

BBS 数据的高级检索

由于软件业务复杂度的提升,对软件开发人员和数据库管理员来说,高级检索是需要进一步完善和提高的技能。高级检索主要包括汇总数据、分组查询、连接查询、组合查询及子查询等。

【项目描述】

本项目根据网站论坛业务需求,利用聚合函数对数据进行汇总;按照分组条件对数据进行分组检索;实现多个表或视图之间满足一定条件的连接查询或子查询。

【项目目标】

- ④ 会用聚合函数实现数据汇总。
- ④ 正确使用分组查询。
- ④ 熟练使用连接查询。
- ④ 正确使用组合查询。
- ④ 熟练使用子查询。

5.1 背景知识

5.1.1 汇总查询

BBS 数据库中存储了发帖信息,为了知道帖子的总数,帖子点击数的最高值、最低值、平均值的情况,T-SQL 提供了聚合函数,方便数据分析和报表的生成。

聚合函数对数据表中的某一列值或一组值执行计算,并返回单个值。聚合函数经常与 SELECT 语句的 GROUP BY 子句一起使用,用于检索数据,进行数据分析和生成报表。除了 COUNT() 函数以外,聚合函数一般都会忽略空值,如表 5-1 所示。

表 5-1 T-SQL 常用的聚合函数

| 函 数 | 说 明 |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| AVG([ALL DISTINCT]表达式) | 返回组中各值的平均值,ALL 是默认值,对所有值求平均值;DISTINCT 指定 AVG 返回排除了重复值的平均值 |
| SUM([ALL DISTINCT]表达式) | 返回表达式中所有值的和,ALL 是默认值,对所有的值求和;DISTINCT 指定 SUM 返回排除了重复值的和 |
| MAX([ALL DISTINCT]表达式) | 返回表达式的最大值 |
| MIN([ALL DISTINCT]表达式) | 返回表达式的最小值 |
| COUNT (([ALL DISTINCT] 表达式 *)) | 返回组中的项数。COUNT (*) 返回组中的项数,包括 NULL 值和重复项。COUNT(ALL 表达式) 对组中的每一行都计算 expression 并返回非空值的数量。COUNT(DISTINCT 表达式) 对组中的每一行都计算 expression 并返回唯一非空值的数量 |

例如,要查询发帖表 Topic 帖子的数量,就可以利用 COUNT() 函数查询出帖子总数,如图 5-1 所示。

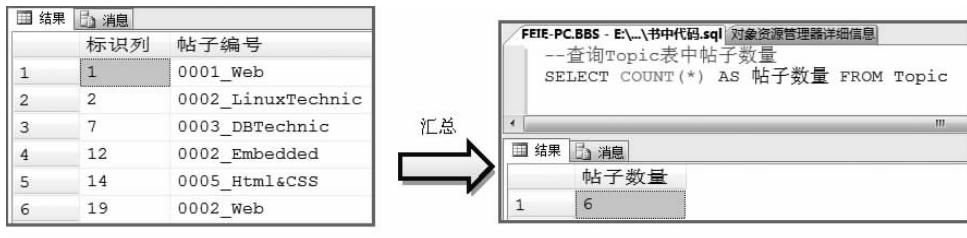


图 5-1 汇总查询

5.1.2 数据分组

利用 T-SQL 聚合函数可以汇总数据,统计发帖表 Topic 帖子的数量。但如果要统计每个板块的帖子数量情况,或每个用户的发帖情况,就需要用到数据分组查询。

分组就是按照某一列的值或者某几列的组合值,将结果集分成若干组进行统计,每组在指定列上或列组合上都有相同的值。

例如,在发帖表 Topic 中统计每个板块的帖子的数量情况,如图 5-2 所示。

The diagram illustrates the process of data grouping. On the left, a table with columns '帖子编号' (Post ID) and '版块编号' (Forum Section ID) contains 8 rows of data. An arrow labeled '分组' (Grouping) points to the right, where a second table shows the result of a grouped query. This second table has columns '版块编号' (Forum Section ID) and '帖子数量' (Post Count), with 5 rows representing the aggregated counts for each unique forum section.

| 帖子编号 | 版块编号 |
|-------------------|------|
| 0001_Web | 1 |
| 0002_Web | 1 |
| 0002_LinuxTechnic | 2 |
| 0002_LinuxTechnic | 2 |
| 0003_DBTechnic | 7 |
| 0002_Embedded | 10 |
| 0001_Embedded | 10 |
| 0005_Html&CSS | 11 |

| 版块编号 | 帖子数量 |
|------|------|
| 1 | 2 |
| 2 | 2 |
| 7 | 1 |
| 10 | 2 |
| 11 | 1 |

图 5-2 数据分组查询

5.1.3 连接查询

关系数据库将数据分解为多个表,从而更方便数据的存储和处理。如果数据存储在多个表中,那么要用一条 SELECT 语句检索出数据,就需要用到连接查询。

连接查询的功能非常强大,在关系数据库中利用连接查询,可以从多个数据表中查询满足一定条件的记录。

在 SQL Server 中,连接查询分为内部连接、外部连接和交叉连接。

1. 内部连接

内部连接(inner join)也称等值连接,它是基于两个表之间的某一列的值相等而进行的连接。通常两个表之间要建立主外键关系,根据某一列的数值进行比较,检索满足条件的数据。这种连接方式是最常用的。

例如,查询帖子主题及所属板块的名称,如图 5-3 所示。这部分信息分别存储在发帖表 Topic 和板块信息表 Section 中,要通过所属板块编号 TSectionID 等外键值来连接查询。

2. 外部连接

在内部连接两个表时,结果集中体现的是两个表中都存在的数据行,但有时会出现在一个表中没有关联的数据行的情况,为了在结果集中能够将其检索出来,需要使用外部连接查询。

外部连接(outer join)可分为左外连接(left outer join)、右外连接(right outer join)和全外连接(full outer join)。左外连接可以检索出左表中所有数据行,右外连接可以检索出右表中所有数据行,全外连接可以检索出左、右两表中所有的数据行。

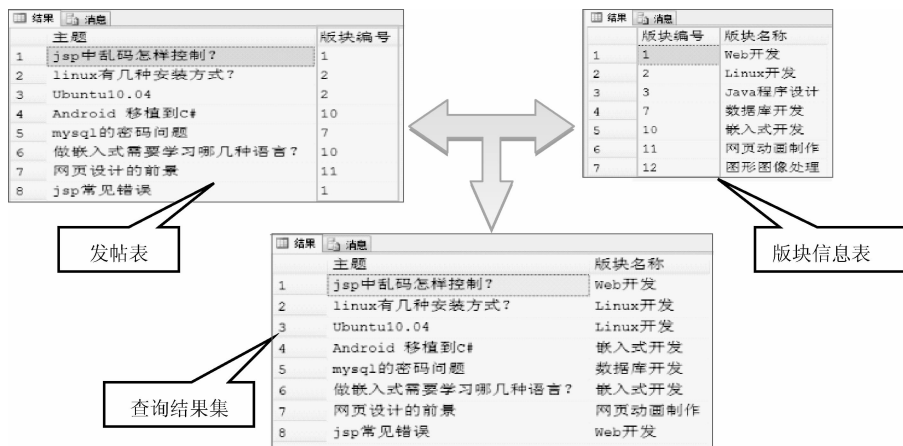


图 5-3 内部连接查询

例如,查询所有注册用户的发帖情况,如图 5-4 所示。用户信息表 Users 中的注册用户未必全部发帖,要查询用户信息表 Users 中所有用户在发帖表 Topic 中的发帖情况,如果将用户信息表 Users 作为左表,可以采用左外连接查询。



图 5-4 左外连接查询

3. 交叉连接

交叉连接(cross join)在两个表没有连接条件限制的情况下将其组合起来,从而产生连接所涉及表的笛卡尔积。笛卡尔积就是将第一个表的行数乘以第二个表的行数从而得到其结果集行数的乘积。

例如,将发帖表 Topic 中的帖子主题列与版块信息表 Section 中的版块名称列交叉连接,如图 5-5 所示。交叉连接后返回的结果集中主题列和版块名称列中有的信息不符合实际情况。

| 主题 | 版块名称 |
|----------------|----------|
| jsp中乱码怎样控制? | Web开发 |
| linux有几种安装方式? | Web开发 |
| Ubuntu10.04 | Web开发 |
| Android 移植到C# | Web开发 |
| mysql的密码问题 | Web开发 |
| 做嵌入式需要学习哪几种语言? | Web开发 |
| 网页设计的前景 | Web开发 |
| jsp常见错误 | Web开发 |
| jsp中乱码怎样控制? | Linux开发 |
| linux有几种安装方式? | Linux开发 |
| Ubuntu10.04 | Linux开发 |
| Android 移植到C# | Linux开发 |
| mysql的密码问题 | Linux开发 |
| 做嵌入式需要学习哪几种语言? | Linux开发 |
| 网页设计的前景 | Linux开发 |
| jsp常见错误 | Linux开发 |
| jsp中乱码怎样控制? | Java程序设计 |
| linux有几种安装方式? | Java程序设计 |
| Ubuntu10.04 | Java程序设计 |
| Android 移植到C# | Java程序设计 |

不正确匹配

图 5-5 交叉连接查询部分数据

5.1.4 组合查询

多数 SQL 语句只能从一个表或多个表中利用一条 SELECT 语句返回一个结果集,但多条 SELECT 语句执行,只能返回多个单一结果集。使用 UNION 语句可以把两个或两个以上的查询结果集合并为一个结果集。

例如,查询所有发帖人和回帖人的编号和帖子主题,如图 5-6 所示。查询发帖表 Topic 中的发帖人编号 TUserID 和主题 TTopic 信息,与回帖表 Reply 中的回帖人编号 RUserID 和主题 RTopic 信息进行合并,返回一个结果集。



图 5-6 组合查询

5.1.5 子查询

子查询是一个嵌套在 SELECT、INSERT、UPDATE 或 DELETE 语句或其他子查询中的查询。子查询还可以嵌套在 WHERE 子句或 HAVING 子句内。子查询也称为内部查询,而包含子查询的语句称为外部查询。子查询的 SELECT 查询总是使用圆括号括起来。子查询一般分为两种,即返回单值的查询和返回值列表的查询。

例如,检索所属版块为“Web 开发”的所有主题和所在版块编号,如图 5-7 所示。利用子查询在版块信息表中根据版块名称“Web 开发”查询其版块编号为 1,将子查询结果用于检索发帖信息表中所在版块编号为 1 的所有帖子的用户编号和主题。

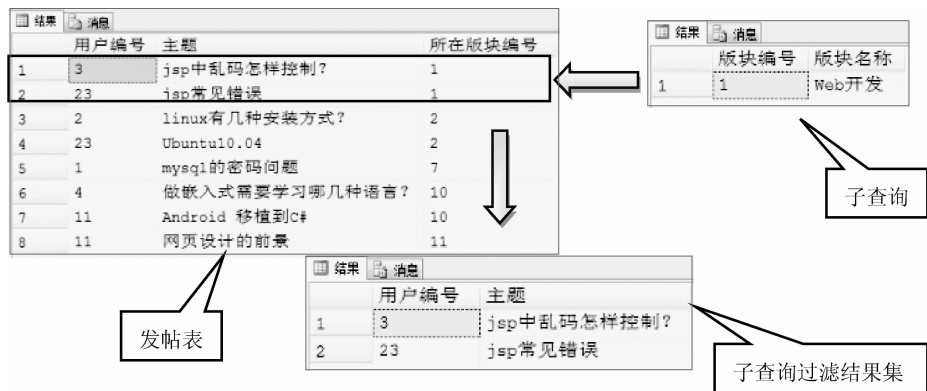


图 5-7 子查询

5.2 典型工作任务

5.2.1 使用聚合函数汇总数据

1. 任务描述

①统计发帖表 Topic 中所有帖子点击数 TClickCount 的最大值、最小值、总和和平均值。

②查询发帖表 Topic 帖子的数量以及发帖人的数量。

2. 技能要求

①熟练使用聚合函数对数据进行汇总。

②正确使用 DISTINCT 关键字实现不重复统计。

3. 任务实现

1)使用聚合函数进行数据汇总

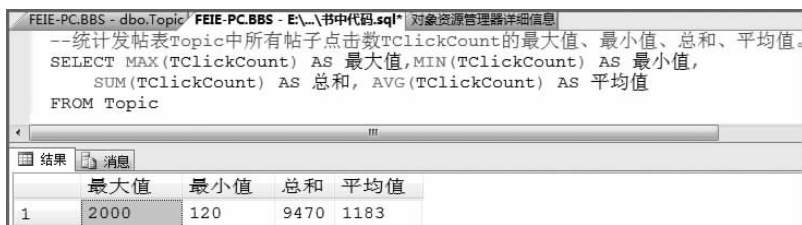
①在 SQL Server Management Studio 工具栏中单击“新建查询”按钮,选择当前可用数

数据库为 BBS。也可以右击 BBS 数据库结点,在弹出的快捷菜单中选择“新建查询”选项。

②统计发帖表 Topic 中所有帖子点击数 TClickCount 的最大值、最小值、总和和平均值。在打开的查询输入窗格中输入以下语句。

```
SELECT MAX(TClickCount) AS 最大值,MIN(TClickCount) AS 最小值,SUM(TClickCount)
    AS 总和,AVG(TClickCount) AS 平均值
FROM Topic
```

③单击“执行”按钮,结果集如图 5-8 所示。



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

```
--统计发帖表Topic中所有帖子点击数TClickCount的最大值、最小值、总和、平均值。
SELECT MAX(TClickCount) AS 最大值,MIN(TClickCount) AS 最小值,
    SUM(TClickCount) AS 总和,AVG(TClickCount) AS 平均值
FROM Topic
```

Below the query window is a results grid with the following data:

| | 最大值 | 最小值 | 总和 | 平均值 |
|---|------|-----|------|------|
| 1 | 2000 | 120 | 9470 | 1183 |

图 5-8 使用聚合函数进行数据汇总

在使用聚合函数汇总数据时,应注意以下几点。

①AVG()函数只能以特定数值列的列名为参数,返回该列的平均值,多个列必须用多个 AVG()函数。

②SUM()函数用于返回特定数值列的和,也可以用于计算列表表达式的和。

③MAX()函数和 MIN()函数一般分别用来返回最大和最小的数值或日期值。

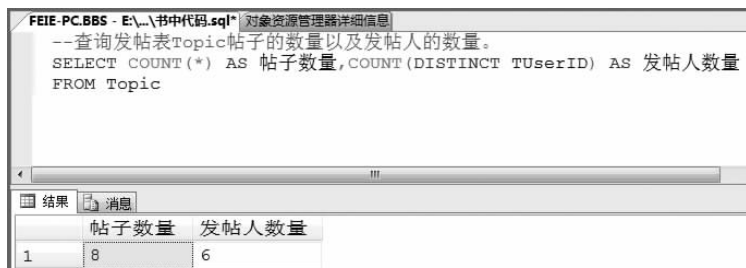
④AVG()、SUM()、MAX()和 MIN()函数都会忽略值为 NULL 的数据行。

2)使用 DISTINCT 关键字

①查询发帖表 Topic 中帖子的数量以及发帖人的数量。在查询输入窗格中输入以下语句。

```
SELECT COUNT(*) AS 帖子数量,COUNT(DISTINCT TUserID) AS 发帖人数量
FROM Topic
```

②单击“执行”按钮,结果集如图 5-9 所示。



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

```
--查询发帖表Topic帖子的数量以及发帖人的数量。
SELECT COUNT(*) AS 帖子数量,COUNT(DISTINCT TUserID) AS 发帖人数量
FROM Topic
```

Below the query window is a results grid with the following data:

| | 帖子数量 | 发帖人数量 |
|---|------|-------|
| 1 | 8 | 6 |

图 5-9 使用 DISTINCT 关键字

使用 DISTINCT 关键字时应注意以下两点。

①DISTINCT 关键字不能单独用于 COUNT(*)语句,需要指定列名。

②如果数据行存在多个 NULL 值,那么在使用 DISTINCT 关键字进行统计时,结果将为一个。

5.2.2 使用 GROUP BY 分组检索

1. 任务描述

- ①统计发帖表 Topic 中每个版块的回帖数量情况。
- ②检索回帖表 Reply 中每个帖子的回帖数量大于等于 2 的帖子信息。
- ③检索回帖表 Reply 中每个版块和每个回复主题的回帖信息。

2. 技能要求

- ①熟练使用 GROUP BY 分组进行检索。
- ②正确使用 HAVING 关键字过滤分组数据。
- ③遵循 SELECT 子句的顺序。

3. 任务实现

1) 简单分组

①统计发帖表 Topic 中每个版块的回帖数量情况。在 SQL Server Management Studio 查询窗口中输入以下 T-SQL 语句。

```
SELECT TSectionID AS 版块编号,COUNT(TopicNumber) AS 帖子数量
FROM Topic
GROUP BY TSectionID
```

②单击“执行”按钮,结果集如图 5-10 所示。

| | 版块编号 | 帖子数量 |
|---|------|------|
| 1 | 1 | 2 |
| 2 | 2 | 2 |
| 3 | 7 | 1 |
| 4 | 10 | 2 |
| 5 | 11 | 1 |

图 5-10 使用 GROUP BY 分组进行数据汇总

分组查询要先确定分组的列,统计函数是对每一行数据进行分组后,再统计数值。在结果集中可以看到根据版块编号 TSectionID 统计的每个版块的发帖情况,因此,在 GROUP BY 子句后面为分组的列名 TSectionID。

2) 过滤分组

①查询回帖表 Reply 中每个帖子的回帖数量大于等于 2 的帖子信息。在查询输入窗格中输入以下语句。

```
SELECT RTopicID AS 帖子编号,COUNT(RClickCount) AS 帖子数量
FROM Reply
GROUP BY RTopicID
```


HAVING COUNT(RClickCount) >= 2

②单击“执行”按钮,结果集如图 5-11 所示。

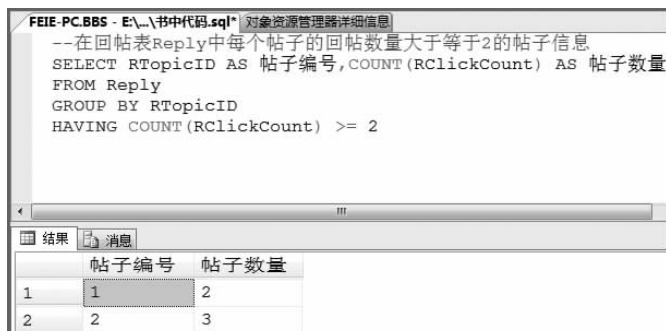


图 5-11 使用 HAVING 子句过滤分组

回帖表是按照回复的主贴编号 RTopicID 进行分组的,过滤每个分组帖子的数量大于等于 2 的帖子信息时并没有使用 WHERE 子句,因为 WHERE 子句用于过滤数据行,而不是分组。SQL 提供了 HAVING 子句用于过滤分组数据。WHERE 子句中所有可用的通配符、运算符和表达式在 HAVING 子句中也都可以使用。

SELECT 语句中子句的顺序如下。

SELECT * |列名 1[, ...,列名 n]|计算表达式

FROM 表名

[WHERE 行过滤条件]

[GROUP BY 列名 1 [, ...,列名 n]

[HAVING 分组过滤条件]

[ORDER BY 排序表达式 [ASC|DESC]

3) 多个分组

①检索回帖表 Reply 中每个版块和每个回复主题的回帖信息。在查询输入窗格中输入以下语句。

SELECT RSectionID AS 所属版块编号,RTopicID AS 回复帖子编号,

COUNT(RClickCount) AS 帖子数量

FROM Reply

GROUP BY RSectionID,RTopicID

②单击“执行”按钮,结果集如图 5-12 所示。

按照多列进行分组时,先根据 GROUP BY 子句中第一列 RSectionID 进行分组,如果所属版块编号 RSectionID 相同,但回复帖子编号 RTopicID 不同,那么再按第二列 RTopicID 进行分组。

GROUP BY 子句的使用规则如下。

①GROUP BY 子句中的每一列都是检索列或计算表达式,不能使用聚合函数。

②GROUP BY 子句中的列不能使用别名。



图 5-12 使用多个分组检索数据

③在 SELECT 子句检索列中,除了聚合计算列,每一列都应在 GROUP BY 子句中
出现。

④如果分组列中有 NULL 值,将其作为一个分组返回。

5.2.3 使用连接查询

1. 任务描述

①在发帖表 Topic 和版块信息表 Section 中查询帖子主题 TTopic 及所属版块名称 SName 等信息。

②在用户信息表 Users 和发帖表 Topic 中查询所有注册用户的发帖情况。

③在回帖表 Reply 和发帖表 Topic 中查询所有帖子的回帖信息。

④查询发帖表 Topic 中的帖子主题与版块信息表 Section 中的版块名称首字符相同的
信息。

2. 技能要求

①正确选择不同的连接类型。

②正确使用两个以上表的连接。

③连接查询中使用表的别名。

3. 任务实现

1) 内部连接

①在发帖表 Topic 和版块信息表 Section 中查询帖子主题 TTopic 及所属版块的名称 SName 等信息。在查询输入窗格中输入以下语句。

```
SELECT TTopic AS 主题, SName AS 版块名称
FROM Section INNER JOIN Topic
ON Section.SectionID=Topic.TSectionID
```

②单击“执行”按钮,结果集见图 5-3。

内部连接等价于下列语句。

```
SELECT TTopic AS 主题,SName AS 版块名称  
FROM Section,Topic  
WHERE Section.SectionID=Topic.TSectionID
```

帖子主题 TTopic 及所属版块名称 SName 等信息分别存储在发帖表 Topic 和版块信息表 Section 中,并且根据版块编号等值测试产生结果集,因此两个表的关系由 FROM 子句中的 INNER JOIN 指定,连接条件由 ON 子句给出。内部连接相当于 WHERE 子句两个表的等值查询。

也可以完成 3 个表的连接查询,在发帖表 Topic、回帖表 Reply 和版块信息表 Section 中查询帖子主题 TTopic、回复主题 RTopic 及所属版块的名称 SName 等信息。T-SQL 语句如下。

```
SELECT TTopic AS 主题,RTopic AS 回复的主题,SName AS 版块名称  
FROM Topic INNER JOIN Reply  
ON Topic.TopicID=Reply.RTopicID  
INNER JOIN Section  
ON Topic.TSectionID=Section.SectionID
```

内部连接的语法格式如下。

```
FROM 表 1 [INNER] JOIN 表 2 ON 连接条件  
[...[INNER] JOIN 表 n] ON 连接条件
```

2) 外部连接

(1) 左外连接。

①在用户信息表 Users 和发帖表 Topic 中查询所有注册用户的发帖情况。在查询输入窗格中输入以下语句。

```
SELECT UserName AS 用户名称,TTopic AS 主题  
FROM Users LEFT OUTER JOIN Topic  
ON Users.UserID=Topic.TUserID
```

②单击“执行”按钮,结果集如图 5-4 所示。

在此任务中,不是所有用户都一定发过帖子,为了使没有发过帖子的用户也能被检索到,将发帖主题信息以 NULL 值匹配。因此,FROM 子句采用左外连接方式,连接条件由 ON 子句指定。其中,Users 表作为左表,Topic 表作为右表,左外连接可以将左表中所有数据行检索出来,右表按着条件匹配数据行,如果没有则以 NULL 匹配。

(2) 右外连接。

①在回帖表 Reply 和发帖表 Topic 中查询所有帖子的回帖信息。在查询输入窗格中输入以下语句。

```
SELECT RTopic AS 回复的主题,TTopic AS 发帖主题  
FROM Reply RIGHT OUTER JOIN Topic  
ON Reply.RTopicID=Topic.TopicID
```

②单击“执行”按钮,结果集如图 5-13 所示。



```

FEIE-PC.BBS - E:\...书\代码\sql 对象资源管理器详细视图
--在回帖表Reply和发帖表Topic中, 查询所有发帖的回帖的信息
SELECT RTopic AS 回复的主题,TTopic AS 发帖主题
FROM Reply RIGHT OUTER JOIN Topic
ON Reply.RTopicID= Topic.TopicID

```

| | 回复的主题 | 发帖主题 |
|---|---------------|---------------|
| 1 | jsp中乱码怎样控制? | jsp中乱码怎样控制? |
| 2 | jsp标签的使用 | jsp中乱码怎样控制? |
| 3 | linux有几种安装方式? | linux有几种安装方式? |
| 4 | linux安装方式之我见 | linux有几种安装方式? |
| 5 | linux安装 | linux有几种安装方式? |
| 6 | Ubuntu的操作手册 | Ubuntu10.04 |
| 7 | NULL | Android 移植到C# |
| 8 | NULL | 网页设计框架 |

图 5-13 使用右外连接检索数据

在此任务中,不是所有的帖子都有回复,为了使没有回复的帖子也能被检索到,将回帖表中的回复主题信息以 NULL 值匹配。因此,FROM 子句采用右外连接方式,连接条件由 ON 子句指定。其中,Reply 表作为左表,Topic 表作为右表,右外连接可以将右表中所有数据行检索出来,左表按着条件匹配数据行,如果没有则以 NULL 匹配。

(3)全外连接。

①查询发帖表 Topic 中帖子主题与版块信息表 Section 中版块名称首字符相同的信息。在查询输入窗格中输入以下语句。

```

SELECT TTopic AS 主题,SName AS 版块名称
FROM Topic FULL OUTER JOIN Section
ON LEFT(Topic.TTopic,1)=LEFT(Section.SName,1)

```

②单击“执行”按钮,结果集如图 5-14 所示。

```

FEIE-PC.BBS - E:\...书\代码\sql 对象资源管理器详细视图
--发帖表Topic中帖子主题与版块信息表Section中版块名称首字符相同的名称
SELECT TTopic AS 主题,SName AS 版块名称
FROM Topic FULL OUTER JOIN Section
ON LEFT(Topic.TTopic,1) = LEFT(Section.SName,1)

```

| | 主题 | 版块名称 |
|----|----------------|----------|
| 1 | jsp中乱码怎样控制? | Java程序设计 |
| 2 | linux有几种安装方式? | Linux开发 |
| 3 | Ubuntu10.04 | NULL |
| 4 | Android 移植到C# | NULL |
| 5 | mysql的密码问题 | NULL |
| 6 | 做嵌入式需要学习哪几种语言? | NULL |
| 7 | 网页设计的前景 | 网页动画制作 |
| 8 | jsp常见错误 | Java程序设计 |
| 9 | 数据库实例的创建 | 数据库开发 |
| 10 | 网页设计框架 | 网页动画制作 |
| 11 | NULL | Web开发 |
| 12 | NULL | 嵌入式开发 |
| 13 | NULL | 图形图像处理 |

图 5-14 使用全外连接检索数据



此任务中的每个表都有不能匹配的数据行,因此,FROM子句采用全外连接方式,连接条件由ON子句指定。其中,Topic表作为左表,Section表作为右表,全外连接可以将左右两表所有数据行都检索出来,如果没有则以NULL匹配。

连接查询可以采用表的别名,用于限定在两个或多个表中重复的列名,或多次使用相同的表,以缩短语句书写长度。此任务中可以使用表的别名,代码如下。

```
SELECT TTopic AS 主题,SName AS 版块名称  
FROM Topic AS a FULL OUTER JOIN Section AS b  
ON LEFT(a.TTopic,1)=LEFT(b.SName,1)
```

外部连接的语法格式如下。

左外连接:

```
FROM 表 1 LEFT [OUTER] JOIN 表 2 ON 连接条件
```

右外连接:

```
FROM 表 1 RIGHT [OUTER] JOIN 表 2 ON 连接条件
```

全外连接:

```
FROM 表 1 FULL [OUTER] JOIN 表 2 ON 连接条件
```

说明: 在指定表进行连接查询时,这种处理是非常耗费资源的。因此在使用表连接时要考虑哪些表是必要的。

5.2.4 使用组合查询

1. 任务描述

- ①在发帖表 Topic 和回帖表 Reply 中查询所有发帖人和回帖人的编号和帖子主题。
- ②在发帖表 Topic 和回帖表 Reply 中查询发帖人编号为 3,4,11 和回帖人编号为 4,11,31 的帖子主题。
- ③在发帖表 Topic 和回帖表 Reply 中查询所有发帖人和回帖人的编号和帖子主题,并按照发帖人编号进行升序排列。

2. 技能要求

- ①正确使用 UNION 关键字。
- ②正确使用 WHERE 子句过滤组合结果集。
- ③正确使用 ORDER BY 子句过滤组合结果集。

3. 任务实现

1) 简单组合查询

①在发帖表 Topic 和回帖表 Reply 中查询所有发帖人和回帖人的编号和帖子主题。在查询输入窗格中输入以下语句。

```
SELECT TUserID AS 用户编号,TTopic AS 主题  
FROM Topic  
UNION  
SELECT RUserID AS 用户编号,RTopic AS 主题  
FROM Reply
```



②单击“执行”按钮,结果集如图 5-6 所示。

2)限制条件的组合查询

①在发帖表 Topic 和回帖表 Reply 中查询发帖人编号为 3,4,11 和回帖人编号为 4,11,31 的帖子主题。在查询输入窗格中输入以下 T-SQL 语句。

```
SELECT TUserID AS 用户编号, TTopic AS 主题
FROM Topic
WHERE TUserID in ('3','4','11')
UNION
SELECT RUserID AS 用户编号, RTopic AS 主题
FROM Reply
WHERE RUserID in ('4','11','31')
```

②单击“执行”按钮,结果集如图 5-15 所示。

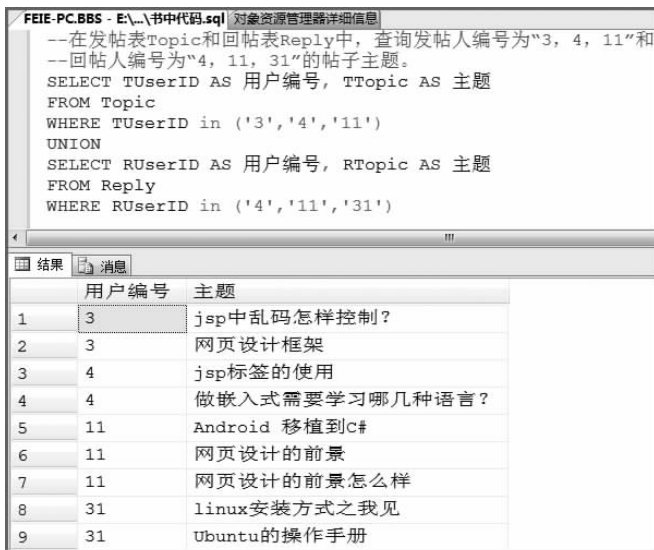


图 5-15 使用限制条件的组合查询

此任务中的两个 SELECT 语句根据每个表的 WHERE 子句过滤数据行,并将结果集进行组合。

3)对组合查询结果排序

①在发帖表 Topic 和回帖表 Reply 中查询所有发帖人和回帖人的编号和帖子主题,并按照发帖人编号进行升序排序。在查询输入窗格中输入以下语句。

```
SELECT TUserID AS 用户编号, TTopic AS 主题
FROM Topic
UNION
SELECT RUserID AS 用户编号, RTopic AS 主题
FROM Reply
ORDER BY TUserID
```

②单击“执行”按钮,结果集如图 5-16 所示。



图 5-16 对组合查询结果排序

本任务在对组合查询排序时,只能使用一条 ORDER BY 子句,并且 ORDER BY 子句只能出现在最后一条 SELECT 语句之后,即只用一种方式对结果集进行排序。

组合查询的语法格式如下。

```
SELECT 列名 1[, ...,列名 n]| 计算表达式
```

```
FROM 表名 1
```

```
[WHERE 检索条件]
```

```
UNION
```

```
...
```

```
SELECT 列名 1[, ...,列名 n]| 计算表达式
```

```
FROM 表名 n
```

```
[WHERE 检索条件]
```

```
[ORDER BY 排序表达式]
```

使用时应注意以下几点。

①UNION 使用规则。使用 UNION 时必须要有两条及两条以上 SELECT 语句,各 Select 语句之间用 UNION 分隔;每个 UNION 查询必须包括相同的列、表达式或聚合函数;列的数据类型必须是相互兼容的。

②使用组合查询的性能问题。理论上执行多条 WHERE 子句和 UNION 子句在性能方面应该没有差别,但在实际中最好先验证一下,然后再选择查询方式。

③使用 UNION 的限制。对于 UNION 子句可以组合的 SELECT 子句的数目,不同的 DBMS 存在不同的标准,需要查询具体的相关文档。

5.2.5 使用子查询过滤数据

1. 任务描述

①检索发帖表 Topic 中所属版块为“Web 开发”的所有主题 TTopic 和所在版块编号 TSectionID。

②在发帖表 Topic 和回帖表 Reply 中查询回帖点击数 RClickCount 排名前 3 的帖子主题 TTopic。

③在版块信息表中查找编号为 2 的版主所发帖子的回帖数量。

④查询没有发过帖子的注册用户信息。

2. 技能要求

①正确使用返回单一值的子查询。

②正确使用返回值列表的子查询。

③正确使用多层嵌套子查询。

④正确使用 EXISTS 关键字。

3. 任务实现

1) 返回单一值的子查询

①检索发帖表 Topic 中所属版块为“Web 开发”的所有主题 TTopic 和所在版块编号 TSectionID。在查询输入窗格中输入以下语句。

```
SELECT TUserID AS 用户编号, TTopic AS 主题, TSectionID AS 所在版块编号
FROM Topic
WHERE TSectionID =
    (SELECT SectionID FROM Section WHERE SName = 'Web 开发')
```

②单击“执行”按钮,结果集如图 5-17 所示。

| 用户编号 | 主题 | 所在版块编号 |
|------|-------------|--------|
| 2 | jsp中乱码怎样控制? | 1 |
| 2 | jsp常见错误 | 1 |

图 5-17 返回单一值的子查询

此任务中满足条件的数据是唯一的,在以内部子查询返回的单一值作为过滤条件时,外部查询中 WHERE 子句的过滤条件表达式要采用“=”运算符。

2) 返回值列表的子查询

①在发帖表 Topic 和回帖表 Reply 中查询回帖点击数 RClickCount 排名前 3 的帖子主题 TTopic。在查询输入窗格中输入以下语句。

```
SELECT TopicID AS 帖子编号, TTopic AS 主题
```



```

FROM Topic
WHERE TopicID IN
    (SELECT TOP 3 RTopicID FROM Reply ORDER BY RClickCount DESC)
    
```

②单击“执行”按钮,结果集如图 5-18 所示。

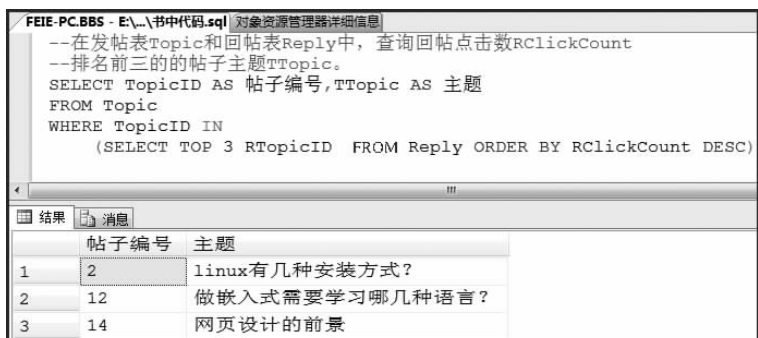


图 5-18 返回值列表的子查询

此任务的内层子查询返回的结果集为值列表,将其作为过滤条件时,外部查询中 WHERE 子句的过滤条件表达式要采用 IN 运算符。IN 表示属于,NOT IN 表示不属于。

在执行内部子查询时,返回结果集的帖子编号是按点击数 RClickCount 由高到低顺序输出的,而在外部查询中按照 IN 逻辑运算符检索时,只是按照值列表中存在的帖子编号 TopicID 检索数据,并没有按点击数 RClickCount 排序。

3) 多层嵌套子查询

①在版块信息表中查找编号为 2 的版主所发帖子的回帖数量。在查询输入窗格中输入以下语句。

```

SELECT COUNT(ReplyID) AS 回帖数量
FROM Reply
WHERE RTopicID IN
    (SELECT TopicID FROM Topic WHERE TUserID='2' AND TSectionID IN
        (SELECT SectionID FROM Section WHERE SMasterID='2'))
    
```

②单击“执行”按钮,结果集如图 5-19 所示。

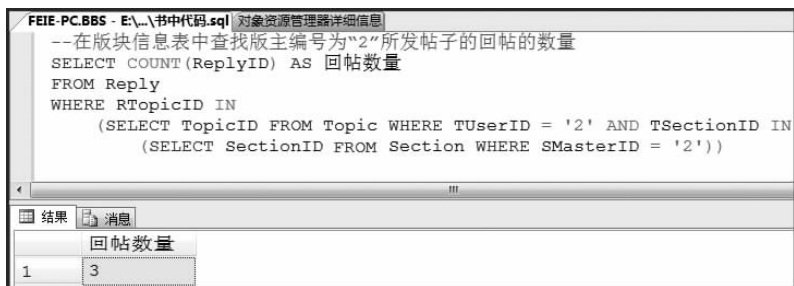


图 5-19 多层嵌套子查询

多层嵌套子查询是从最内层开始执行,由内向外进行处理的。此任务中要查找编号为 2 的版主所发帖子的回帖数量,先在最内层子查询返回编号为 2 的版主所有的版块编号 Sec-

tionID, 然后将其作为过滤条件以检索外面一层子查询所属版块编号 TSectionID, 并且次内层子查询同时满足发帖人编号为 2, 因此在外部查询中 WHERE 子句回复主帖编号满足子查询条件。

4) EXISTS 子查询

① 查询没有发过帖子的注册用户信息。在查询输入窗格中输入以下语句。

```
SELECT UserID AS 用户编号, UserName AS 用户名称
FROM Users
WHERE NOT EXISTS
(SELECT * FROM Topic WHERE Users.UserID=Topic.TUserID)
```

② 单击“执行”按钮, 结果集如图 5-20 所示。



图 5-20 EXISTS 子查询

使用 EXISTS 关键字引入一个子查询时, 对外部查询的数据行相当于进行了一次存在测试。外部查询的 WHERE 子句测试满足子查询的行是否存在。子查询实际上不产生任何数据, 它只返回 True 或 False 值。

查询没有发过帖子的注册用户信息时, 子查询检索满足条件的数据是否存在, 如果发帖表中的 TUserID 与用户信息表中任意行的 UserID 匹配, 外部查询中 WHERE 子句的 NOT EXISTS 关键字将返回 False 值, 即结果集中不存在该行。

5.3 项目小结

本项目主要进行 BBS 网站论坛的高级检索。使用聚合函数实现数据汇总时, 以列名或计算表达式为参数, 如果不重复累计列值, 可以加入 DISTINCT 关键字。使用 GROUP BY 分组检索时, 在 SELECT 子句检索列中, 除了聚合计算列, 每一列都应在 GROUP BY 子句中出现。使用表连接查询时, 要考虑哪些表是必要的, 并根据结果集的要求选择内部连接、左外连接、右外连接还是全外连接。使用组合查询时, 每个 UNION 查询必须包括相同的列、表达式或聚合函数, 同时列的数据类型必须是相互兼容的。使用子查询检索时, 子查询可以返回单一值或多个值, 也可以通过 EXISTS 关键字判断是否有符合条件的数据。



5.4 理论练习

一、选择题

- 关于聚合函数,以下说法错误的是()。
 - SUM()返回表达式中所有数的总和,因此只能用于数字类型的列
 - AVG()返回表达式中所有数的平均值,可以用于数字型和日期型的列
 - MAX()和 MIN()可以用于字符型的列
 - COUNT()可以用于字符型的列
- 关于多表连接查询,以下描述错误的是()。
 - 外连接查询返回的结果集行数可能大于所有符合连接条件的结果集行数
 - 多表连接查询必须使用 JOIN 关键字
 - 内连接查询返回的结果是所有符合连接条件的数据
 - 在 WHERE 子句中指定连接条件可以实现内连接查询
- 现有书目表 Book,包含价格 Price(float)和类别 Type(char)字段,查询各类别的平均价格和类别名称,以下语句正确的是()。
 - SELECT AVG(Price), Type FROM Book GROUP BY Type
 - SELECT COUNT(Price), Type FROM Book GROUP BY Price
 - SELECT AVG(Price), Type FROM Book GROUP BY Price
 - SELECT COUNT(Price), Type FROM Book GROUP BY Type
- 现有顾客表 Customers,包含的数据见下表,若执行 SELECT AVG(Discount) FROM Customers 语句,则输出结果是()。

| CId | CName | Discount |
|-----|-------|----------|
| 1 | jack | NULL |
| 2 | lee | 8 |
| 3 | tom | 7 |
| 4 | chen | 1 |

- 错误提示:不能对 NULL 进行 AVG()操作
 - 16
 - 8
 - 4
- 关于分组查询,以下描述错误的是()。
 - 使用 GROUP BY 进行分组查询
 - 对分组后的条件的筛选必须使用 HAVING 子句
 - HAVING 子句可以与 WHERE 子句同时出现在一个 SELECT 语句中
 - 使用分组查询时,在 SELECT 列表中只能出现被分组的列



二、填空题

1. 在 SQL Server 2005 中连接查询分为_____、_____和_____。
2. 外部连接查询包括_____、_____和_____。
3. 分组查询时使用_____子句,组合查询时使用_____子句。
4. 当子查询返回值为单一值时,运算符采用_____;当子查询返回列表值时,运算符采用_____;当子查询判断数据是否存在时,运算符采用_____。

5.5 项目实战

1. 项目实战 1

在回帖表 Reply 中实现数据的分组和组合查询。

1) 任务说明

- ①在回帖表 Reply 中统计每个版块的回帖数量。
- ②在回帖表 Reply 中查询每个版块回帖数量大于 5 的帖子数量和所属版块编号。
- ③将回帖表 Reply 中的发帖人编号 RUserID 和所属版块编号 RSectionID 与版块信息表 Section 中的版主编号 MasterID 与版块编号 SectionID 进行组合。

2) 技能要点

- ①正确使用聚合函数。
- ②正确使用 GROUP BY 子句分组。
- ③正确使用 HAVING 子句过滤分组。
- ④正确使用 UNION 组合数据。

3) 实现思路

- ①用表主键值统计数量可以用 COUNT(列名),也可以用 COUNT(*),因为主键唯一,所以函数值是相同的。
- ②进行组合查询时,选择的列数据类型要相互兼容。

2. 项目实战 2

在发帖表 Topic、回帖表 Reply 和用户信息表 Users 中实现联合查询。

1) 任务说明

- ①查询发帖表 Topic 中所有发帖人的名称。
- ②在回帖表 Reply 中查询回复主题为“jsp 常见错误”和发帖人名称为 chunyu 的帖子。
- ③查询版块信息表 Section 中所有版块的发帖表信息。

2) 技能要点

- ①正确使用内部连接查询。
- ②正确使用外部连接查询。



3)实现思路

- ①通过表之间的主外键实现连接查询。
- ②在使用内部连接查询过滤数据时,应将相应语句写到 ON 关键字后。
- ③左外连接查询中以全部检索的数据表作为左表,右外连接查询中以全部检索的数据表作为右表。

3. 项目实战 3

在发帖表 Topic、回帖表 Reply 和用户信息表 Users 中实现子查询。

1)任务说明

- ①查询回帖表 Reply 中回帖人的名称为 xiaolongnv 的帖子信息。
- ②在回帖表 Reply 中查询回复主题为“jsp 常见错误”和发帖人名称为 chunyu 的帖子。
- ③在发帖表 Topic 中查询是否存在版块名称为“数据库开发”的帖子,并按发帖时间 TReleaseTime 降序排列。

2)技能要点

- ①正确使用子查询。
- ②正确使用嵌套子查询。
- ③正确使用 EXISTS 关键字。

3)实现思路

- ①实现子查询由内向外分析检索条件。
- ②注意嵌套子查询的内外连接关系。
- ③[NOT] EXISTS 子查询判断是否有符合条件的数据,子查询返回值为 True 或 False。