



多表复杂查询



1

多表连接查询概述

2

多表连接查询类型

3

嵌套查询

4

联合查询



连接查询的概念

- ☑ 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。



多表连接查询的类别

- ☑ 多表连接查询中，连接可分为以下几类：
- ① 内连接；
 - ② 交叉连接；
 - ③ 自连接；
 - ④ 外连接；
 - ⑤ 复合条件连接。



多表连接查询的类别

- ☑ SQL Server 2008实现连接查询的方法包括以下两种：
 - ① 在FROM或WHERE子句中指定连接条件；
 - ② 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。



连接条件的位置

- ☑ 通常在FROM子句中指定连接条件。这是因为：将连接条件放置在FROM子句中，更益于将连接条件与WHERE子句中指定的搜索条件分隔开来。



表别名

- ☑ 可以为数据表定义表别名。表别名的定义要放在FROM子句中。表别名定义的语法格式如下：
- ☑ `SELECT <目标项列表> FROM <表名称> [ AS ] <表别名> [ , ... n ]`



表别名的作用

- ☑ 表别名一旦定义，便可在任何地方使用它，来替代表的原名。
- ☑ 表别名常用于连接查询中，用来简化表名，缩短查询语句的长度。表别名还可用于自连结查询中，用来避免字段列前缀的重名现象。



JOIN连接的语法格式

- ☑ 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 ] ] [ ; ]
```




多表复杂查询

1

多表连接查询概述



2

多表连接查询类型

3

嵌套查询

4

联合查询



内连接的概念

- ☑ 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。



内连接的实现方法

- ☑ 可有以下两种方法实现内连接的查询语句：
 - ① 使用[INNER] JOIN关键字实现内连接，并用ON关键字指定连接条件；
 - ② 不使用JOIN关键字实现内连接。



使用JOIN实现内连接

- ☑ 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] [;]



使用JOIN实现内连接

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。

T-SQL代码：

```
Use Library
```

```
GO
```

```
select Reader.*,ReaderType.*
```

```
from Reader inner join ReaderType on Reader.TypeID=ReaderType.TypeID
```




使用JOIN实现内连接

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。

查询结果：

结果

消息

	RID	Rname	Rsex	Rbirthday	Sdept	TypeID	Lendnum	TypeID	Typename	LimitNum	LimitDays
1	2000186010	张子建	男	2000-09-03	计算机	1	0	1	教师	6	90
2	2000186011	赵良宇	男	2001-03-04	机电	1	1	1	教师	6	90
3	2003216008	张英	女	2000-05-06	经管	2	0	2	职员	4	60
4	2004060003	李亚茜	女	2002-09-02	计算机	1	0	1	教师	6	90
5	2004216010	任灿灿	女	1999-01-02	人文社科	1	1	1	教师	6	90
6	2011216117	孟霞	女	1998-02-03	人文社科	3	0	3	学生	3	30
7	2012216008	杨淑华	女	2002-01-05	计算机	3	1	3	学生	3	30
8	2012216009	程鹏	男	2001-03-09	计算机	3	2	3	学生	3	30
9	2019216001	韩梅梅	女	2001-07-09	计算机	3	0	3	学生	3	30
10	2019216002	李雷	男	2000-08-09	计算机系	3	2	3	学生	3	30



使用JOIN实现内连接

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）。

T-SQL代码：

```
select Reader.RID,Rname,Bname,LendDate,SReturnDate  
from Reader join Borrow on Reader.RID=Borrow.RID  
join book on book.BID=Borrow.BID  
where Rname='韩梅梅' and ReturnDate is null
```

结果		消息			
	RID	Rname	Bname	LendDate	SReturnDate
1	2019216001	韩梅梅	Python核心编程	2020-05-03	2020-06-02



使用JOIN实现内连接

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）”。

T-SQL代码：

```
select Reader.RID ,Rname,Typename,Rdept,ReturnDate,Bname
from book join Borrow on book.BID=Borrow.BID
      join Reader on Reader.RID=Borrow.RID
      join ReaderType on ReaderType.TypeID=Reader.TypeID
where Bname='Python编程快速上手-让繁琐工作自动化'
```

结果		消息				
	RID	Rname	Typename	Rdept	ReturnDate	Bname
1	2019216001	韩梅梅	学生	计算机	2020-05-04	Python编程快速上手-让繁琐工作自动化
2	2019216002	李雷	学生	计算机	NULL	Python编程快速上手-让繁琐工作自动化



使用JOIN实现内连接

案例2：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。

T-SQL代码：

```
Select Reader.RID,Rname,Reader.Rdept,Reader.Lendnum,ReaderType.TypeName,  
ReaderType.LimitNum,ReaderType.LimitDays,book.Bname,Borrow.LendDate,Borrow.  
ReturnDate,Borrow.SReturnDate  
from Reader join ReaderType on Reader.TypeID=ReaderType.TypeID  
join Borrow on Reader.RID=Borrow.RID  
join book on book.BID=Borrow.BID
```



使用JOIN实现内连接

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。

查询结果：

	RID	Rname	Rdept	Lendnum	Typename	LimitNum	LimitDays	Bname	LendDate	ReturnDate	SReturnDate
1	2000186010	张子建	计算机	0	教师	6	90	SAP基础教程	2014-05-30	2014-07-14	2014-08-28
2	2000186011	赵良宇	机电	1	教师	6	90	数据库应用技术	2014-11-26	NULL	2015-02-24
3	2004216010	任灿灿	人文社科	1	教师	6	90	SQL Server 2008从入门到精通	2014-11-15	NULL	2015-02-13
4	2011216117	孟霞	人文社科	0	学生	3	30	C#入门经典	2014-04-09	2014-05-02	2014-05-09
5	2012216008	杨淑华	计算机	1	学生	3	30	SAP财务管理大全	2014-10-28	NULL	2014-11-27
6	2012216009	程鹏	计算机	2	学生	3	30	SAP基础教程	2014-11-17	NULL	2014-12-17
7	2012216009	程鹏	计算机	2	学生	3	30	SQL Server 2008基础教程	2014-10-30	NULL	2014-11-29
8	2019216001	韩梅梅	计算机	1	学生	3	30	Python核心编程	2020-05-03	NULL	2020-06-02
9	2019216001	韩梅梅	计算机	1	学生	3	30	Python编程快速上手-让繁琐工作自动化	2020-05-03	2020-05-04	2020-06-02
10	2019216002	李雷	计算机系	1	学生	3	30	Python编程快速上手-让繁琐工作自动化	2020-05-04	NULL	2020-06-03



不使用JOIN实现内连接

- ☑ 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>



不使用JOIN实现内连接

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）。

T-SQL代码：

```
select Reader.RID,Rname,Bname,LendDate,SReturnDate  
from Reader,Borrow,book  
where Rname='韩梅梅' and ReturnDate is null  
and Reader.RID=Borrow.RID and book.BID=Borrow.BID
```

结果		消息			
	RID	Rname	Bname	LendDate	SReturnDate
1	2019216001	韩梅梅	Python核心编程	2020-05-03	2020-06-02



两种连接方式的性能比较

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。

T-SQL代码:

```
set showplan_all on
```

```
GO
```

```
select Reader.RID,Rname,Bname,LendDate,SReturnDate
```

```
from Reader join Borrow on Reader.RID=Borrow.RID
```

```
join book on book.BID=Borrow.BID
```

```
where Rname='韩梅梅' and ReturnDate is null
```

```
GO
```

```
select Reader.RID,Rname,Bname,LendDate,SReturnDate
```

```
from Reader,Borrow,book
```

```
where Rname='韩梅梅' and ReturnDate is null
```

```
and Reader.RID=Borrow.RID and book.BID=Borrow.BID
```

```
GO
```

```
set showplan_all off
```



两种连接方式的性能比较

结果 消息																
Argument		DefinedValues		EstimateRows	EstimateIO	EstimateCPU	AvgRowSize	TotalSubtreeCost	OutputList							
1		NULL		1.166667	NULL	NULL	NULL	0.009900847	NULL							
2	JTER REFERENCES:([Library].[dbo].[Borrow].[BID])	NULL		1.166667	0	4.876667E-06	77	0.009900847	[Library].[							
3	JTER REFERENCES:([Library].[dbo].[Reader].[RID])	NULL		1.166667	0	4.876667E-06	44	0.00658652	[Library].[							
4	3JECT:([Library].[dbo].[Reader].[PK__Reader__014...	[Library].[dbo].[Reader].[RID], [Library].[dbo].[...		1	0.003125	0.000168	25	0.003293	[Library].[							
5	3JECT:([Library].[dbo].[Borrow].[PK__Borrow__1DE...	[Library].[dbo].[Borrow].[BID], [Library].[dbo].[...		1.166667	0.003125	0.0001582833	29	0.003283283	[Library].[							
6	3JECT:([Library].[dbo].[book].[PK__book]), SEEK:([Li...	[Library].[dbo].[book].[Bname]		1	0.003125	0.0001581	53	0.00330945	[Library].[							

	StmtId	NodeId	Parent	PhysicalOp	LogicalOp	Argument	DefinedValues	EstimateRows	EstimateIO	EstimateCPU	AvgRowSize	TotalSubtreeCost	OutputList			
1	ate,SReturnDa...	1	1	0	NULL	NULL	1	1.166667	NULL	NULL	NULL	0.009900847	NULL			
2	ERENCES:([Li...	1	2	1	Nested L...	Inner Join	OUTER...	NULL	1.166667	0	4.876667E-...	77	0.009900847	[Library]...		
3	EFERENCES:(...	1	3	2	Nested L...	Inner Join	OUTER...	NULL	1.166667	0	4.876667E-...	44	0.00658652	[Library]...		
4	Library].[dbo].[R...	1	4	3	Clustered...	Clustere...	OBJEC...	[Library].[dbo...	1	0.003125	0.000168	25	0.003293	[Library]...		
5	Library].[dbo].[B...	1	5	3	Clustered...	Clustere...	OBJEC...	[Library].[dbo...	1.166667	0.003125	0.00015828...	29	0.003283283	[Library]...		
6	rary].[dbo].[boo...	1	6	2	Clustered...	Clustere...	OBJEC...	[Library].[dbo...	1	0.003125	0.0001581	53	0.00330945	[Library]...		



交叉连接的概念

- ☑ 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。



交叉连接的实现方法

- ☑ 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：
 - ① 使用JOIN关键字实现连接查询；
 - ② 不使用JOIN关键字实现连接查询。



使用JOIN的连接查询

- ☑ 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的连接查询

- ☑ 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>`



交叉连接案例

案例7：将读者表和读者类型表交叉连接。

T-SQL代码：

```
select *
```

```
from Reader cross join ReaderType
```

或

```
select *
```

```
from Reader,ReaderType
```



自连接的概念

- ☑ 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。



自连接案例

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。

T-SQL代码：

```
select x.RID,x.BID,x.ReturnDate  
from Borrow x join Borrow y on x.RID=y.RID  
where x.ReturnDate is null and y.ReturnDate is null  
and x.BID != y.BID
```



自连接案例

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。

查询结果：

	RID	BID	ReturnDate
1	2012216009	TP311.138/136	NULL
2	2012216009	F270.7/455	NULL
3	2019216002	TP311.56/23	NULL
4	2019216002	TP311.138/78	NULL



外连接的引入

- ☑ 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。



外连接的执行过程

- ☑ 外连接的执行过程是：以主表的每行记录去匹配从表的记录行，符合查询条件与连接条件的数据将被添加到查询结果集中；若主表的记录行在从表中没有与之相匹配的记录行，则保留主表的记录行，将从表中相应记录行对应的列值填充NULL 值，并将主从表的行数据连接起来，追加到查询结果集中。



外连接的实现方法

- ☑ 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]]



外连接的类型

- ☑ 根据<外连接类型>值的不同，外连接可分为以下三种：
- ① 左外连接；
 - ② 右外连接；
 - ③ 完全外连接。



左外连接的概念

- ☑ 左外连接是查询结果集中除包含满足条件的记录行外，还包含左表中不满足条件的记录行。
- ☑ 左外连接中以左表作为主表，当左表中不满足条件的记录行与右边从表记录行进行拼接组合时，右表相应列值被填充为NULL值。



左外连接的实现语法

- ☑ 实现左外连接的主要语法格式如下：
- ☑ `SELECT <目标项列表> FROM <表名1> LEFT [OUTER] JOIN <表名2> ON <连接条件>`



左连接案例

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。

T-SQL代码：

```
select Reader.*,Borrow.RID,Borrow.BID  
from Reader left join Borrow on Reader.RID=Borrow.RID
```



左连接案例

结果

消息

	RID	Rname	Rsex	Rbirthday	Rdept	TypeID	Lendnum	RID	BID
1	2000186010	张子建	男	2000-09-03	计算机	1	0	2000186010	F270.7/455
2	2000186011	赵良宇	男	2001-03-04	机电	1	1	2000186011	TP311.138/125
3	2003216008	张英	女	2000-05-06	经管	2	0	NULL	NULL
4	2004060003	李亚茜	女	2002-09-02	计算机	1	0	NULL	NULL
5	2004216010	仕灿灿	女	1999-01-02	人文社科	1	1	2004216010	TP311.138/235
6	2011216117	孟霞	女	1998-02-03	人文社科	3	0	2011216117	TP312/429
7	2012216008	杨淑华	女	2002-01-05	计算机	3	1	2012216008	F275.3/65
8	2012216009	程鹏	男	2001-03-09	计算机	3	2	2012216009	F270.7/455
9	2012216009	程鹏	男	2001-03-09	计算机	3	2	2012216009	TP311.138/136
10	2019216001	韩梅梅	女	2001-07-09	计算机	3	1	2019216001	TP311.56/22
11	2019216001	韩梅梅	女	2001-07-09	计算机	3	1	2019216001	TP311.56/23
12	2019216002	李雷	男	2000-08-09	计算机	3	1	2019216002	TP311.138/78
13	2019216002	李雷	男	2000-08-09	计算机	3	1	2019216002	TP311.56/23

查询已成功执行。

张英和李亚茜两位读者没有借阅过图书。



右外连接的概念

- ☑ 右外连接与左外连接正好相反，是查询结果集中除包含满足条件的记录行外，还包含右表中不满足条件的记录行。
- ☑ 右外连接中以右表作为主表，当右表中不满足条件的记录行与左边从表记录行进行拼接组合时，左表相应列值被填充为NULL值。



右外连接的实现语法

- ☑ 实现右外连接的主要语法格式如下：
- ☑ `SELECT <目标项列表> FROM <表名1> RIGHT [OUTER] JOIN <表名2> ON <连接条件>`



右连接案例

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。

T-SQL代码：

```
select  
Borrow.RID,Borrow.BID,book.BID,book.Bname,book.Author  
from Borrow right join book on Borrow.BID=book.BID
```



	RID	BID	BID	Bname	Author
1	NULL	NULL	F270.7/34	ERP从内部集成开始	陈启申
2	2000186010	F270.7/455	F270.7/455	SAP基础教程	黄佳
3	2012216009	F270.7/455	F270.7/455	SAP基础教程	黄佳
4	NULL	NULL	F270.7/56	ERP系统的集成应用	金蝶软件
5	2012216008	F275.3/65	F275.3/65	SAP财务管理大全	王纹
6	NULL	NULL	TP183/4	MATLAB神经网络43个案例分析	王小川
7	2000186011	TP311.138/125	TP311.138/125	数据库应用技术	周慧
8	2012216009	TP311.138/136	TP311.138/136	SQL Server 2008基础教程	Robin Dewson
9	NULL	NULL	TP311.138/230	SQL Server 2008数据库设计与实现	Louis Davidson
10	2004216010	TP311.138/235	TP311.138/235	SQL Server 2008从入门到精通	Mike Hotek
11	2019216002	TP311.138/78	TP311.138/78	数据库系统概论	萨师煊
12	2019216001	TP311.56/22	TP311.56/22	Python核心编程	Wesley Chun
13	2019216001	TP311.56/23	TP311.56/23	Python编程快速上手-让繁琐工作自动化	Al Sweigart
14	2019216002	TP311.56/23	TP311.56/23	Python编程快速上手-让繁琐工作自动化	Al Sweigart
15	2011216117	TP312/429	TP312/429	C#入门经典	Karli Watson

连接结果加入了表“**Book**”中不满足等值条件的第**1、4、6、9**行，表“**Borrow**”中相应的列为**NULL**，说明图书《**ERP从内部集成开始**》、《**ERP系统的集成应用**》、和《**数据库系统概论**》、

《**MATLAB神经网络43个案例分析**》四本书没有被借出过，《**SAP基础教程**》、《**Python编程快速上手**》被借出过两次。



完全外连接的概念

- ☑ 完全外连接是查询结果集中除包含满足条件的记录行外，还包含左表与右表中不满足条件的记录行。
- ☑ 当左(右)表中不满足条件的记录行与右(左)表中记录行进行拼接时，右(左)表相应列值被填充为NULL值。
- ☑ 完全外连接能够获取左、右表中都不满足条件的记录行，并对这些不满足条件的列值填充空值。完全外连接的功能可由左外连接与右外连接配合起来共同完成。



完全外连接的实现语法

- ☑ 实现完全外连接的主要语法格式如下：
- ☑ `SELECT <目标项列表> FROM <表名1> FULL [OUTER] JOIN <表名2> ON <连接条件>`



完全连接案例

案例11：借阅和读者全外连接。

T-SQL代码：

```
SELECT Reader.*, Borrow.RID, Borrow.BID  
FROM Borrow FULL OUTER JOIN Reader --全外连接  
ON Borrow.RID=Reader.RID
```

查询结果：

查询结果与**左**外连接相同，因为借阅表中读者的编号均在读者表中存在，都能形成连接。



复合条件连接的概念

- ☑ 在多表连接查询中，如果在WHERE子句中，或在ON关键字所携带的连接条件中，包含由两个或两个以上的条件表达式组成的复合条件，则称这类连接为复合条件连接。



多表复杂查询

1

多表连接查询概述

2

多表连接查询类型

3

嵌套查询

4

联合查询





嵌套查询的概念

- ☑ 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。



父查询与子查询

- ☑ 在嵌套查询结构中，外部的SELECT语句称为父查询或外层查询，内部的SELECT语句称为子查询或内层查询。
- ☑ 子查询中还可以嵌套更深一级的子查询，因此嵌套查询的嵌套级别可以达到多级。
- ☑ 采用子查询有时会提高算法的时间和空间效率，但算法不易读懂，读者应权衡利弊进行选择。



[NOT] IN谓词的用法

- ☑ 在构造的嵌套查询中，当子查询的返回结果为集合类型时，常用IN或NOT IN关键字作为关联内、外层查询块的谓词，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) [;]
- ☑ 当没有用EXISTS 引入子查询时，在子查询的SELECT投影列表中只能指定一个表达式。



[NOT] IN谓词查询的逻辑

- ☑ [NOT] IN谓词构成的嵌套查询的处理逻辑为：
- ☑ 内层查询先求解，以此解作为构建外层查询查询条件基础；外层查询通过集合谓词将一个列名或表达式与内层查询返回的结果集进行比较，根据列值或表达式的值是否等于内层查询的结果集中的某个值来决定外层查询的条件返回值是TRUE还是FALSE，从而进一步决定哪些记录能够被放入最终的查询结果集。



[NOT] IN谓词的用法

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）。

T-SQL代码：

```
select distinct RID --父查询  
from Borrow
```

```
where BID in ( select BID from book --子查询  
where Publisher='人民邮电出版社' )
```

结果		消息
		RID
1	2000186010	
2	2000186011	
3	2012216009	
4	2019216001	
5	2019216002	



[NOT] IN谓词的用法

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。

T-SQL代码：

```
select * from Reader  
where RID not in ( select RID from Borrow )
```

结果		消息			
	RID	Rname	Rsex	Rbirthday	Rdept
1	2003216008	张英	女	2000-05-06	经管
2	2004060003	李亚茜	女	2002-09-02	计算机



ANY/ALL谓词的用法

- ☑ ANY或ALL关键字必须与比较运算符联合使用，用来对比较运算符的运算范围进行特定的语义修饰。
- ☑ 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) [;]
- ☑ 与[NOT] IN子查询类似，在子查询的SELECT投影列表中只能指定一个表达式。



ANY/ALL与比较运算符组合

运算符 +ALL	语义	运算符 +ANY	语义
= ALL	等于子查询结果集中的所有值	= ANY	等于子查询结果集中的某个值
!=ALL 或<> ALL	不等于子查询结果集中的所有值	!=ANY 或<> ANY	不等于子查询结果集中的某个值
> ALL	大于子查询结果集中的所有值	> ANY	大于子查询结果集中的某个值
>= ALL	大于等于子查询结果集中的所有值	>= ANY	大于等于子查询结果集中的某个值
< ALL	小于子查询结果集中的所有值	< ANY	小于子查询结果集中的某个值
<= ALL	小于等于子查询结果集中的所有值	<= ANY	小于等于子查询结果集中的某个值
!> ALL	不大于子查询结果集中的所有值	!> ANY	不大于子查询结果集中的某个值
!< ALL	不小于子查询结果集中的所有值	!< ANY	不小于子查询结果集中的某个值



ANY/ALL谓词查询的逻辑

- ☑ ANY/ALL谓词构成的嵌套查询的处理逻辑为：
- ☑ 子查询先求解，得到单值数据集；父查询通过比较运算符与ANY或ALL谓词，将列名或表达式与子查询返回的数据集中的某个或所有元素进行比较；根据比较的结果，返回TRUE或FALSE，从而进一步决定哪些记录能够被放入最终的查询结果集。



ANY/ALL谓词的用法

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。

T-SQL代码：

```
SELECT * FROM Reader  
WHERE Rtime<(select min(Rtime) from Reader where  
Rdept='计算机' )
```

结果		消息						
	RID	Rname	Rsex	Rbirthday	Rdept	TypeID	Lendnum	Rtime
1	2000186011	赵良宇	男	2001-03-04	机电	1	1	2011-09-15



ANY/ALL谓词的用法

PS: 使用ANY或ALL谓词的查询, 可以用包含集合运算符或聚集函数的查询结构来替换, 两种方法在功能上完全等价, 但后者在查询效率上更高或语法结构上更简洁。

T-SQL代码:

```
SELECT * FROM Reader
```

```
WHERE Rtime < ALL (select Rtime from Reader where  
Rdept=
```

结果 消息

	RID	Rname	Rsex	Rbirthday	Rdept	TypeID	Lendnum	Rtime
1	2000186011	赵良宇	男	2001-03-04	机电	1	1	2011-09-15



ANY/ALL与等价的聚集函数对照

运算符 +ALL	等价的聚集函数	运算符 +ANY	等价的聚集函数
> ALL	>MAX()	> ANY	>MIN()
>= ALL	>=MAX()	>= ANY	>=MIN()
< ALL	<MIN()	< ANY	<MAX()
<= ALL	<=MIN()	<= ANY	<=MAX()
!> ALL	<=MIN()	!> ANY	<=MAX()
!< ALL	>=MAX()	!< ANY	>=MIN()



ANY/ALL谓词的用法

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。

T-SQL代码：

```
select * from book  
where Publisher!='人民邮电出版社' and Price >ANY (  
select Price from book where Publisher='人民邮电出版社')
```



ANY/ALL谓词的用法

T-SQL代码：聚集函数替换代码

```
select * from book
```

```
where Publisher!='人民邮电出版社' and Price > (
```

```
select MIN(Price) from book where Publisher='人民邮电出版社')
```

结果		消息						
	BID	ISBN	Bname	Author	Publisher	PublishYear	Price	LentOut
1	F270.7/34	9787121010613	ERP从内部集成开始	陈启申	电子工业出版社	2005	45.00	0
2	F275.3/65	9787302104575	SAP财务管理大全	王纹	清华大学出版社	2005	41.40	1
3	TP183/4	9787512412026	MATLAB神经网络43个案例分析	王小川	北京航空航天大学出版社	2013	48.00	1
4	TP311.138/235	9787302247784	SQL Server 2008从入门到精通	Mike Hotek	清华大学出版社	2011	53.10	1
5	TP311.138/78	9787040406641	数据库系统概论	萨师煊	高等教育出版社	2014	39.60	1
6	TP312/429	9787302444060	C#入门经典	Karli Watson	清华大学出版社	2016	88.20	0



[NOT] EXISTS谓词的用法

- ☑ [NOT] EXISTS关键字通常用在子查询块的前面，用来构造一种条件判断。这种条件判断的实质是进行一次存在测试，外层查询的WHERE子句测试内层查询返回的数据集是否存在着行记录。
- ☑ 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) [;]



[NOT] EXISTS谓词的用法

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。

T-SQL代码：

```
select * from book where not exists  
( select * from Borrow where book.BID=Borrow.BID )
```

	BID	ISBN	Bname	Author	Publisher	PublishYear	Price	LentOut
1	F270.7/34	9787121010613	ERP从内部集成开始	陈启申	电子工业出版社	2005	45.00	0
2	F270.7/56	9787302109921	ERP系统的集成应用	金蝶软件	清华大学出版社	2005	31.50	0
3	TP183/4	9787512412026	MATLAB神经网络43个案例分析	王小川	北京航空航天大学出版社	2013	48.00	0
4	TP311.138/230	9787115215543	SQL Server 2008数据库设计与实现	Louis Davidson	人民邮电出版社	2009	89.00	0



[NOT] EXISTS谓词的用法

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。

T-SQL代码：

```
select * from book where not exists  
( select * from Borrow where book.BID=Borrow.BID )
```

等价于：

```
select distinct RID  
from Borrow join book on Borrow.BID=book.BID  
where Publisher='人民邮电出版社'
```



[NOT] EXISTS谓词的用法

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。

查询结果：

结果		消息
RID		
1	2000186010	
2	2000186011	
3	2012216009	
4	2019216001	
5	2019216002	



用比较运算符的嵌套查询

- ☑ 当子查询的返回结果为某一单值类型时，常用比较运算符来关联内、外层查询块。
- ☑ 通过比较运算符关联内、外层查询块的嵌套查询用法如下：
- ☑ `SELECT-FROM WHERE <列名或表达式> OP (子查询) [;]`
- ☑ 其中OP为=、!=、<>、>、>=、<、<=、!>、!<等比较运算符。



子查询在其他语句中的应用

- ☑ 子查询除用于SELECT语句外，还可以出现在INSERT、UPDATE、DELETE等T-SQL语句中。
- ☑ 因此，任何允许使用表达式的地方都可以使用子查询。



子查询在UPDATE语句

案例18：计算读者表“Reader”中的已借书数量列

T-SQL代码：

```
select update Reader
```

```
set Lendnum=
```

```
(select COUNT(*) from Borrow --从借阅表中统计每个读者的  
借书册数
```

```
where Reader.RID=Borrow.RID and ReturnDate is null)
```




子查询在UPDATE语句

案例19：计算借阅表“Borrow”中的应还日期列

T-SQL代码：

```
select update Reader
```

```
set Lendnum=
```

```
(select COUNT(*) from Borrow --从借阅表中统计每个读者的  
借书册数
```

```
where Reader.RID=Borrow.RID and ReturnDate is null)
```



多表复杂查询

1

多表连接查询概述

2

多表连接查询类型

3

嵌套查询

4

联合查询





联合查询的概念

- ☑ 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。



联合查询的运算符

- ☑ 联合查询的运算符主要包括以下三种：
- ① 求并 (UNION) ；
 - ② 求交 (INTERSECT) ；
 - ③ 求差 (EXCEPT) 。



实现联合查询的语法

- ☑ 实现联合查询的语法格式如下：
- ☑ $\langle \text{SELECT 语句1} \rangle \{ \{ \text{UNION} \mid \text{INTERSECT} \mid \text{EXCEPT} \} \langle \text{SELECT 语句2} \rangle \} [\dots n]$



实现求并联合查询的语法

- ☑ UNION (并) 运算符能够将两个或两个以上的查询数据集顺序连接, 合并为一个结果数据集并显示输出。
- ☑ 实现求并联合查询的语法格式如下:
- ☑ `<SELECT语句1> { UNION [ALL]
<SELECT语句2> } [, ... n]`



实现求交联合查询的语法

- ☑ INTERSECT (交) 操作将两个或多个查询结果数据集的交集作为联合查询的结果数据集返回并输出。
- ☑ 实现求交联合查询的语法格式如下:
- ☑ $\langle \text{SELECT 语句1} \rangle \{ \text{INTERSECT} \langle \text{SELECT 语句2} \rangle [, \dots n]$



实现求差联合查询的语法

- ☑ EXCEPT (差) 操作将对左右两个查询语句的结果数据集求差集，并将求取的结果数据集作为联合查询的结果数据集返回并输出。
- ☑ 实现求差联合查询的语法格式如下：
- ☑ <SELECT语句1> EXCEPT <SELECT语句2>



下一章再见!