项目 3　利用标准 SQL 语言操作数据库

【项目要点】

(1) 通过 SELECT 语句从数据库中查询数据。

(2) 通过 INSERT 语句向数据库中插入数据。

(3) 通过 UPDATE 语句更新数据库的数据。

(4) 通过 DELETE 语句删除数据库中的数据。

任务 3.1　认知 SQL 语言

SQL(structured query language,结构化查询语言)是一种用于数据库查询和设计的语言,可以实现对关系数据库中数据的存取、查询、更新和管理。结构化查询语言是高级的非过程化编程语言,允许用户在高层数据结构(即数据表)上工作。它不要求用户指定对数据的存放方法,也不需要用户了解具体的数据存放方式,所以对具有完全不同底层结构的不同数据库系统,可以使用相同的结构化查询语言作为数据输入与管理的接口。结构化查询语言语句可以嵌套,这使它具有极大的灵活性和强大的功能。SQL 是最重要的关系数据库操纵语言,并且它的影响已经超出数据库领域,得到其他领域的重视和采用,如人工智能领域的数据检索、第四代软件开发工具中嵌入 SQL 的语言等。

SQL 是 1986 年 10 月由美国国家标准局通过的数据库语言美国标准,接着,国际标准化组织(ISO)颁布了 SQL 正式国际标准。1989 年 4 月,ISO 提出了具有完整性特征的 SQL89 标准,1992 年 11 月又公布了 SQL92 标准,在此标准中,把数据库分为三个级别:基本集、标准集和完全集。

各种不同的数据库对 SQL 语言的支持与标准存在着细微的不同,这是因为有的产品的开发先于标准的公布,另外,各产品开发商为了达到特殊的性能或新的特性,需要对标准进行扩展。目前已有 100 多种遍布在从微机到大型机上的数据库产品,其中包括 DB2、SQL/DS、Oracle、Informix、SyBase、SQL Server、dBase4、Paradox、Access 等。

SQL 语言基本上独立于数据库本身、使用的机器、网络、操作系统,基于 SQL 的 DBMS 产品可以运行在从个人机、工作站到基于局域网、小型机和大型机的各种计算机系统上,具有良好的可移植性,可以看出标准化的工作是很有意义的。早在 1987 年就有些有识之士预测 SQL 的标准化是“一场革命”,是“关系数据库管理系统的转折点”。数据库和各种产品都使用 SQL 作为共同的数据操纵语言和标准的接口,使不同数据库系统之

间的互操作有了共同的基础，进而实现异构机、各种操作环境的共享与移植。SQL 语言是一种非过程化语言，它允许用户在高层的数据结构（数据表或关系）上操作，而不对单个数据行进行操作。在使用 SQL 语言的过程中，用户完全不用考虑诸如数据的存储格式、数据的存储路径等复杂的问题。用户所需要做的只是使用 SQL 语言提出自己的要求，至于如何实现这些要求，则是关系数据库管理系统的任务。

1974 年，在 IBM 公司圣约瑟研究实验室研制的大型关系数据库管理系统 SYSTEM R 中，使用 SEQUEL 语言（由 BOYCE 和 CHAMBERLIN 提出），后来在 SEQUEL 的基础上发展了 SQL 语言。SQL 语言是一种交互式查询语言，允许用户直接查询存储数据，但它不是完整的程序语言，如它没有 do 或 for 类似的循环语句，但它可以嵌入到另一种语言中，也可以借用 VB、C、Java 等语言，通过调用级接口（call level interface）直接发送到数据库管理系统。SQL 基本上是域关系演算，但也可以实现关系代数操作。

SQL 是一个综合的、通用的、功能极强又简洁易学的关系数据库语言，它包含六个部分。

1. 数据定义语言

数据定义语言（data definition language，DDL），其语句主要包括命令动词 CREATE、DROP 和 ALTER，可在数据库中创建新表或删除表（CREAT TABLE 或 DROP TABLE）、为数据表添加索引等。

2. 数据操纵语言

数据操纵语言（data manipulation language，DML），其语句包括动词 INSERT（插入）、UPDATE（更新）和 DELETE（删除），它们分别用于添加、修改和删除表中的行。也称为动作查询语言。

3. 数据查询语言

数据查询语言（data query language，DQL），其语句也称为“数据检索语句”，用以从表中获得数据，确定数据怎样在应用程序给出。保留字 SELECT、FROM 是 DQL 用得最多的动词，其他 DQL 常用的保留字有 WHERE、ORDER BY、GROUP BY 和 HAVING。这些 DQL 保留字常与其他类型的 SQL 语句一起使用。

4. 事务处理语言

事务处理语言（transaction processing language，TPL），其语句能确保被 DML 语句影响的表的所有行及时得以更新。TPL 语句包括 BEGIN TRANSACTION、COMMIT 和 ROLLBACK。

5. 数据控制语言

数据控制语言（data control language，DCL），其语句通过 GRANT 或 REVOKE 获得权限，确定单个用户和用户组对数据库对象的访问。某些 RDBMS 可用 GRANT 或

REVOKE 控制对表单个列的访问。

6. 游标控制语言

游标控制语言(cursor control language,CCL),它的语句如 DECLARE CURSOR、FETCH INTO 和 UPDATE WHERE CURRENT 用于对一个或多个表单独行的操作。

任务 3.2　利用数据定义语言创建、修改、删除数据库

数据定义语言包括数据库的创建与删除,数据表的创建、修改及删除,视图的创建、修改及删除等功能。

3.2.1　创建数据库的语句

创建数据库的语句——CREATE DATABASE 的语法格式如下。

```
CREATE DATABASE database_name
[
ON
    <[PRIMARY] (NAME = logical_file_name,
    FILENAME = 'os_file_name'
    [,SIZE = size]
    [,MAXSIZE = max_size]
    [,FILEGROWTH = growth_increment])
    >[,...n]
]
[
LOG ON
    <(NAME = logical_name,
    FILENAME = ' os_file_name'
    [,SIZE = size]
    [,MAXSIZE = max_size]
    [,FILEGROWTH = growth_increment])
    >[ ,...n]
]
```

语句中各部分内容的含义说明如下。

(1) 语句中大写字母为 SQL 语言保留的关键字,在输入语句时,不能有拼写错误,相邻的关键字之间至少有一个空格字符。

(2) database_name:指定数据库名,由用户根据需要自行定义,其命名规则与高级语言的标识符相同,最大长度为 128 个字符。

(3) ON 子句:指定数据库中的文件组和数据文件属性。

(4) LOG ON 子句:指定日志文件属性。

本书中语句书写格式说明如下。

(1) 大写字母表示关键字,运用时不能有拼写错误,小写字母表示用户给定的名称或数值。

(2) < >表示必选项。

(3) | 表示多项选一。

(4) …n 表示可有多个同类项。

(5) [ ]表示可选项。

数据库名在 SQL Server 数据库管理系统中必须是唯一的,不能重复,在同一台数据库服务器中无法建立重名的数据库。在同一个用户数据库中,logical_file_name 也不能有重名; os_file_name 是操作系统文件名,在同一台数据库服务器中无法建立重名的操作系统文件。

【例 3-1】 使用 CREATE DATABASE 命令创建一个名称为 JWGL 的数据库,数据库所使用的文件组、文件均采用 SQL Server 数据库管理系统的默认值。

操作步骤如下。

在 SSMS 窗口中,单击标准工具栏上的"新建查询"按钮,打开 SQL Query 标签页,然后,在 SQL Query 标签页中输入下面的命令。

```
CREATE DATABASE JWGL
GO
```

GO 是批处理结束标记(一种程序结束标记)。

SQL 语言命令输入完毕,单击标准工具栏上的"执行"按钮。如果 SQL 语句没有错,Microsoft SQL Server Management Studio 把该语句发送到后台的数据库引擎,由数据库引擎执行。语句执行完毕,数据库引擎向 SSMS 窗口返回"命令已成功完成"信息,如图 3-1 所示。

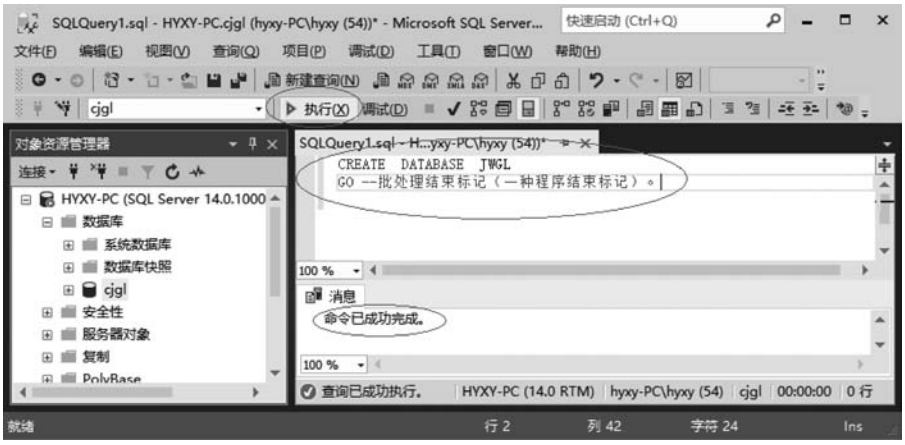

图 3-1　在 SQL Query 标签页中输入并执行 SQL 语句的 SSMS 窗口

右击 SSMS 窗口左侧栏中的"数据库"节点,在弹出的快捷菜单中选择"刷新"命令,如图 3-2 所示。

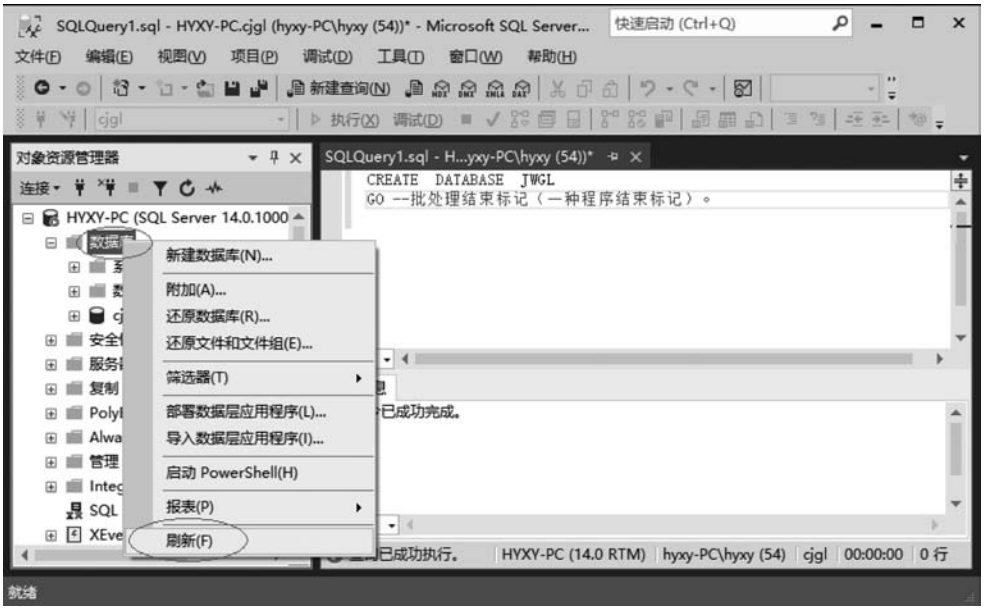

图 3-2 “数据库”节点快捷菜单

展开“数据库”节点,在它的下一级节点中可观察到新生成的数据库 JWGL,如图 3-3 所示。

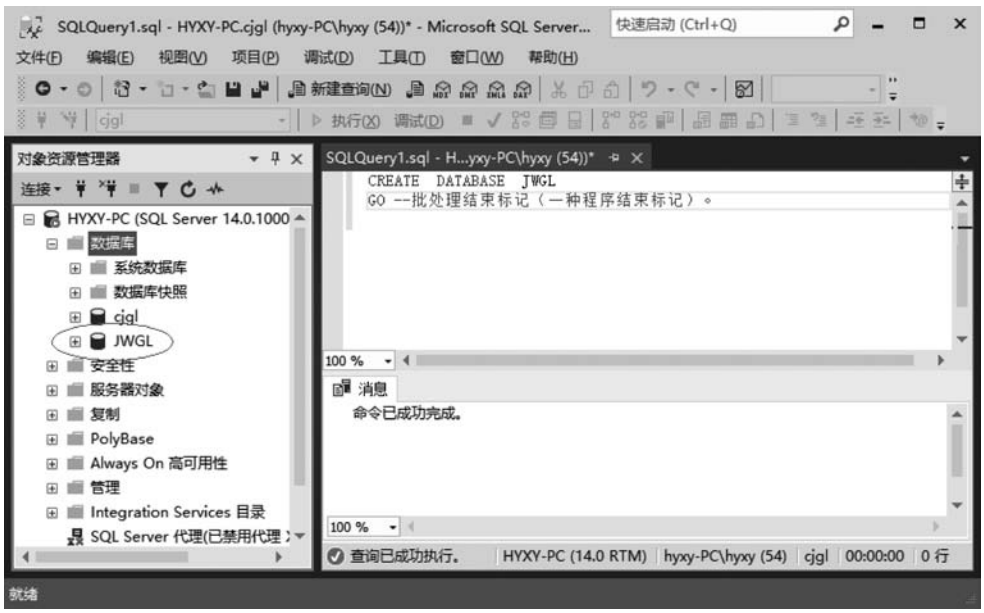

图 3-3 新生成的 JWGL 数据库

【例 3-2】 使用 CREATE DATABASE 语句创建一个逻辑名为“货物管理”的数据库,数据库的主数据文件的逻辑名为“货物管理_data”,物理文件名为“货物管理_data. mdf”,

初始容量为 5MB,存储在 D:\下,文件的最大容量为 10MB,文件增量为 1MB; 事务日志文件的逻辑名称为“货物管理_log”,物理文件名为“货物管理_log.ldf”,初始容量为 2MB,存储在 D:\下,最大容量为 15MB,增长速度为 1MB。

在 SQL Query 标签页中输入下面的语句并执行即可完成任务。

```
CREATE DATABASE 货物管理
ON
(
    NAME = 货物管理_data,
    FILENAME = 'C:\货物管理_data.mdf',
    SIZE = 3,
    MAXSIZE = 10,
    FILEGROWTH = 1
)
LOG ON
(
    NAME = 货物管理_log,
    FILENAME = 'C:\货物管理_log.ldf',
    SIZE = 2MB,
    MAXSIZE = 15MB,
    FILEGROWTH = 1MB
)
GO
```

3.2.2　删除数据库的语句

删除数据库是指将数据库及其全部内容从系统中删除。DROP DATABASE 语句用于删除数据库,它的简单语法格式如下。

```
DROP DATABASE <数据库名>
```

【例 3-3】 使用 DROP DATABASE 语句删除 JWGL 数据库。

在 SQL Query 标签页中输入下面的语句并执行即可完成任务。

```
USE master
GO
DROP DATABASE JWGL
GO
```

3.2.3　备份还与原数据库

1. 备份数据库的语法格式

```
BACKUP DATABASE "数据库逻辑名" TO DISK = '备份文件物理名'
```

【例 3-4】 把 cjgl 数据库备份到 E 盘,备份文件名为 cjgl.bak。

在 SQL Query 标签页中输入下面的语句并执行即可完成任务。

```
USE master
GO
BACKUP DATABASE cjgl TO DISK = 'E:\cjgl.bak'
GO
```

2. 还原数据库的语法格式

```
RESTORE DATABASE "数据库逻辑名" FROM DISK = '备份文件物理名'
```

【例 3-5】 先删除 cjgl 数据库,然后利用 E 盘的备份文件将其还原。

在 SQL Query 标签页中输入下面的语句并执行即可完成任务,该语句执行结果如图 3-4 所示。

```
USE master
GO
DROP DATABASE cjgl
GO
RESTORE DATABASE cjgl FROM DISK = 'E:\cjgl.bak';
GO
```

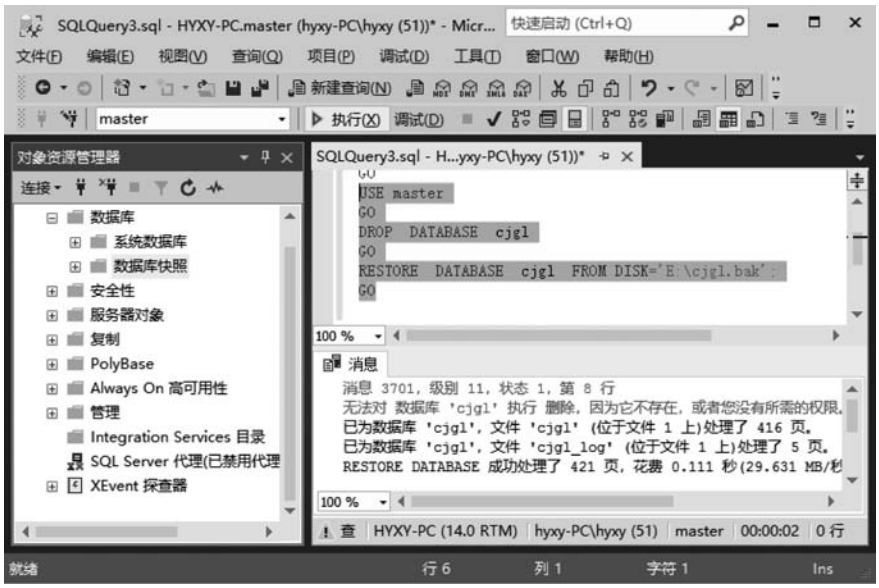

图 3-4 还原数据库语句的执行结果

3.2.4 创建数据表

使用 CREATE TABLE 语句可以创建数据表。创建数据表的简单语法格式如下。

```
CREATE TABLE 表名
(
    列名 1 列 1 的属性 [PRIMARY KEY|UNIQUE] [NULL|NOT NULL]
    [,列名 2 列 2 的属性 [PRIMARY KEY|UNIQUE] [NULL|NOT NULL] [,...n]]
)
```

其中,列的属性包括列的数据类型、列的长度、列上的约束等;PRIMARY KEY 表示主键;UNIQUE 表示唯一索引;NULL 表示列值允许为空;NOT NULL 表示列值不允许为空。

【例 3-6】 在例 3-2 中创建的货物管理数据库中建立供应商表和零件表。其中供应商表的表结构为 S(Sno, Sname, Status, City),其中的 Sno、Sname、Status、City 分别表示供应商代码、供应商名称、供应商状态、供应商所在城市。零件表的表结构为 P(Pno, Pname, Color, Weight, City),其中的 Pno、Pname、Color、Weight、City 分别表示零件号、零件名、颜色、重量及产地。其中,数据表要满足如下要求。

(1) 供应商代码不能为空,且值是唯一的,供应商的名称也是唯一的。

(2) 零件号不能为空,且值是唯一的。零件名不能为空。

根据上述分析,用 SQL 语句建立供应商表、零件表的语句如下。

```
USE 货物管理
GO
CREATE TABLE S
(
    Sno CHAR(5) NOT NULL UNIQUE,
    Sname CHAR(30) UNIQUE,
    Status CHAR(8),
    City CHAR(20)
)
GO
CREATE TABLE P
(
    Pno CHAR(6) NOT NULL UNIQUE,
    Pname CHAR(30) NOT NULL,
    Color CHAR(8),
    Weight NUMERIC (6,2),
    City CHAR(20)
)
GO
```

3.2.5　修改数据表

在基本表建立并使用一段时间后,可能需要根据实际要求对基本表的结构进行修改,即通过更改、添加、删除列和约束,或者通过启用或禁止约束和触发器来对当前数据库中的数据表进行修改。ALTER TABLE 语句用于修改基本表,它的语法格式如下。

```
ALTER TABLE <表名>
```

```
[ADD <新列名><数据类型>[完整性约束条件]]
[DROP <完整性约束名>][ALTER COLUMN <列名><数据类型>]
```

【例 3-7】 给 cjgl 数据库中的 xs 表增加 1 个新列：奖学金等级，类型为 tinyint，允许为空，然后在表 xs 中删除名为奖学金等级的列。

在 SQL Query 标签页中输入并执行如下的 SQL 语句。

```
USE cjgl
GO
ALTER TABLE xs ADD 奖学金等级 tinyint null
GO
```

执行完上述 SQL 语句后，查看“奖学金等级”列是否已增加(通过数据表 xs 的右键快捷菜单中的“编辑前 200 行”命令)。

```
ALTER TABLE xs DROP COLUMN 奖学金等级
GO
```

执行完上述 SQL 语句后，查看“奖学金等级”列是否已删除。

3.2.6 给数据表创建索引

1. 使用 CREATE INDEX 语句创建索引

使用 CREATE INDEX 语句可以在一个已有表上创建索引，一个表可以创建多个索引。

语法格式如下。

```
CREATE [UNIQUE] INDEX 索引名 ON 表名(列名 [ASC|DESC],...)
```

其中：

(1) 索引名在一个表中必须是唯一的。

(2) 列名表示创建索引的列名。

(3) UNIQUE 表示创建的是唯一性索引。

CREATE INDEX 语句并不能创建主键。

【例 3-8】 在 cjgl 数据库的 xs 数据表的“系名”列上建立一个升序索引，索引名称为 depart_xs。

```
USE cjgl
GO
CREATE INDEX depart_xs ON xs(系名 ASC)
GO
```

可以在一个索引的定义中包含多个列，中间用逗号隔开，但是它们要属于同一个表，这样的索引称为复合索引。

【例 3-9】 在 cjgl 数据库的 xs 数据表的“班级名”列和“系名”列上建立一个复合索引

cl_de_xs。

```
USE cjgl
GO
CREATE INDEX cl_de_xs ON xs(班级名,系名);
GO
```

2. 在创建数据表的同时创建索引

在前面两种情况下,索引都是在数据表创建之后创建的。索引也可以在创建表时一起创建。在创建数据表的 CREATE TABLE 语句中可以包含索引的定义。

语法格式如下。

```
CREATE TABLE 表名 ( 列名, ... | [索引项])
```

其中,索引项语法格式如下。

```
PRIMARY KEY (列名,...)                    /* 主键 */
|< INDEX|KEY > [索引名] (列名,...)         /* 索引 */
|UNIQUE[INDEX] [索引名] (列名,...)         /* 唯一性索引 */
|[FULLTEXT] [INDEX] [索引名] (列名,...)    /* 全文索引 */
```

KEY 通常是 INDEX 的同义词。在定义列的时候,可以将某列定义为 PRIMARY KEY,但如果当主键由多个列组成,定义列时无法定义此主键,必须在语句最后加上一个 PRIMARY KEY(列名,...)子句。

【例 3-10】 在 cjgl 数据库中创建 score 数据表的同时,在 sNo 和 cNo 列上创建联合主键。

```
USE cjgl
GO
CREATE TABLE score
(
    sNo char(10) NOT NULL,
    cNo char(2) NOT NULL,
    score float NULL,
    PRIMARY KEY(sNo,cNo)
)
GO
```

3. 使用 DROP INDEX 语句删除索引

语法格式如下。

```
DROP INDEX 索引名 ON 表名
```

【例 3-11】 在 cjgl 数据库中,删除数据表 xs 中的复合索引 cl_de_xs。

```
USE cjgl
GO
```