数据定义

本章以学生-课程数据库为例来介绍SQL的数据定义、数据查询和数据操纵语句。本节主要讲解对表结构和索引结构的定义。

学生-课程数据库

学生-课程数据库中包括以下三个表：

- 学生表：Student (Sno, Sname, Age, Gender, Dept)；
- 课程表：Course (Cno, Cname, Credit)；
- 学生选课表：SC (Sno,Cno,Grade)；

表中的主码以下划线进行表示。各个表中的数据示例如图5.2所示

学生 Student

学号 Sno	姓名 Sname	性别 Gender	年龄 Age	系 Dept
2022001	沐辰	男	19	计算机
2022123	浩宇	男	18	计算机
2022191	若汐	女	18	数学
2022267	依诺	女	19	金融

课程 Course

课程号 Cno	课程名 Cname	学分 Credit
1	高数	4
2	C语言	3
3	数据库	4

选课 SC

学号 Sno	课程号 Cno	成绩 Grade
2022001	1	92
2022001	3	85
2022191	2	88
2022191	3	90

图 5.2 学生-课程数据库的数据示例

表的定义、修改与删除

（1） 定义表

使用学生-课程数据库之前，首先需要定义数据库中的基本表，SQL语言使用**CREATE TABLE**语句来定义表结构，其基本格式如下：

```
CREATE TABLE <表名> ( <列名><数据类型> [列级完整性约束条件]
                    [,<列名><数据类型> [列级完整性约束条件]]
                    ........
                    [,<表级完整性约束条件>] )
```

建表时需要定义表名、所有的属性列名和列的数据类型，同时还可以定义与列或者与表有关的完整性约束条件。SQL标准支持了多种数据类型，表5.1列出了几种常用数据类型。要注意，不同的关系数据库管理系统中支持的数据类型不完全相同。如果完整性约束条件涉及多个属性列，则必须定义在表级上，否则既可以定义在列级也可以定义在表级。

表的定义会被存入系统的数据字典中。数据字典其实是关系数据库管理系统中的系统表，它记录了数据库中所有的定义信息，包括表结构、完整性约束，索引，视图，用户的操作权限以及统计信息等。当用户操作表中数据时，关系数据库管理系统会根据数据字典中的内容自动检查该操作是否违背了相关定义。

下例中给出了学生表、课程表和学生选课表的定义：

> [例5.1] 创建学生表Student。
> CREATE TABLE Student
> (Sno CHAR(9) PRIMARY KEY, /*列级完整性约束条件，Sno是主码*/
> Sname CHAR(20) NOT NULL, /*列级完整性约束条件，Sname不能取空值*/
> Age INT,
> Gender CHAR(2),
> Dept CHAR(10)
>);

> [例5.2] 创建课程表Course。
> CREATE TABLE Course
> (Cno CHAR(4) PRIMARY KEY, /*列级完整性约束条件，Cno是主码*/
> Cname CHAR(40) UNIQUE, /*列级完整性约束条件，Cname取唯一值*/
> Credit SMALLINT
>);

> [例5.3] 创建学生选课表SC。
> CREATE TABLE SC
> (Sno CHAR(9),
> Sno CHAR(4),
> Grade SMALLINT,
> PRIMARY KEY(Sno,Cno), /*表级完整性约束条件，Sno和Cno共同构成主码*/
> FOREIGN KEY(Sno) REFERENCES Student(Sno), /*表级完整性约束条件，Sno是外码，被参照表是Student*/
> FOREIGN KEY(Cno) REFERENCES Course(Cno) /*表级完整性约束条件，Cno是外码，被参照表是Course*/
>);

学生表和课程表的主码约束只涉及一个属性，所以可以定义在列级上，而学生选课表的主码约束涉及两个属性，所以必须定义在表级上。外码约束只有表级约束，它用于实现表之间的参照完整性。上例中的外码约束表示，学生选课表的学号依赖于学生表的学号，学生选课表的课程号依赖于课程表的课程号，这符合现实意义，即选课的学生一定来自于学生表中的学生，学生选修的课程一定来自于课程表。

（2） 修改表

随着应用环境和应用需求的变化，有时需要修改已建立好的基本表。SQL语言使用**ALTER TABLE**语句修改基本表，其基本格式如下：

```
ALTER TABLE <表名>
[ ADD [COLUMN] <新列名><数据类型> [完整性约束] ]
[ ADD <表级完整性约束> ]
[ DROP [COLUMN] <列名> [CASCADE | RESTRICT] ]
[ DROP CONSTRAINT <完整性约束名> [CASCADE | RESTRICT] ]
[ ALTER COLUMN <列名><数据类型> ];
```

修改基本表时可以增加新列、新的列级完整性约束条件，新增表级完整性约束条件，删除表中的列，删除完整性约束条件和修改原有列的定义。删除列和完整性约束条件时可以添加关键字CASCADE和RESTRICT。CASCADE表示级联删除，即删除列和约束条件的同时删除引用该列和该约束条件的其他对象；RESTRICT则指如果该列或该约束条件被其他对象引用，则不能删除。

虽然关系数据库管理系统提供了修改基本表的功能，但是在应用程序开发中并不提倡使用这些操作。因为对于关系数据库而言，一旦允许程序访问数据库中的数据，进行交互时，它的假设前提就是数据的组织形式或者表的模式已经定义好并且已具有完整的约束条件。

下面给出了修改表的一些例子：

> [例5.4] 给学生表新增“入学时间”列，数据类型为日期型，列约束条件为非空。
> ALTER TABLE Student ADD Entrance DATE NOT NULL;

> [例5.5] 将学生表中性别的数据类型由字符串改为整数。
> ALTER TABLE Student ALTER COLUMN Gender INT;

（3） 删除表

当不再需要某个基本表时，SQL语言使用DROP TABLE语句删除基本表，其基本格式如下：

```
DROP TABLE <表名> [CASCADE | RESTRICT] ;
```

其中，CASCADE关键字指级联删除，即删除基本表的同时删除基本表的相关依赖对象，如索引、触发器，有的关系数据库管理系统还会同时删除视图。读者可以查阅使用产品的用户手册，了解具体的删除策略；RESTRICT表示如果预删除的基本表被其他表的约束所引用（如，FOREIGN KEY）或者基本表有视图、触发器、存储过程和函数时，则基本表不能被删除。默认情况下设置为RESTRICT。

下例给出了删除学生表的SQL语句：

> [例5.6] 删除学生表。
> DROP TABLE Student CASCADE;

基本表被删除时，表的定义以及表中的数据都将一起被删除。由于学生选课表SC通过外码Sno引用了学生表Student，因此Student表被删除的同时SC也将被级联删除。

索引的定义与删除

同文档数据库管理系统一样，关系数据库管理系统也提供了索引功能来加快数据的查询。用户可以根据应用需求在基本表上建立一个或者多个索引。常用的关系数据库索引类型有顺序文件上的索引、B+树索引、散列索引和位图索引等。当使用索引功能查询数据时，关系数据库管理系统会自动选择合适的索引类型来实现快速查找。

（1） 建立索引

SQL语言使用CREATE INDEX语句来创建索引，其基本格式如下：

```
CREATE [UNIQUE] [CLUSTER] INDEX <索引名>
ON <表名> ( <列名> [ASC | DESC]
```

```
        [, <列名> [ASC | DESC]] ....);
```

索引可以建立在表的一列或者多列上，各列名之间用逗号隔开。创建索引时，还可以指定每个列索引值的排列次序，ASC表示升序排列，DESC表示降序排列，默认情况下为ASC。关键字UNIQUE表示建立的索引为唯一索引，即索引的每一个索引值只对应唯一的元组。CLUSTER表示建立的索引为聚簇索引，即索引中的索引值与对应的元组存放在连续的物理块中。通常，一个基本表只能拥有一个聚簇索引。关系数据库系统会默认为基本表的主键创建索引。创建的索引定义会被写入数据字典中。

下例给出了在学生表、课程表和学生选课表上创建索引的定义：

> [例5.7] 在学生表上按学号升序建唯一索引。
> CREATE UNIQUE INDEX Stusno ON Student(Sno);
> [例5.8] 在课程表上按课程号升序建唯一索引。
> CREATE UNIQUE INDEX Coucno ON Course(Cno);
> [例5.9] 在学生选课表上按学号升序和课程号降序建唯一索引。
> CREATE UNIQUE INDEX SCno ON SC(Sno ASC, Cno DESC);

（2）修改索引

对已经建立的索引，SQL语言使用ALTER INDEX语句来进行修改，其基本格式如下：

```
ALTER INDEX <旧索引名>  RENAME TO <新索引名>;
```

> [例5.10] 将学生选课表的SCno索引名改为SCSno。
> ALTER INDEX SCno RENAME TO SCSno;

（3）删除索引

SQL语言使用DROP INDEX来删除不必要的索引，其基本格式如下：

```
DROP INDEX <索引名> ;
```

> [例5.11] 删除学生表的Stusno索引。
> DROP INDEX Stusno;

删除索引时，数据库管理系统会同时将索引的定义从数据字典中删除。