

计软实验四：数据库的创建与 SQL 查询

前言：

《计算机软件技术基础》课程中的数据库部分主要涉及以下内容：数据库系统基础知识、数据模型（尤其是 E-R 模型，关系模型）、关系型数据库、数据库设计过程。

基础知识章节整体介绍了数据库系统的发展历程和基本概念，应该意识到，数据库的目的就是为了方便存储、管理、查询数据，它是现今众多计算机应用系统的核心。我们使用的网站、APP 等等，给用户提供了一个可视的界面以及相关的功能，其中的内容都来自于背后的数据库支持，因此数据库也常被称为后台数据库。

接下来，**数据模型章节**回到很原始的问题，即“如何组织数据”，并从抽象概念（E-R 模型）和数据模型（层次模型、网状模型、关系模型）的角度进行了回答。关系模型从中脱颖而出。因此，我们现在提到数据库，默认的便是基于关系模型构建的数据库，各种实际业务处理中常用的如 Oracle、SQLserver、MySQL、SQLite 也都是关系数据库。

关系型数据库如此成功，离不开其基础理论的支持，所以课件中的 6.3 节首先简单介绍了关系模型的理论知识。一方面，三类完整性约束确保了数据库中业务数据的操作正确性和相容性；另一方面，关系代数为查询操作提供了理论支持，它是 SQL 语言的数学基础。我们所编写的 SQL 语言都可以翻译为关系代数上的运算。

SQL 语言简单易学，即使你对关系代数并不了解、对完整性约束等概念理解尚不透彻，也可以轻松掌握 SQL 的语法，学会用 SQL 进行数据查询。

使用 SQL 语言创建数据库，进行基本的 SQL 查询，这便是我们本次实验的主要内容。

而完整的**数据库设计过程**，则留给有兴趣的同学自行探索实践。

一、 实验目的

1. 熟悉 SQL sever、SSMS 软件界面、基本操作方法。
2. 掌握数据库、数据表的创建方法，数据的添加方法。
3. 掌握基本 SQL 查询语句。

二、 实验内容

1. 在 SSMS(SQL Server Management Studio) 中创建数据库、数据表，添加数据；
2. 在所创建的数据库中进行 SQL 查询。
3. 根据前两部分的学习，自行完成测验题目。

三、 实验步骤

实验步骤分为五部分：

1. SSMS 软件界面，
2. 使用命令创建数据库，
3. 使用命令创建数据表
4. 使用命令添加数据，
5. SQL 查询。

前四部分用于熟悉 SSMS 软件、掌握创建数据库的基本流程，后续 SQL 查询将以此为基础。在实际操纵过程中，应当理解各个步骤的含义，熟悉示例数据库中 STUDENT、COURSE、SCORE 这三张表中各列数据的名称、含义，有助于后续 SQL 查询中对语句的快速理解。

第五部分 SQL 查询是实验要点，对 SQL 的基本查询、统计语法通过**实例**进行讲解，**图**片中给出了**具体代码和相应注释**。SQL 语法简单，按照示例依次实现即可理解。实验过程中，可以自行设计查询要求进行实验，熟悉语法。

此外，蓝色字体中给出了各部分对应的相关知识，加深对实验内容的理解，明白实验内容在实际 SQL 应用中的意义。

1. SSMS 软件界面

【相关知识】

SQL sever 和 SSMS 下载

SQL Server 是 Microsoft 公司推出的关系型数据库管理系统。

SSMS (SQL Server Management Studio)是用于管理 SQL Server 基础架构的集成环境。

简而言之，电脑上需要先后安装 SQL sever 和 SSMS 之后才可以进行数据库开发等工作。

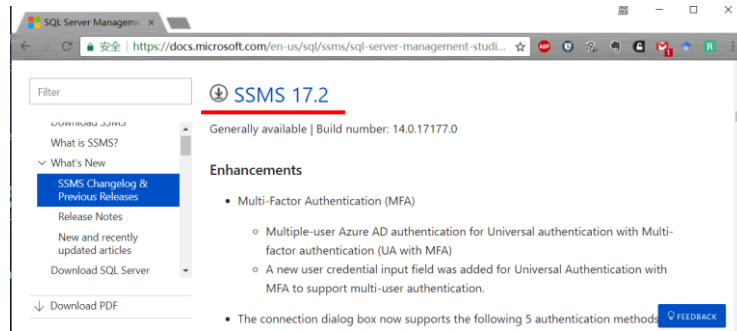
SQL sever 2016

自行下载安装

SSMS 17.2

下载网址：<https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-changelog-ssms#previous-ssms-releases>

页面如图所示：



点击 SSMS (Microsoft SQL Server Management Studio) 图标运行软件；

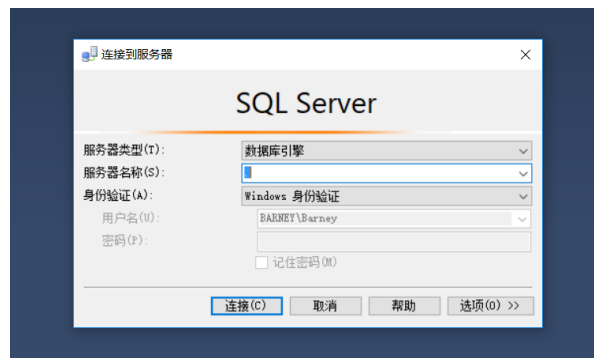


图 1-1 连接到服务器

如图 1-1 , 服务器类型选择“数据库引擎”, 服务器名称输入“.”即可 , 身份验证为“windows 身份验证”, 点击“连接”。

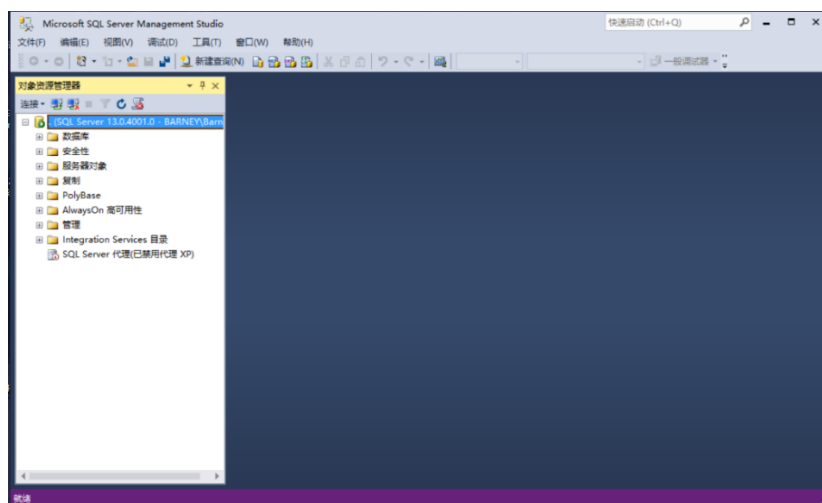


图 1-2 SSMS 界面

进入 SQL server 2016 Management Studio 界面 ,如图 1-2。左边为“对象资源管理器” ,也是我们使用最多的界面之一。点击“数据库”的展开选项“+”即可看到已存在的数据库列表。

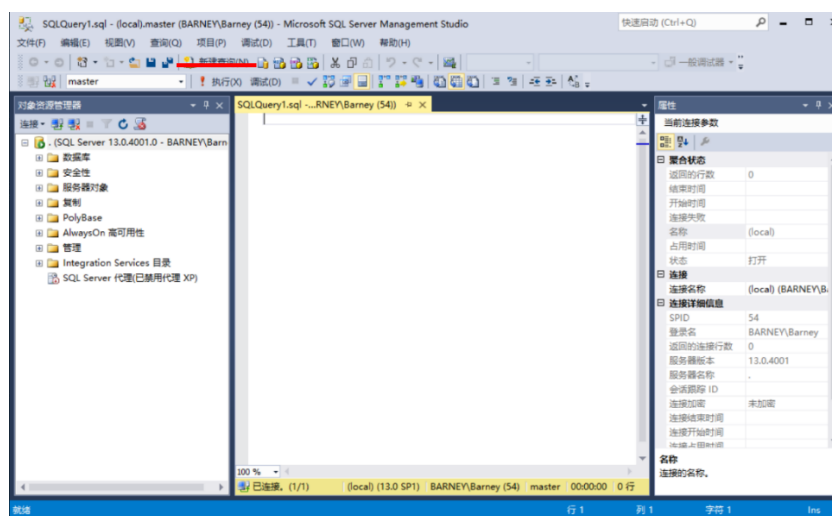


图 1-3 查询窗口

点击“新建查询”，出现图 1-3 所示的查询窗口，也就是我们将频繁使用的 SQL 语句输入创窗口。

2. 使用命令创建数据库

【相关知识】

数据库包括数据文件和日志文件 ,其中数据文件又可以包括主要数据文件和次要数据文件。

数据文件包括数据和对象，例如表、索引、存储过程和视图；

日志文件包含恢复数据库中的所有事物所需的信息。

主要数据文件：包含数据库的启动信息，并指向数据库的其他文件。每个数据库有一个主要文件。建议扩展名为“mdf”。

次要数据文件：用于将数据分散到多个不同的物理磁盘上以提高访问速度。次要文件是可选的，建议文件扩展名为“ndf”。

日志文件：保存用于恢复数据库的日志信息。每个数据库必须至少有一个日志文件。建议文件扩展名为“ldf”。

此外，每个数据库有一个名为 PRIMARY 的文件组。此文件组包含主要数据文件和未放入其他文件组的次要文件。创建对象时若没有指定文件组，则将被分配给 PRIMARY 文件组。

(上机实验中，我们只涉及包含一个主要数据文件、一个日志文件的数据库，默认存在于 PRIMARY 文件组中 ;只涉及表的创建 ,关于“索引、存储过程、视图”等知识可自行学习。)

图 2-1 为创建数据库的命令。阅读代码理解含义，并在查询窗口输入 (注意以学号命名)，点击“执行”运行代码。

在对象资源管理器中右击“数据库”，点“刷新”，即可看到新创建的数据库，如图 2-2。

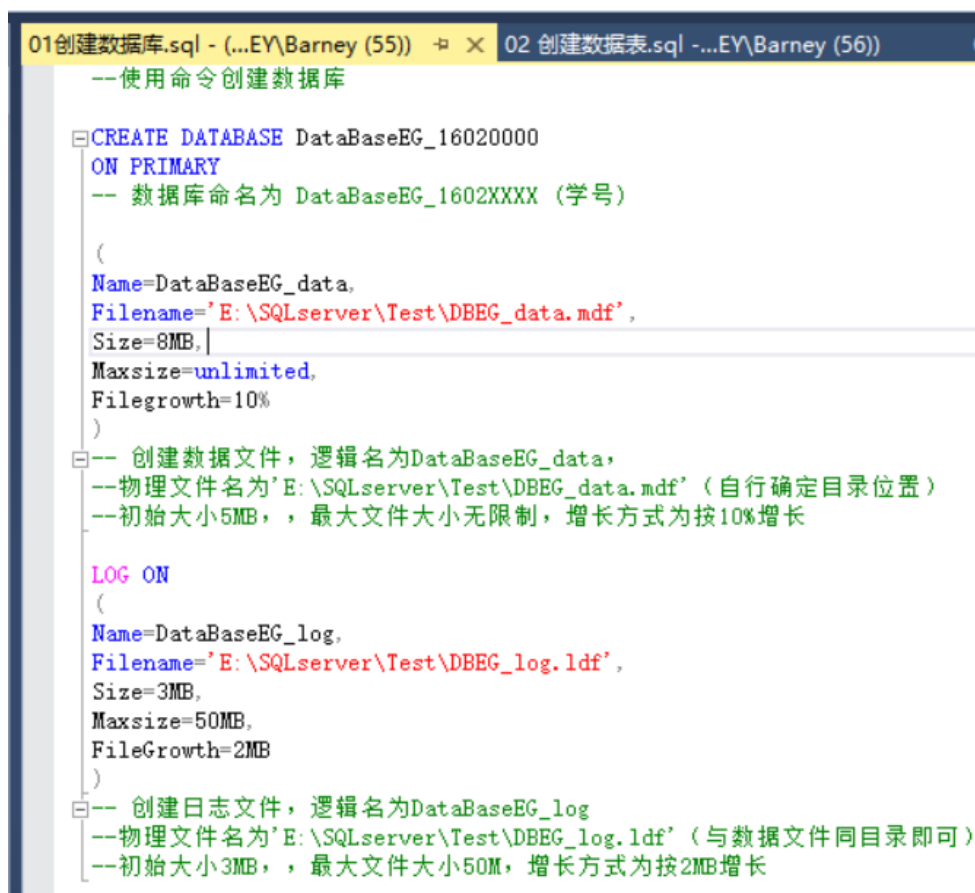


图 2-1 使用命令创建数据库

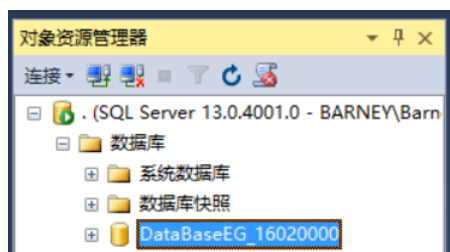


图 2-2 新创建的数据库

另外，使用命令删除数据库也很简单，在查询窗口输入“DROP database DataBaseEG_16020000”并执行即可。SQL 对大小写并不敏感。为了维护和阅读方便，习惯性地关键字大写，其他小写，并且注意换行和缩进。

3. 使用命令创建数据表

【相关知识】

表是储存数据的数据库对象，用来保存各种数据。

表的组织形式与 Excel 类似，按行和列的格式组织。

每一行代表一条记录，每一列代表记录中的一个字段。

(上机实验中，我们只按照设定好的例子进行基本的数据表创建、主键、外键设定。实际应用中需要根据需要进行数据表的设计等工作，感兴趣的同学可查阅资料自行学习实践。)

图 3-1 为创建数据表的命令。阅读代码理解含义，并自行在查询窗口输入 (注意以学号命名)，点击“执行”运行代码。(左右两图为同一个文件)

在对象资源管理器中右击展开数据库下的表目录，刷新，即可看到新创建的数据表，如图 2-2。

右击图 3-2 中的数据库关系图，点击“新建关系库数据图”，将三个表全部选中并添加，即可看到当前所设计表之间的关系。如图



图 3-1 使用命令创建数据表- 依次创建 STUDENT、COURSE、SCORE 表，设置外键

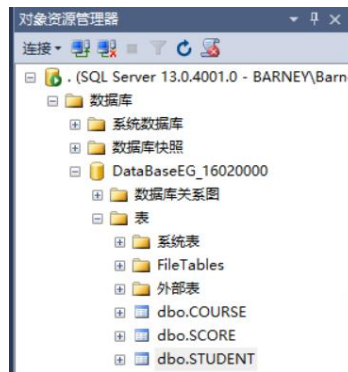


图 3-2 新创建的数据表

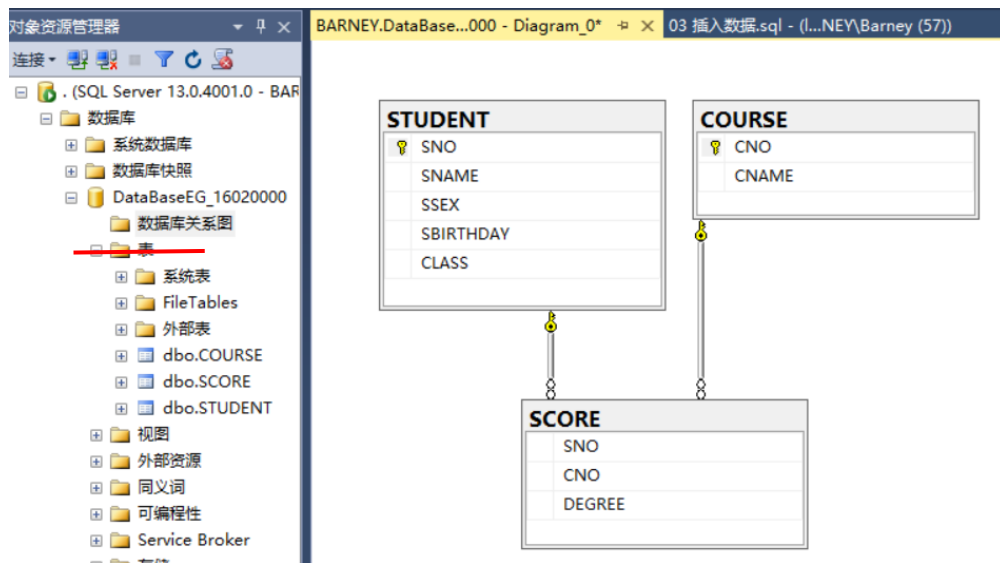
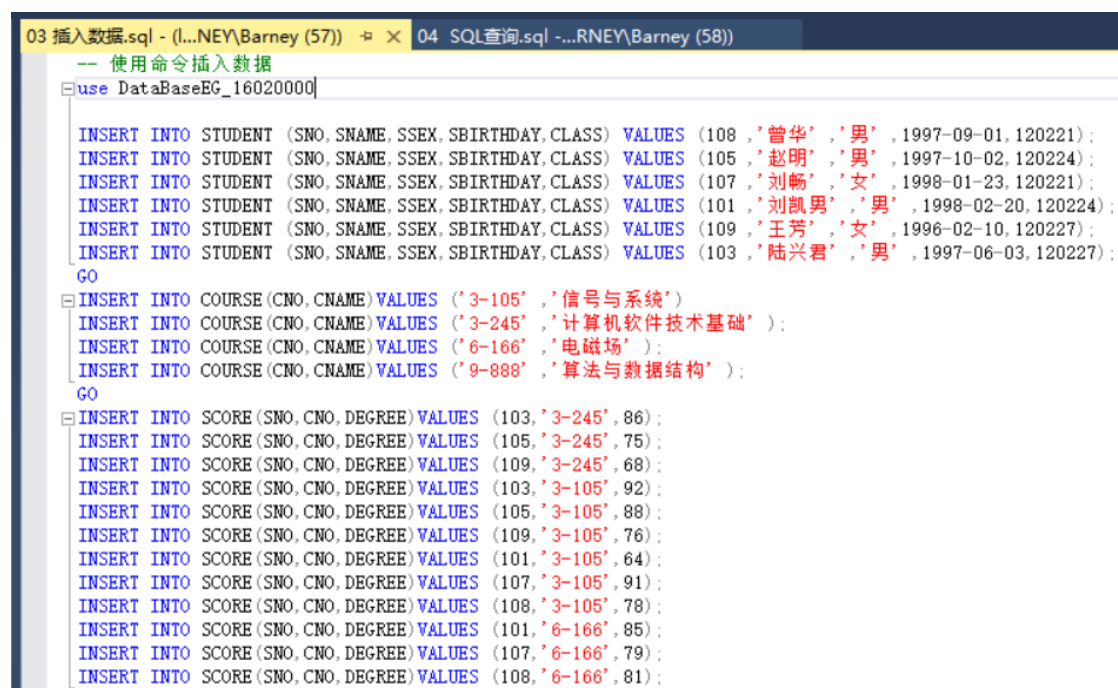


图 3-3 数据库关系图

4. 使用命令添加数据

图 4-1 为添加数据的命令。阅读代码理解含义，并自行在查询窗口输入，点击“执行”运行代码。执行命令是注意底部窗口的执行反馈信息，确保执行成功。

至此，数据已经输入完成。可以通过在对象资源管理器中右击某一数据表，选择“编辑前 200 行”，查看输入的数据。如图 4-2 所示。



```
03 插入数据.sql - (\\...NEY\Barney (57))  X  04 SQL查询.sql - ...RNEY\Barney (58))
-- 使用命令插入数据
use DataBaseEG_16020000

INSERT INTO STUDENT (SNO, SNAME, SSEX, SBIRTHDAY, CLASS) VALUES (108, '曾华', '男', 1997-09-01, 120221);
INSERT INTO STUDENT (SNO, SNAME, SSEX, SBIRTHDAY, CLASS) VALUES (105, '赵明', '男', 1997-10-02, 120224);
INSERT INTO STUDENT (SNO, SNAME, SSEX, SBIRTHDAY, CLASS) VALUES (107, '刘畅', '女', 1998-01-23, 120221);
INSERT INTO STUDENT (SNO, SNAME, SSEX, SBIRTHDAY, CLASS) VALUES (101, '刘凯男', '男', 1998-02-20, 120224);
INSERT INTO STUDENT (SNO, SNAME, SSEX, SBIRTHDAY, CLASS) VALUES (109, '王芳', '女', 1996-02-10, 120227);
INSERT INTO STUDENT (SNO, SNAME, SSEX, SBIRTHDAY, CLASS) VALUES (103, '陆兴君', '男', 1997-06-03, 120227);
GO

INSERT INTO COURSE (CNO, CNAME) VALUES ('3-105', '信号与系统');
INSERT INTO COURSE (CNO, CNAME) VALUES ('3-245', '计算机软件技术基础');
INSERT INTO COURSE (CNO, CNAME) VALUES ('6-166', '电磁场');
INSERT INTO COURSE (CNO, CNAME) VALUES ('9-888', '算法与数据结构');
GO

INSERT INTO SCORE (SNO, CNO, DEGREE) VALUES (103, '3-245', 86);
INSERT INTO SCORE (SNO, CNO, DEGREE) VALUES (105, '3-245', 75);
INSERT INTO SCORE (SNO, CNO, DEGREE) VALUES (109, '3-245', 68);
INSERT INTO SCORE (SNO, CNO, DEGREE) VALUES (103, '3-105', 92);
INSERT INTO SCORE (SNO, CNO, DEGREE) VALUES (105, '3-105', 88);
INSERT INTO SCORE (SNO, CNO, DEGREE) VALUES (109, '3-105', 76);
INSERT INTO SCORE (SNO, CNO, DEGREE) VALUES (101, '3-105', 64);
INSERT INTO SCORE (SNO, CNO, DEGREE) VALUES (107, '3-105', 91);
INSERT INTO SCORE (SNO, CNO, DEGREE) VALUES (108, '3-105', 78);
INSERT INTO SCORE (SNO, CNO, DEGREE) VALUES (101, '6-166', 85);
INSERT INTO SCORE (SNO, CNO, DEGREE) VALUES (107, '6-166', 79);
INSERT INTO SCORE (SNO, CNO, DEGREE) VALUES (108, '6-166', 81);
```

图 4-1 使用命令添加数据

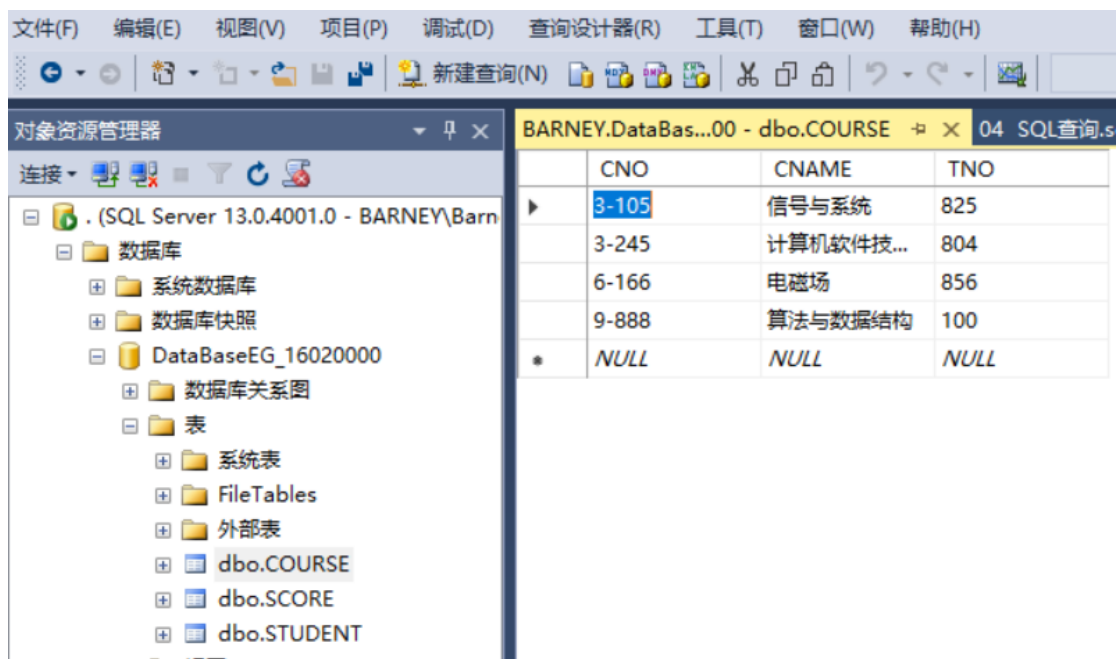


图 4-2 查看已输入的数据

5. SQL 查询

【相关知识】

SQL 是用于访问和处理数据库的标准的计算机语言。我们前边进行数据库、数据表创

建、数据输入使用的便是 SQL 语言。

SQL 语言分为两个部分：数据操作语言 (DML) 和 数据定义语言 (DDL)。

SQL (结构化查询语言)是用于执行查询的语法。但是 SQL 语言也包含用于更新、插入和删除记录的语法。查询和更新指令构成了 SQL 的 DML 部分：

- **SELECT** - 从数据库表中获取数据 (接下来实验的主要内容)
- **UPDATE** - 更新数据库表中的数据
- **DELETE** - 从数据库表中删除数据
- **INSERT INTO** - 向数据库表中插入数据 (第 4 步中插入数据有使用)

SQL 的数据定义语言 (DDL) 部分使我们有能力创建或删除表格。我们也可以定义索引 (键), 规定表之间的链接, 以及施加表间的约束。

SQL 中最重要的 DDL 语句:

- **CREATE DATABASE** - 创建新数据库 (第 2 步中创建数据库有使用)
- **ALTER DATABASE** - 修改数据库
- **CREATE TABLE** - 创建新表 (第 3 步中创建数据表有使用)
- **ALTER TABLE** - 变更 (改变) 数据库表 (第 3 步中设置外键有使用)
- **DROP TABLE** - 删除表
- **CREATE INDEX** - 创建索引 (搜索键)
- **DROP INDEX** - 删除索引

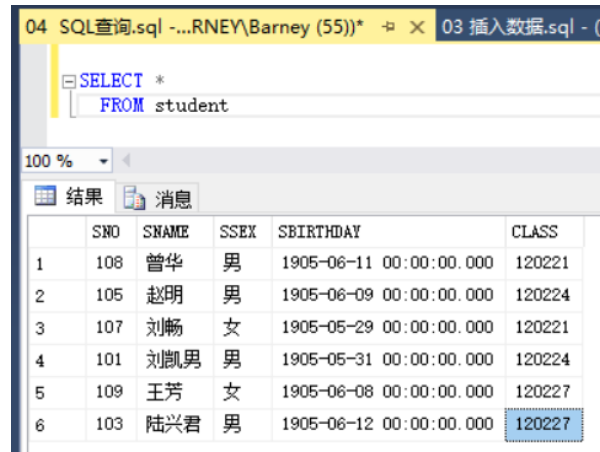
(索引、视图等知识我们实验中未涉及, 感兴趣的同学可查阅资料自行学习实践。)

接下来, 我们将通过实例学习 SELECT 语句的语法。在自己电脑上输入图片中对应代码进行实践。

5.1 基本语法

SELECT 列名称 FROM 表名称

(1) SELECT 后边跟“*”表示查询所有列



The screenshot shows a SQL query window with the following content:

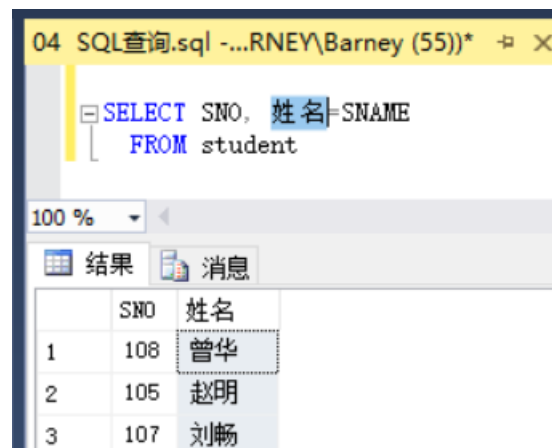
```
04 SQL查询.sql -...RNEY\Barney (55))* X 03 插入数据.sql - (0
SELECT *
FROM student
```

The results are displayed in a table with the following columns: SNO, SNAME, SSEX, SBIRTHDAY, and CLASS.

	SNO	SNAME	SSEX	SBIRTHDAY	CLASS
1	108	曾华	男	1905-06-11 00:00:00.000	120221
2	105	赵明	男	1905-06-09 00:00:00.000	120224
3	107	刘畅	女	1905-05-29 00:00:00.000	120221
4	101	刘凯男	男	1905-05-31 00:00:00.000	120224
5	109	王芳	女	1905-06-08 00:00:00.000	120227
6	103	陆兴君	男	1905-06-12 00:00:00.000	120227

图 5.1-1

(2) 查询多个列，之间用逗号隔开；在列名前可以指定显示结果中列的名称



The screenshot shows a SQL query window with the following content:

```
04 SQL查询.sql -...RNEY\Barney (55))* X
SELECT SNO, 姓名=SNAME
FROM student
```

The results are displayed in a table with the following columns: SNO and 姓名.

	SNO	姓名
1	108	曾华
2	105	赵明
3	107	刘畅

图 5.1-2

(3) 可以在语句中加入运算，例如 SCORE 表中所有学生分数加 5：

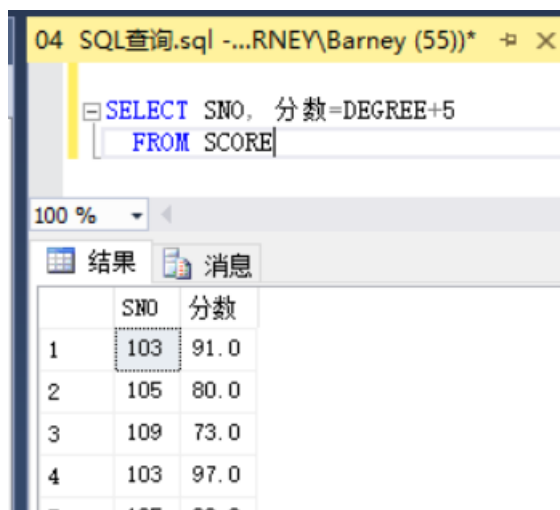


图 5.1-3

(4) 使用 ORDER BY 语句可以将结果按照指定的列排序，默认升序 (ASC)。

使用 DESC 关键字可改为降序。下列左图为成绩升序排列，右图为降序。

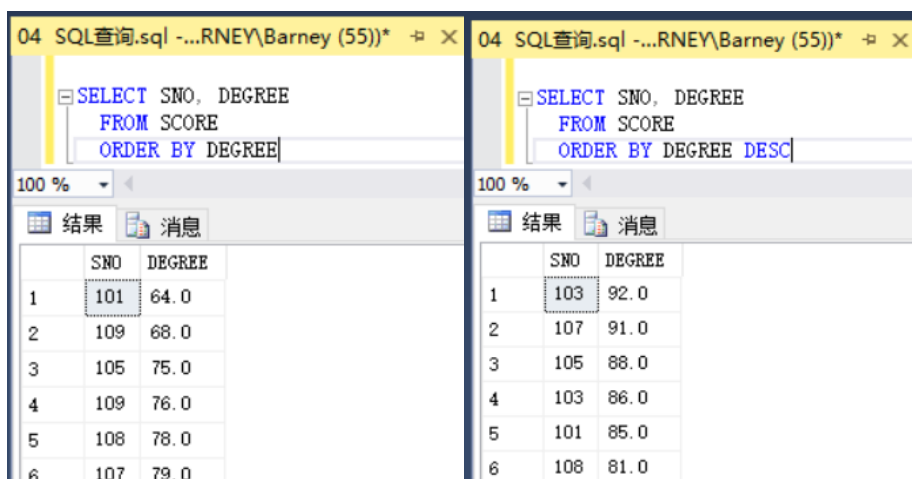
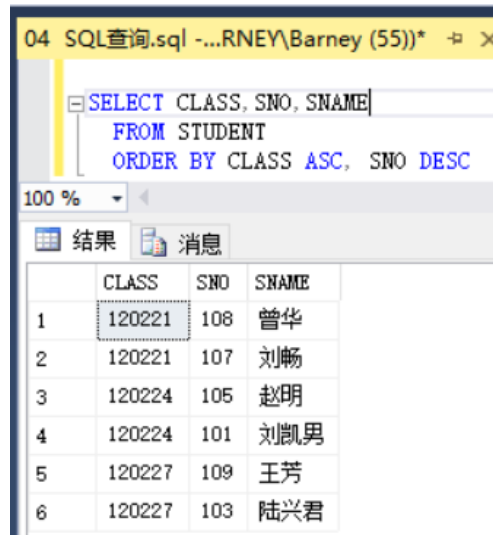


图 5.1-4-1

图 5.1-4-2

(5) 多个列联合排序，可分别指定。下图先按照班级升序，再按照学号降序。

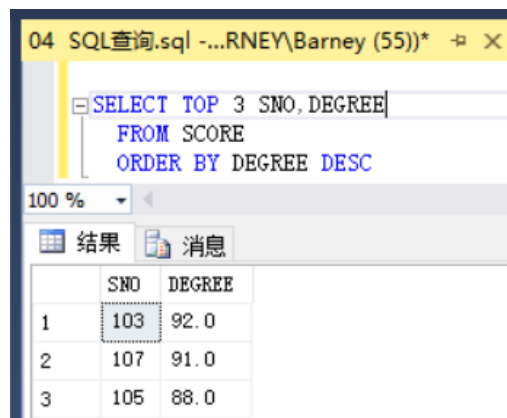


```
SELECT CLASS, SNO, SNAME
FROM STUDENT
ORDER BY CLASS ASC, SNO DESC
```

	CLASS	SNO	SNAME
1	120221	108	曾华
2	120221	107	刘畅
3	120224	105	赵明
4	120224	101	刘凯男
5	120227	109	王芳
6	120227	103	陆兴君

图 5.1-5

- (6) 使用 TOP n 关键字查询符合条件的前 n 条数据。例如选择分数最高的三条记录。



```
SELECT TOP 3 SNO, DEGREE
FROM SCORE
ORDER BY DEGREE DESC
```

	SNO	DEGREE
1	103	92.0
2	107	91.0
3	105	88.0

图 5.1-6

5.2 SELECT 配合 WHERE 语句进行条件查询

SELECT 列名称 FROM 表名称

[WHERE <查询条件>]

- (1) 通过逻辑关系“= > >= < <= AND OR BETWEEN ”进行特定的筛选。

下列图片给出部分示例。

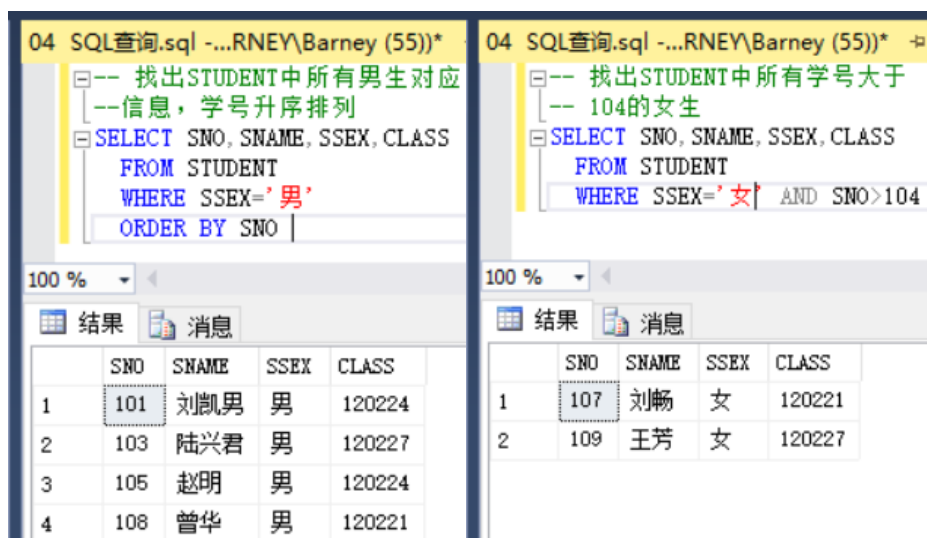


图 5.2-1-1

图 5.2-1-2

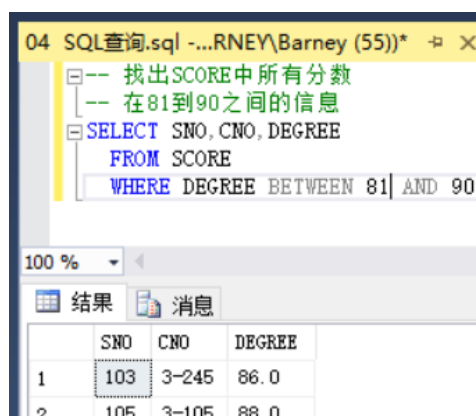


图 5.2-1-3

(2) 通过“LIKE”配个“%”（表示任意长度字符串）或“_”（代表单个字符串）进行模糊查询。示例如下：

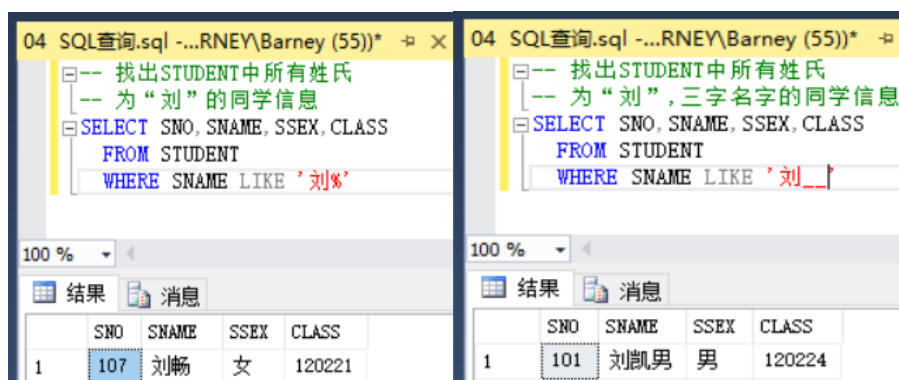


图 5.2-2

5.3 聚合函数

通过聚合函数“COUNT、MAX、MIN、SUM、AVG”等，可实现数据的简单统计、筛选。示例如下



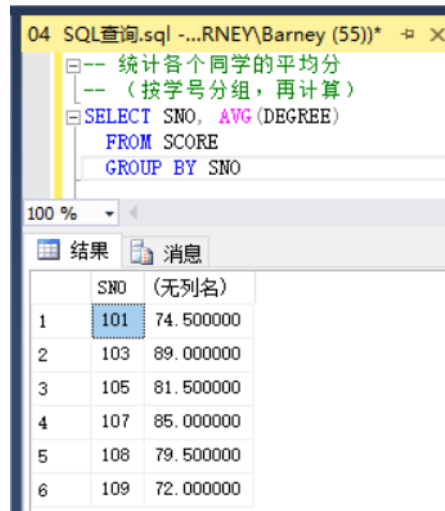
图 5.2-3

5.4 统计汇总：GROUP BY、HAVING 字句

(1) 使用“GROUP BY”语句对数据进行统计汇总，

“GROUP BY”表示根据一定规则进行分组，以便于对各个分组的数据进行处理；常与

聚合函数一起使用。示例如下：



04 SQL查询.sql -...RNEY\Barney (55))*

```
-- 统计各个同学的平均分  
-- (按学号分组, 再计算)  
SELECT SNO, AVG(DEGREE)  
FROM SCORE  
GROUP BY SNO
```

100 %

结果 消息

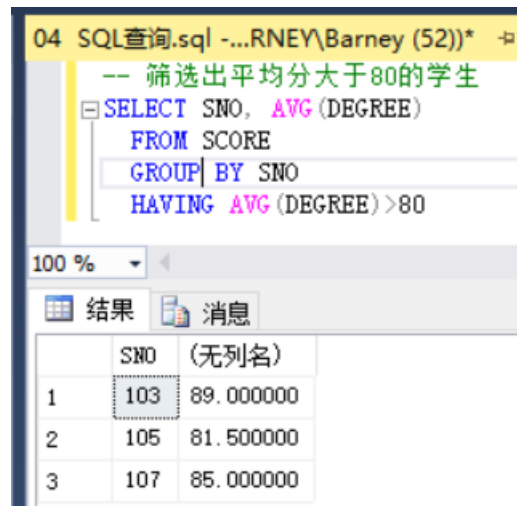
	SNO (无列名)	
1	101	74.500000
2	103	89.000000
3	105	81.500000
4	107	85.000000
5	108	79.500000
6	109	72.000000

图 5.2-4

(2) 使用“HAVING”语句对数据进行统计汇总

“HAVING”语句与“WHERE”功能类似，都可以设定条件以使查询结果满足一定条件限制。但“HAVING”可以用于对聚合函数进行筛选，“WHERE”则不可以。

例如(4)中求出不同学生的平均分，为了进一步限制，只显示平均分大于80分的学生，实现代码如下：



04 SQL查询.sql -...RNEY\Barney (52))*

```
-- 筛选出平均分大于80的学生  
SELECT SNO, AVG(DEGREE)  
FROM SCORE  
GROUP BY SNO  
HAVING AVG(DEGREE) > 80
```

100 %

结果 消息

	SNO (无列名)	
1	103	89.000000
2	105	81.500000
3	107	85.000000

图 5.2-5

5.5 多表查询

以上各个示例均是单张表中信息的查询。实际应用中，多张表的联合查询是非常常见的。例如我们需要从 STUDENT 表中获取学生信息，同时从 SCORE 表中调出学生的成绩信息。

多表查询有两张方法，一种是利用 WHERE 语句，另一种是利用 join...on 语句。下边四幅图中，前两张图给出 WHERE 语句查询两张、三张表的方法；后两张图给出利用 join...on 查询两张、三张表的方法。

The screenshot shows a SQL query window with the following text:

```
-- 联合查询STUDENT, SCORE表
-- 显示姓名、班级、课程号、成绩信息
SELECT SNAME, CLASS, CNO, DEGREE -- 两个表中各自独有的列名，直接列出
FROM STUDENT, SCORE -- 列出联合查询的两张表
WHERE STUDENT.SNO = SCORE.SNO -- 利用多表中的共有信息（SNO）进行筛选
ORDER BY CLASS
```

Below the query, the results are displayed in a table:

	SNAME	CLASS	CNO	DEGREE
1	刘畅	120221	3-105	91.0
2	刘畅	120221	6-106	79.0
3	曾华	120221	3-105	78.0
4	曾华	120221	6-106	81.0

图 5.2-6-1 使用基本语法同时查询两个表

The screenshot shows a SQL query window with the following text:

```
-- 联合查询STUDENT, SCORE, COURSE表
-- 显示姓名、班级、课程号、成绩信息
SELECT SNAME, CLASS, SCORE.CNO, DEGREE, CNAME
-- 两个表中各自独有的列名，直接列出；SNO为共有列，需要指定，格式为“表名.列名”
FROM STUDENT, SCORE, COURSE -- 列出联合查询的两张表
WHERE STUDENT.SNO = SCORE.SNO AND SCORE.CNO = COURSE.CNO
-- 利用多表中的共有信息（SNO、CNO）进行筛选
ORDER BY CNAME
-- 按照课程升序
```

Below the query, the results are displayed in a table:

	SNAME	CLASS	CNO	DEGREE	CNAME
1	刘凯男	120224	6-166	85.0	电磁场

图 5.2-6-2 使用基本语法同时查询三个表

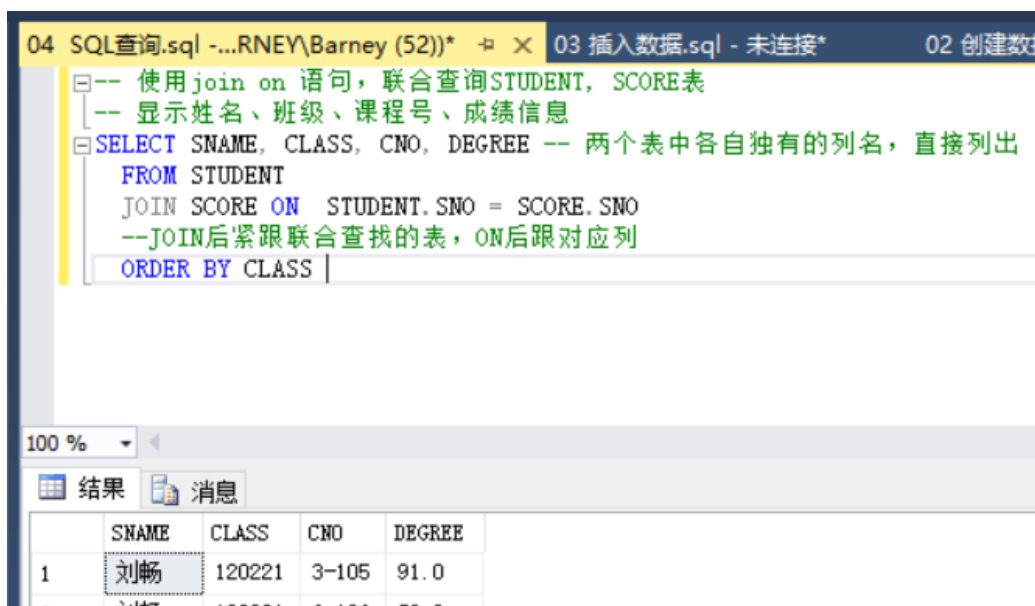


图 5.2-6-3 使用 join...on 语句同时查询两个表

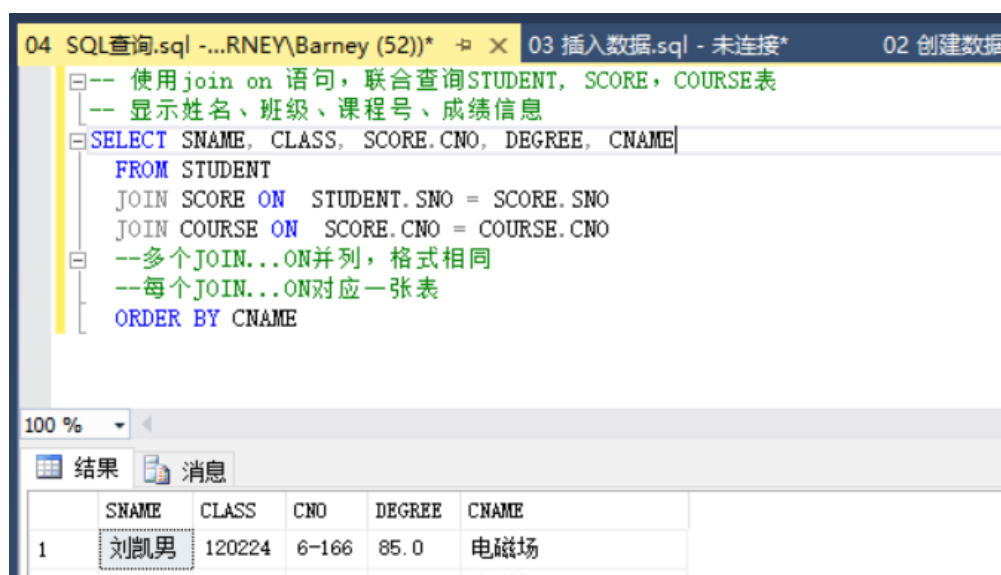


图 5.2-6-4 使用 join...on 语句同时查询三个表

总结：

SQL 查询的基本格式为：

```

SELECT [ALL/DISTINCT] < * / 目标列或目标列表达式> [<, 目标列或目标列表达式>]...
FROM <表名或视图名> [<, 表名或视图名>]...
[WHERE <条件表达式>]
[GROUP BY <列名1>]
[ORDER BY <列名2> [ASC/DESC]]:
--在< >中的项不可缺，[ ] 中的项可选。

```

基本的 SELECT 语句配合 WHERE 语句，可以实现按条件查询；

通过聚合函数 (MAX、AVG 等) 配合 GROUP BY、HAVING 语句 , 可以实现对数据的统计汇总 ;

通过 WHERE 或 JOIN...ON 语句 , 可以实现多表联合查询。

四、 测验题目

使用 INSERT 命令 , 在 STUDENT 表中插入你的信息 , 其中学号为 123 , 其余信息按照你的个人信息填写。

完成下列查询操作 : (所有查询代码编辑在一个查询文件中 , 方便检查)

- (1) 显示 STUDENT 表的全部信息 , 各列名称改为中文信息“学号、姓名、性别、生日、班级” , 按照学号降序排列。
- (2) 筛选出 STUDENT 表中和你姓名**字数**相同的学生的下列信息 : 学号、姓名、班级。
先按照班级升序 , 再按照学号降序排列。
- (3) 筛选出 STUDENT 表中和你相同性别、学号大于 105 的学生的全部信息 , 按照学号升序排列。
- (4) 统计 COURSE 表中共有几门课程。
- (5) 计算每门课程的平均分 , 显示下列信息 : 课程编号、课程名称、平均分 , 按照课平均分降序排列。
- (6) 筛选出所有平均分在 80-100(含) 之间的学生 , 显示下列信息 : “学号 姓名 班级 平均分” , 按照课平均分降序排列。