**1、检索**

**单个列**：SELECT prod\_id

FROM products;

**多个列**：在关键字后面给出多个列名，以逗号分割

SELECT prod\_id, prod\_name, prod\_price

FROM Products;

**所有列**：SELECT \*

FROM Products;

**2、排序**

**单列排序**：SELECT prod\_name

FROM Products

ORDER BY prod\_name;

**多列排序**（首先按价格，然后按名称排序）：

SELECT prod\_id, prod\_price, prod\_name

FROM Products

ORDER BY prod\_price, prod\_name;

**按相对列位置进行排序**：

SELECT prod\_id, prod\_price, prod\_name

FROM Products

ORDER BY 2, 3;(数字与前面列名相对应)

**指定单列排序方向**（降序排序，指定DESC关键字）：

SELECT prod\_name

FROM Products

ORDER BY prod\_price DESC;

**指定多列排序方向**（DESC只作用于其前面的列名）：

SELECT prod\_name

FROM Products

ORDER BY prod\_price DESC, prod\_name;

**3、过滤**

WHERE子句在表名（FROM）之后给出

**简单过滤**：

SELECT prod\_name, prod\_price

FROM Products

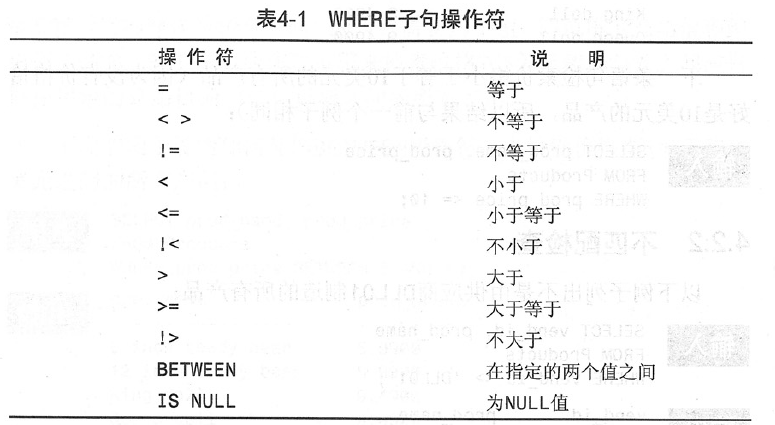
WHERE prod\_price = 3.49;

**范围过滤**：

SELECT prod\_name, prod\_price

FROM Products

WHERE prod\_price BETWEEN 5 AND 10;



**高级过滤**：

AND OR操作符（AND优先级最高，解决冲突可用括号）：

SELECT prod\_name, prod\_price

FROM Products

WHERE (vend\_id = 'DLL01' OR vend\_id = 'BRS01') AND prod\_price >= 10;

IN操作符：用来指定条件范围，范围内的每个条件都可以匹配，功能同OR

SELECT prod\_name, prod\_price

FROM Products

WHERE vend\_id IN ('DLL01','BRS01')

ORDER BY prod\_name;

NOT操作符：

SELECT prod\_name

FROM Products

WHERE NOT vend id = 'DLL01'

ORDER BY prod\_name;

**4、用通配符进行过滤（LIKE操作符）**

百分号（%）：表示任何字符出现任意次数

SELECT prod\_id, prod\_name

FROM Products

WHERE prod\_name LIKE'Fish%';

下划线：只能匹配单个字符

SELECT prod\_id, prod\_name

FROM Products

WHERE prod\_name LIKE '\_ inch teddy bear';

方框号：匹配框号中的单个字符，相反前面用^符号

SELECT cust\_contact

FROM Customers

WHERE cust\_contact LIKE'[JM]%'

ORDER BY cust\_contact;

**5、计算字段**

拼接：用+和||符号将多个字段拼接在一起，MySQL只支持concat函数

SELECT vend\_name + '(' + vend\_country + ')'

FROM Vendors

ORDER BY vend\_name;

SELECT vend\_name || '(' || vend\_country || ')'

FROM Vendors

ORDER BY vend\_name;

SELECT CONCAT (vend\_nae, ' (', vend\_country, ') ')

去空格：上述操作会保存填充为列宽的文本值，使用rtrim去空格

SELECT RTRIM (vend\_name) + ' (' + RTRIM (vend\_country) + ') '

FROM Vendors

ORDER BY vend\_name;

给连接后的新字段起名（alias）：

SELECT RTRIM (vend\_name) + ' (' + RTRIM (vend\_country) + ') ' AS vend\_title

FROM Vendors

ORDER BY vend\_name;

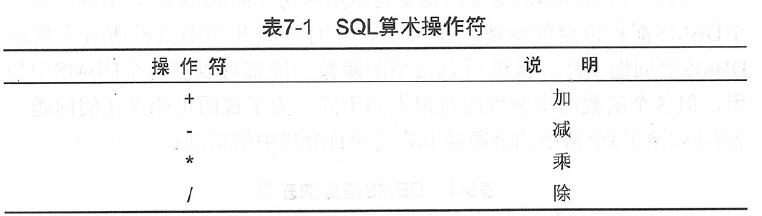
算术计算：添加新字段，值为两个字段乘积，并起名

SELECT prod\_id, quantity, item\_price,

quantity \* item\_price AS expanded\_price

FROM OrderItems

WHERE order\_num = 20008



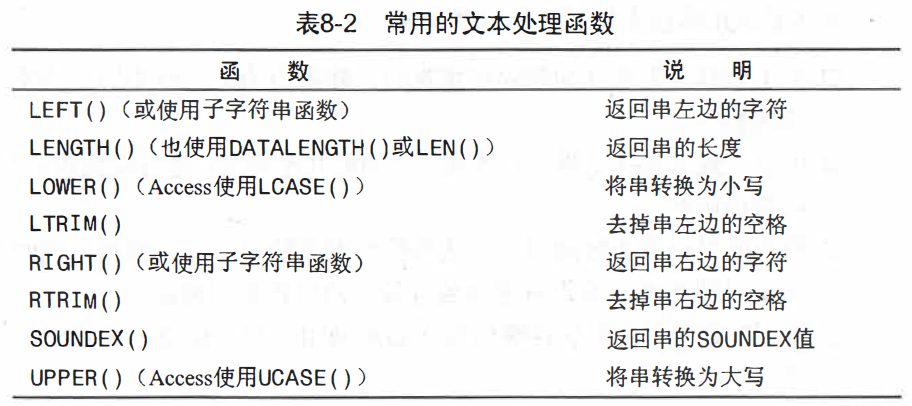
**6、使用函数**

文本处理函数：UPPER函数将字符转化为大写

SELECT vend\_name, UPPER (vend\_name) AS vend\_name\_upcase

FROM Vendors

ORDER BY vend\_name;



日期和时间处理函数：year返回日期一部分，将order\_date列中返回年份

SELECT order\_num

FROM Orders

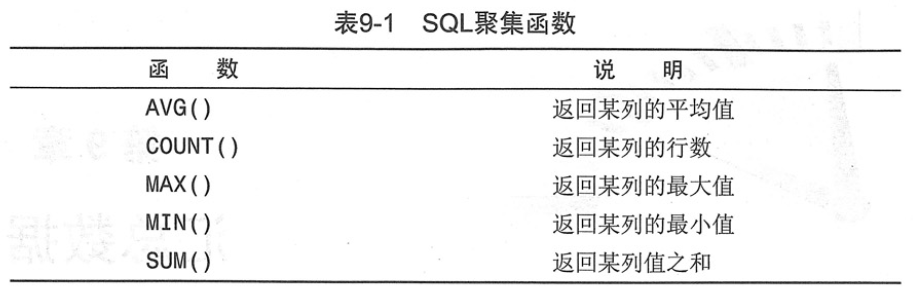
WHERE YEAR (yy, order\_date) = 2004;

数值处理函数：



**7、汇总数据**

聚集函数（默认聚集all）：



**聚集不同值**：指定distinct参数，重复的价格只统计一个

SELECT AVG (DISTINCT prod\_price) AS avg\_price

FROM Products

WHERE vend\_id = ‘DLL01’

**组合聚集函数**：select可包含多个聚集函数

SELECT COUNT(\*) AS num\_items

MIN (prod\_price) AS price\_min,

MAX (prod\_price) AS price\_max,

AVG (prod\_price) AS price\_avg

**8、分组数据**

**数据分组**：group by将数据分为多个逻辑组，以便对每个组进行聚集计算，下面对每个vend\_id进行count

select vend\_id, count (\*) as num\_prods

from products

group by vend\_id;

**过滤分组**：having对每个分组进行条件限制，该分组行数大于等于2（where用于行级过滤，having用于分组过滤，仅与group by结合用）

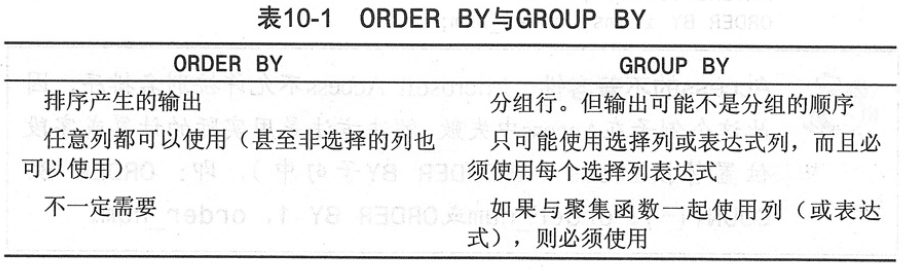
select cust\_id, count (\*) as orders

from orders

group by cust\_id

having count (\*) >= 2;

**分组和排序**：



（一般在给出group by子句，应该也有order by子句，保证数据正确排序）

**子句顺序**：

