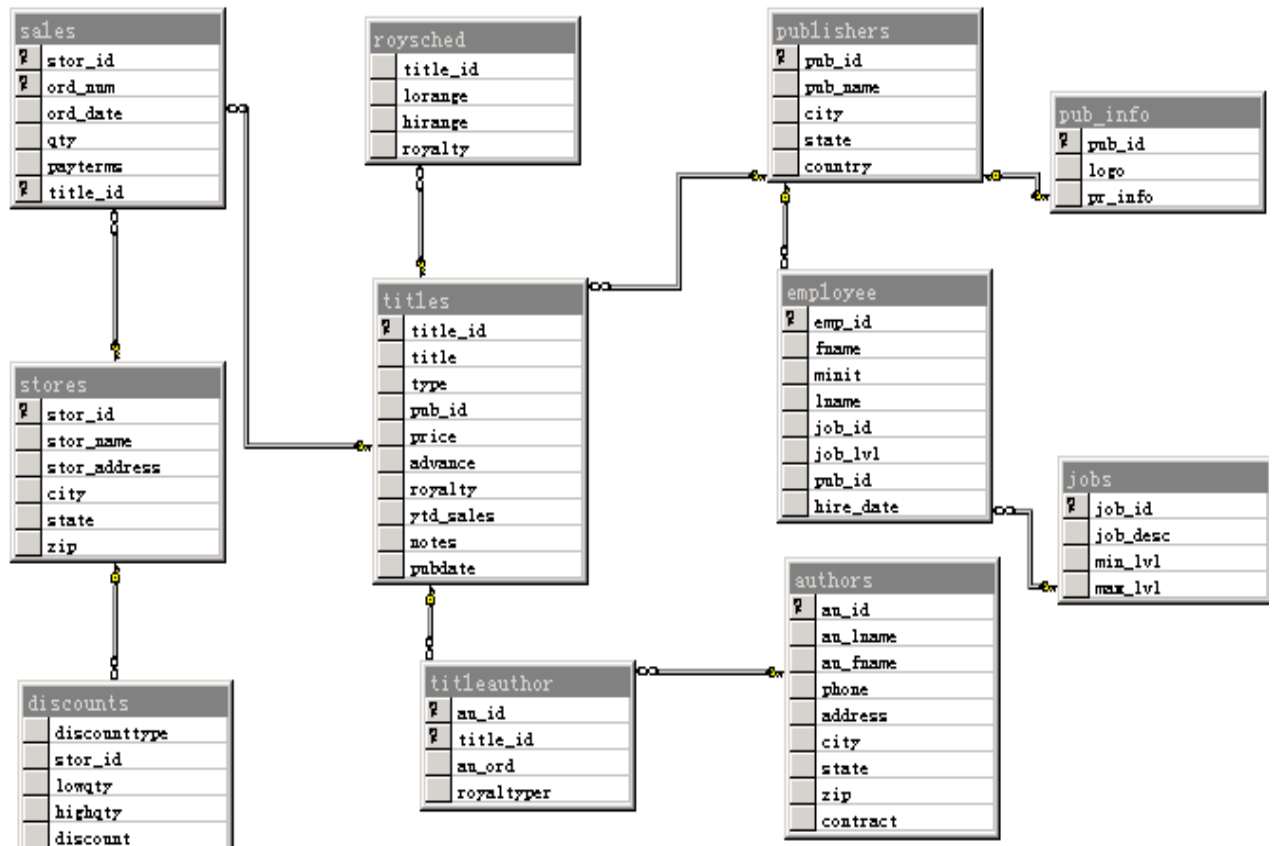
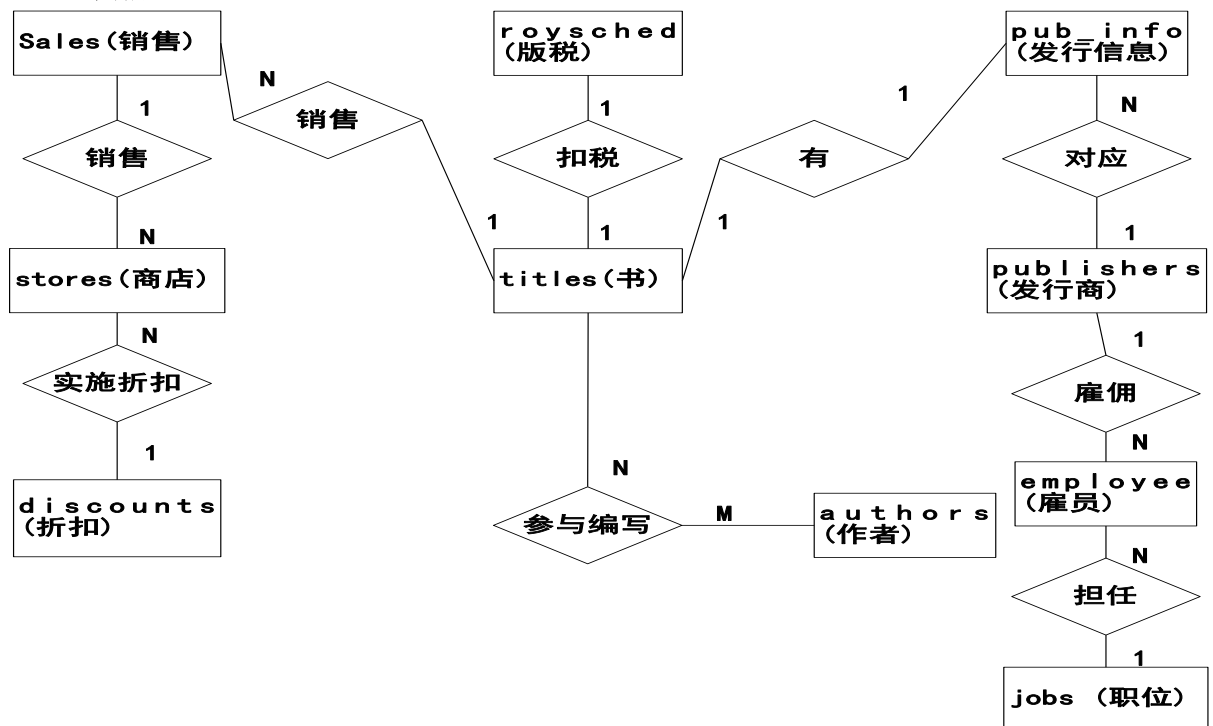


该实验以 SQL Server 2000 系统自带的 pubs 数据库为例，以一个图书出版公司为模型。

● SQL SERVER200 实体关系图描述



● E-R 图描述



(1) 该系统中数据库基本表如下：

Authors:

属性名	数据类型	含义说明	可为空	检查	键/索引
au_id	Id	作者编号	否	是 ¹	主键
au_lname	varchar(40)	作者姓	否		
au_fname	varchar(20)	作者名	否		
phone	char(12)	电话	否		
address	varchar(40)	地址	是		
city	varchar(20)	所在城市	是		
state	char(2)	所在州	是		
zip	char(5)	邮编	是	是 ²	
contract	Bit	是否签约	否		

1 **au_id** CHECK 约束定义为 (**au_id** LIKE '[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9][0-9][0-9]').

2 **zip** CHECK 约束定义为 (**zip** LIKE '[0-9][0-9][0-9][0-9][0-9]').

=====

discounts

属性名	数据类型	含义说明	可为空	检查	键/索引
discounttype	varchar(40)	折扣类型	否		
stor_id	char(4)	商店编号	是		外键 stores(stor_id)
lowqty	Smallint	数量下限	是		
highqty	Smallint	数量上限	是		
discount	Float	折扣	否		

=====

Employee

属性名	数据类型	含义说明	可为空	默认值	检查	键/索引
emp_id	Empid	职工编号	否		是 ¹	主键
fname	varchar(20)	职工名	否			
minit	char(1)		是			
lname	varchar(30)	职工姓	否			
job_id	Smallint	工作编号	否	1		外键 jobs(job_id)
job_lvl	Tinyint		否	10		
pub_id	char(4)	出版社编号	否	'9952'		外键 publishers(pub_id)
Hire_date	Datetime	工作日期	否	GETDATE()		

CHECK 约束定义为：

(**emp_id** LIKE '[A-Z][A-Z][A-Z][1-9][0-9][0-9][0-9][0-9][FM]') OR

(**emp_id** LIKE '[A-Z]-[A-Z][1-9][0-9][0-9][0-9][0-9][FM]')。

Jobs

属性名	数据类型	含义说明	可为空	检查	键/索引
job_id	Smallint	工作编号	否		主键
job_desc	varchar (50)	工作描述	否		
min_lvl	Tinyint		否	是 ¹	
max_lvl	Tinyint		否	是 ²	

(1) min_lvl CHECK 约束定义为 (min_lvl >= 10)。

(2) max_lvl CHECK 约束定义为 (max_lvl <= 250)。

pub_info

属性名	数据类型	含义说明	可为空	检查	键/索引
pub_id	char(4)	出版社编号	否		主键，外键 publishers(pub_id)
logo	Image	标志图	是		
pr_info	Text	出版信息	是		

Publishers

属性名	数据类型	含义说明	可为空	检查	键/索引
pub_id	char(4)	出版社编号	否	是 ¹	主键
pub_name	varchar(40)	出版社名称	是		
city	varchar(20)	所在城市	是		
state	char(2)	所在州	是		
country	varchar(30)	所在国家	是		

1 pub_id CHECK 约束定义为

(pub_id = '1756' OR (pub_id = '1622' OR (pub_id = '0877' OR (pub_id = '0736' OR (pub_id = '1389')))) OR (pub_id LIKE '99[0-9][0-0]'))。

roysched

属性名	数据类型	含义说明	可为空	检查	键/索引
title_id	Tid	书编号	否		外键 titles(title_id)
lorange	Int	低	是		
hirange	Int	高	是		
royalty	Int	版权	是		

Sales

属性名	数据类型	含义说明	可为空	键/索引
stor_id	char(4)	商店编号	否	组合主键，聚集索引,外键 stores(stor_id)

ord_num	varchar(20)	订单编码	否	组合主键，聚集索引
ord_date	Datetime	订购日期	否	
qty	Smallint	数量	否	
payterms	varchar(12)	付款方式	否	
title_id	Tid	书编号	否	组合主键，聚集索引,外键 titles(title_id)

titles

属性名	数据类型	含义说明	可为空	检查	键/索引
title_id	Tid	书编号	否		主键
title	varchar(80)	书名	否		
type	char(12)	类型	否		
pub_id	char(4)	出版社编号	是		外键 publishers (pub_id)
price	Money	价格	是		
advance	Money	预付款	是		
royalty	Int	版税	是		
Ytd_sales	Int	年销售量	是		
notes	varchar(200)	简介	是		
pubdate	Datetime	出版日期	是		

Stores

属性名	数据类型	含义说明	可为空	检查	键/索引
stor_id	char(4)	商店编号	否		主键
stor_name	varchar(40)	商店名称	是		
stor_address	varchar(40)	商店地址	是		
city	varchar(20)	所在城市	是		
state	char(2)	所在州	是		
zip	char(5)	邮编	是		

titleauthor

属性名	数据类型	含义说明	可为空	检查	键/索引
au_id	id	作者编号	否		组合主键，聚集索引,外键 authors(au_id)
title_id	tid	书编号	否		组合主键，聚集索引,外键 titles(title_id)
au_ord	tinyint		是		

royaltyper	int	版权百分比	是		
------------	-----	-------	---	--	--

(2) 练习内容

目的 1:

1. 加深对表间关系的理解。
2. 理解数据库中数据的查询方法和应用。
3. 学会各种查询的异同及相互之间的转换方法。

内容 1:

1. 查询所有作者的作者号、姓名信息
2. 查询所有作者的姓名、作者号信息，并在每个作者的作者号前面显示字符串“身份证号：”，表明显示的信息是身份证信息
3. 查询在 CA 州的作者姓名和城市
4. 查询出版日期在 1992.1.1-2000.12.31 之间的书名和出版日期(查询 1991 年出版的书)
5. 查询每个出版社出版的书
6. 查询某店销售某书的数量
7. 查询有销售记录的所有书信息，包括书的编号、书名、类型和价格
8. 查询已销售书的信息
9. 显示所有的书名（无销售记录的书也包括在内）
10. 查询已销售书的信息（书号、书名、作者等）
11. 查询所有出版商业（business）书籍的出版社的名称

目的 2:

1. 理解数据库中数据的其他查询方法和应用；
2. 学会各种查询要求的实现。

内容 2:

在实验 1 的基础上，练习查询语句的使用，包括计算列、求和、最大、最小值、各类选择条件、字符匹配、分组和排序，体会各种查询的执行过程，为简单综合应用打下良好的基础。

1. 查询书名以 T 开头或者出版社号为 0877，而且价格大于 16 的书的信息。
2. 按照类型的升序和价格的降序（在类型相同时）显示书的信息（书名、作者、出版社、类型、价格）
3. 查询销售量大于 30 的书名及销售数量
4. 查询在 2002.1.1 到 2002.10.31 间，每本书的销售总额
5. 查询所有作者的所在城市和州名，要求没有重复信息
6. 计算多少种书已被订价
7. 查询每本书的书名、作者及它的售书总量
8. 计算所有书的平均价格
9. 查询价格最高的书的书名、作者及价格

目的 3:

1. 加深对数据库相关性质的理解；

2. 各种约束性理解;
3. 学会数据库中数据的更新的方法。

内容 3:

1. 参照以上各表给出的主键、外键设置的设置要求, 在自己创建的表中进行相应的设置。
2. 向 authors 表中插入一行作者信息 (具体值自定)
3. 数量超过 100 的商店增加 10% 的折扣
4. 删除 2001.10.3 的订单
5. 删除 1 中所建立的索引
6. 建立 CA 州作者所著书的视图 (包括作者号、姓名、所在州、书名、价格、出版日期)
7. 建立付款方式是现金 (cash) 的订单视图
8. 建立 CA 州的所有商店的视图

目的 4:

1. 在查询分析器中, 练习使用 IN、比较符、ANY 或 ALL 等操作符进行查询。
2. 练习使用 EXISTS 操作符进行嵌套查询操作

内容 4:

1. 在 pubs 数据库的 titleauthor 和中, 用 IN 谓词查询来自 ‘CA’ 州(在 authors 表中) 的作家的全部作品 (title_id) 和作家的代号 (au_id)。
2. 在 pubs 数据库中, 用比较运算符引出的子查询找出在名称为 “Algodata Infosystems” 的出版社所在城市中的作者的姓名 (au_lname, au_fname)
3. 在 pubs 数据库中的 titles 表中, 查询价格大于所有类型 (TYPE) 为 “business” 的图书价格的书名 (title) 和价格 (price)
4. 在 pubs 数据库的 sales 表中查找所有销售量大于所有图书平均销售量 avg (qty) 的书的代号 (title_id) 及销售量 (qty)。
5. 用带有 IN 的嵌套查询, 查询来自城市 (city) 为 “London” 的客户所订的订单信息 (customers 和 orders 表)。
6. 用带有 IN 的嵌套查询, 查询 Northwind 数据库中的产品表 (Products) 中来自国家为 “Germany” (在供应商表 (Suppliers) 表中) 的供货商供应的产品信息 (包括 Productid, Productname, categoryid, unitprice)。
7. 使用 EXISTS 子查询在 Pubs 数据库 titles 表及 publishers 表中查询 New Moon Books 出版社所出版的图书名称 (title)

目的 5:

1. 分类汇总。

内容 5:

1. 找出 pubs 数据库 titles 表中计算机类图书中价格最高的图书的价格。
2. 查询 titles 表中有几类图书。
3. 按照州进行分类, 查找每个州有几名作者。

4. 要求按照出版商 id 进行分类, 查找每个出版商的书到目前为止的销售额总和 (ytd_sales)。
5. 在 pubs 数据库的 titles 表中, 找出平均价格大于 18 美元的书的种类。
6. 在 pubs 数据库的 titles 表中, 找出最高价大于 20 美元的书的种类。
7. 找出 title_id 和 pub_name 的对应关系。
8. 找出 title_id, title 和 pub_name 的对应关系。
9. 查询每个作者的编号, 姓名, 所出的书的编号, 并对结果排序。
10. 从 authors 表中选择 state,city 列, 从 publisher 表中选择 state,city 列, 并把两个查询的结果合并为一个结果集, 并对结果集按 city 列、state 列进行排序。
11. 对上面的查询语句作修改, 保留所有重复的记录。
12. 显示所有来自 CA 州的作家的全部作品和作家代号。(使用 IN, 和连接两种方法)
13. 查找由位于以字母 B 开头的城市中的任一出版商出版的书名: (使用 exists 和 in 两种方法)

一、简单查询学生选课数据

- 1、列出全部学生的信息。
- 2、列出信息系全部学生的学号及姓名。
- 3、列出所有已被选修的选修课的课号。
- 4、求 c01 号课成绩大于 80 分的学生的学号及成绩, 并按成绩由高到低列出。
- 5、列出非信息系学生的名单。
- 6、查询成绩在 70~80 分之间的学生选课得分情况
- 7、列出选修 c01 号课或 c03 号课的全体学生的学号和成绩。
- 8、列出所有 95 级学生的学生成绩情况。
- 9、列出成绩为空值(或不为空值)的学生的学号和课号。
- 10、求出所有学生的总成绩。
- 11、列出每个学生的平均成绩。
- 12、列出各科的平均成绩、最高成绩、最低成绩和选课人数。

1 SELECT * FROM 学生

2 SELECT 学号,姓名 FROM 学生 WHERE 专业='信息系'

3 SELECT DISTINCT 课号 FROM 选修课

4 SELECT 学号,成绩 FROM 选课 WHERE 课号='01' AND 成绩>80 ORDER BY 成绩 DESC

5 方法一: SELECT 姓名 FROM 学生 WHERE 专业<>'信息系'

方法二: SELECT 姓名 FROM 学生 WHERE NOT 专业='信息系'

方法三: SELECT 姓名 FROM 学生 WHERE 专业!='信息系'

6 方法一: SELECT * FROM 选课 WHERE 成绩>=70 AND 成绩<=80

方法二: SELECT * FROM 选课 WHERE 成绩 BETWEEN 70 AND 80

不在此范围内的查询:(注意写出和以下语句等价的语句)

SELECT * FROM 选课 WHERE 成绩 NOT BETWEEN 70 AND 80

7 方法一: SELECT 学号,成绩 FROM 选课 WHERE 课号='c01' OR 课号='c03'

方法二: SELECT 学号,成绩 FROM 选课 WHERE 课号 IN ('c01','c03')

相反条件查询: SELECT 学号,成绩 FROM 选课 WHERE 课号 NOT IN ('c01','c03')

8 SELECT * FROM 选课 WHERE 学号 LIKE '95%'

SELECT * FROM 选课 WHERE 学号 LIKE '95____'

相反条件查询: SELECT * FROM 选课 WHERE 学号 NOT LIKE '98%'

9 答案一: SELECT 学号,课号 FROM 选课 WHERE 成绩 IS NULL

答案二: SELECT 学号,课号 FROM 选课 WHERE 成绩 IS NOT NULL

10 SELECT SUM(成绩) AS 总成绩 FROM 选课

11 SELECT 学号,AVG(成绩) AS 平均成绩 FROM 选课 GROUP BY 学号

12 SELECT 课号,AVG(成绩) AS 平均成绩,MAX(成绩) AS 最高分,
MIN(成绩) AS 最低分,COUNT(学号) AS 选课人数 FROM 选课 GROUP BY 课号

目的 4:

2. 在查询分析器中,练习使用 IN、比较符、ANY 或 ALL 等操作符进行查询。

3. 练习使用 EXISTS 操作符进行嵌套查询操作

请完成以下习题:

14. 在 pubs 数据库的 titleauthor 和中,用 IN 谓词查询来自 'CA' 州(在 authors 表中)的作家的全部作品(title_id)和作家的代号(au_id)。

```
select title_id,au_id
from titleauthor
```



```

where au_id in (select au_id
                 from authors
                 where state='CA')

```

在 pubs 数据库中，用比较运算符引出的子查询找出在名称为 “Algodata Infosystems” 的出版社所在城市中的作者的姓名（au_lname, au_fname）

```

select au_lname, au_fname
from authors
where city= (select city
              from publishers
              where pub_name='Algodata Infosystems')

```

在 pubs 数据库中的 titles 表中，查询价格大于所有类型(TYPE)为 “business” 的图书价格的书名（title）和价格(price)

```

select title,price
from titles
where price>all (select price
                  from titles
                  where type='business')

```

在 pubs 数据库的 sales 表中查找所有销售量大于所有图书平均销售量 avg（qty）的书的代号(title_id)及销售量（qty）。

```

select title_id ,qty
from sales
where qty>all(select avg(qty)
              from sales
              )

```

用带有 IN 的嵌套查询，查询来自城市（city）为 “London” 的客户所订的订单信息（customers 和 orders 表）。

```

select *
from orders
where customerID in (select customerID
                     from customers
                     where city='london')

```

用带有 IN 的嵌套查询，查询 Northwind 数据库中的产品表（Products）中来自国家为 “Germany” （在供应商表（Suppliers）表中）的供货商供应的产品信息（包括 Productid, Productname, categoryid, unitprice）。

```

SELECT Productid,Productname, categoryid,unitprice
from Products
where Productid in(select supplierID
                   from Suppliers
                   where country='Germany')

```

2、提高操作实验

练习使用 EXISTS 操作符进行嵌套查询操作。请完成以下习题：

使用 EXISTS 子查询在 Pubs 数据库 titles 表及 publishers 表中查询 New Moon Books 出版社所出版的图书名称（title）

```
select title
      from titles
     where exists (select *
                  from publishers
                  where pub_name='New Moon Books')
```

T-SQL 高级查询课堂练习及答案

--练习 1

--找出 pubs 数据库 titles 表中计算机类图书中价格最高的图书的价格。

```
USE pubs
GO
SELECT max(price) FROM titles
where type='popular_comp'
GO
```

--练习 2

--查询 titles 表中有几类图书。

```
USE pubs
GO
SELECT count(distinct type) FROM titles
GO
```

--练习 3

--按照州进行分类，查找每个州有几名作者。

```
USE pubs
GO
SELECT state, count(*) FROM authors
group by state
order by 1
GO
```

--练习 4

--要求按照出版商 id 进行分类，查找每个出版商的书到目前为止的销售额总和 (ytd_sales)。

```
USE pubs
GO
SELECT pub_id, sum(ytd_sales) FROM titles
group by pub_id
order by 1
```

GO

--练习 5

--在 pubs 数据库的 titles 表中，找出平均价格大于 18 美元的书的种类。

USE pubs

GO

SELECT pub_id,avg(price) '平均价格' FROM titles

GROUP BY pub_id

HAVING avg(price) > 18

GO

--练习 6

--在 pubs 数据库的 titles 表中，找出最高价大于 20 美元的书的种类。

USE pubs

GO

SELECT type,max(price) '平均价格' FROM titles

GROUP BY type

HAVING max(price) > 20

GO

--练习 7

--

找出 title_id 和 pub_name 的对应关系。

Use pubs

go

Select titles.title_id, publishers.pub_name

From titles JOIN publishers

ON titles.pub_id=publishers.pub_id

Go

--练习 8

--找出 title_id, title 和 pub_name 的对应关系。

Use pubs

go

Select titles.title_id, titles.title, publishers.pub_name

From titles JOIN publishers

ON titles.pub_id=publishers.pub_id

Go

--练习 9

--查询每个作者的编号，姓名，所出的书的编号，并对结果排序。

Use pubs

go

```

Select authors.au_id,
       authors.au_fname + '.' + authors.au_lname 'name',
       titleauthor.title_id
From authors JOIN titleauthor
ON authors.au_id=titleauthor.au_id
order by authors.au_id
go

```

10. 从 authors 表中选择 state,city 列，从 publisher 表中选择 state,city 列，并把两个查询的结果合并为一个结果集，并对结果集按 city 列、state 列进行排序。

```

use pubs
go
select state,city from publishers
union
select state,city from authors
order by 1,2

```

11. 对上面的查询语句作修改，保留所有重复的记录。

```

use pubs
go
select state,city from publishers
union all
select state,city from authors
order by 1,2

```

12. 显示所有来自 CA 州的作家的全部作品和作家代号。（使用 IN，和连接两种方法）

```

use pubs
go
select title_id,au_id
from titleauthor
where au_id in
( select au_id from authors
where state = 'CA')
order by title_id
go

use pubs
go
select t.title_id,t.au_id
from titleauthor t join authors a
on t.au_id = a.au_id

```

```
where a.state = 'CA'
order by title_id
go
```

13. 查找由位于以字母 B 开头的城市中的任一出版商出版的书名：(使用 exists 和 in 两种方法)

```
USE pubs
GO
SELECT title
FROM titles
WHERE EXISTS
    (SELECT *
     FROM publishers
     WHERE pub_id = titles.pub_id
     AND city LIKE 'B%')
GO
```

```
USE pubs
GO
SELECT title
FROM titles
WHERE pub_id IN
    (SELECT pub_id
     FROM publishers
     WHERE city LIKE 'B%')
GO
```