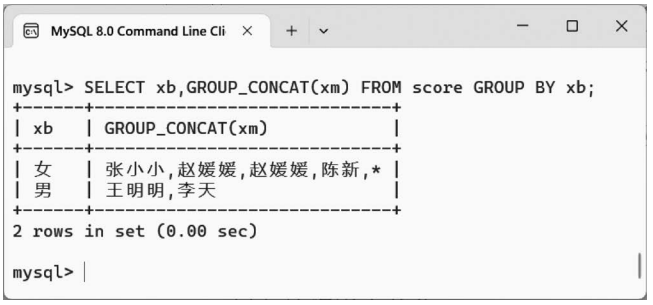

图 5-26　对 score 表统计男、女生人数及具体人名

（2）输入命令。

```
SELECT xb,GROUP_CONCAT(xm ORDER BY xm DESC) FROM score GROUP BY xb;
```

实现对 score 表对姓名按照拼音字母排列，从大到小的顺序输出，执行结果如图 5-27 所示。

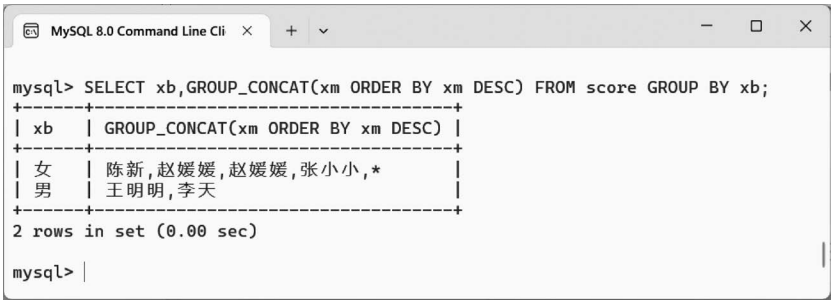

图 5-27　对 score 表对姓名按照拼音字母排列，从大到小的顺序输出

（3）输入命令。

```
SELECT xb,GROUP_CONCAT(DISTINCT xm) FROM score GROUP BY xb;
```

实现对 xm 字段去重管理，如图 5-28 所示。

【例 5-9】　显示当前系统的日期和时间，并以各种不同的格式显示输出。

具体操作步骤如下：

（1）输入命令。

```
SELECT NOW();
```

实现显示当前系统的日期和时间，执行结果如图 5-29 所示。

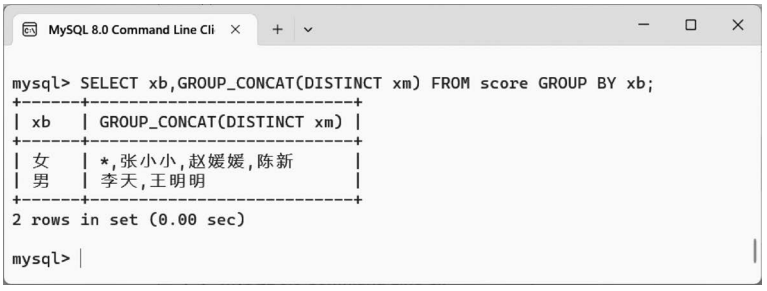

图 5-28　对 score 表实现 xm 字段去重

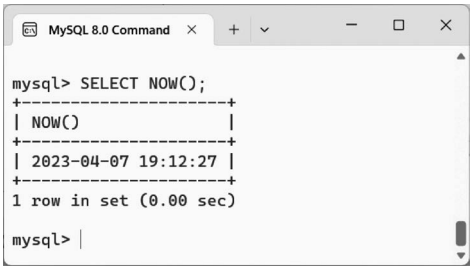

图 5-29　显示当前系统的日期和时间

（2）输入命令。

```
SELECT DATE_FORMAT(NOW(),'%y,%m,%d');
```

实现年份以 2 位数字表示，并以逗号方式隔开显示当前系统的日期，执行结果如图 5-30 所示。

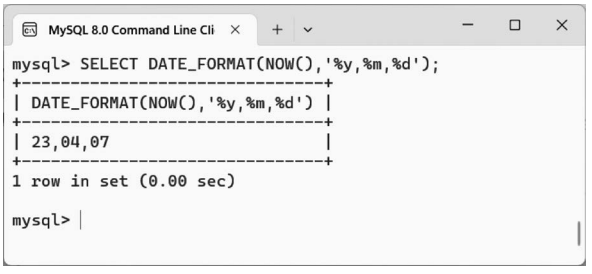

图 5-30　年份以 2 位数字并以逗号方式隔开的系统日期

（3）输入命令。

```
SELECT DATE_FORMAT(NOW(),'%c,%j,%b,%M,%a,%u');
```

实现查询显示当前的数字月份、一年中的第几天、缩写的月份名字、缩写的星期名字以及一年中的周数，执行结果如图 5-31 所示。

【例 5-10】　对 score 表实现将 xm、xb、jtdz 字段连接成新的字符串输出，再对其按照空 2 格的方式输出并求其对应字符串的长度，最后对 jtdz 字段截取对应的省份。

具体操作步骤如下：

（1）输入命令。

```
SELECT CONCAT(xm,xb,jtdz) FROM score;
```

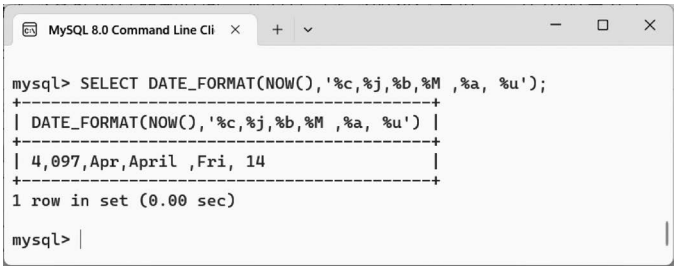

图 5-31　以指定格式查询显示当前系统的日期和时间

实现对 score 表中 xm、xb、jtdz 字段实现连接成新的字符串输出，执行结果如图 5-32 所示。

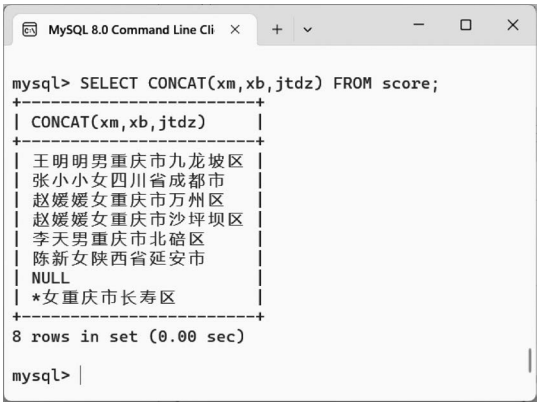

图 5-32　实现对 score 表中 xm、xb、jtdz 字段实现连接成新的字符串输出

（2）输入命令。

```
SELECT CONCAT_WS(' ',xm,xb,jtdz) FROM score;
```

实现对 score 表中 xm、xb、jtdz 字段实现连接成新的字符串，对其按照空 2 格的方式输出，执行结果如图 5-33 所示。

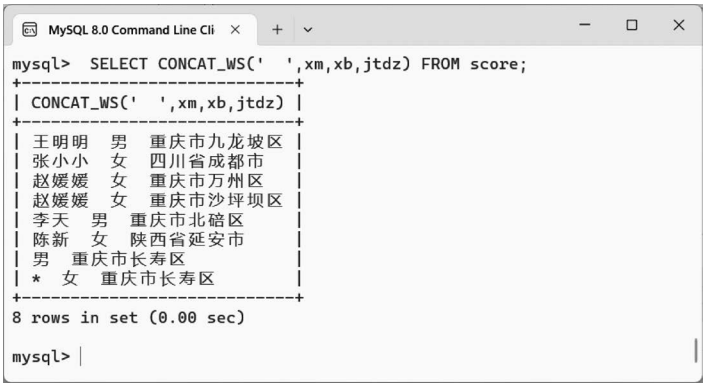

图 5-33　实现对 score 表中 xm、xb、jtdz 字段连接并按照空 2 格的方式输出

（3）输入命令。

```
SELECT LENGTH(CONCAT_WS(' ',xm,xb,jtdz)) FROM score;
```

实现对 score 表中 xm、xb、jtdz 字段实现连接成新的字符串，对其按照空 2 格的方式输出并求其长度，执行结果如图 5-34 所示。

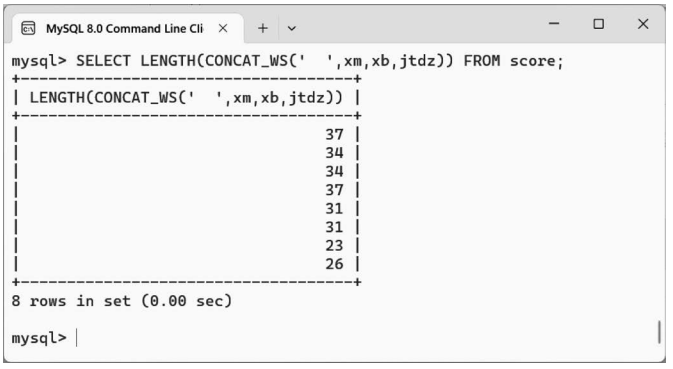

图 5-34　实现对 score 表按照空 2 格的方式对 xm、xb、jtdz 字段连接并求其长度

（4）输入命令。

```
SELECT SUBSTR(jtdz,1,3) FROM score;
```

实现对 score 表 jtdz 字段截取省份，执行结果如图 5-35 所示。

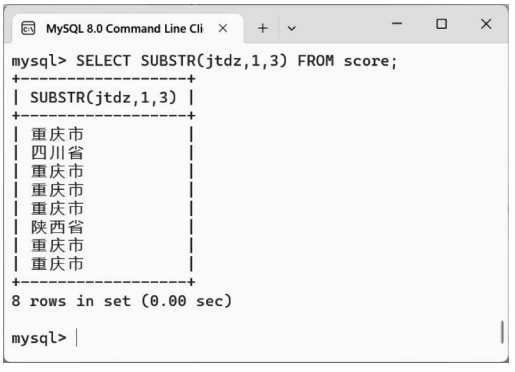

图 5-35　实现对 score 表 jtdz 字段截取对应的省份

【例 5-11】 先对 score 表增加 age 和 sum_score 字段，然后根据 csrq 字段求每人的 age 值，根据每人的各科成绩求其总和 sum_score，再对 sum_score 进行判断，350 分以上显示优秀，其余为合格等级。

具体操作步骤如下：

（1）输入命令。

```
ALTER TABLE score ADD age INT;
```

实现对 score 表增加 age 字段，执行结果如图 5-36 所示。

（2）输入命令。

```
UPDATE score SET age = (DATE_FORMAT(NOW(),'%Y') - DATE_FORMAT(csrq,'%Y'));
```

实现根据 csrq 字段求每人的 age 值，执行结果如图 5-37 所示。

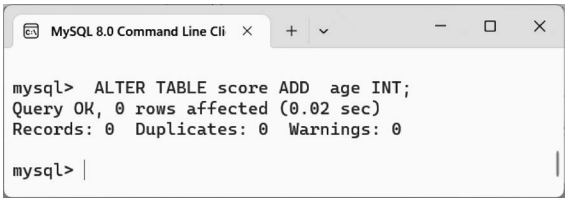

图 5-36　实现对 score 表增加 age 字段

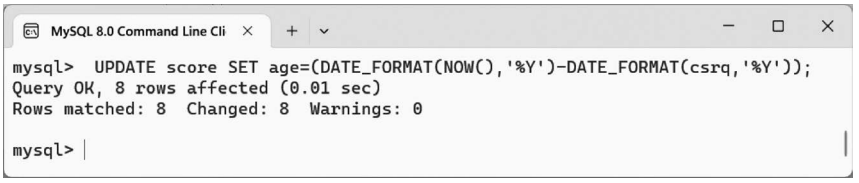

图 5-37　根据 csrq 字段求每人的 age 值

(3) 输入命令。

```
ALTER TABLE score ADD sum_score FLOAT;
```

实现对 score 表增加 sum_score 字段，执行结果如图 5-38 所示。

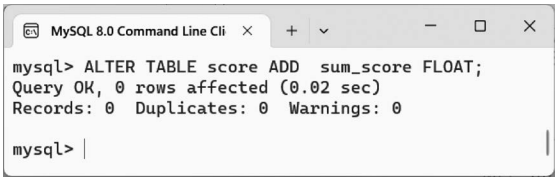

图 5-38　对 score 表增加 sum_score 字段

(4) 输入命令。

```
UPDATE score SET sum_score = math + mysql + english + chinese;
```

实现对 sum_score 字段计算出每人的各科总分，执行结果如图 5-39 所示。

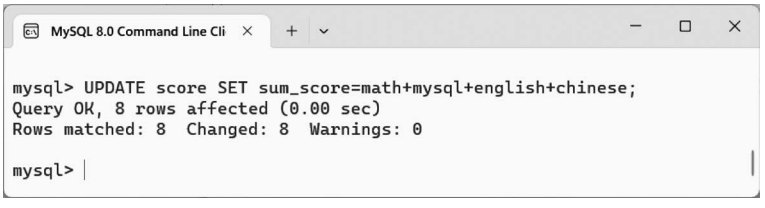

图 5-39　对 sum_score 字段计算出每人的各科总分

(5) 输入命令。

```
SELECT * FROM score;
```

查询显示 score 表中所有信息，执行结果如图 5-40 所示。

(6) 输入命令。

```
SELECT xm,sum_score,IF(sum_score＞350,'优秀','合格') FROM score;
```

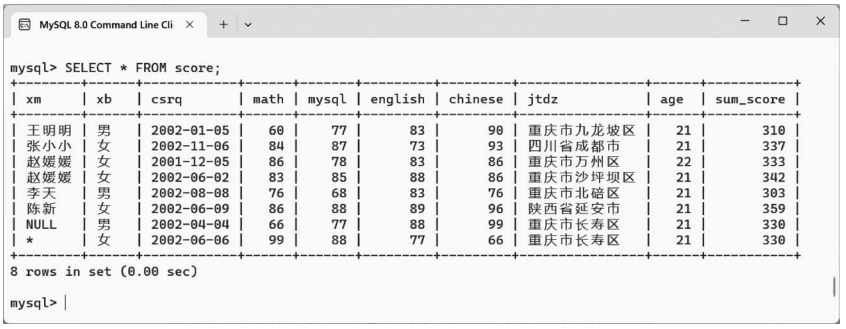

图 5-40　查询显示 score 表中所有信息

实现对 score 表中的 sum_score 字段评定等级，执行结果如图 5-41 所示。

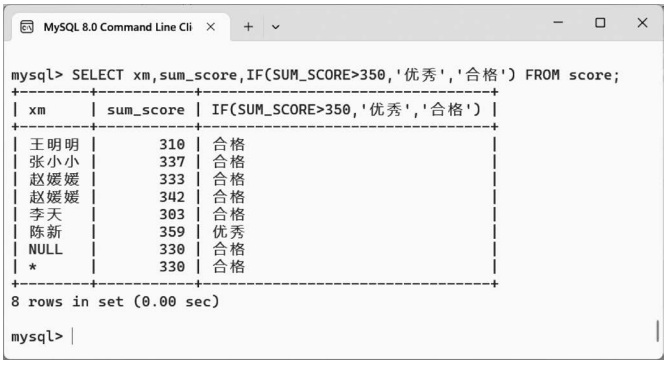

图 5-41　对 score 表中的 sum_score 字段评定等级

实训巩固

1. 在 MySQL 客户端查看各种不同的数据类型的取值范围。

2. 创建一个 teacher 表，包含 th、xm、csrq 和 zhicheng 4 个字段，其中设置 zhicheng(职称)可选值有“教授”“副教授”“高级工程师”“一级技师”“讲师”“助教”。

3. 向 teacher 表输入 10 条学生记录。

4. 对 teacher 表统计男、女生人数。

5. 对 teacher 表统计男、女生人数及具体人名信息。

6. 将 teacher 表对 th 字段按照从大到小的顺序输出，再对 xm 字段实现去重管理。

知识拓展

SET 类型的值可以取列表中的一个元素或多个元素的组合。取多个元素时，不同元素之间用逗号隔开。

1. SET 类型特征

SET 类型的值最多只能是由 64 个元素构成的组合，根据成员的不同，存储也有所不同：

1～8 成员的集合，占 1 字节。

9～16 成员的集合，占 2 字节。

17～24 成员的集合，占 3 字节。

25～32 成员的集合，占 4 字节。

33～64 成员的集合，占 8 字节。

同 ENUM 类型一样，列表中的每个值都有一个顺序排列的编号。MySQL 中存入的是这个编号，而不是列表中的值。

2. SET 类型元素值录入

插入记录时，可以使用 SET 类型中的值，也可以用索引值的方式，并且 SET 字段中的元素顺序无关紧要。记录存入 MySQL 数据库后，数据库系统会自动按照定义时的顺序显示。如果插入的成员中有重复，则只存储一次。

3. SET 类型索引值录入

当对 SET 类型利用索引值录入时，系统将按照二进制的方式，从右向左依次对应。

SET 类型：低位(右)→ 高位(左)。

例如，在例 5-5 中 xuesheng 表中 zhiwu 字段，SET('班长','团支书','纪律委员','生活委员','学生会干事','组织部部长','文体部部长','学生会主席')则：

（1）当插入索引值 4 时，对应的数值位为 00000100。当插入索引值 4 时，zhiwu 字段的描述信息如表 5-14 所示。

表 5-14　插入索引值 4 时，zhiwu 字段的描述信息

班长	团支书	纪律委员	生活委员	学生会干事	组织部部长	文体部部长	学生会主席
0	0	1	0	0	0	0	0

因此，插入的是“纪律委员”。

（2）当插入索引值 5 时，依次对应的数值位为 00000101。当插入索引值 5 时，zhiwu 字段的描述信息如表 5-15 所示。

表 5-15　插入索引值 5 时，zhiwu 字段的描述信息

班长	团支书	纪律委员	生活委员	学生会干事	组织部部长	文体部部长	学生会主席
1	0	1	0	0	0	0	0

因此，插入的是“班长，纪律委员”。

（3）当插入索引值 7 时，对应的数值位为 00000111。当插入索引值 7 时，zhiwu 字段的描述信息如表 5-16 所示。

表 5-16　插入索引值 5 时，zhiwu 字段的描述信息

班长	团支书	纪律委员	生活委员	学生会干事	组织部部长	文体部部长	学生会主席
1	1	1	0	0	0	0	0

因此，插入的是“班长，团支书，纪律委员”。

课后习题

一、选择题

1. 查看 MySQL 数据库支持的所有的数据类型所需要的命令为（　　）。

A. HELP DATA TYPES；　　B. HELP INT；

C. HELP DATA；　　D. 以上都不正确

2. MySQL 中查看双精度类型的取值范围所使用的命令为（　　）。

A. HELP INT；　　B. HELP FLOAT；

C. HELP DOUBLE；　　D. HELP TINYINT；

3. MySQL 中枚举类型的关键字是（　　）。

A. SET　　B. ENUM　　C. meiju　　D. 以上都不正确

4. （　　）函数可以实现对类别统计并详细显示每个类别所包含的具体信息。

A. GROUP_CONCAT()　　B. GROUP()

C. GROUP_BY()　　D. 以上都不是

5. MySQL 中对字段实现去重的关键字是（　　）。

A. ORDER　　B. GROUP　　C. DISTINCT　　D. SELECT

二、填空题

1. MySQL 的整数类型有 5 种，分别是_____、小整型（SMALLINT）、_____、普

通整型(INT|INTEGER)和大整型(BIGINT)。

2. MySQL 包含了大量并且丰富的函数,具体包含有_____、数值型函数、字符串型函数、_____和流程控制函数等。

3. 实现表中对姓名、性别字段按照以逗号格式隔开的方式实现连接的函数是_____。

4. 查询显示系统的日期并以一年中的第几天以及星期几所对应的函数是_____。

5. 实现对表中数据统计计数对应的函数是_____。

三、简答题

1. 简述 MySQL 包含几类数据类型,分别有哪些。

2. 简述 MySQL 包含几类常用函数,分别有哪些。

3. 简述 MySQL 中 ENUM 和 SET 类型的相同点和不同点。

4. 简述 MySQL 中使用聚合函数时的注意事项。

5. 简述 MySQL 日期时间函数中的输出格式及作用。