



第 7 讲 数据库查询

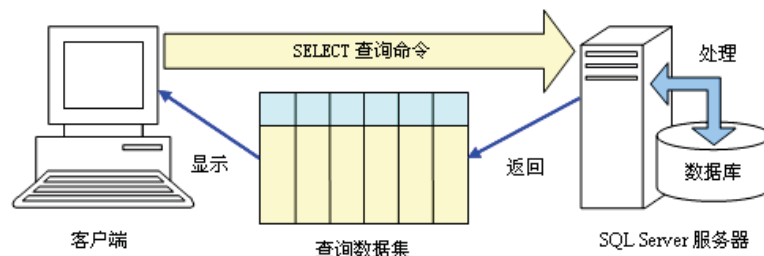
课时内容	数据库查询	授课类型	理论+实践	课时	16
教学目标	☒ SELECT 数据查询语句的语法格式和使用。 ☒ 投影查询、连接查询和选择查询。 ☒ 分组统计查询、限定查询、排序查询和保存查询。				
教学重点	☒ SELECT 数据查询语句的语法格式和使用。 ☒ 投影查询、连接查询和选择查询的熟练应用。 ☒ 分组统计查询、限定查询、排序查询和保存查询的正确应用。				
教学难点	☒ 连接查询。 ☒ 选择查询中逻辑表达式的应用。 ☒ 分组统计查询、限定查询、排序查询。				
教学设计	1. 教学思路：（1）启发讨论。本单元的教学内容是数据库技术中最重要的部分，SELECT 数据查询语句表面看起来就是一句话，似乎比较简单，但实际应用起来有很多困惑。因此，倡导启发式与讨论式教学方法，注重学生思维的训练，帮助学生树立积极、自信和严谨的解决问题的态度，重点传授问题解决的步骤，以提高解决问题的能力。（2）大量任务训练。标准 SQL 的 SELECT 数据查询语句与 T-SQL 的 SELECT 数据查询语句其语法和用法大同小异。因此，通过完成本单元大量的任务，熟练掌握 SELECT 数据查询语句的语法和用法有着至关重要的意义。 2. 教学手段：（1）案例贯穿。本学习情境采用全程案例贯穿法进行实施，本教学单元的所有任务结果都将存储在上一单元建立的数据库中，并将为下一单元的任务训练打下基础。（2）任务驱动。本学习情境采用任务驱动法进行实施，每位学生的每项工作任务必须正确完成，才能进行到下一个任务，在工作任务的驱动和全程任务考核评价的监控下，培养作风严谨和坚持不懈的职业责任感。（3）通过课堂练习和课后习题拓展学生的实际应用能力。 3. 教学资料及要求：学习通课程资源平台提供了教学视频、课件及单元测验。				
教学方法	问题导向法 讲授法 实例演示法 实践法				
教学内容及过程提要					
一、复习上次课内容 二、新课引入 三、授课内容 核心知识点 1.SELECT 数据查询语句的语法格式和使用。 2.投影查询、连接查询和选择查询。 3.分组统计查询、限定查询、排序查询和保存查询。四、实验内容 五、小结					
教学内容及过程					
复习： 回顾关系模型的三个专门运算（投影、选择和连接） 新课导入：					

自主探究问题导向：1、如何从教务管理系统中查询成绩？
2、如何查找一本书是否被借出？

授课内容：

一、查询的概念

查询(Query)又称为检索，是数据库最核心、最基本的操作之一。查询操作用来从数据表或视图中迅捷地搜索并提取所需数据。



查询命令运行原理

二、语法格式

SELECT 语句的基本语法格式如下：

```
SELECT [ ALL | DISTINCT ] [ TOP ( <数值表达式> ) [ PERCENT ] ] <目标项列表>
[ INTO <新表名> ]
FROM <源表名或视图名> [ ,...n
[ WHERE <查询条件表达式> ]
[ GROUP BY <分组表达式>
[ HAVING <条件表达式> ] ]
[ ORDER BY <排序表达式> [ ASC | DESC ] ]
[ COMPUTE { 聚合函数( <列名/表达式> ) } [ ,...n ] [ BY <分类表达式> ]
```

三、无条件查询单一表

无条件查询单一表是指查询操作仅限于对一个数据表进行，并且不使用条件查询子句筛选记录。

案例 1：查询学生信息表中的学号、姓名、性别和出生日期。

案例 2：查询课程信息表中的所有课程信息。

案例 3：从 student 数据库的课程信息表里查询出学分对应的课程学时（假设每 18 学时计 1 学分）。

案例 4：从 student 数据库的课程信息表里查询出学分对应的课程学时，并将“学时”作为单位输出。

案例 5：从 student 数据库的课程信息表里查询出学分对应的课程学时，并将“学时”作为列单独内容输出。

案例 6：从 student 数据库学生信息表中查询学生的学号、姓名、性别和年龄，要求年龄列的列标题为“年龄”。

案例演示，通过案例讲解目标输出项的内容

四、过滤查询结果记录

案例演示，通过案例讲解目标输出项的内容

1、保留或过滤相同数据行

案例 7：从学生成绩表中查询哪些学生选修了课程，列出学生的学号（要求学号唯一，不重复）。

2、选定相邻若干数据行

案例 8：显示“案例 7”的查询结果中的前 5 条记

案例 9：显示“案例 7”的查询结果中的前 20%的记录。

五、条件查询筛选数据行

1、条件查询概念

可以通过 SELECT 语句的 WHERE 子句构造记录的筛选条件，选择出满足条件的部分记录，组成查询数据集，从而避免检索出数据表中所有的数据行，这就是条件查询。

2、语法格式

WHERE 子句一般放在 FROM 子句后面。基本语法格式如下：

SELECT <目标项列表>

FROM <表名或视图名>

WHERE <查询条件表达式>

WHERE 子句的运算符：

运算符类型	运算符	作用与意义
比较运算符	=, !=, <>, >, >=, !>, <, <=, !<	比较两个值的大小
逻辑运算符	AND, OR, NOT	用来构造多重复合条件
范围运算符	BETWEEN AND, NOT BETWEEN AND	判断值是否在范围内
集合运算符	IN, NOT IN	判断值是否在列表集合中
空值运算符	IS NULL, IS NOT NULL	测试字段是否为空值
字符匹配运算符	LIKE, NOT LIKE	用于模糊查询

3、案例演示讲解：

(1) 简单查询

案例 10：从教师信息表中查询职称为教授的教师信息。

案例 11：从学生成绩表中查询考试成绩不及格的学生及课程信息。

(2) 复合条件查询

案例 12：从教师信息表中查询职称为教授的女教师信息。

案例 13：从教师信息表中查询职称为教授和副教授的教师信息。

案例 14：查询年龄在 20~28 岁之间（包括 20 和 28 岁）的学生的学号和姓名，年龄(通过计算得到的输出列)

案例 15：从教师信息表中查询职称为教授和副教授的教师信息。

(3) 模糊查询

模糊查询是指在一些只针对字符型数据或日期时间型数据进行查询的应用中，无法给出精确的检索条件，只能提供一些并不完全确定的线索来搜索信息。

用 LIKE 或 NOT LIKE 关键字可以构造特殊的条件表达式，用于检索与特定字符串匹配的字符型或日期时间型数据。模糊查询的条件表达式语法格式如下：

匹配表达式 [NOT] LIKE '<匹配模式串>'

通配符：

通配符	符号描述	通配符表示的对象	示例(假定查询的为 Readers 表)
%	百分号	代表零个或多个任意字符串	姓名 LIKE '刘%', 查询姓刘的所有记录
			姓名 LIKE '%小%', 查询所有姓名中含“小”字的记录
_	下划线	代表一个任意字符，一个汉字要对应两个下划线	姓名 LIKE '徐__', 查询姓徐且名为单字的所有记录
[]	方括号表示集合或范围	代表方括号中列出的或属于指定范围内的任何单一字符	姓名 LIKE '[赵孙李]%', 查询姓赵、孙或李的所有记录
[^]	括在方括号中的^	代表不在方括号中列出或不属于指定范围内的任何单一字符	姓名 LIKE '[^马]%', 查询不姓马的所有记录

案例 16：查询姓张的同学信息。

案例 17：查询名字中倒数第二个字为“晓”的同学信息。

案例 18：查询以“计算机_”开头，倒数第 2 个字为“信”的课程的信息。

案例 19：把不是姓张和姓王的同学都找出来。

六、查询的处理子句

1、查询中的排序

(1) 概念

SELECT 语句返回的查询数据集中的记录通常是按表中记录的物理顺序排列的。

可通过排序手段来改变查询数据集中记录的显示顺序，也就是对数据排序的意义。

(2) 语法格式

通过在 SELECT 语句中使用 ORDER BY 子句对查询数据集中的记录按照一个或多个列的值进行排序输出。排序依照的列又称为排序关键字。

ORDER BY 子句的格式为：
ORDER BY { <排序表达式> [ASC | DESC] } [...n]

案例 1：查询学生信息表“计算机软件 1 班”的同学信息，按性别升序，出生日期降序排列。

2、聚合函数的应用

(1) 概念

聚合函数也称为聚集函数或分组函数，这类函数将数据表在某一字段列上的值或与列相关的表达式值作为一个整体，进行某些特定的运算，或对一组记录中指定的列值或与列相关的表达式值进行某些特定的运算，并返回单个的数值做为函数值。

函数参数可以为字段列名、常量或由算术运算符连接而成的表达式。函数返回值作为新的一列出现在查询结果集中。

(2) 聚合函数

函数名	功能	应用格式
COUNT	计算并返回记录个数	COUNT(*)
	计算并返回指定列或表达式中值项的个数	COUNT([ALL DISTINCT] <列名/表达式>)
SUM	计算并返回指定列或表达式中值项的总和	SUM([ALL DISTINCT]<列名/表达式>)
AVG	计算并返回指定列或表达式中值项的平均值	AVG([ALL DISTINCT]<列名/表达式>)
MAX	计算并返回指定列或表达式中值项的最大值	MAX([ALL DISTINCT]<列名/表达式>)
MIN	计算并返回指定列或表达式中值项的最小值	MIN([ALL DISTINCT]<列名/表达式>)
VAR	计算并返回给定列或表达式值的统计方差	VAR([ALL DISTINCT]<列名/表达式>)
STDEV	计算并返回给定列或表达式值的统计标准偏差	STDEV([ALL DISTINCT]<列名/表达式>)

(3) 案例

案例 2：统计学生信息表中的男生人数。

案例 3：从学生信息表中统计出班级个数，每个班级只统计一次。

案例 4：从学生成绩表中统计出学习了“001”号课程的学生人数，总成绩，平均成绩，最高分，最低分，标准差，方差。

1、 查询中实施分组

(1) 概念

GROUP BY 子句则根据一列或多列的值对查询数据集进行分组，将数据在指定列上具有相同值的记录划分在一起，组成一组。

对查询数据集分组的目的是为了细化聚合函数的作用对象，并借助聚合函数对每一组的记录进行统计，从而最终生成汇总的数据集。

(2) 语法格式

GROUP BY 子句的语法格式如下：

GROUP BY [ALL] <分组表达式> [,...n] [WITH { CUBE | ROLLUP }] [HAVING <条件表达式>]

GROUP BY 子句用来对查询操作实施分组处理，HAVING 子句则设置筛选条件。

(3) 案例

案例 5：从学生信息表中统计出各专业的学生数。要求查询结果显示专业名称和人数两列。

案例 6：统计学生信息表中的男女生人数。

案例 7：从学生信息表中统计出各专业的学生数。要求查询结果显示专业名称和人数两列，且只显示专业人数大于 5 的。

案例 8：从学生成绩表中找出学习了三门课以上的学生。要求查询结果显示学号和课程数。

4、查询中实施分类汇总

(1) 概念

COMPUTE 子句用于对查询数据集中的记录进行分类汇总，并产生一个额外的汇总记录行，插入到查询结果数据集的末尾。COMPUTE 子句通过聚合函数，对列或表达式进行统计计算。

(2) 语法格式

COMPUTE 子句的语法格式如下：

[COMPUTE { { SUM | AVG | COUNT | MAX | MIN | VAR | STDEV } (<列名/表达式>) } [,...n] [BY { <分类表达式> } [,...n]]

(3) 案例

案例 9：对学生成绩表中的成绩进行统计，统计所有学生考试的总成绩和平均成绩。

案例 10：对学生成绩表中的每个学生统计他所学课程的门数，总成绩和平均成绩。

5、保存查询结果为表

(1) 概念

查询结果不仅可以显示查看，还可以作为一个数据表永久保存起来。

INTO 子句用来将查询结果数据集保存到指定名称的数据表中。INTO 子句的语法格式如下：

INTO <新表名>

(2) 案例

案例 11：查询学生中成绩小于 60 分的课程号并将查询结果以新表的形式建立。

七、多表复杂查询

1、概念

连接查询通过构造连接字段条件，从多个具有一定关联的表中检索所需的数据。

用于连接查询的多个表之间必须要有某种联系，通常表现为这些表之间存在着意义相同的字段列，这是连接查询存在的必要性。

2、分类

多表连接查询中，连接可分为以下几类：

内连接；

交叉连接；

自连接；

外连接；

复合条件连接。

3、连接类型

SQL Server 实现连接查询的方法包括以下两种：

在 FROM 或 WHERE 子句中指定连接条件；

使用 JOIN 关键字实现连接查询。即：ANSI（American National Standards Institute—美国国家标准学会）连接语法形式。

4、语法格式

JOIN 关键字实现连接的语法格式如下：

```
SELECT <目标项列表> FROM <表名 1>[连接类型] JOIN <表名 2> [ON (<连接条件>)] [ WHERE <查询条件表达式> ] [ GROUP BY <分组表达式> [ HAVING <条件表达式> ] ] [ ORDER BY <排序表达式> [ ASC | DESC ] ] [;]
```

5、内连接

（1）概念

内连接是用来组合两个表的最常见操作。

内连接需要构造一个称为连接条件的表达式，该表达式包含两个表共有的字段列与比较运算符。

进行内连接查询时，根据查询条件和连接条件，对两个表中的记录进行筛选，将所有满足条件的数据行组合起来，生成查询结果数据集，查询结果集中仅包含那些满足连接条件的数据行。

（2）实现方式

可有以下两种方法实现内连接的查询语句：

使用[INNER]JOIN 关键字实现内连接，并用 ON 关键字指定连接条件；

不使用 JOIN 关键字实现内连接。

（3）语法格式

使用 JOIN 关键字实现内连接的命令格式：

```
SELECT <目标项列表> FROM <表名 1> [INNER] JOIN <表名 2> ON <连接条件> [ WHERE <查询条件表达式> ] [ GROUP BY <分组表达式> [ HAVING <条件表达式> ] ] [ ORDER BY <排序表达式> [ ASC | DESC ] ] [;]
```

案例 1：从图书管理数据库“Library”中查询每个读者的详细信息（读者及读者类型），允许有重复列。

案例 2：查询“韩梅梅”还要哪些书没有归还（显示借书号，读者名，书名、借阅日期、应归还日期）。

案例 3：查询借阅过《Python 编程快速上手-让繁琐工作自动化》书的读者的详细信息（包括读者号，读者名，读者类型和所属部门，借阅时间，书名）。

案例 4：查询每个读者的详细信息（包括读者及借阅图书的信息），不允许有重复列。

不使用 JOIN 关键字实现内连接的命令格式：

```
SELECT <目标项列表> FROM <表名 1>, <表名 2> WHERE [<表名 1>.<字段列名 1> <运算符> [<表名 2>.<字段列名 2>
```

案例 5：查询“韩梅梅”还要哪些书没有归还（显示借书号，读者名，书名、借阅日期、应归还日期）。

案例 6：查询“韩梅梅”还要哪些书没有归还（显示借书号，读者名，书名、借阅日期、应归还日期）。比较两种实现方式的性能。

6、交叉连接

(1) 概念

交叉连接又称为非限制连接，其实质是将两个或多个参与连接的表不加任何限制地组合起来，形成结果集。

交叉连接对应于集合运算意义上的广义笛卡尔积，运算结果集的列为所有参与表的字段列之和，记录行数为所有参与表的行数之乘积值。

等价于 FROM 表名 1，表名 2 之后不加 WHERE 连接条件逻辑表达式。

一般没有意义。

(2) 实现方式

针对两个表的情况，实现交叉连接查询时，根据是否包含 JOIN 关键字，可有以下两种实现方法：

使用 JOIN 关键字实现连接查询；

不使用 JOIN 关键字实现连接查询。

(3) 语法格式

使用 JOIN 关键字实现连接查询的命令格式：

SELECT * FROM <第 1 个表名> CROSS JOIN <第 2 个表名>

不使用 JOIN 关键字实现连接查询的命令格式：

SELECT <表名 1>.*, <表名 2>.* FROM <表名 1>, <表名 2>

案例 7：将读者表和读者类型表交叉连接。

7、自连接

(1) 概念

一个表与自身进行连接，这种连接称为自连接。

自连接中，对于执行不同引用对象的同一个表，为避免使用相同表名而导致错误，必需为表命名别名，以示区别，使该表在逻辑上表现为两个表。

自连接能够同时操作同一个表的不同行，可以被看作是内连接的一种特例。

(2) 案例

案例 8：从图书管理数据库“Library”中查询出借了两本以上图书的读者的借书信息。

8、外连接

(1) 概念

当在应用中需要显示某个特定表的全部记录行，即使这些记录行并不满足查询条件与连接条件，此时就要用到外连接。

外连接操纵的两个表有主从之分，主表需要遵循一定的约束限制，而对从表却可以不进行任何限制。

(2) 实现方法

外连接需在 SELECT 语句中使用 OUTER JOIN 关键字。实现外连接的语法格式如下：

SELECT <目标项列表> FROM <表名 1> <外连接类型> [OUTER] JOIN <表名 2> ON <连接条件>
[WHERE <查询条件表达式>] [GROUP BY <分组表达式>] [HAVING <条件表达式>] [ORDER BY
<排序表达式> [ASC | DESC]]

(3) 类型

根据<外连接类型>值的不同，外连接可分为以下三种：

左外连接；

右外连接；
完全外连接。

(4) 案例

案例 9: 从表 “Reader” 和表 “Borrow” 中查询出读者的借阅情况，包括没有借过书的读者情况。

案例 10: 从表 “Borrow” 和表 “Book” 中查询出图书被借阅的情况，包括没有被借的图书情况。

案例 11: 借阅和读者全外连接。

八、嵌套查询

1、概念

嵌套查询又称子查询，是多个 SELECT 语句的一种嵌套包含结构，即在一个查询块的 WHERE 子句或 HAVING 子句中允许包含另一个查询块。

嵌套查询是 T-SQL 语句的扩展，能够将多个简单查询语句通过嵌套关系关联起来，创建出更为复杂的查询结果。

(1) [NOT] IN 谓词的用法

用法如下：

[NOT] IN 关键字用法如下：

SELECT-FROM WHERE <列名或表达式> [NOT] IN (内层/子查询) [;]

案例 12: 从图书管理数据库 “Library” 中查询出借阅过 “人民邮电出版社” 出版图书的读者编号（不包括重复的行）。

案例 13: 从图书管理数据库 “Library” 中查询没有借阅过图书的读者信息。

(2) ANY/ALL 谓词的用法

ANY 或 ALL 关键字用法如下：

SELECT-FROM WHERE <列名或表达式> <比较运算符> [ANY | ALL] (子查询) [;]

案例 14: 从图书管理数据库 “Library” 中查询比计算机学院所有读者办证时间都早的读者信息。

案例 15: 从图书管理数据库 “Library” 中查询比人民邮电出版社的某一本馆藏图书定价高的那些其他出版社（不在包括人民邮电出版社）出版的馆藏图书信息。

(3) [NOT] EXISTS 谓词的用法

带有 EXISTS 或 NOT EXISTS 谓词的嵌套查询用法如下：

SELECT-FROM WHERE <列名或表达式> [NOT] EXISTS (子查询) [;]

案例 16: 从图书管理数据库 “Library” 中查询那些从未被借阅的图书信息。

案例 17: 从图书管理数据库 “Library” 中，用 EXISTS 子查询查询出借阅了 “人民邮电出版社” 出版图书的读者编号。

案例 17: 从图书管理数据库 “Library” 中，用 EXISTS 子查询查询出借阅了 “人民邮电出版社” 出版图书的读者编号。

(4) 子查询在 UPDATE 语句

案例 18: 计算读者表 “Reader” 中的已借书数量列

案例 19: 计算借阅表 “Borrow” 中的应还日期列

九、嵌套查询

1、概念

联合查询也称集合查询，是一种以两个或多个 SELECT 查询结果数据集为操作对象，通过对这些查询结果数据集实施并、交、差等集合运算，获取更加复杂、更加综合的数据结果的一种查询方

法。 2、运算符 联合查询的运算符主要包括以下三种： 求并(UNION)； 求交(INTERSECT)； 求差(EXCEPT)。 3、语法格式 实现联合查询的语法格式如下： <SELECT 语句 1> { { UNION INTERSECT EXCEPT } <SELECT 语句 2> } [,...n]	
小结	1. 能够充分认识到 SELECT 数据查询是数据库的重要技术，是数据库的主要应用目的。熟练掌握各种数据查询为数据库应用系统的开发奠定基础。 2. 能够根据项目需求分析对表进行投影查询、连接查询、选择查询、分组统计查询、限定查询、排序查询、保存查询。 3. 能够阅读并熟练书写 SELECT 数据查询语句和有关参数
作业	1: 按照学号将数据集进行分组，然后分别统计各个组的平均成绩，并显示平均成绩大于 60 分的学生的学号和平均成绩。 2: 查询学生中成绩小于 75 分的课程号、成绩以及这些学生单人成绩的总和。 3: 查询平均成绩大于 60 分的这部分学生的平均成绩以及他们的总平均成绩。
上机实验内容	1、完成课堂所有案例 2、思考并完成作业
课程小计	