第三讲 SQL基础

教学内容

- SQL简介
  - SQL语言分类
  - SQL数据类型
- 使用SQL管理数据库
  - Create Database
  - Alter Database
  - Drop Database
- 使用SQL管理数据表
  - Create Table
  - Alter Table
  - Drop Table
- 记录操作语句
  - Insert、Update、Delete

3.1 SQL简介

- SQL是结构化查询语言(Structured Query Language)的简称。
- SQL最早的是IBM的圣约瑟研究实验室为其关系数据库管理系统SYSTEM R开发的一种查询语言，它的前身是SQUARE语言。
- SQL语言结构简洁，功能强大，简单易学，自从IBM公司1981年推出以来，得到了广泛的应用。
- Oracle,Sybase,Informix,MySQL大型的数据库管理系统，Visual Foxpro,Access等单机数据库都支持SQL语言作为查询语言。

3.1.1 SQL语言分类

- 数据定义语言DDL(Data Definition Language)
  - CREATE、ALTER、DROP
- 数据操作语言DML(Data Manipulation Language)
  - INSERT、UPDATE、DELETE
- 数据控制语言DCL(Data Control Language)
  - GRANT、REVOKE
- 数据查询语言DQL(Data Query Language)
  - SELECT

3.2 使用SQL创建数据库

（1）创建用户数据库的最简形式

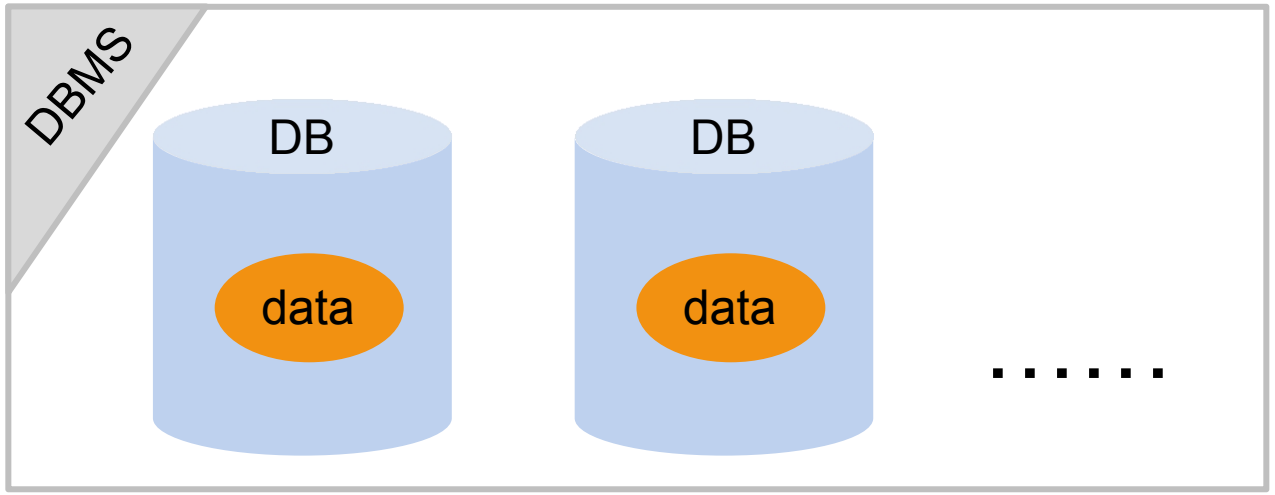
- 语法
  - **CREATE DATABASE database_name**
- 示例：创建数据库（**StudScore_DB2**）
  - **CREATE DATABASE StudScore_DB2**

使用SQL创建数据库

- 创建数据库

CREATE DATABASE 数据库名称 [库选项];

- 数据库名称：由字母、数字和下划线组成的任意字符串。
- 库选项：用于设置此数据库的相关特性，如字符集CHARSET。



DBMS：数据库管理系统　　DB：数据库　　data：数据

创建数据库

- 创建数据库
  - **CREATE DATABASE StudScore_DB; # UTF8**
- 创建指定字符集的数据库
  - **CREATE DATABASE StudScore_DB02 CHARACTER SET GBK;**
- 查看数据库字符集
  - **SHOW CHARACTER SET;**
  - **show variables like '%character%';**
- 不存在即创建数据库
  - **CREATE DATABASE IF NOT EXISTS StudScore_DB03 DEFAULT CHARSET GBK;**

数据库命名的规则

- 不能与其它数据库重名。
- 名称由任意字母、阿拉伯数字、下划线（_）和“$”组成，但不能使用单独的数字。
- 不能使用**MySQL**关键字作为数据库名、表名。
- **Windows**下数据库名、表名的不区分大小写、**Linux**下区分

修改数据库

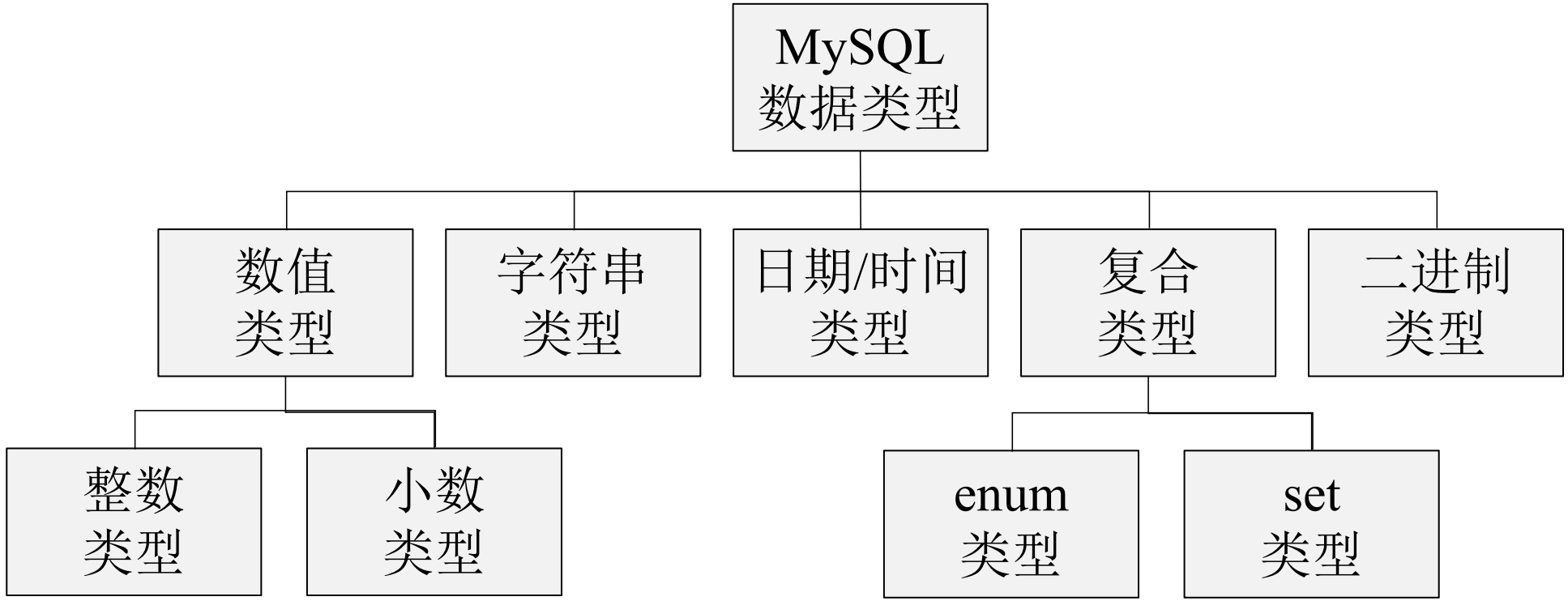
- 语法
  - **ALTER DATABASE [数据库名]**

    [DEFAULT] CHARACTER SET <字符集> | [DEFAULT] COLLATE <比较规则>;
- 注意
  - **[],<>,|** 在具体命令中不能出现

使用数据库

- 选择数据库
  - **USE 数据库名;**
  - **例：USE StudScore_DB03;**
- 删除数据库
  - **DROP DATABASE 数据库名;**
  - **例：DROP DATABASE StudScore_DB03;**

MySQL数据类型



整数类型

类型	大小	范围（有符号）	范围（无符号）	用途
TINYINT	1 字节	(-128，127)	(0，255)	小整数值
SMALLINT	2 字节	(-32 768，32 767)	(0，65 535)	大整数值
MEDIUMINT	3 字节	(-8 388 608，8 388 607)	(0，16 777 215)	大整数值
INT或INTEGER	4 字节	(-2 147 483 648，2 147 483 647)	(0，4 294 967 295)	大整数值
BIGINT	8 字节	(-9223372036854775808，9223372036854775 807)	(0，18 446 744 073 709 551 615)	极大整数值

age tinyint unsigned #非负

小数类型

类型	大小	范围（有符号）	范围（无符号）	用途
FLOAT	4 字节	-3.402 823 466 E+38 ~-1.175 494 351 E-38	0和1.175 494 351 E-38 ~3.402 823 466 E+38	单精度浮点数值
DOUBLE	8 字节	-1.797 693 134 862 315 7 E+308 ~-2.225 073 858 507 201 4 E-308)	0和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	DECIMAL (length, precision)	length决定小数的最大位数 precision用于设置小数位数	length决定小数的最大位数 precision用于设置小数位数	小数值

- **StudScore DECIMAL(5,2); #小数取值范围：-999.99～999.99**

日期/时间类型

类型	大小(字节)	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’	YYYY-MM-DD HH:MM:SS	混合日期和时间值，时间戳

字符串类型

类型	大小	用途
CHAR(n)	0-255字节	定长字符串
VARCHAR(n)	0-65535 字节	变长字符串
TINYTEXT	0-255字节	短文本字符串
TEXT	0-65 535字节	长文本数据
MEDIUMTEXT	0-16 777 215字节	中等长度文本数据
LONGTEXT	0-4 294 967 295字节	极大文本数据

- 字符串类型必须使用单引号，
- 如：**Teacher_name ='孙悟空'**

二进制类型

类型	大小	用途
Binary(n)	0-255字节	较短的二进制
VARBinary(n)	0-65535 字节	较长的二进制
Bit(n)	0-64字节	短二进制
TINYBLOB	0-255字节	较短的二进制
BLOB	0-65 535字节	图片、声音等文件
MEDIUMBLOB	0-16 777 215字节	图片、声音、视频等文件
LONGBLOB	0-4 294 967 295字节	图片、声音、视频等文件

复合类型

类型	最大值	说　明
Enum (“value1”, “value2”, …)	65535	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

- **Gender enum('男','女')**
- **Interesting set('唱歌','游泳','网球')**

3.3 使用SQL语句管理表

- 创建表
  - 创建简单表
  - 创建带Check约束的表
  - 创建带外键的表
  - 创建带复合主键的表
- 修改表
  - 添加列
  - 删除列
- 删除表
  - **DROP TABLE**

3.3.1 创建表

- **Create table语句语法简化形式**

```
Create table tablename
(
      column1 datatype [constraint],
      column2 datatype [constraint],
      …
      columnN datatype [constraint],
       [constraint Condition]
) Other options;
```

1.创建学生信息表(StudInfo)

字段名称	数据类型	字段长度	空值	PK	字段描述	举例
StudNo	Varchar	15		Y	学生学号	20191152070
StudName	Varchar	20			学生姓名	李明
StudGender	Char	2			学生性别	男
StudBirthDay	Date		Y		出生年月	2000-10-3
ClassID	Varchar	10			班级编号	20191152

```
Create Table StudInfo(
   StudNo VARCHAR(15) PRIMARY KEY COMMENT '学号',
   StudName VARCHAR(20) not null COMMENT '姓名',
   StudGender Char(2) not null COMMENT '性别',
   StudBirthDay Date null COMMENT '生日',
   ClassID VARCHAR(10) not null COMMENT '班级编号'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

字段的NULL值与DEFAULT值

- **NULL**
  - 表示字段的数据值未知或不可用。
  - 并不表示零（数字值或二进制值）、零长度的字符串或空白（字符值）。
- **Default**
  - 当数据表设计时某个字段设有默认值，在数据录入时，该字段若不输入，则以默认值来填充该字段。

创建带Check约束的学生信息表

- 约束性别为男或女

```
--Drop Table StudInfo
Create Table StudInfo(
   StudNo VARCHAR(15) PRIMARY KEY,
   StudName VARCHAR(20) not null,
   StudGender Char(2) DEFAULT '男'
         Check(StudGender IN('男', '女')),
   StudBirthDay Date null,
   ClassID VARCHAR(10) not null
);
```

创建带ENUM字段类型的学生信息表

- ENUM性别为男或女

```
--Drop Table StudInfo
Create Table StudInfo(
   StudNo VARCHAR(15) PRIMARY KEY,
   StudName VARCHAR(20) not null,
   StudGender ENUM('男', '女') not null
                DEFAULT '男',
   StudBirthDay Date null,
   ClassID VARCHAR(10) not null
);
```

2.班级信息表(ClassInfo)

字段名称	数据类型	字段长度	空值	PK	字段描述	举例
ClassID	Varchar	10		Y	班级编号	20191152
ClassName	Varchar	50			班级名称	计算机2019
ClassDesc	Varchar	100	Y		班级描述	计算机怎样

```
CREATE TABLE ClassInfo(
        ClassID varchar(10) primary key,
        ClassName varchar(50) not null,
        ClassDesc varchar(100) null
);
```

3.创建带关系即外键的学生信息表

```
--Drop Table StudInfo
CREATE TABLE StudInfo(
    StudNo varchar(15) primary key ,
    StudName varchar(20) NOT NULL ,
    StudGender char(2)
        Check(StudGender='男' or StudGender='女' ),
    StudBirthDay Date NULL ,
    ClassID varchar(10) not null ,
    Constraint FK_CID Foreign key(ClassID)
        references ClassInfo(ClassID)
);
```

4.创建学生成绩信息表(StudScoreInfo)

字段名称	数据类型	字段长度	约束	PK	字段描述	举例
StudNo	Varchar	15		Y	学生学号	20191152070
CourseID	Varchar	15		Y	课程编号	A0101
StudScore	Decimal	4,1	[0,100]		学生成绩	80.5

```
Create Table StudScoreInfo(
        StudNo varchar(15),
        CourseID varchar(15),
        StudScore Decimal(4,1) default 0
          Check(StudScore>=0 and StudScore<=100),
        Constraint PK_S_C Primary key (StudNo,CourseID)
);
```

5.创建自增字段数据表

```
CREATE TABLE Test(
    AID int not null auto_increment,
    AName varchar(30) not null,
    primary key (AID)
);

INSERT INTO TEST (AName)  values ('孙悟空'), ('猪八戒');
INSERT INTO TEST (AName)  values ('唐  僧'), ('沙和尚');
```

查看数据表结构

- 显示表结构：
  - **DESCRIBE/DESC** 表名
  - **SHOW CREATE TABLE** 表名 查看表的详细信息
- 查看所有数据表
  - **show tables;**

3.3.2 修改数据表

- **ALTER TABLE 表名 alter_spec [, alter_spec ...];**
- **Alter_spec:**

ADD [COLUMN] col_name column_definition [FIRST | AFTER col_name] --添加新字段
| ADD [CONSTRAINT [symbol]] PRIMARY KEY (index_col_name,...) --添加主键
| ADD [CONSTRAINT [symbol]]UNIQUE (index_col_name,...) --添加唯一索引
| ADD [CONSTRAINT [symbol]] FOREIGN KEY (index_col_name,...)
reference (index_col_name,...) --添加外键
| ALTER [COLUMN] col_name {SET DEFAULT literal | DROP DEFAULT} --修改默认值
| CHANGE [COLUMN] old_col_name new_col_name column_definition --修改字段名及数据类型
| MODIFY [COLUMN] col_name column_definition --修改字段类型
| DROP [COLUMN] col_name --删除字段
| DROP PRIMARY KEY --删除主键
| DROP INDEX index_name --删除索引名称
| DROP FOREIGN KEY fk_symbol --删除外键
| table_options --更改表的其它选项

ALTER TABLE语句允许对表进行多个修改操作，其子句间用逗号分隔。

修改数据表

1.修改表名

- **rename table 旧表名 to 新表名**
- **alter table 旧表名 rename 新表名**

2.增加新列

- 语法
  - **ALTER TABLE table_name ADD column_name datatype [first | after 旧字段名];**
- 示例
  - **Alter Table StudScoreInfo Add UniqueCol varchar(20) null;**

3. 删除列

- 语法
  - **ALTER TABLE table_name DROP column_name;**
- 示例
  - **Alter Table StudScoreInfo Drop UniqueCol;**

修改数据表 cont.

4.修改字段名及数据类型

- 语法
  - **alter table 表名 change 旧字段名 新字段名 数据类型;**
- 示例
  - **alter table StudScoreInfo change UniqueCol NewUniqueCol decimal(5,2);**

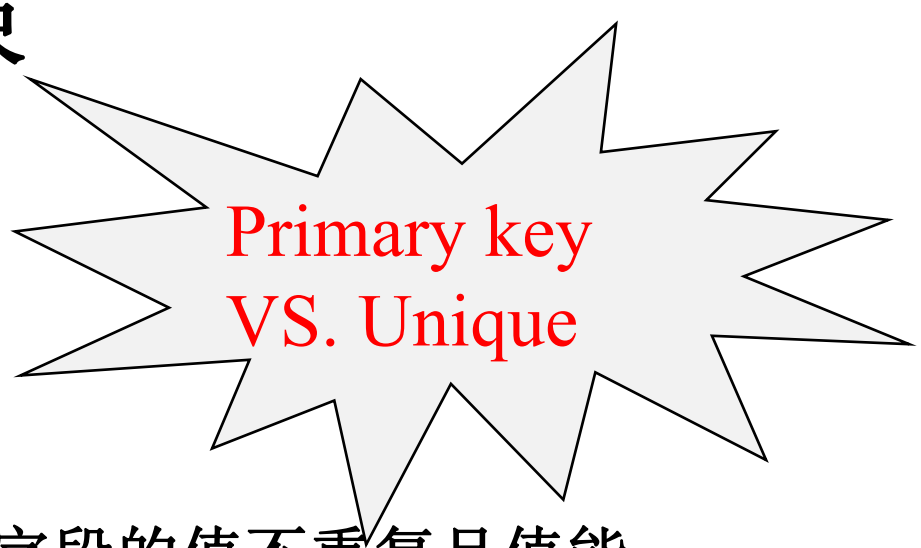
5.修改字段数据类型

- 语法
  - **alter table 表名 modify 旧字段名 新数据类型;**
- 示例
  - **alter table StudScoreInfo modify NewUniqueCol date;**

数据完整性

- 三种数据完整性约束
  - 实体完整性：通过主键约束或者唯一性约束实现
  - 参照完整性：通过外键约束实现
  - 用户自定义的完整性：通过非空约束、检查约束和默认值约束实现
- 添加主键、外键、唯一性约束条件的语法格式：
  - **ALTER TABLE** 表名 **ADD [CONSTRAINT [**约束名**]]** 约束类型(字段名)

主键/唯一性约束



- 添加主键
  - **alter table 表名 add primary key (字段名);**
- 删除主键
  - **alter table 表名 drop PRIMARY KEY;**
- 唯一性约束（**unique**）：用于保证表中某个字段的值不重复且值能为空（ **null**），一个表可以定义多个唯一性约束。
- 添加唯一性约束
  - **alter table 表名 add [constraint [约束名]] unique (字段名);**
- 删除唯一性约束
  - **alter table 表名 drop index 唯一索引名;**

添加或删除外键约束

- 添加外键
  - **ALTER TABLE 表名 ADD CONSTRAINT 外键名 foreign KEY(外键字段) references 关联表名(关联字段) [on delete 级联选项][on update 级联选项;**
- 删除外键
  - **alter table 表名 drop foreign Key 约束名**

外键约束级联选项

参数名称	功能描述
cascade	父表记录的删除(delete)或修改(update)操作，会自动删除或修改子表中与之对应的记录
set null	父表记录的删除(delete)或修改(update)操作，会将子表中与之对应记录的外键自动设置为null值
no action	父表记录的删除(delete)或修改(update)操作，如果子表存在与之对应的记录，那么删除或修改操作将失败
restrict	与no action功能相同，是默认设置，也是最安全的设置

非空/默认值约束

- 添加非空约束
  - **alter table 表名 modify 字段名 数据类型 not null;**
- 取消非空约束
  - **alter table 表名 modify 字段名 数据类型 null;**
- 添加默认值约束
  - **alter table 表名 alter 字段名 set default 默认值;**
- 删除默认值约束
  - **alter table 表名 alter 字段名 drop default;**

修改表的其它选项

- alter table 表名 engine=新的存储引擎类型
- alter table 表名 default charset=新的字符集
- alter table 表名 auto_increment=新的初始值
- alter table 表名 pack_keys=新的压缩类型

- 注意：pack_keys仅限于MyISAM

关键字说明

- PRIMARY KEY用于定义主键。也可以使用多字段来定义主键。
- COMMENT注释该字段的含义。
- NOT NULL是非空约束。
- DEFAULT为该字段加默认值，可以减少输入次数。
- AUTO_INCREMENT为自增型属性，一般用作主键，数值会自动加1
- 当表中有外键字段时用CONSTRAINT设置外键。
- ENGINE=InnoDB是设置该表的存储引擎，DEFAULT CHARSET=utf8是设置该表的默认字符集。

3.3.3 删除数据表

- 语法
  - **DROP TABLE table_name**
- 示例
  - **Drop Table StudScoreInfo**
- 注意
  - 先删除外键表，再删除主键表，即先删子表，再删父表。

3.4 使用SQL语句维护数据

- 添加记录
  - Insert
- 修改记录
  - Update
- 删除记录
  - Delete
  - Truncate Table

添加记录（Insert）

- INSERT语句用于向数据表中添加记录。
- 语法
  - INSERT [INTO] tablename [(column { ,column})]
  - VALUES (columnvalue [{,columnvalue}]);
  - INSERT [INTO] tablename [(columnname list)]
  - VALUES (valuelist1),… (valuelistN);
- 示例：向学生成绩表添加记录

```
Insert Into StudScoreInfo
   (StudNo,CourseID,StudScore)
Values
  ('20191152070','A0101',80.5)
```

特殊字符序列转义

- “\”开头，且字符序列大小写敏感
- 如：**O'Neil**

MySQL中的特殊字符序列	转义后的字符
\”	双引号（“）
\’	单引号（‘）
\\	反斜线（\）
\n	换行符
\r	回车符
\t	制表符
\0	ASCII 0(NUL)
\b	退格符
_	_
\%	%

更新记录（update）

- update语句更新或修改满足给定条件的记录
- 格式
  - update tablename
  - set column1 = value1 [,column2 = value2...]
  - [WHERE {search_condition}];
- 示例:更新学号为20191152004的姓名为李丽，性别为女

```
Update StudInfo
Set StudName= '李丽',StudGender= '女'
Where StudNo= '20191152004';
```

- 注意
  - 使用UPDATE语句时，关键是要设定好where子句。

删除记录（delete）

- DELETE删除数据库表格中的行或记录。
- 格式
  - DELETE FROM tablename
  - [WHERE {search_condition}]
- 示例：删除学号为20191152004学生信息

```
DELETE FROM StudInfo
WHERE StudNo='20191152004';
```

- 注意
  - where从句指定删除记录的判断条件。
  - 不加where从句，删除表格中的所有记录。

TRUNCATE TABLE命令

- **TRUNCATE TABLE命令语法如下：**
  - **TRUNCATE [TABLE] table_name;**
  - 注：不能清空父表
- 删除学生成绩表(StudScoreInfo)所有记录。
  - **TRUNCATE TABLE StudScoreInfo;**

记录操作语句简化形式

- （1）添加新记录Insert

```
INSERT [INTO] TableName
  [(FieldsList)]
VALUES
  (ValuesList);
```

- （2）更新记录update

```
UPDATE TableName
SET FieldName1=Values1,FieldName2=Values2
WHERE FieldName=Condition;
```

- （3）删除记录delete

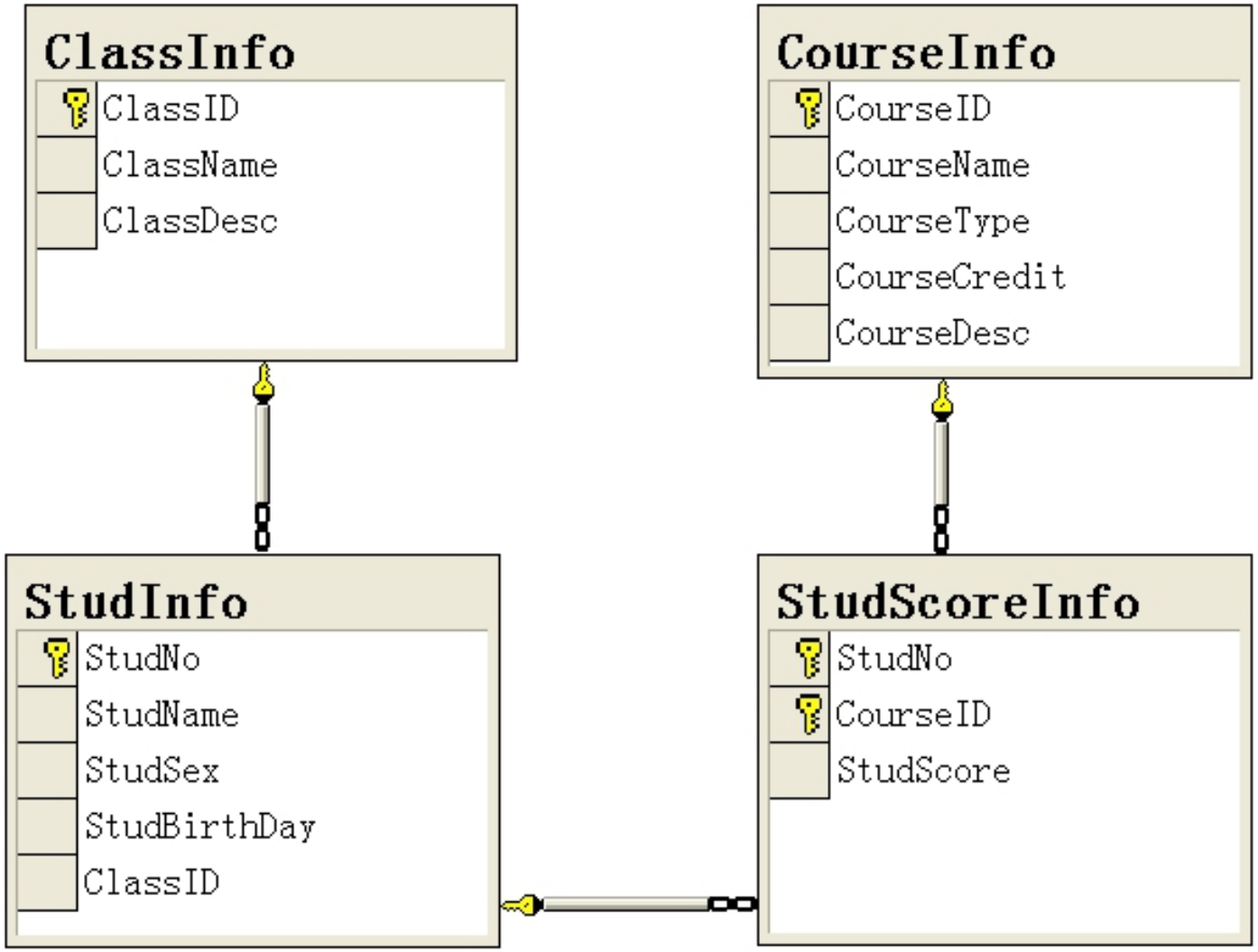
```
DELETE [FROM] tablename
WHERE fieldname=condition;
```

创建课程信息表(CourseInfo)

字段名称	数据类型	字段长度	是否为空	PK	字段描述	举例
CourseID	Varchar	15		Y	课程编号	A0101
CourseName	Varchar	50			课程名称	MySQL
CourseType	Varchar	10			课程类别	C
CourseCredit	Decimal	3,1			课程学分	2.5
CourseDesc	Varchar	100	Y		课程描述	MySQL

```
Create Table CourseInfo(
        CourseID varchar(15) Primary key,
        CourseName varchar(50) not null,
        CourseType varchar(10) not null,
        CourseCredit Decimal(3,1) not null,
        CourseDesc varchar(100)
);
```

示例数据库数据表间关系图



下次课内容

- SELECT select_list
- FROM table_list
- [WHERE search_conditions]
- [GROUP BY group_by_list]
- [HAVING search_conditions]
- [ORDER BY order_list [ASC | DESC]]
- [LIMIT *[offset,]* count]