第4章 SQL语句

学习目标

- 掌握数据库创建
- Mysql数据库表的管理
- 表的数据操作
- 掌握SQL语言

学习内容

1.1 数据库创建

1.2 数据库表的管理

1.3 表的操作

1.4 SQL语句

1.5 本章小结

1. 创建数据库

- 在创建数据库时，数据库命名有以下几项规则：
  - 不能与其他数据库重名，否则将发生错误。
  - 名称可以由任意字母、阿拉伯数字、下划线（_）和“$”组成，可以使用上述的任意字符开头，但不能使用单独的数字，否则会造成它与数值相混淆。
  - 名称最长可为64个字符，而别名最多可长达256个字符
  - 不能使用MySQL关键字作为数据库名、表名。
- 在默认情况下，Windows下数据库名、表名的大小写是不敏感的，而在Linux下数据库名、表名的大小写是敏感的。如果为了便于数据库在平台间进行移植，可以采用小写来定义数据库名和表名。

1. 创建数据库

- 创建数据库语法结构

  使用create database或create schema命令可以创建数据库。其语法结构如下。

  create {database|schema}[if not exists]databasename

  [default]character set charset_name

  |[default]collate collation_name;

- 创建数据库。创建数据库是指在数据库系统中划分一块空间，用来存储相应的数据。这是进行表操作的基础，也是进行数据库管理的基础。MySQL中，创建数据库是通过SQL语句create database实现的。

1.1 创建数据库

1. 创建数据库

【例1】通过create database语句创建一个名称为mysql2024的数据库。

```
mysql> create  database if not exists mysql2024;
```

【例2】创建教务管理数据库teaching2024，并指定字符集为gb2312，校对原则为gb2312_chinese_ci。

```
mysql> create database teaching2024
-> default character set gb2312
-> default collate gb2312_chinese_ci;
```

1.1 管理数据库

1. 查看数据库

查看新创建的数据库 。成功创建数据库后，可以使用show databases 命令查看数据库，也可以在指定路径（或数据库的默认存放位置）下查看数据库。

1.1 管理数据库

打开数据库。数据库创建后，若要操作一个数据库，还需要使其成为当前的数据库，即打开数据库。可以使用USE语句打开一个数据库，使其成为当前默认数据库。

例如， 选择名称为mysql2024的数据库，设置其为当前默认的数据库。

命令和运行结果如下：

```
mysql> use mysql2024;
        Database changed
```

1.1 管理数据库

修改数据库。数据库创建后，如果需要，可以修改数据库的参数。

修改数据库的语法格式如下：

```
alter {database | schema} [db_name]
[default] character set charset_name
|[default] collate collation_name;
```

【例3】将mysql2024库修改字符集为gb2312，校对原则为gb2312_chinese_ci

命令和运行结果如下：

```
mysql> alter database mysql2024
    -> default character set gb2312
    -> collate gb2312_chinese_ci;
Query OK, 1 row affected (0.00 sec)
```

1.1 管理数据库

显示数据库结构。如果查看已有数据库的相关信息，例如MySQL版本id号、默认字符集等信息，使用MySQL命令实现。

【例4】显示数据库teaching2024的结构信息。

命令和运行结果如下：

```
mysql> show create database teaching2024;
```

1.1 管理数据库

删除数据库 。删除数据库是指在数据库系统中删除已经存在的数据库。删除数据库之后，原来分配的空间将被收回。删除数据库语法格式如下：

```
drop database [if exists] db_name
```

例如，删除mysqltest库命令如下：

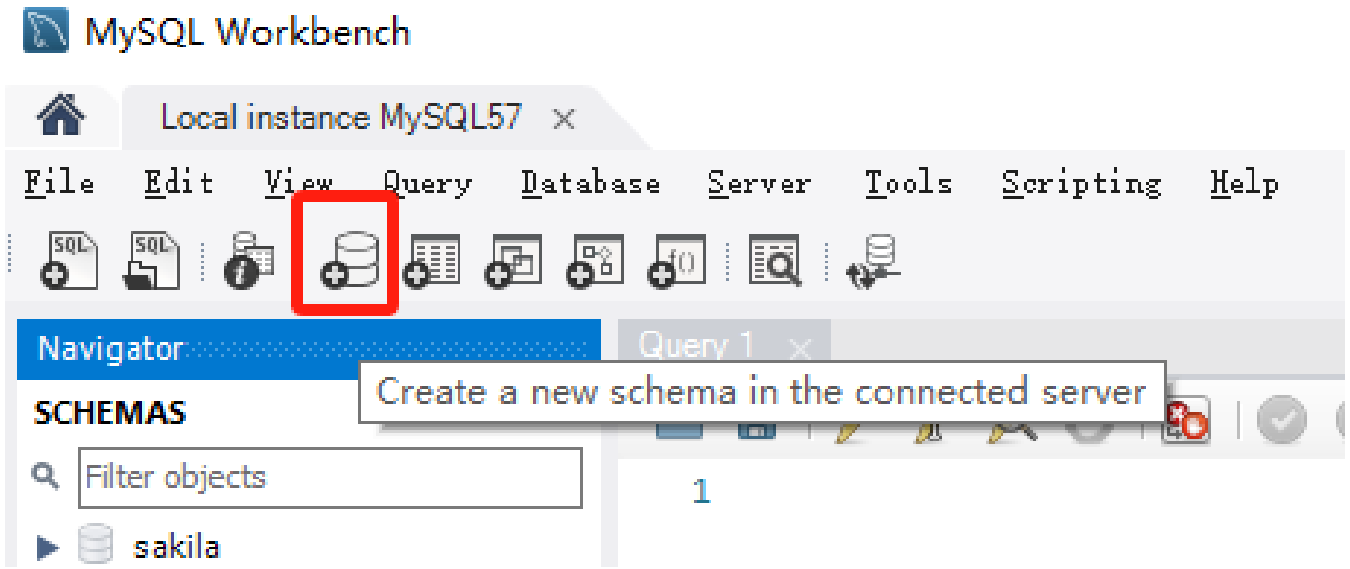
```
mysql> drop database mysqltest;
```

需要注意的是，删除数据库会删除该数据库中所有的表和所有数据。因此，删除数据库前最好存有备份。

1.1 Mysql workbench管理数据库

1.使用MySQL Workbench创建数据库

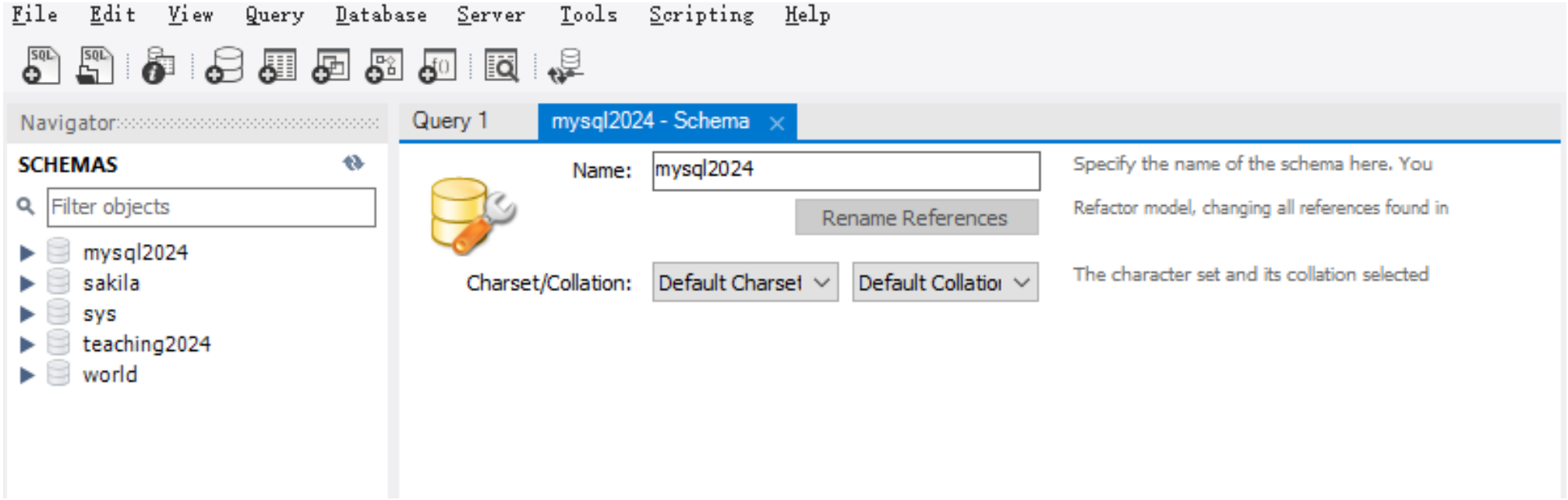
在数据库操作主界面中，单击如图所示的提示为Create a new schema in the connected server的创建数据库按钮，Schema在这里就是数据库的意思。



1.1 Mysql workbench管理数据库

1.使用MySQL Workbench创建数据库

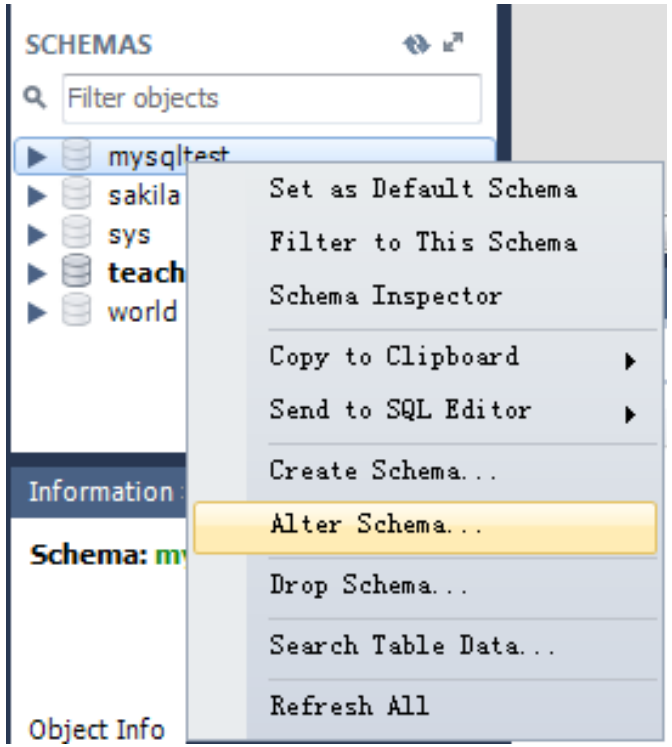
进入如图所示设置数据库参数界面。在输入数据库名mysql2024，选择字符集和排序规则。然后，单击Apply按钮。



1.1 Mysql workbench管理数据库

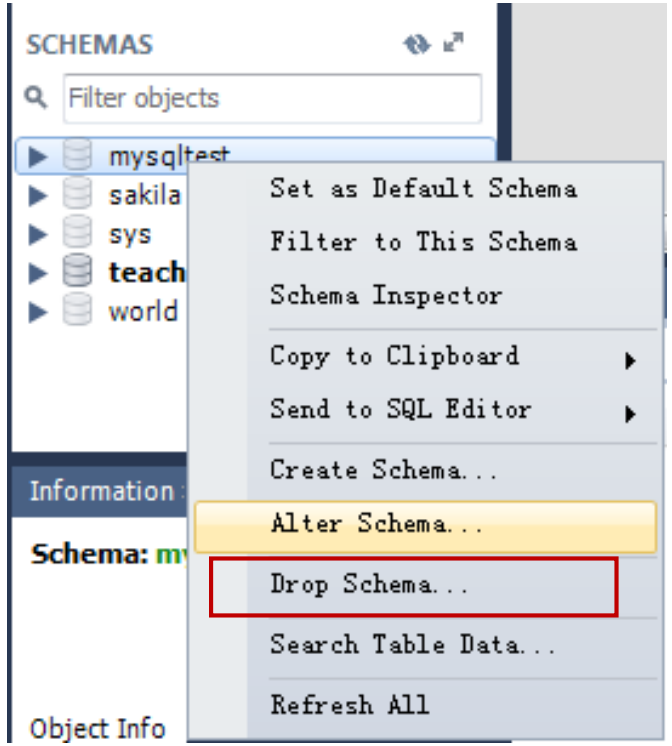
2．修改数据库参数

在数据库操作界面的schemas区域，右击要修改的数据库mysql2024，如图所示。执行弹出菜单中的Alter schema命令。



3．删除数据库参数

在数据库操作界面的schemas区域，右击要修改的数据库mysql2024，如图所示。执行弹出菜单中的Drop schema命令。



1.2 数据库表的管理

创建数据库的为了存储、管理和查询数据，而表是存储数据最重要的载体。表是MySQL数据库中最重要的数据库对象，也是构建高性能数据库的基础。数据表设计的优劣将影响磁盘空间使用效率、数据处理时内存的利用率以及数据的查询效率。

1.2 数据库表的管理

在MySQL数据库系统中，可以按照不同的标准对表进行分类。

- 按照表的用途分类。

①系统表：用于维护MySQL服务器和数据库正常工作的数据表。例如，系统数据库mysql中就存在若干系统表。

②用户表：由用户自己创建的、用于各种数据库应用系统开发的表。

③分区表：分区表是将数据水平划分为多个单元的表，这些单元可以分布到数据库中的多个文件组中。在维护整个集合的完整性时，使用分区可以快速而有效地访问或管理数据子集，从而使大型表或索引更易于管理。

1.2 数据库表的管理

- 按照表的存储时间分类。

①永久表：包括SQL Server的系统表和用户数据库中创建的数据表，该类表除非人工删除，否则一直存储在介质中。

②临时表：临时表只有创建该表的用户在用来创建该表的连接中可见。临时表关联的连接被关闭时，临时表自动地被删除。如果服务器关闭，则所有临时表会被清空、关闭。

1.2 数据库表的管理

1. InnoDB存储引擎的表空间

InnoDB表空间分为共享表空间和独享表空间两种类型。

- 表空间的基本概念

(1) 共享表空间。MySQL服务实例承载着的数据库的所有InnoDB表的数据、索引、各种元数据以及事务回滚（undo）信息，全部存放在共享表空间文件中。

(2) 独立表空间。每一个表都将会以独立的文件方式来进行存储，每一个表都有一个.frm表描述文件，还有一个.ibd文件。该文件包括一个表单独的数据内容以及索引内容。

1.2 数据库表的管理

1. InnoDB存储引擎的表空间

- 查看数据库的表空间

  利用如下命令可以查看数据库的表空间。

  mysql> show variables like 'InnoDB_data%';

  - 表空间有四个文件组成：ibdata1、ibdata2、ibdata3、ibdata4，每个文件的大小为10M，当每个文件都满了的时候，ibdata4会自动扩展；
  - 不管是共享表空间和独立表空间，都会存在InnoDB_data_file文件，因为这些文件不仅仅要存放数据，而且还要存储事务回滚（undo）信息。

1.2 数据库表的管理

1. InnoDB存储引擎的表空间

- 共享表空间和独立表空间的比较
  - 共享表空间的特点。
    - 表空间可以分成多个文件存放在一起方便管理。
    - 多个表及索引在表空间中混合存储，当数据量非常大的时候，表做了大量删除操作后表空间中将会有大量的空隙，特别是对于统计分析，对于经常删除操作的这类应用最不适合用共享表空间。
    - 共享表空间分配后不能回缩。

1.2 数据库表的管理

1.InnoDB存储引擎的表空间

- 共享表空间和独立表空间的比较
  - 独立表空间的特点。
  - 每个表都有独立的表空间，每个表的数据和索引都会存在自己的表空间中，可以实现单表在不同的数据库中移动。
  - Drop table操作自动回收表空间，如果对于统计分析或是日值表，删除大量数据后可以通过命令“alter table TableName engine=innodb; ”回收不用的空间。
  - 对于使用独立表空间的表，不管怎么删除，表空间的碎片不会太严重的影响性能，而且还有机会处理。
  - 单表增加过大，当单表占用空间过大时，存储空间会不足。

1.2 数据库表的管理

1. InnoDB存储引擎的表空间

- 共享表空间和独立表空间之间的转换

(1) 查看当前数据库的表空间管理类型。可以通过如下命令查看。

```
mysql> show variables like "InnoDB_file_per_table";
```

ON代表独立表空间管理，OFF代表共享表空间管理

(2) 修改数据库的表空间管理方式。修改InnoDB_file_per_table的参数值（InnoDB_file_per_table=1 为使用独占表空间，InnoDB_file_per_table=0 为使用共享表空间）即可，但是修改不能影响之前已经使用过的共享表空间和独立表空间；

1.3 创建数据库表

创建表的语法结构

- 表决定了数据库的结构， 表是存放数据的地方，一个库需要什么表，各数据库表中有什么样的列，是要合理设计的。
- 创建表的语法结构如下。

```
create [temporary]table[if not exists]table_name
[([column_definition], ...|[index_definition])]
[table_option][select_statement];
```

其中，column_definition：字段的定义。包括指定字段名、数据类型、是否允许空值，指定默认值、主键约束、唯一性约束、注释字段名、是否为外键，以及字段类型的属性等。字段的定义具体格式描述如下：

```
col_name  type  [not null | null] [default default_value]
[auto_increment] [unique [key] | [primary] key]
[comment 'string'] [reference_definition]
```

1.3 创建数据库表

1. 利用SQL语句创建数据表

教务管理数据库mysql2024将创建5张表：student(学生表)、course（课程表）、score（成绩表）、teacher（教师表）和teach_course（纽带表）。

按照表所示的学生信息表结构创建student表。

列序号	字段名	类型	取值说明	列含义
1	studentno	char(11)	主键	学生学号
2	sname	char(8)	否	学生姓名
3	sex	enum (2)	否	性别
4	birthdate	date	否	出生日期
5	entrance	int(3)	否	入学成绩
6	phone	varchar(12)	否	电话
7	Email	varchar(20)	否	电子信箱

1.3 创建数据库表

```
create table if not exists student
(
studentno  char(11) not null comment'学号’,
sname char(8) not null comment'姓名’,
 sex enum('男', '女')  not null default'男' comment'性别’,
birthdate date not null comment'出生日期’,
entrance int(3)  null comment'入学成绩’,
phone varchar(12) not null comment'电话’,
Email varchar(20) not null comment'电子信箱’,
primary key (studentno)
);
```

1.3 创建数据库表

2. 利用create table命令建立课程信息表course，表结构如表所示。

列序号	字段名	类型	取值说明	列含义
1	courseno	char(6)	主键	课程编号
2	cname	char(20)	否	课程名称
3	type	char(8)	否	类别
4	period	int(2)	否	总学时
5	exp	int(2)	否	实验学时
6	term	int(2)	否	开课学期

1.3 创建数据库表

```
create table if not exists course
(
courseno  char(6) not null,
cname  char(6) not null,
type char(8) not null,
period int(2) not null,
exp int(2) not null,
term int(2) not null,
primary key (courseno)
);
```

1.3 创建数据库表

3. 利用create table命令建立学生分数表score，表结构如表所示。该表中主键由两个列构成。

列序号	字段名	类型	取值说明	列含义
1	studentno	char(11)	主键	学号
2	courseno	char(6)		课程编号
3	daily	float(3,1)	否	平时成绩
4	final	float(3,1)	否	期末成绩

```
create table if not exists score
    (studentno  char(11) not null,
    courseno  char(6) not null,
    daily float(3,1) default 0,
    final float(3,1) default 0,
    primary key (studentno , courseno)
    );
```

1.3 创建数据库表

4.利用create table命令建立教师信息表teacher，表结构如表所示。

列序号	字段名	类型	取值说明	列含义
1	teacherno	char(6)	主键	教师编号
2	tname	char(8)	否	教师姓名
3	major	char(10)	否	专业
4	prof	char(10)	是	职称
5	department	char(16)	否	院系部门

```
create table if not exists teacher
(teacherno  char(6) not null comment '教师编号',
tname  char(8) not null comment'教师姓名',
major  char(10) not null comment '专业',
prof char(10) not null comment '职称',
department char(16) not null comment '部门',
primary key (teacherno)
);
```

1.3 创建数据库表

5. 为了完善teaching2024数据库的表间联系，创建表结构如表所示的纽带表teach_course。

```
mysql> create table if not exists teach_course
       (teacherno char(6) not null,
       courseno  char(6) not null,
       primary key (teacherno,courseno)
       );
```

列序号	字段名	类型	取值说明	列含义
1	teacherno	nchar(6)	主键	教师编号
2	courseno	nchar(6)		课程编号

1.3 创建数据库表

- 说明：
  - (1) 主键设置。primary key表示设置该字段为主键。
  - (2) 添加注释。comment'学号'表示对studentno字段增加注释为“学号”
  - (3) 字段类型的选择。sex enum('男'，'女')表示sex字段的字段类型是enum，取值范围为'男'和 '女'。对于取值固定的字段可以设置数据类型为enum。例如，在course表的type字段表示的是课程的类型，一般是固定的几种类型。因此，也可以把该字段的定义写成：type enum ('必修课'，' 选修课') default '必修课'。
  - (4) 默认值的设置。default'男'表示默认值为“男”。
  - (5) 设置精度。Score表中的daily float(3,1)表示精度为4，小数位1位。
  - (6) 如果没有指定是null或是not null，则列在创建时假定为null。

1.3 创建数据库表

◆ 设置表的属性值自动增加

- 在MySQL数据表中，一个整数列可以拥有一个附加属性auto_increment。其主要用于为表中插入的新记录自动生成唯一的序列编码
- 默认的情况下，该字段值是从1开始自增，也可自定义开始值。一个数据表只能有一个字段使用auto_increment约束，且该字段必须为主键的一部分。可以是任何整数类型（tinyint、samllint、int、bigint等）。
- 设置属性值字段增加的基本语法规则如下：

  属性名 数据类型 auto_increment

1.3 创建数据库表

在mysql2024库中，创建选课表sc，选课号sc_no是自动增量，选课时间默认为当前时间，其他字段分别是学号、课程号和教师号。

```
mysql>create table sc
        (sc_no int(6) not null auto_increment,
        studentno  char(11) not null,
        courseno  char(6) not null,
        teacherno char(6) not null,
        sc_time timestamp not null default now(),
        primary key (sc_no) );
```

1.4 查看数据库表

查看表

- 数据表创建后，就可以用show tables命令查询已创建的表的情况。也可以查看表结构，即是指查看数据库中已存在的表的定义。查看表结构的语句包括describe语句和show create table语句。
- 通过这两个语句，可以查看表的字段名、字段的数据类型、完整性约束条件等。

(1) 查看已经创建的表。命令和运行结果如下：

mysql>show tables;

```
mysql> show tables;
+--------------------+
| Tables_in_mysql2024 |
+--------------------+
| course             |
| sc                 |
| score              |
| student            |
| teach_course       |
| teacher            |
+--------------------+
6 rows in set (0.00 sec)

mysql>
```

1.4 查看数据库表

（2）查看表基本结构语句describe。MySQL中，describe语句可以查看表的基本定义。其中包括，字段名、字段数据类型、是否为主键和默认值等。

describe语句的命令如下：

mysql> describe student;

（3） 查看表详细结构语句show create table。MySQL中，show create table语句可以查看表的详细定义。show create table语句的命令如下：

mysql> show create table course;

```
mysql> describe student;
+-----------+-----------------+------+-----+---------+-------+
| Field     | Type            | Null | Key | Default | Extra |
+-----------+-----------------+------+-----+---------+-------+
| studentno | char(11)        | NO   | PRI | NULL    |       |
| sname     | char(8)         | NO   |     | NULL    |       |
| sex       | enum('男','女') | NO   |     | 男      |       |
| birthdate | date            | NO   |     | NULL    |       |
| entrance  | int(3)          | YES  |     | NULL    |       |
| phone     | varchar(12)     | NO   |     | NULL    |       |
| Email     | varchar(20)     | NO   |     | NULL    |       |
+-----------+-----------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

```
mysql> show create table course;
+--------+-----------------------------------------
----------------------------------------
-----------------------+
| Table  | Create Table
                       |
+--------+-----------------------------------------
----------------------------------------
-----------------------+
| course | CREATE TABLE `course` (
  `courseno` char(6) NOT NULL,
  `cname` char(6) NOT NULL,
  `type` char(8) NOT NULL,
  `period` int(2) NOT NULL,
  `exp` int(2) NOT NULL,
  `term` int(2) NOT NULL,
  PRIMARY KEY (`courseno`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+--------+-----------------------------------------
----------------------------------------
-----------------------+
1 row in set (0.00 sec)

mysql>
```

1.5 修改数据库表

- 修改数据库表

alter table用于更改原有表的结构。例如，可以增加或删减字段、重新命名字段或表，还可以修改默认字符集。

(1) 增加字段。在创建表时，表中的字段就已经定义完成。如果要增加新的字段，可以通过alter table语句进行增加。增加表的字段，可以实现如下功能：

- 增加无完整性约束条件的字段。
- 增加有完整性约束条件的字段。
- 表的第一个位置增加字段。
- 表的指定位置之后增加字段。

1.5 修改数据库表

- 修改表语法格式。修改数据库表语法格式如下：

```
alter [ignore] table tbl_name
alter_specification [, alter_specification] ...
alter_specification:
add [column] column_definition [first | after col_name ] //添加字段
|alter [column]col_name{set default literal|drop default}//修改字段默认值
|change [column] old_col_name column_definition          //重命名字段
[first|after col_name]
|modify [column]column_definition[first|aftercol_name]  //修改字段数据类型
```

1.5 修改数据库表

|drop [column] col_name //删除列
|rename [TO] new_tbl_name //对表重命名
|order by col_name //按字段排序
|convert TO character set charset_name[collate collation_name]
//将字符集转换为二进制
|[default] character set charset_name [collate collation_name]
//修改表的默认字符集

【例】 在student表的Email列后面增加一列address。

```
mysql>alter table student
    ->add address varchar(30) not null after Email;
```

1.5 修改数据库表

- 修改表名。表名可以在一个数据库中唯一的确定一张表。数据库系统通过表名来区分不同的表。MySQL中，修改表名是通过SQL语句alter table实现的。

  【例】 将表sc重名为se_course。

  mysql> alter table sc rename to se_course;

- 修改字段的数据类型。alter table语句也可以修改字段的数据类型。

【例】修改course表的type字段，因为该字段一般是取固定值。因此，也可以把该字段的定义写成：type enum (‘必修课，’选修课‘) default ’必修课‘。

```
mysql> alter table course
    -> modify type enum('必修','选修') default '必修';
```

1.5 修改数据库表

- 删除字段。删除字段是指删除已经定义好的表中的某个字段。MySQL中，alter table语句也可以删除表中的字段。

删除student表的字段address。

```
mysql> alter table student  drop address;
```

1.5 修改数据库表

- 删除数据库表，删除表是指删除数据库中已存在的表。删除表时，会删除表中的所有数据。因此，在删除表时要特别注意。
- MySQL中通过drop table语句来删除表。删除表的语法格式如下：

  drop table table_name

【例】在mysql2024数据库中创建表example，然后删除example表。

```
create table example
(
today datetime,name char(20)
);
```

mysql> desc example;

Field	Type	Null	Key	Default	Extra
today	datetime	YES		NULL	
name	char(20)	YES		NULL	

mysql> drop table example;

1.5 修改数据库表

临时表的管理

- Mysql临时表适合当工作在非常大的表上时，偶尔需要运行很多查询获得一个大量数据的小的子集，不是对整个表运行这些查询，而是让MySQL每次找出所需的少数记录，将记录选择到一个临时表可能更快些，然后多这些表运行查询。
- 创建临时表很容易，给正常的create table语句加上temporary关键字即可。例如，创建临时表tmp_emp1。

```
mysql> create temporary table tmp_emp1
-> (name varchar(10) not null,
-> value integer not null
->  );
```

1.5 修改数据库表

临时表的管理

- 临时表将在连接MySQL期间存在。断开时，MySQL将自动删除表并释放所用的空间。当然你可以在仍然连接的时候删除临时表并释放空间。删除方法与一般用户表相同。

  drop table tmp_table

  说明：

  （1）创建临时表你必须有create temporary table 权限。

  （2）show tables语句不会列举临时表。

  （3）不能用rename来重命名一个临时表。

1.6 索引建立与删除

- 建立索引的目的：加快查询速度
- 关系数据库管理系统中常见索引：
  - 顺序文件上的索引
  - B+树索引（参见爱课程网3.2节动画《B+树的增删改》）
  - 散列（hash）索引
  - 位图索引
- 特点：
  - B+树索引具有动态平衡的优点
  - HASH索引具有查找速度快的特点

1.6 索引建立与删除

◆语句格式

CREATE [UNIQUE] [CLUSTER] INDEX <索引名>

ON <表名>(<列名>[<次序>][,<列名>[<次序>]]...);

- <表名>：要建索引的基本表的名字
- 索引：可以建立在该表的一列或多列上，各列名之间用逗号分隔
- <次序>：指定索引值的排列次序，升序：ASC，降序：DESC。缺省值：ASC
- UNIQUE：此索引的每一个索引值只对应唯一的数据记录
- CLUSTER：表示要建立的索引是聚簇索引

1.6 索引建立与删除

[例] 为mysql2024数据库中的Student表建立索引。Student表按学号升序建唯一索引.

```
mysql> create UNIQUE INDEX  stusno ON student(studentno ASC);
```

删除这个索引：

```
mysql> drop index stusno on student;
```

修改索引名：

```
mysql> alter table student rename index stusno to stusnono;
```

1.7 数据类型

- SQL中域的概念用数据类型来实现
- 定义表的属性时需要指明其数据类型及长度
- 选用哪种数据类型
  - ✓取值范围
  - ✓要做哪些运算

1.7 数据类型

数据类型	含义
CHAR(*n*),CHARACTER(*n*)	长度为n的定长字符串
VARCHAR(*n*), CHARACTERVARYING(*n*)	最大长度为n的变长字符串
CLOB	字符串大对象
BLOB	二进制大对象
INT，INTEGER	长整数（4字节）
SMALLINT	短整数（2字节）
BIGINT	大整数（8字节）
NUMERIC(*p*，*d*)	定点数，由*p*位数字（不包括符号、小数点）组成，小数后面有*d*位数字
DECIMAL(*p*, *d*), DEC(*p*, *d*)	同NUMERIC
REAL	取决于机器精度的单精度浮点数
DOUBLE PRECISION	取决于机器精度的双精度浮点数
FLOAT(*n*)	可选精度的浮点数，精度至少为*n*位数字
BOOLEAN	逻辑布尔量
DATE	日期，包含年、月、日，格式为YYYY-MM-DD
TIME	时间，包含一日的时、分、秒，格式为HH:MM:SS
TIMESTAMP	时间戳类型
INTERVAL	时间间隔类型

1.7 数据类型

1. 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。字符串类型包括char、varchar、blob、text、enum和set。字符串类型可以分为2类：普通的文本字符串类型（char和varchar）和特殊类型（set和enum）。它们之间都有一定的区别，取值的范围不同，应用的地方也不同。

(1) 普通的文本字符串类型，即char和varchar类型，char列的长度被固定为创建表所声明的长度，取值在1~255之间；varchar列的值是变长的字符串，取值和char一样。下面介绍普通的文本字符串类型如表所示。

1.7 数据类型

类型	取值范围	说明
[national] char(m) [binary|ASCII|unicode]	0~255个字符	固定长度为m的字符串，其中m的取值范围为0~255。National关键字指定了应该使用的默认字符集。Binary关键字指定了数据是否区分大小写（默认是区分大小写的）。ASCII关键字指定了在改列中使用latin1字符集。Unicode关键字指定了使用UCS字符集
char	0~255个字符	Char(m)类似
[national] varchar(m) [binary]	0~255个字符	长度可变，其他和char(m)类似

1.7 数据类型

1. 字符串类型

（2）特殊类型set和enum，介绍如表所示。

类型	最大值	说明
Enum (“value1”, “value2”, ...)	65 535	该类型的列只可以容纳所列值之一或为null
Set (“value1”, “value2”, ...)	64	该类型的列可以容纳一组值或为null

1.7 数据类型

1. 字符串类型

说明：在创建表时，使用字符串类型时应遵循以下原则：

(1) 从速度方面考虑，要选择固定的列，可以使用char类型。

(2) 要节省空间，使用动态的列，可以使用varchar类型。

(3) 要将列中的内容限制在一种选择，可以使用enum类型。

(4) 允许在一个列中有多于一个的条目，可以使用set类型。

(5) 如果要搜索的内容不区分大小写，可以使用text类型。

1.7 数据类型

2. 数字类型总体可以分成整型和浮点型两类。

- 整数类型。整数类型是数据库中最基本的数据类型。标准SQL中支持integer和smallint这两类整数类型。这些类型包括准确数字的数据类型（numeric、decimal、integer和smallint），还包括近似数字的数据类型（float、real和double precision）。其中的关键词int是integer的同义词。

1.7 数据类型

数据类型	取值范围	说明	单位
tinyint	符号值：-127~127 无符号值：0~255	最小的整数	1字节
bit	符号值：-127~127 无符号值：0~255	最小的整数	1字节
bool	符号值：-127~127 无符号值：0~255	最小的整数	1字节
smallint	符号值：- 32768~32767 无符号值：0~65535	小型整数	2字节
mediumint	符号值：- 8388608~8388607 无符号值：0~16777215	中型整数	3字节
int	符号值：- 2147683648~2147683647 无符号值：0~4294967295	标准整数	4字节
bigint	符号值：- 92233720368547758O8~92233720368547758O7 无符号值：0~18446744073709551615	大整数	8字节

1.7 数据类型

- 浮点数据类型。MySQL中使用浮点数类型和定点数类型来表示小数。浮点数类型包括单精度浮点数（float型）和双精度浮点数（double型）。定点数类型就是decimal型，关键词dec是decimal的同义词。

数据类型	取值范围	说明	单位
float	**+(-)3.402823466E+38**	**单精度浮点数**	**8或4字节**
double	**+(-)1.7976931348623157E+308 +(-)2.2250738585072014E-308**	**双精度浮点数**	**8字节**
decimal	**可变**	**一般整数**	**自定义长度**

3. 日期和时间数据类型

日期与时间类型是为了方便在数据库中存储日期和时间而设计的。

- MySQL中有多种表示日期和时间的数据类型。其中，year类型表示年份；date类型表示日期；time类型表示时间；datetime和timestamp表示日期和时间。其中的每种类型都有其取值的范围，如赋予它一个不合法的值，将会被“0”代替。下面介绍日期和时间数据类型，如表2-6所示。

1.7 数据类型

类型	取值范围	说明
date	**1000-01-01**	日期，格式**YYYY-MM-DD**
time	**-838:58:59　835:59:59**	时间，格式**HH：MM：SS**
datetime	**1000-01-01 00:00:00** **9999-12-31 23:59:59**	日期和时间，格式**YYYY-MM-DD HH：MM：SS**
timestamp	**1970-01-01 00:00:00** **2037年的某个时间**	时间标签，在处理报告时使用显示格式取决于**M**的值
year	**1901-2155**	年份可指定两位数字和四位数字的格式

4. 二进制类型

- 二进制类型是在数据库中存储二进制数据的数据类型。二进制类型包括binary、varbinary、bit、tinyblob、blob、mediumblob和longblob类型。tinytext、longtext和text等适合存储长文本的类型，也放在这里介绍。
- 其中，text和blob类型。它们的大小可以改变，text类型适合存储长文本，而blob类型适合存储二进制数据，支持任何数据，例如文本、声音和图像等。下面介绍text和blob类型，如表2-7所示。

1.7 数据类型

类　型	最大长度（字节数）	说　明
tinyblob	**2^8~1(225)**	**小blob字段**
tinytext	**2^8~1(225)**	**小text字段**
blob	**2^16~1(65 535)**	**常规blob字段**
text	**2^16~1(65 535)**	**常规text字段**
mediumblob	**2^24~1(16 777 215)**	**中型blob字段**
mediumtext	**2^24~1(16 777 215)**	**中型text字段**
longblob	**2^32~1(4 294 967 295)**	**长blob字段**
longtext	**2^32~1(4 294 967 295)**	**长text字段**

4. 二进制类型

- 二进制类型是在数据库中存储二进制数据的数据类型。二进制类型包括binary、varbinary、bit、tinyblob、blob、mediumblob和longblob类型。tinytext、longtext和text等适合存储长文本的类型，也放在这里介绍。
- 其中，text和blob类型。它们的大小可以改变，text类型适合存储长文本，而blob类型适合存储二进制数据，支持任何数据，例如文本、声音和图像等。下面介绍text和blob类型，如表2-7所示。

1.7 数据类型

类　型	最大长度（字节数）	说　明
tinyblob	2^8~1(225)	小blob字段
tinytext	2^8~1(225)	小text字段
blob	2^16~1(65 535)	常规blob字段
text	2^16~1(65 535)	常规text字段
mediumblob	2^24~1(16 777 215)	中型blob字段
mediumtext	2^24~1(16 777 215)	中型text字段
longblob	2^32~1(4 294 967 295)	长blob字段
longtext	2^32~1(4 294 967 295)	长text字段

1.8 表的数据操作

- MySQL数据表分为表结构（Structure）和数据记录（Record）2部分。前面创建表的操作，仅仅是创建了表结构，表结构即决定表拥有哪些字段以及这些字段的名称、数据类型、长度、精度、小数位数、是否允许空值（null）、设置默认值和主键等。
- MySQL语言一般通过insert、update和delete等3种DML语句对表进行数据的添加、更新和删除数据操作，并以此维护和修改表的数据。

1.8 表的数据操作

1. 表记录的插入

为数据表输入数据的方式有多种，常见的有通过命令方式添加行数据的，也可以通过程序实现表数据的添加。可以通过insert into、replace into语句插入，也可以使用load data in file方式将保存在文本文件中的数据插入到指定的表。

2. 使用insert into| replace语句添加数据

(1) insert into| replace语句语法格式：

insert | replace[into]table_name[(col_name,...)]

values({expr|default},...),(...),...

|set col_name ={expr|default}, ...

1.8 表的数据操作

【例】利用**insert into**命令向表**student**中插入多行数据。

```
mysql> insert into student values
-> ('18122221324','何白露',
-> '女','2000/12/4','879','13178978999','heyy@sina.com '),
-> ('18125111109','敬横江',
-> '男','2000/3/1','789','  15678945623','jing@sina.com '),
-> ('18125121107','梁一苇',
-> '女','1999/9/3','777','  13145678921','bing@126.com '),
-> ('18135222201','凌浩风',
-> '女','2001/10/6','867','15978945645','tang@163.com '),
-> ('18137221508','赵临江',
-> '男','2000/2/13','789','12367823453','ping@163.com  '),
-> ('19111133071','崔依歌',
-> '女','2001/6/6','787','  15556845645','cui@126.com '),
```

1.8 表的数据操作

- 使用insert into语句添加数据

【例】利用replace into命令向表course中插入多行数据。

```
mysql> replace into course values
    -> ('c05103','电子技术','必修','64','16','2'),
    -> ('c05109','C语言','必修','48','16','2'),
    -> ('c05127','数据结构','必修','64','16','2'),
    -> ('c05138','软件工程','选修','48','8','5'),
    -> ('c06108','机械制图','必修','60','8','2'),
    -> ('c06127','机械设计','必修','64','8','3'),
    -> ('c06172','铸造工艺','选修','42', '16','6'),
    -> ('c08171','会计软件','选修','32','8','8');
```

1.8 表的数据操作

- 使用insert into语句添加数据

说明：

（1）使用insert语句可以向表中插入一行数据，也可以插入多行数据，最好一次插入多行数据，各行数据之间用“，”分隔。

（2）values子句：包含各列需要插入的数据清单，数据的顺序要与列的顺序相对应。若表名后不给出列名，则在values子句中要给出每一列（除identity和 timestamp类型的列）的值，如果列值为空，则值必须置为null，否则会出错。

（3）如果向表中添加已经存在的学号（已经设为主键）的记录，因此将出现主键冲突错误。例如，插入已经存在的学号19112111208记录，结果如下。

```
mysql> insert into student values('19112111208','韩小雨',
    -> '女','2001/2/14','666','15878945612','han@163.com ');
ERROR 1062 (23000): Duplicate entry '19112111208' for key 'primary'
```

(4) replace into向表中插入数据时，首先尝试插入数据到表中，如果发现表中已经有此行数据（根据主键或者唯一索引判断），则先删除此行数据，然后插入新的数据，否则，直接插入新数据。

(5) 还可以向表中插入其他表的数据，这也是成批插入数据的一种方式。但要求两个表要有相同的结构。具体操作将在以后介绍。其语法格式如下，：

```
insert into  table name1 select  * from table name2;
```

1.8 表的数据操作

表记录的插入

利用load data语句将数据装入数据库表中

【例】假设teacher表的数据已放在“d:\teacher.txt”中，现将teaching.txt的数据插入到teacher表中：

```
mysql>load data local infile "d:\\teacher.txt" into table teacher;
        Query OK, 9 rows affected, 8 warnings (0.03 sec)
        Records: 9  Deleted: 0  Skipped: 0  Warnings: 0
mysql> select *  from teacher;                    //输出表的记录
```

1.8 表的数据操作

说明：

(1) teacher.txt各行文本之间要用制表符<Tab>分隔，每行最后也加<Tab>分隔符。

(2) "d:\\teacher.txt":要用“\\”,表示斜线。

(3) 以此类推，可以将score表和teach_course表的数据载入。