**show columns from table\_name;**

**describe table\_name;**

**select Name, countryID, countryGDP from country;**

**distinct**

**limit、limit 5，5**

**order by**

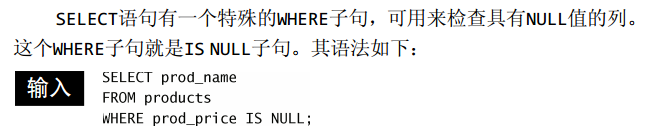
**使用全限定**

**select Code, Name, GNP from country limit 20;**

**select Code, Name, GNP from country order by GNP DESC limit 20;**

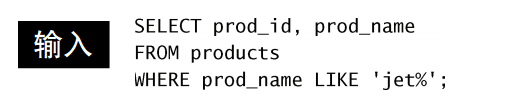
**数据根据WHERE子句中指定的搜索条件进行过滤**

使用BETWEEN时，必须指定两个值,所需范围的低端值和高端值。这两个值必须用AND关键字分隔,BETWEEN匹配范围中所有的值，包括指定的开始值和结束值

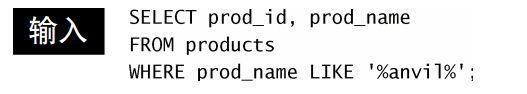


MySQL允许给出多个WHERE子句。这些子句可以两种方式使用：以AND子句的方式或OR子句的方式使用

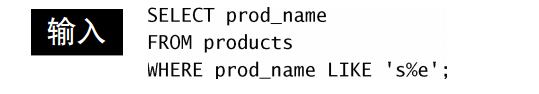
%代表搜索模式中给定位置的0个、 1个或多个字符



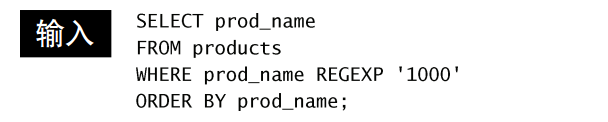
此例子使用了搜索模式'jet%'。在执行这条子句时，将检索任意以jet起头的词。 %告诉MySQL接受jet之后的任意字符，不管它有多少字符。注意where之后的条件加了单引号



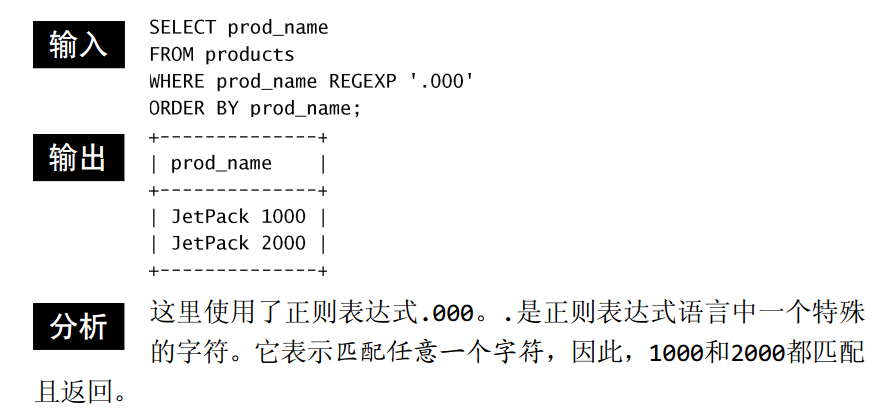
如果你想找出名称包含anvil的所有产品，可构造一个通配符搜索模式



例子找出以s起头以e结尾的所有



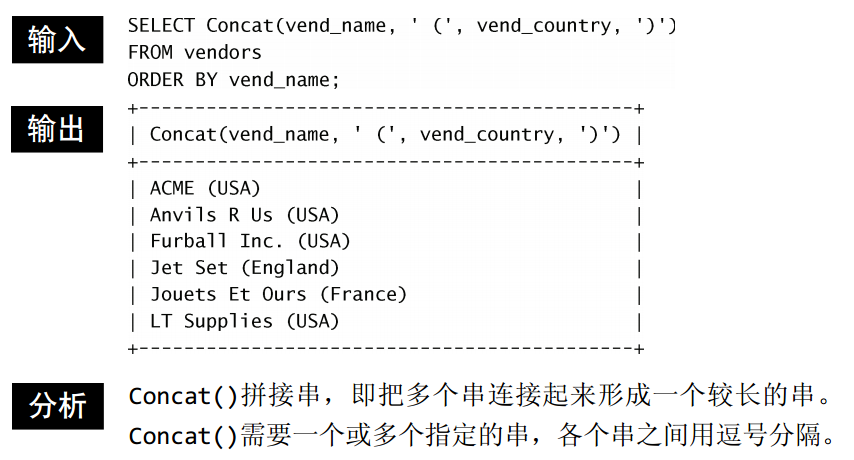
语句检索列prod\_name包含文本1000的所有行



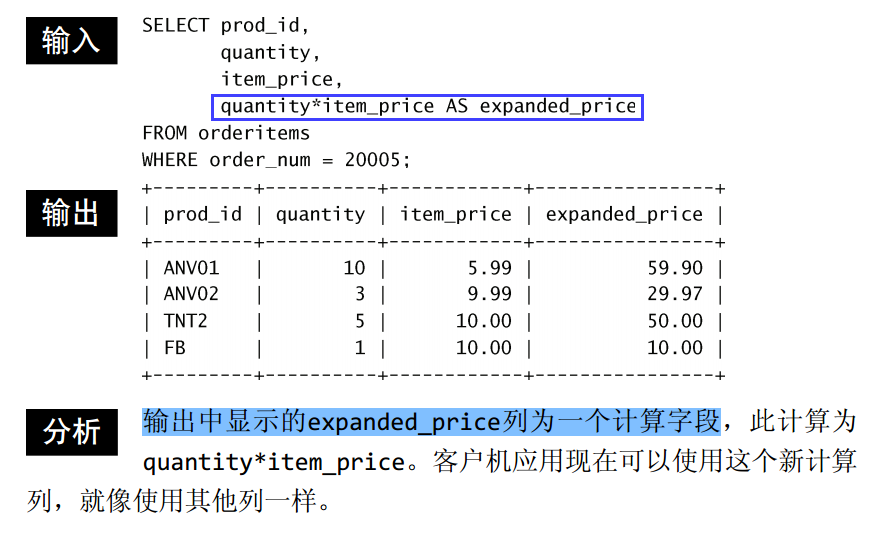
LIKE匹配整个列。如果被匹配的文本在列值中出现， LIKE将不会找到它，相应的行也不被返回（除非使用通配符）。而REGEXP在列值内进行匹配，如果被匹配的文本在列值中出现， REGEXP将会找到它，相应的行将被返回。这是一个非常重要的差别

\\\匹配\

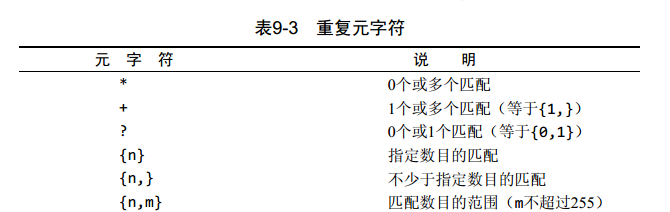
[**\\.匹配**](file:///\\.匹配)**.**



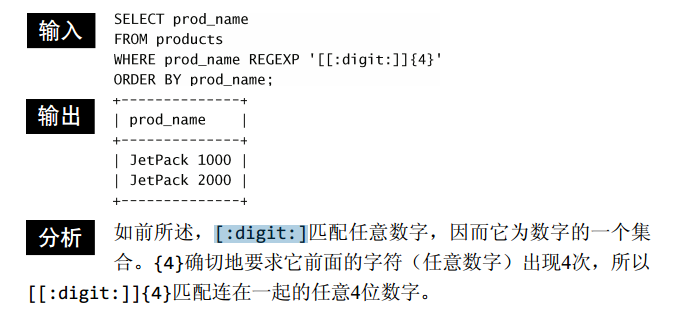
MySQL除了支持RTrim()（正如刚才所见，它去掉串右边的空格），还支持LTrim()（去掉串左边的空格）以及Trim()（去掉串左右两边的空格）



如何测试计算 SELECT提供了测试和试验函数与计算的一个很好的办法。虽然SELECT通常用来从表中检索数据，但可以省略FROM子句以便简单地访问和处理表达式。例如， SELECT3\*2;将返回6， SELECT Trim('abc');将返回abc，而SELECT Now()利用Now()函数返回当前日期和时间。通过这些例子，可以明白如何根据需要使用SELECT进行试验



sticks?匹配stick和sticks（s后的?使s可选，因为?匹配它前面的任何字符的0次或1次出现）注意？操作的是前面的s



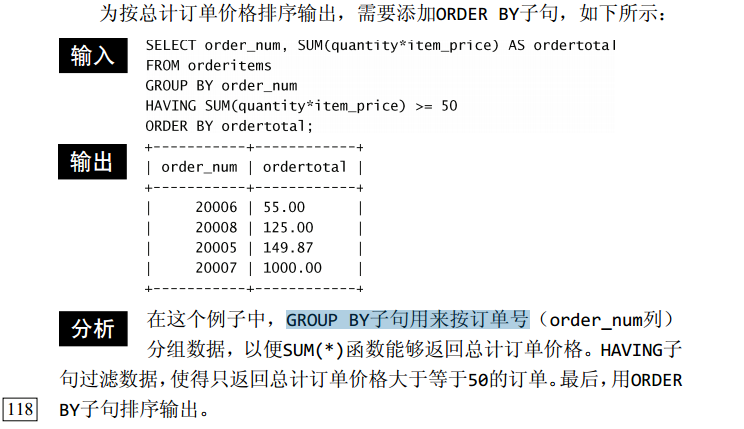
还可以是

SELECT prod\_name FROM products

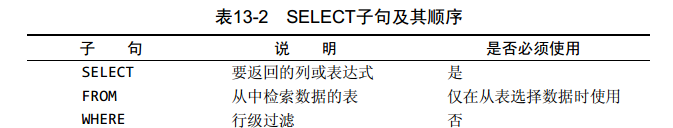
WHERE prod\_name REGEXP ‘[0-9]{4}’ ORDER BY prod\_name;

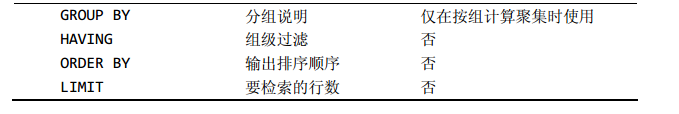
^的双重用途 ^有两种用法。在集合中（用[和]定义），用它来否定该集合，否则，用来指串的开始处, ^[] 串的开始的位置 [^ …] 否定([]内的)该集合

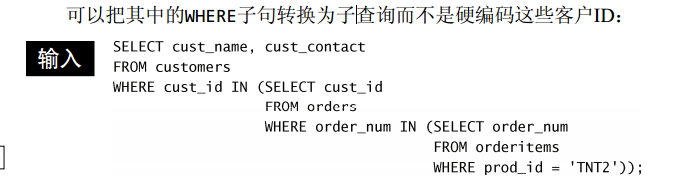
SOUNDEX可以理解为某种模糊匹配算法，考虑了类似发音字符和音节，使得能对串进行发音比较，而不是精确地字符进行比较



Group by 按照某个条件进行分组，然后基于分组进行聚集

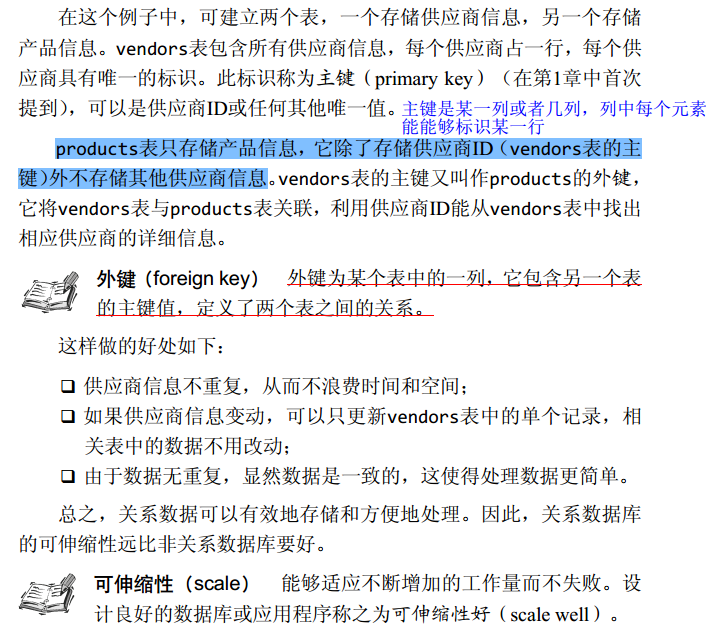




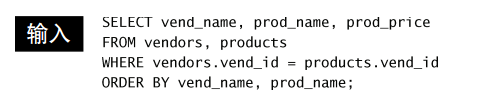


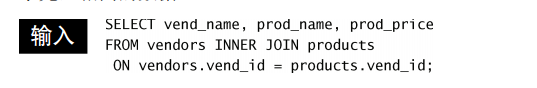
在产品名称中查找TNT2的产品，返回产品编号，在返回的产品编号中查找顾客ID，然后在返回的顾客ID内查找顾客的相关信息。

为了执行上述SELECT语句， MySQL实际上必须执行3条SELECT语句。最里边的子查询返回订单号列表，此列表用于其外面的子查询的WHERE子句。外面的子查询返回客户ID列表，此客户ID列表用于最外层查询的WHERE子句。最外层查询确实返回所需的数据



创建联结





WHERE子句作为过滤条件，它只包含那些匹配给定条件（这里是联结条件）的行

由没有联结条件的表关系返回的结果为笛卡儿积。检索出的行的数目将是第一个表中的行数乘以第二个表中的行数，第一个表中的每个行将与第二个表中的每个行配对，而不管它们逻辑上是否可以配在一起

应该保证所有联结都有WHERE子句，否则MySQL将返回比想要的数据多得多的数据

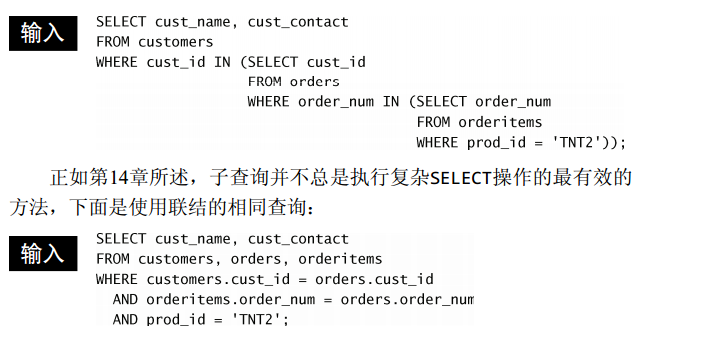
关系表的设计就是要保证把信息分解成多个表，一类数据一个表。各表通过某些常用的值（即关系设计中的关系relational 互相关联

主键：表中的一列或者若干列，列中的每个元素能唯一确定表中的一行

外键(foreign key):外键为某个表中的一列，它包含另一个表的主键值，定义了两个表之间的关系，一个表的外键是另一个表的主键

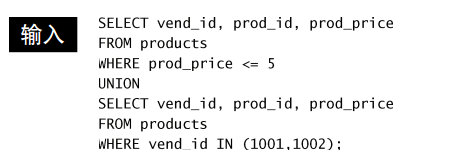
创建联结的基本规则：首先列出所有要进行联结的表，然后定义表之间的关系

注意使用全限定名，避免字段的二义性，使用WHERE字句，避免MySQL将返回比想要的数据多得多的数据。



这个查询中返回数据需要使用3个表。但这里我们没有在嵌套子查询中使用它们，而是使用了两个联结。这里有3个WHERE子句条件。前两个关联联结中的表，后一个过滤产品TNT2的数据

创建组合查询Union



UNION指示MySQL执行两条SELECT语句，并把输出组合成单个查询结果集