

本文档对课程内常见 SQL 语法作出整理。

基本规则：

1. SQL 不区分大小写；
2. SQL 语句末尾以分号结束，缩进不影响结果；
3. SQL 注释以两个减号开头，例如「-- 注释内容」。

## 目录

<b>1 数据定义</b>	<b>2</b>
1.1 Create 创建	2
1.2 Drop 删除	3
1.3 Alter 修改	4
<b>2 数据操纵</b>	<b>4</b>
2.1 Insert 数据插入	4
2.2 Delete 数据删除	5
2.3 Update 数据修改	5
<b>3 数据查询</b>	<b>5</b>
3.1 Select 查询语句	5
3.2 条件的千变万化	8
<b>4 数据控制</b>	<b>8</b>
4.1 Grant 授予权限	8
4.2 Revoke 撤回权限	9
<b>5 约束完整性命名</b>	<b>9</b>
<b>6 触发器</b>	<b>10</b>
<b>7 事务操作</b>	<b>11</b>

# 1 数据定义

数据定义语句包括 CREATE、DROP、ALTER 语句，一般为对数据库、表、视图等宏观概念的定义。

## 1.1 Create 创建

**创建表** 以下语句用于创建表。

```
1 Create Table 表名(  
2     列名 数据类型 列级完整性约束,  
3     列名 数据类型 列级完整性约束,  
4     ...  
5     列名 数据类型 列级完整性约束,  
6 表级完整性约束);
```

```
1 Create Table Course(  
2     Cno Char(4) Primary Key,  
3     Cname Char(40),  
4     Cpno Char(4),  
5     Ccredit Smallint,  
6     Foreign Key (Cpno) References Course(Cno)  
7 );
```

- Primary Key 表示主键约束；
- Foreign Key 表示外键约束，其中的 References 指向对应表的对应主属性。

**创建模式** 以下语句用于创建模式，较少使用。

```
1 Create Schema 模式名 Authorization 用户名;
```

**创建视图** 以下语句用于创建视图。

```
1 Create View 视图名(列, 列, ..., 列) As  
2     子查询  
3 [With Check Option];
```

With Check Option 若被使用，则在视图增、删、改元组时会自动加上子查询中的条件。

```
1 Create View CS_Student_View(Sno, Sname, Sage) As
2     Select Sno, Sname, Sage From Student Where Sdept = 'CS'
3 With Check Option;
```

建立计算机系学生的学号、姓名、年龄视图，其中对视图增、删、改元组时会自动加上 Sdept = 'CS' 的条件。

**创建索引** 以下语句用于创建索引。

```
1 Create [Unique] [Cluster] Index 索引名
2 On 表名(列 [次序], 列 [次序], ..., 列 [次序])
```

```
1 Create Unique Index Student_Index
2 On Student(Sno Desc);
```

对学生表按照学号列降序建立一个唯一索引。

## 1.2 Drop 删除

**删除表** 以下语句用于删除表。Restrict 表示限制条件，Cascade 表示级联删除，默认 Restrict（下同）。

```
1 Drop Table 表名 [Restrict/Cascade];
```

```
1 Drop Table Student Cascade;
```

**删除模式** 以下语句用于删除模式。删除模式会将其定义的表也删除。

```
1 Drop Schema 模式名 [Restrict/Cascade];
```

**删除视图** 以下语句用于删除视图。

```
1 Drop View 视图名 [Restrict/Cascade];
```

**删除索引** 以下语句用于删除索引。

```
1 Drop Index 索引名;
```

## 1.3 Alter 修改

**修改表** 以下语句用于修改表中列。几个语句之间是独立的。

```
1 Alter Table 表名 Add Column 列名 数据类型 列级完整性约束;  
2 Alter Table 表名 Drop Column 列名 [Cascade/Restrict];  
3 Alter Table 表名 Alter Column 列名 数据类型 ;
```

```
1 Alter Table Student Add Column S_entrance Date Not Null; -- 新增入学时间列  
2 Alter Table Student Drop Column Sage Cascade; -- 级联删除年龄列  
3 Alter Table SC Alter Column Grade Int; -- 修改成绩列
```

以下语句用于修改表中的完整性约束。

```
1 Alter Table 表名 Add 表级完整性约束;  
2 Alter Table 表名 Drop Constraint 完整性约束名 [Cascade/Restrict];
```

```
1 Alter Table Student Add Unique(Sname); -- 增加姓名唯一性  
2 Alter Table Student Drop Constraint sp_ibfk_1; -- 删除某个特定约束
```

**修改索引** 以下语句用来对索引重命名。

```
1 Alter Index 旧索引名 Rename To 新索引名;
```

## 2 数据操纵

数据操纵语句包括 INSERT、DELETE、UPDATE 语句，一般为对元组等微观概念的操纵。

### 2.1 Insert 数据插入

**插入元组** 以下语句用于插入元组。数据的顺序需要与列的顺序一致（而列的顺序可以重排）。

```
1 Insert Into 表名(列, 列, ..., 列)  
2 Values (数据, 数据, ..., 数据);
```

```
1 Insert Into Student(Sno, Sname, Sage, Ssex, Sdept) Values  
2 ("22001", "小明", 20, "男", "CS"),  
3 ("22002", "小红", 19, "女", "IS"),  
4 ("22003", "小刚", 21, "男", "AI");
```

**插入子查询结果** 以下语句通过子查询结果用于插入元组。较少使用。

```
1 Insert Into 表名(列, 列, ..., 列)
2     子查询;
```

```
1 Insert Into Student2(Sno, Sname, Sage)
2     Select Sno, Sname, Sage From Student Where Sdept = "CS";
```

## 2.2 Delete 数据删除

```
1 Delete From 表名 Where 条件;
```

```
1 Delete From SC Where Sno In (
2     Select Sno From Student Where Sdept = 'IS' );
```

删除信息系学生的选课记录。

## 2.3 Update 数据修改

Update 可以用于表中数据修改，也可以用于视图中数据修改。

```
1 Update 表名/视图名
2 Set 列名=表达式, 列名=表达式, ..., 列名=表达式
3 Where 条件;
```

```
1 Update Student
2 Set Sage=Sage+2, Sdept="CS"
3 Where Sno="22001";
```

# 3 数据查询

## 3.1 Select 查询语句

数据查询语句也即 SELECT 语句，用途十分广泛，复习时以教师课件为准。

### 版本 1 （基础款）

```
1 Select 目标列
2 From 表名/视图名
3 Where 条件;
```

```
1 Select * From Student Where Sage>=20;
```

查询年龄大于等于 20 岁的学生的全部信息。

## 版本 2 （带排序）

```
1 Select 目标列  
2 From 表名/视图名  
3 Where 条件  
4 Order By 排序属性 [Asc/Desc];
```

```
1 Select * From Student Where Sdept="CS" Order By Sage Desc;
```

查找计算机系学生的全部信息，按照年龄降序排列。

## 版本 3 （指定去重）

```
1 Select [All/Distinct] 目标列  
2 From 表名/视图名  
3 Where 条件;
```

```
1 Select All Sno From SC Where Grade>=60;
```

在选课表中查询及格学生的学号，同一学号输出次数取决于及格科目数。如果把其中的 All 换成 Distinct，那么同一学号仅输出一次。

## 版本 4 （带 Group by 子句）

```
1 Select 目标列  
2 From 表名/视图名  
3 Where 全局条件  
4 Group By 列名  
5 Having 组内条件（可能采用聚集函数）;
```

```
1 Select Sno From SC Where Grade>=60  
2 Group By Sno Having Count(*)>=3;
```

查询及格课程数大于等于 3 门的学生的学号。此查询的过程是：

1. 先在 SC 表中选出  $\text{Grade} \geq 60$  的元组，排除不及格元组；
2. 再对这些元组按照 Sno 进行分组，相同 Sno 的分成一组；

3. 随后将一组内记录条数  $\geq 3$  条的组号 (即 Sno) 筛选出来, 返回结果。

```
1 Select Sno,Count(*),Avg(Grade) From SC Where Grade>=60
2 Group By Sno Having Avg(Grade)<80;
```

查询均分小于 80 分的学生的学号, 仅统计及格科目。此查询的过程是:

1. 先在 SC 表中选出  $\text{Grade} \geq 60$  的元组, 排除不及格元组;
2. 再对这些元组按照 Sno 进行分组, 相同 Sno 的分成一组;
3. 随后将一组内平均分小于 80 分的组号 (即 Sno), 以及及格科目数和平均分筛选出来, 返回结果。
4. 需要注意的是: 计算平均分之前, 就已经把不及格的科目排除在外了。

## 版本 5 (外连接)

```
1 Select 目标列 From 左表
2 Left/Right Outer Join 右表 On(连接约定)
3 Where 条件;
```

```
1 Select Student.Sno, Sname, Ssex, Sage, Sdept, Cno, Grade From Student
2 Left Outer Join SC On(Student.Sno = SC.Sno)
3 Where Sdept='CS';
```

将 Student 表的学号、姓名、性别、年龄、院系与 SC 表的课程号、成绩连接起来, 使用左外连接 (即使 Student 表中有的元组在 SC 表没有任何值, 也必须保留, 这时的 Cno 和 Grade 字段返回 NULL), 且仅限于计算机专业。

## 版本 6 (带别名)

```
1 Select 目标列 别名 (或不止一个)
2 From 表名 别名, 表名 别名, ..., 表名 别名
3 Where 条件;
```

```
1 Select S1.Sno "学号", S1.Sname "姓名", S1.Sage "年龄"
2 From Student S1, Student S2
3 Where S1.Sdept = S2.Sdept And S2.Sname = "刘晨";
```

查询与刘晨在同一个系学习的学生, 这里的 S1,S2 即为 Student 的别名, 「学号」、「姓名」和「年龄」是查询返回结果的别名。这个例子其实是自身连接。

## 版本 7 （集合查询）

```
1 子查询甲
2 Intersect/Union/Except
3 子查询乙；
```

```
1 Select * From Student Where Sage=20
2 Intersect
3 Select * From Student Where Sdept="CS";
```

查询计算机系的 20 岁学生的全部信息。

## 3.2 条件的千变万化

条件语句是千变万化的，其中可能的变体有：

- 比较：= > < >= <= != <>，可能添加 Not
- 确定范围：(Not) Between ... And ...
- 确定集合：(Not) In
- 字符匹配：(Not) Like，匹配字符串中 % 代表任意个字符（包括 0 个），\_ 代表 1 个字符。可能使用 Escape 转码字符。
- 空值：Null
- 逻辑：And、Or、Not
- 量词：All、Any
- 存在谓词：(Not) Exists

# 4 数据控制

数据控制语句包括 GRANT、REVOKE 语句，一般为对用户数据库权限的管理。默认情况下，作出数据控制的角色是数据库管理员（DBA）或者其他有权限操作的用户。

## 4.1 Grant 授予权限

```
1 Grant 权限（或为 All Privileges）
2 On 对象类型 对象名
3 To 用户（或为 Public）
4 [With Grant Option];
```



```
1 Grant Select, Update(Sno)
2 On Table SC
3 To User1
4 With Grant Option;
```

将对选课表的查询和修改学号权限授予用户 1，且允许其再授予二级权限。

## 4.2 Revoke 撤回权限

```
1 Revoke 权限（或为 All Privileges）
2 On 对象类型 对象名
3 From 用户（或为 Public）[Cascade/Restrict]
```

```
1 Revoke All Privileges
2 On Table SC
3 From User1;
```

撤回用户 1 对选课表的全部权限。注意 Revoke 语句用介词 From，而 Grant 语句用介词 To。

## 5 约束完整性命名

Constraint（约束完整性）是一种数据对象，一般在插入或删除操作中使用到。本节相当于给定义的约束完整性显式命名，其格式为

```
1 Constraint 约束条件名 约束条件内容
```

其中约束条件内容包括 Not Null、Unique、Primary Key、Foreign Key、Check 等。

**插入操作的完整性命名 举个例子：**

```
1 Create Table Student(
2     Sno Numeric(6)
3     Constraint C1 Check (Sno Between 90000 And 99999),
4     Sname Char(20)
5     Constraint C2 Not Null,
6     Sage Smallint
7     Constraint C3 Check (Sage>0 And Sage<30),
```

```

8      Ssex Char(2)
9          Constraint C4 Check (Ssex in ("男","女")),
10         Constraint C5 Primary Key(Sno) -- 表级完整性约束
11 );

```

**修改操作的完整性命名** 举两个例子：

```

1 Alter Table Student Drop Constraint C2; -- 删除 C2 即「名字非空」完整性
2 Alter Table Student Add Constraint C2 Not Null; -- 再加回来

```

## 6 触发器

触发器也称为「事件—条件—动作」规则。

**创建触发器** 在一个表定义触发器的格式如下：

```

1 Create Trigger 触发器名
2 Before/After 触发事件 On 表名
3 Referencing New/Old Row As 变量
4 For Each Row/Statement
5 When(触发条件)
6     触发动作体

```

其含义为：

1. 如果在某个表上发生了某种触发事件（Update、Insert 等），那么在触发事件之前（或之后），需要执行触发动作体；
2. 显式定义旧元组与新元组为某一变量，以便触发动作体使用该变量；
3. 在触发时，对于每一行（或每条语句），若触发条件为真，则执行一次触发动作体。

```

1 Create Trigger Salary_Trigger
2 Before Insert Or Update On Teacher
3 Referencing New Row As NewTuple
4 For Each Row
5 Begin
6     If NewTuple.Job="教授" And NewTuple.Sal<4000 Then
7         NewTuple.Sal=4000;
8     End If;

```

```
9 End;
```

该触发器会在插入或更新教师表之前生效，如果教授的工资低于 4000 元则自动转换为 4000 元。这个例子仅供演示，MySQL 不直接支持这种写法。

**删除触发器** 删除一个触发器用 Drop 即可实现。

```
1 Drop Trigger 触发器名 On 表名;
```

## 7 事务操作

事务是数据库的不可分割的基本单位，具有原子性、一致性、隔离性、持续性。事务定义语句一般如下：

```
1 Begin Transaction;  
2 若干条语句;  
3 Commit/Rollback;
```

在 MySQL 中，开头语句为 Start Transaction;。

```
1 Start Transaction;  
2 Update Student Set Sage=Sage-1 Where Sdept='CS';  
3 Select * From Student;  
4 Rollback;  
5 Select * From Student;
```

对计算机系学生的年龄减 1，然后回退。回退以后的数据还原至事务开始前的状态，相当于在回退前对计算机系学生的年龄加 1。这是 MySQL 语法，开头语句用 Begin 还是 Start 随数据库语法而确定。