

第3章

chapter 3

SQL 语言

学习目标

结构化查询语言(structured query language, SQL)是关系数据库的标准语言,本章主要讲授 SQL 语言在数据库中的应用。目前,几乎所有的关系型数据库管理系统,如 Oracle、Sybase、Microsoft SQL Server 和 Access 等均采用 SQL 语言标准。因此,本章的教学目标主要有两个,一是要求读者掌握对数据库的基本操作,并了解数据库管理系统的基本功能;二是要求读者熟练掌握 SQL 查询语句,并运用 SQL 语句完成对数据库的操作。

学习方法

本章重在实验,因此要求读者结合课堂讲授的知识,强化上机实训,把课堂上学过的有关概念进行实训,以便加深理解,达到学习目标。

学习指南

本章的重点是 3.2 节、3.3 节、3.4 节、3.5 节、3.8 节和 3.9 节,难点是 3.3 节、3.4 节和 3.5 节。

本章导读

建议围绕下列问题进行学习:

- (1) SQL 查询语句对关系代数运算做了哪些扩展?
- (2) 连接查询包括哪些? 它们分别用于什么地方?
- (3) 在使用分组聚合查询时需要注意什么?
- (4) 相关子查询与非相关子查询的概念是什么?
- (5) 如何理解存在量词以及存在量词在 SQL 查询中的重要地位?
- (6) 如何理解查询表的概念? 查询表与子查询有何异同点?
- (7) 在 SQL 的 DDL 语句中如何实现完整性约束? 实现完整性约束的方法有哪些?
- (8) 如何将数据库的对象(如基本表、索引)建立在特定的物理文件上?
- (9) 将数据库对象建立在特定的物理文件上,其目的是什么?
- (10) 建立视图的目的是什么? 视图主要用于什么操作?

3.1 SQL 概述

SQL 语言于 1974 年由 Boyce 等提出,并于 1975—1979 年在 IBM 公司研制的 System R 数据库管理系统上实现,现已成为国际标准。

自从 SQL 成为国际标准以来,很多数据库厂商都对 SQL 语言进行了再开发和扩展,但是包括查询 SELECT、插入 INSERT、修改 UPDATE、删除 DELETE、创建 CREATE 以及对象删除 DROP 在内的标准 SQL 语句仍然可被用来完成几乎所有的数据库操作。

3.1.1 SQL 发展

SQL 语言是关系数据库的标准语言,是数据库领域中一个主流语言,它经历了如下几个阶段。

(1) SQL-86: 第一个 SQL 标准,由美国国家标准局(American National Standard Institute, ANSI)公布,1987 年国际标准化组织(International Organization for Standardization, ISO)通过。该标准也称为 SQL-1。

(2) SQL-92: 在 1992 年,由 ISO 和 ANSI 对 SQL-86 进行了重新修订,发布了第二个 SQL 标准 SQL-92,该标准也称为 SQL-2。

(3) SQL-99: 随着信息技术的应用,数据库理论和技术得到了广泛的应用和发展。在 1999 年,ISO 发布了反映最新数据库理论和技术标准 SQL-99,该版本在 SQL-2 的基础上,扩展了诸多功能,包括递归、触发、面向对象技术等。该标准也称为 SQL-3。

(4) SQL-2003: 该标准是最新的标准,也称 SQL-4,于 2003 年发布,包括 9 个部分。

- ① ISO/IEC 9075—1: Framework(SQL/Framework);
- ② ISO/IEC 9075—2: Foundation(SQL/Foundation);
- ③ ISO/IEC 9075—3: Call Level Interface(SQL/CLI);
- ④ ISO/IEC 9075—4: Persistent Stored Modules(SQL/PSM);
- ⑤ ISO/IEC 9075—9: Management of External Data(SQL/MED);
- ⑥ ISO/IEC 9075—10: Object Language Bindings(SQL/OLB);
- ⑦ ISO/IEC 9075—11: Information and Definition Schemas(SQL/Schemata);
- ⑧ ISO/IEC 9075—13: Java Routines and Types Using the Java Programming Language(SQL/JRT);
- ⑨ ISO/IEC 9075—14: XML-Related Specifications(SQL/XML)。

目前,许多数据库厂商都支持 SQL-92 的绝大多数标准,以及 SQL-99 和 SQL-2003 的部分标准,并对 SQL 语言进行了扩展。这些扩展的 SQL 语言,不仅遵循标准 SQL 语言规定的功能,而且还增强了许多功能,并赋予 SQL 不同的名字,如 Oracle 产品将 SQL 称为 PL/SQL, Sybase 和 Microsoft SQL Server 产品将 SQL 称为 Transact-SQL。

SQL 语言由 4 部分组成,包括数据定义语言 DDL、数据操纵语言 DML、数据控制语言 DCL 和其他,其功能如下:

(1) 数据定义语言(data definition language,DDL): 主要用于定义数据库的逻辑结构,包括数据库、基本表、视图和索引等,扩展 DDL 还支持存储过程、函数、对象、触发器等定义。DDL 包括 3 类语言,即定义、修改和删除。

(2) 数据操纵语言(data manipulation language,DML): 主要用于对数据库的数据进行检索和更新,其中更新操作包括插入、删除和修改数据。

(3) 数据控制语言(data control language,DCL): 主要用于对数据库的对象进行授权、用户维护(包括创建、修改和删除)、完整性规则定义和事务定义等。

(4) 其他: 主要是嵌入式 SQL 语言和动态 SQL 语言的定义,规定了 SQL 语言在宿主语言中使用的规则。扩展 SQL 还包括数据库数据的重新组织、备份与恢复等功能。

3.1.2 SQL 特点

SQL 语言因其简单、灵活、易掌握,受到了广大用户的接受,SQL 语言既可以作为交互式数据库语言使用,也可以作为程序设计语言的子语言使用,它是一个兼有关系代数和元组演算特征的语言,其特点如下所述。

1. 综合统一

(1) SQL 语言集数据定义语言 DDL,数据操纵语言 DML 和数据控制语言 DCL 的功能于一体,语言风格统一,可以独立完成数据库生命周期中的全部活动,包括定义关系模式、录入数据以及建立数据库、查询、更新、维护、数据库重构、数据库安全性控制等一系列操作,这就为数据库应用系统开发提供了良好的环境。例如用户在数据库投入运行后,还可根据需要随时地逐步地修改模式,并不影响数据库的运行,从而使系统具有良好的可扩充性。

(2) 在关系模型中实体和实体间的联系均用关系表示,这种数据结构的单一性带来了数据操作符的统一,即对实体及实体间联系的每一种操作(如查找、插入、删除和修改)都只需要一种操作符。

2. 高度非过程化

非关系数据模型的数据操纵语言是面向过程的语言,在完成某项操作请求时必须指定存取路径。而用 SQL 语言进行数据操作,用户只需提出“做什么”,而不必指明“怎么做”,因此用户无须了解存取路径,存取路径的选择以及 SQL 语句的操作过程由系统自动完成。这不仅大大减轻了用户负担,而且有利于提高数据独立性。

3. 面向集合的操作方式

SQL 语言采用集合操作方式,其操作对象、操作结果都是元组的集合。而非关系数据模型采用的是面向记录的操作方式,其操作对象是一条记录。

(1) 非关系数据模型采用的是面向记录的操作方式,任何一个操作的对象都是一条记录。例如: 查询所有平均成绩在 80 分以上的学生姓名,用户必须说明完成该操作请求的具体处理过程,即如何用循环结构按照某条路径一条一条地把满足条件的学生记录读出来。

(2) SQL 语言采用集合操作方式,不仅查询操作的对象是元组的集合,而且一次插入、删除和更新操作的对象也可以是元组的集合。

4. 同一种语法结构提供两种使用方式

(1) SQL 语言既是自含式语言,又是嵌入式语言,且在两种不同的使用方式下,SQL 语言的语法结构基本上是一致的。

(2) 作为自含式语言,它能够独立地用于联机交互的使用方式,用户可以在终端键盘上直接键入 SQL 命令对数据库进行操作。

(3) 作为嵌入式语言,SQL 语句能够嵌入到高级语言(如 Java、VC、VB、Delphi 等)程序中,供程序员设计程序时使用。

5. 语言简洁,易学易用

SQL 语言功能极强,但十分简洁,易学易用。SQL 语言的动词非常少,主要包括:

- (1) 数据查询: SELECT;
- (2) 数据定义: CREATE、DROP、ALTER;
- (3) 数据操纵: INSERT、UPDATE、DELETE;
- (4) 数据控制: GRANT、REVOKE。

3.1.3 SQL 查询基本概念

SQL 语言支持关系数据库管理系统的三级模式结构,其中外模式对应视图和部分基本表,模式对应基本表,内模式对应存储文件,如图 3-1 所示。

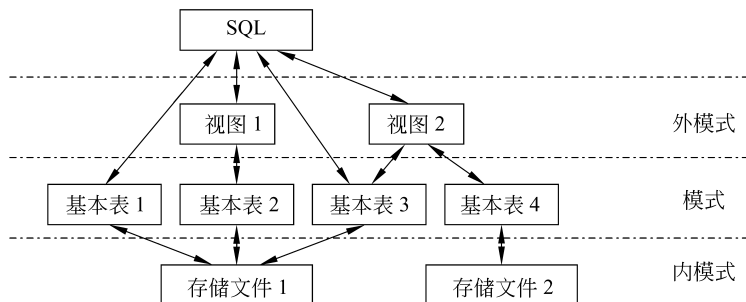


图 3-1 SQL 对关系数据库模式的支持

用户使用 SQL 语言可对基本表、视图和查询表进行操作。

1) 基本表

数据库中独立存在的表称为基本表。在 SQL 中一个关系对应一个基本表,一个(或多个)基本表对应一个存储文件;一个表可以带若干索引,索引也存放在存储文件中。

2) 视图

视图是指从一个或几个基本表(或视图)中导出的表,是虚表,只存放视图的定义而不存放对应的数据。

3) 查询表

查询表是指查询结果对应的表。

4) 存储文件

存储文件是指数据库中存放关系的物理文件,其逻辑结构组成了关系数据库的内模式,其物理结构对用户是透明的。

3.2 简单查询

一条 SQL 语句可以完成用若干条宿主语言才能完成的功能。SQL 语言中最重要的部分是查询语句,查询语句的执行离不开数据库模式。本章所用的数据库为学生成绩管理数据库 ScoreDB,其数据库模式如图 3-2~图 3-5 所示,其中带下划线的属性或属性集合为主码,斜体的属性为外码;其实例数据如图 3-6~图 3-9 所示。

班级编号	班级名称	所属学院	年级	班级人数
<u>classNo</u>	className	institute	grade	classNum
char(6)	varchar(30)	varchar(30)	smallint	tinyint

图 3-2 班级表 Class

学号	姓名	性别	出生日期	籍贯	民族	所属班级
<u>studentNo</u>	studentName	sex	birthday	native	nation	<i>classNo</i>
char(7)	varchar(20)	char(2)	datetime	varchar(20)	varchar(30)	char(6)

图 3-3 学生表 Student

课程号	课程名	学分	课时数	先修课程
<u>courseNo</u>	courseName	creditHour	courseHour	<i>priorCourse</i>
char(3)	varchar(30)	numeric	int	char(3)

图 3-4 课程表 Course

学号	课程号	成绩
<u>studentNo</u>	<u>courseNo</u>	score
char(7)	char(3)	numeric

图 3-5 成绩表 Score

	<u>courseNo</u>	courseName	creditHour	courseHour	<i>priorCourse</i>
1	001	高等数学	6	96	NULL
2	002	离散数学	6	96	001
3	003	计算机原理	4	64	NULL
4	004	C 语言程序设计	6	96	003
5	005	数据结构	4	64	004
6	006	数据库系统原理	5	80	005
7	007	操作系统	5	80	003

图 3-6 课程表 Course 的数据

	<u>classNo</u>	className	institute	grade	classNum
1	CP0801	注册会计师 08_01 班	会计学院	2008	NULL
2	CP0802	注册会计师 08_02 班	会计学院	2008	NULL
3	CP0803	注册会计师 08_03 班	会计学院	2008	NULL
4	CS0701	计算机科学与技术 07-01 班	信息管理学院	2007	NULL
5	CS0702	计算机科学与技术 07-02 班	信息管理学院	2007	NULL
6	CS0801	计算机科学与技术 08-01 班	信息管理学院	2008	NULL
7	ER0701	金融管理 07-01 班	金融学院	2007	NULL
8	IS0701	信息管理与信息系统 07-01 班	信息管理学院	2007	NULL
9	IS0801	信息管理与信息系统 08-01 班	信息管理学院	2008	NULL

图 3-7 班级表 Class 的数据

	<u>studentNo</u>	studentName	sex	birthday	native	nation	<u>classNo</u>
1	0700001	李小勇	男	1990-12-21 0:00	南昌	汉族	CS0701
2	0700002	刘方晨	女	1990-11-11 0:00	九江	汉族	IS0701
3	0700003	王红敏	女	1990-10-01 0:00	上海	汉族	IS0701
4	0700004	张可立	男	1991-05-20 0:00	南昌	蒙古族	CS0701
5	0700005	王红	男	1992-03-26 0:00	南昌	蒙古族	CS0702
6	0800001	李勇	男	1990-12-21 0:00	南昌	汉族	CS0801
7	0800002	刘晨	女	1990-11-11 0:00	九江	汉族	IS0801
8	0800003	王敏	女	1990-10-01 0:00	上海	汉族	IS0801
9	0800004	张立	男	1991-05-20 0:00	南昌	蒙古族	CS0801
10	0800005	王红	男	1992-03-26 0:00	南昌	蒙古族	CP0802
11	0800006	李志强	男	1991-12-21 0:00	北京	汉族	CP0802
12	0800007	李立	女	1991-08-21 0:00	福建	畬族	IS0801
13	0800008	黄小红	女	1991-08-09 0:00	云南	傣族	CS0801
14	0800009	黄勇	男	1991-11-21 0:00	九江	汉族	CP0802
15	0800010	李宏冰	女	1992-03-09 0:00	上海	汉族	CP0802
16	0800011	江宏吕	男	1990-12-20 0:00	上海	汉族	CP0802
17	0800012	王立红	男	1990-11-18 0:00	北京	汉族	CS0801
18	0800013	刘小华	女	1991-07-16 0:00	云南	哈尼族	IS0801
19	0800014	刘宏昊	男	1991-09-16 0:00	福建	汉族	IS0801
20	0800015	吴敏	女	1992-01-20 0:00	福建	畬族	CP0802

图 3-8 学生表 Student 的数据

	<i>studentNo</i>	<i>courseNo</i>	<i>score</i>		<i>studentNo</i>	<i>courseNo</i>	<i>score</i>
1	0700001	001	98	21	0800003	001	70
2	0700001	002	82	22	0800003	002	60
3	0700001	006	56	23	0800003	007	50
4	0700002	003	69	24	0800004	001	70
5	0700002	004	87	25	0800004	003	98
6	0700002	005	77	26	0800004	006	89
7	0700002	007	90	27	0800004	007	80
8	0700003	001	46	28	0800005	001	82
9	0700003	002	38	29	0800005	002	80
10	0700003	007	50	30	0800005	004	47
11	0800001	001	70	31	0800005	006	90
12	0800001	002	68	32	0800005	007	82
13	0800001	003	70	33	0800006	001	78
14	0800001	004	58	34	0800006	002	60
15	0800001	005	88	35	0800006	003	69
16	0800001	006	72	36	0800006	004	87
17	0800001	007	71	37	0800006	005	77
18	0800002	001	98	38	0800006	006	56
19	0800002	003	60	39	0800006	007	90
20	0800002	007	46	40	0800007	002	38

图 3-9 成绩管理表 Score 的数据

本章给出的所有例题的操作执行结果都是在 SQL Server 2005 数据库中执行得到的。

3.2.1 投影运算

SQL 查询语句的基本结构包括 3 个子句：SELECT、FROM 和 WHERE，其中：

- SELECT 子句对应于关系代数中的投影运算，用来指定查询结果中所需要的属性或表达式。
- FROM 子句对应于关系代数中的笛卡儿积，用来给出查询所涉及的表，表可以是基本表、视图或查询表。
- WHERE 子句对应于关系代数中的选择运算，用来指定查询结果元组所需要满

足的选择条件。

对于 SQL 语句,SELECT 和 FROM 子句是必需的,其他是可选的,其基本语法为

```
SELECT A1, A2, ..., An
FROM R1, R2, ..., Rm
WHERE P
```

其中: $A_1, A_2, \dots, A_n$ 代表需要查找的属性或表达式; $R_1, R_2, \dots, R_m$ 代表查询所涉及的表; P 代表谓词(即选择条件),如果省略 WHERE 子句,表示 P 为真。SQL 的查询结果中允许包含重复元组,这与关系代数中的投影运算的语义不一样。

SQL 查询的执行过程为:首先对 $R_1, R_2, \dots, R_m$ 执行笛卡儿积,然后在笛卡儿积中选择使得谓词 P 为真的记录,再在 $A_1, A_2, \dots, A_n$ 属性列中进行投影运算,不消除重复元组。如果需要消除重复元组,必须使用专门的关键字 DISTINCT。

上面描述的 SQL 查询执行过程只是逻辑上的,在具体执行时会进行优化处理,查询优化的内容详见第 8 章。

1. 查询指定列

选取表中的全部列或指定列,称为关系代数的投影运算。通过 SELECT 子句确定要查询的属性。

[例 3.1] 查询所有班级的班级编号、班级名称和所属学院。

```
SELECT classNo, className, institute
FROM Class
```

该查询的执行过程是:从 Class 表中依次取出每个元组,对每个元组仅选取 classNo, className 和 institute 3 个属性的值,形成一个新元组,最后将这些新元组组织成一个结果关系输出。该查询的结果如图 3-10 所示。

	classNo	className	institute
1	CP0801	注册会计师08_01班	会计学院
2	CP0802	注册会计师08_02班	会计学院
3	CP0803	注册会计师08_03班	会计学院
4	CS0701	计算机科学与技术07-01班	信息管理学院
5	CS0702	计算机科学与技术07-02班	信息管理学院
6	CS0801	计算机科学与技术08-01班	信息管理学院
7	ER0701	金融管理07-01班	金融学院
8	IS0701	信息管理与信息系统07-01班	信息管理学院
9	IS0801	信息管理与信息系统08-01班	信息管理学院

图 3-10 例 3.1 的查询结果

2. 消除重复元组

SQL 查询默认是不消除重复元组,因为消除重复元组要消耗系统资源。如果需要消除重复元组,可以使用 DISTINCT 关键字。

[例 3.2] 查询所有学院的名称。

```
SELECT institute
FROM Class
```

上述查询不消除重复元组,其查询结果如图 3-11 所示。如果需要消除重复元组,则可使用如下查询,其查询结果如图 3-12 所示。

```
SELECT DISTINCT institute
FROM Class
```

	institute
1	会计学院
2	会计学院
3	会计学院
4	信息管理学院
5	信息管理学院
6	信息管理学院
7	金融学院
8	信息管理学院
9	信息管理学院

图 3-11 例 3.2 的查询结果一

	institute
1	会计学院
2	金融学院
3	信息管理学院

图 3-12 例 3.2 的查询结果二

3. 查询所有列

查询所有的属性列,SQL 可以使用两种方法:一是将所有的列在 SELECT 子句中列出(可以改变列的显示顺序);二是使用 * 符号,* 表示所有属性,此时按照表定义时的顺序显示所有属性。

[例 3.3] 查询所有班级的全部信息。

```
SELECT classNo, className, classNum, grade, institute
FROM Class
```

或

```
SELECT *
FROM Class
```

4. 给属性列取别名

可以为属性列取一个便于理解的列名,如用中文来显示列名,为属性列取别名特别适合那些经过计算的列。

[例 3.4] 查询所有班级的所属学院、班级编号和班级名称,要求用中文显示列名。

```
SELECT institute 所属学院,
       classNo 班级编号, className 班级名称
FROM Class
```

该查询的结果如图 3-13 所示。该查询也可以使用 AS 关键字来取别名,如

	所属学院	班级编号	班级名称
1	会计学院	CP0801	注册会计师08_01班
2	会计学院	CP0802	注册会计师08_02班
3	会计学院	CP0803	注册会计师08_03班
4	信息管理学院	CS0701	计算机科学与技术07-01班
5	信息管理学院	CS0702	计算机科学与技术07-02班
6	信息管理学院	CS0801	计算机科学与技术08-01班
7	金融学院	ER0701	金融管理07-01班
8	信息管理学院	IS0701	信息管理与信息系统07-01班
9	信息管理学院	IS0801	信息管理与信息系统08-01班

图 3-13 例 3.4 的查询结果

```
SELECT institute AS 所属学院,
       classNo AS 班级编号, className AS 班级名称
FROM Class
```

5. 查询经过计算的列

SELECT 子句中可以使用属性、常数、函数和表达式,如果是函数或表达式,则先计算函数或表达式的值,然后将计算的结果显示出来。

[例 3.5] 查询每个班级编号、班级名称以及该班级现在为几年级,并将班级编号中大写字母改为小写字母输出。

```
SELECT lower(classNo)班级编号,
       className, year(getdate())-grade AS 年级
FROM Class
```

其中,函数 lower()表示将大写字母改为小写字母,函数 getdate()可获取当前系统的日期,函数 year()用于提取日期中的年份,其查询结果如图 3-14 所示。

班级编号	className	年级
1	cp0801	注册会计师08_01班
2	cp0802	注册会计师08_02班
3	cp0803	注册会计师08_03班
4	cs0701	计算机科学与技术07-01班
5	cs0702	计算机科学与技术07-02班
6	cs0801	计算机科学与技术08-01班
7	er0701	金融管理07-01班
8	is0701	信息管理与信息系统07-01班
9	is0801	信息管理与信息系统08-01班

图 3-14 例 3.5 的查询结果

3.2.2 选择运算

WHERE 子句可以实现关系代数中的选择运算,用于查询满足选择条件的元组,这是查询中涉及最多的一类查询。WHERE 子句中常用的查询条件如下所示。

- 比较运算: >, >=, <, <=, =, <> (或 !=);
- 范围查询: BETWEEN...AND;
- 集合查询: IN;
- 空值查询: IS null;
- 字符匹配查询: LIKE;
- 逻辑查询: AND, OR, NOT。

1. 比较运算

使用比较运算符 >, >=, <, <=, =, <> (或 !=) 实现相应的比较运算。

[例 3.6] 查询 2007 级的班级编号、班级名称和所属学院。

```
SELECT classNo, className, institute
FROM Class
WHERE grade=2007
```

	classNo	className	institute
1	CS0701	计算机科学与技术07-01班	信息管理学院
2	CS0702	计算机科学与技术07-02班	信息管理学院
3	ER0701	金融管理07-01班	金融学院
4	IS0701	信息管理与信息系统07-01班	信息管理学院

图 3-15 例 3.6 的查询结果

其查询结果如图 3-15 所示。

该查询的执行过程可能有多种方法:一种是全表扫描法,即依次取出 Class 表中的每个元组,判断该元组的 grade 属性值是否等于 2007,若是则将该元组的班级编号、班级名称和所属学院属性取出,形成一个新元组,最后将所有新元组组织为一个结果关系输出,该方法适用于小表,或者该表未在 grade 属性列上建索引。另一种是索引搜索法,如果该表在 grade 属性列上建有索引,且满足条件的记录不多,则可以使用索引搜索法来检索数据。具体使用何种方法由数据库管理系统的查询优化器来选择,详见第 8 章内容。

[例 3.7] 在学生 Student 表中查询年龄大于或等于 19 岁的同学学号、姓名和出生日期。