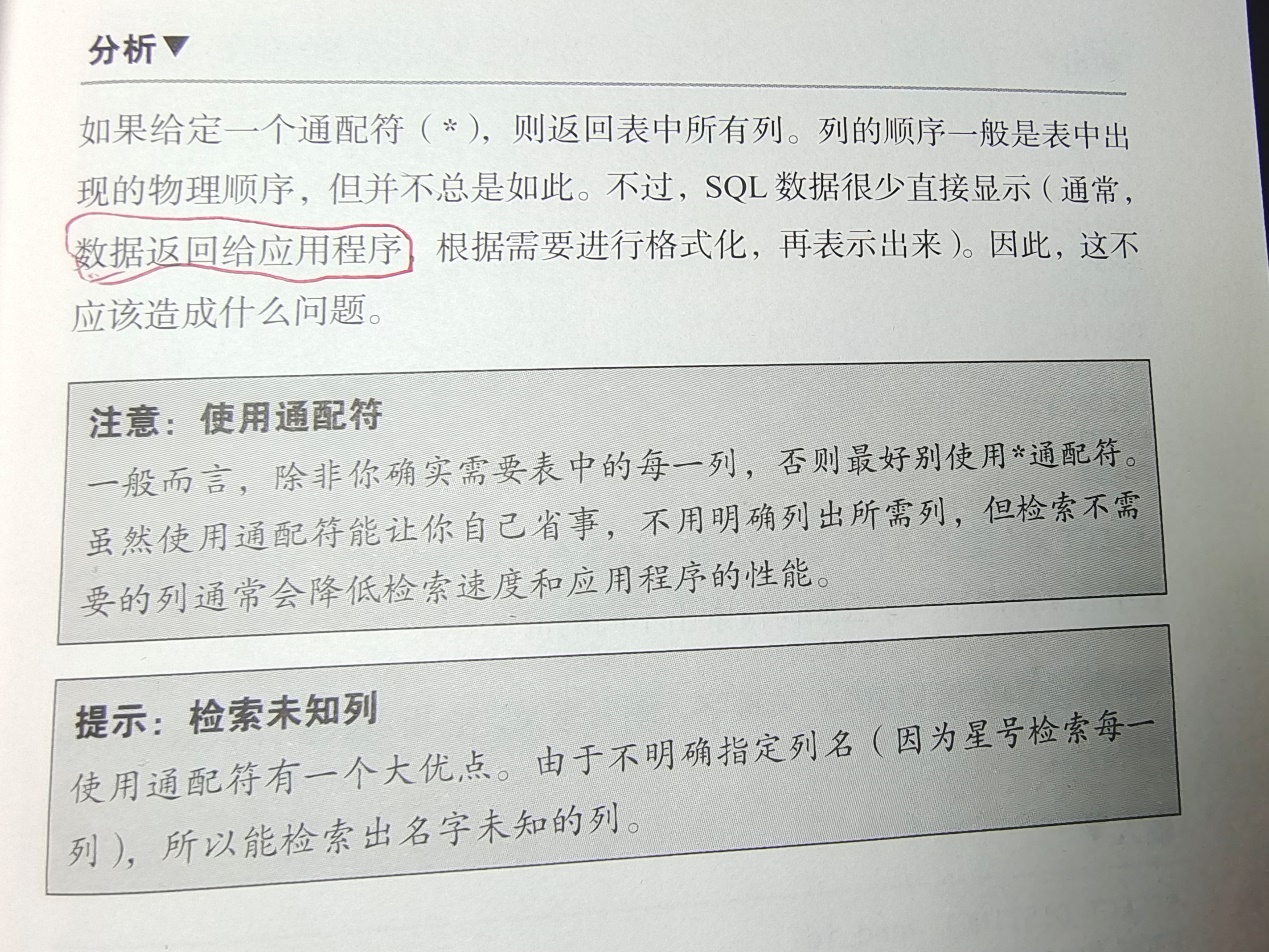
# 第二章 检索数据

1. SQL关键字用大写，而对列名和表名用小写。
2. 将SQL分成多行更容易调试。
3. 选择多个列时，要在列名之间加上逗号
4. 不同数据库显示的数据的对齐格式，小数位数不同。
5. 
6. Select 语句返回所有匹配的行 ，但我不希望每个值重复出现。这时就要用到DSTINCT关键字。DISTINCT关键字要放在列名前面。 DISTINCT作用于你写出来的所有的列。不能只作用部分列。
7. 可以用LIMIT关键字来限制显示多少行，

利用LIMIT 数字 OFFSET 数字 可以指定从哪里开始检索多少行。

第一个数字是检索的行数 第二个数字是指从哪开始。（注：第一个检索的行是第0行）

1. 代码注释 （1）-- --后的文本会被注释掉，但它前面不会

（2）# 使用#后，整行都会被注释。

（3）/\*\*/ 已经 耳熟能详了

# 第三章

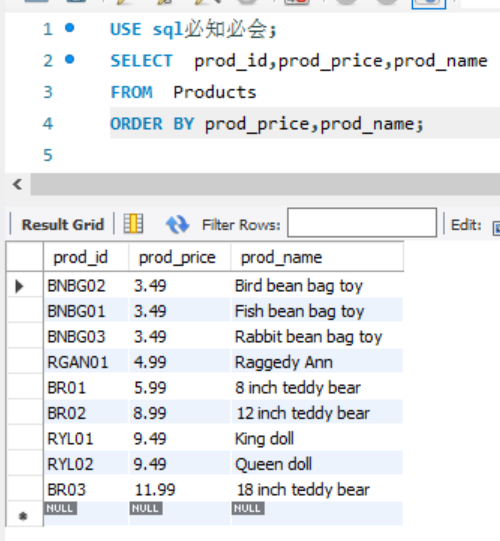
1.使用ORDER BY 进行排序 。注意ORDER BY 子句应是SELECT语句的最后一个子句。

2.按多个列排列时，先按第一个列名进行排列，如果排列中具有相同的此列的值，则按第二个列名对 刚才出现的相同的值 进行排列。

（1）如图，未排序时：



（2）如图，排序后



分析：先按prod\_price 进行排序，如果prod\_price中出现相同的值，例如BNBG02， BNBG01，BNBG03三者的prod\_price相同，则按prod\_name对这三个值进行排序。

3.ORDER BY 除了用列名来指出排列顺序外，还可以用 数字 来指出。但有一个很大的区别，我们知道，

（1）select后跟的列名会被搜寻出来制成一个表，而使用数字时，数字代表的是select后跟列名的位置，而不是完整表中的位置。

（2）而通过列名指定排列顺序，可以用完整表中的所有列名来排列。

如图：因为select语句后只有两个列名，所以数字3会报错。



4.排序方向有两种，升序（asc）与降序(desc),默认为升序。!!!注意当使用降序时，需要对每一个列名后加上desc关键字

# 第四章 过滤数据

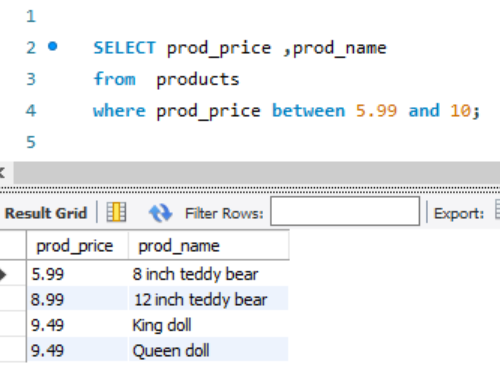
1.一般我们不需要搜寻所有的行，因此需要通过搜索条件（也叫过滤条件）来选择所需要，需要用到where子句。

2.where子句的位置应该放在order by子句的前面。

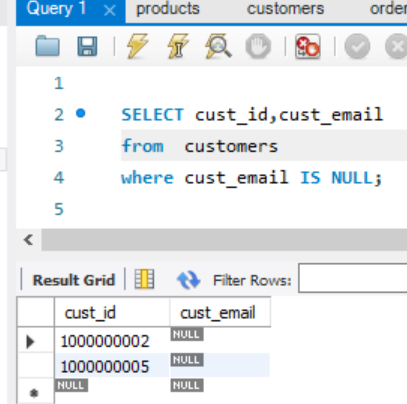
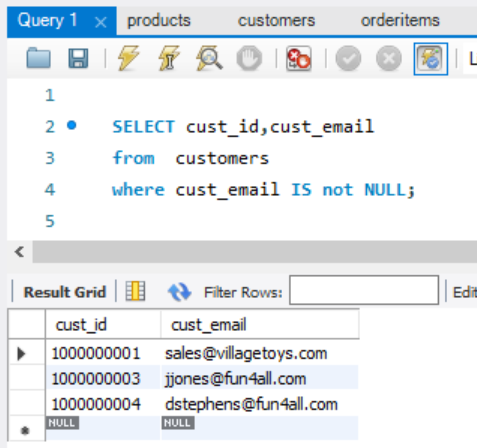
3.where子句有一系列的操作符，像>,=,<,!= 等等参阅不同数据库管理系统的规定。

4.不匹配检查 用！=或<>

5.范围值检查 用between 数字 and 数字



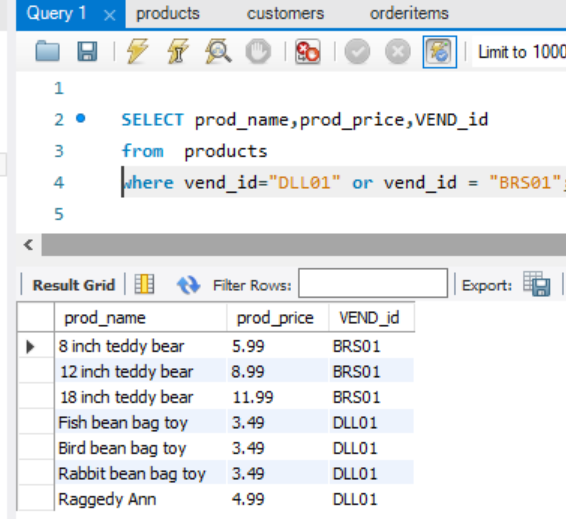
6.空值检查 千万不要用=来判断空不空，要用IS NULL。Is not null 可以用来判断非空。

疑问：between and 与使用> and <,效果一样，但未何还要用。

# 第五章 高级数据过滤

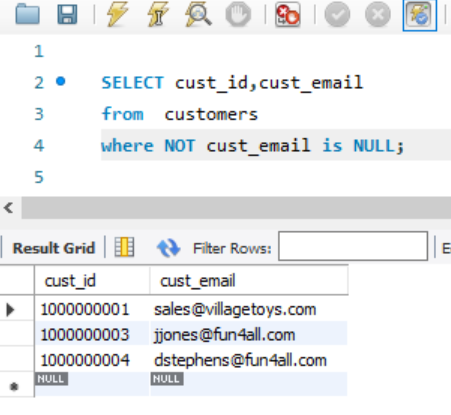
1.在上一章中只是用单一的where子句，子句里面只有一个条件，现在我们可以用and 和 or来联结多个子句，即可以含有多个条件。and是必须满足所有条件，or是满足任一条件。



2.由于and操作符的优先级比or高，因此总是会先优先处理and，在某些情况下以人的思维去理解很容易出错，因此加上括号，括号的优先级比and和or高，总是加上括号，这并没有什么坏处。

3.IN 操作符用来指定条件范围，范围中的每个条件都可以匹配，IN取一组由逗号隔开，括在括号里合法值。有时候好像，它跟用or实现的查询，好像效果相同。但是IN有它自己的优点：（1）条理清晰（2）里面可以用select语句（会在第十一章详细介绍）

4.NOT操作符：否定其后所跟的任何条件。可以跟IN连用，产生妙用，可以轻易找出与条件不匹配的行。



5.用and操作符可以达到跟between and 一样的效果

# 第六章 用通配符进行过滤

1.概念介绍：

通配符（wildcard）：用来匹配值的一部分的特殊字符

搜索模式（search pattern）:由字面值、通配符或两者组合构成的搜索条件。

谓词：从技术上讲，LIKE是谓词，而不是操作符。

2．可以用LIKE谓词来操纵通配符。

3.根据DBMS的不同及其配置，搜索可以区分大小写。

4.百分号（%）通配符，%代表任意字符出现任意次数。

（1）“刘%” 表示 以 刘 开头，不管“刘”后面有多少个字。

（2）“刘%备”表示以“刘”开头 ，以“备”结尾的名字，不管中间有多少字。

假如有一个字符“刘准备 ” ，你觉得还能成功匹配吗？

显然不能，因为这个字符串是以空格结尾的，并不是以“备”结尾。

解决方案：“刘%备%” 这样就可以省略后面的空格。（在第八章会用函数来去掉空格）

注意：%还能匹配0个字符。

5.通配符%看起来像是可以匹配任何东西，但有个例外，那就是NULL，子句where prod\_name LIKE “%” 不会匹配产品名称为NULL的行。

6.下划线（\_）通配符。\_ 只能匹配一个字符。

（1）“刘\_” ，代表 长度为2，以“刘”开头的字符。

7.方括号（[]）通配符，用来指定一个字符集，它必须匹配通配符[]的位置的一个字符。

“[JM]%” 是以J或M字符开头的任意字符串。“[^JM]%”与前面加NOT的效果相同，但更方便。

MySQL.Oracle,DB2,SQLite不支持方括号通配符。

8.使用通配符的技巧：使用通配符搜索很耗费时间

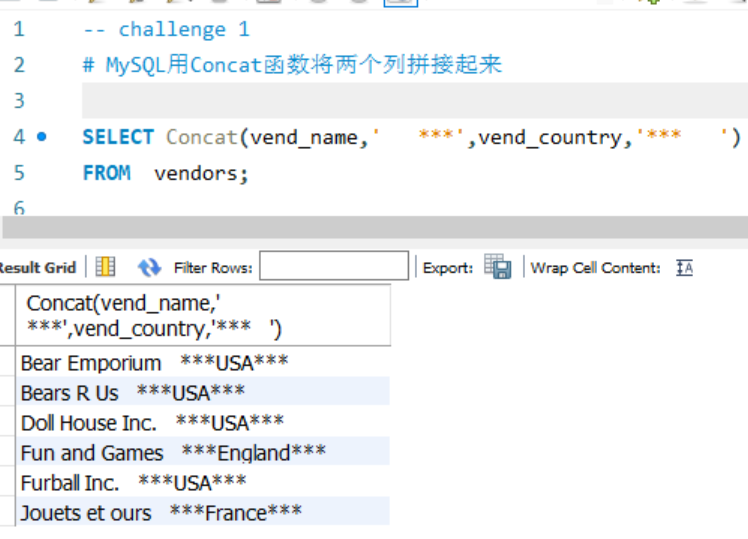
（1）不要过度使用通配符。

（2）尽量不要把它放在搜索模式的开始处

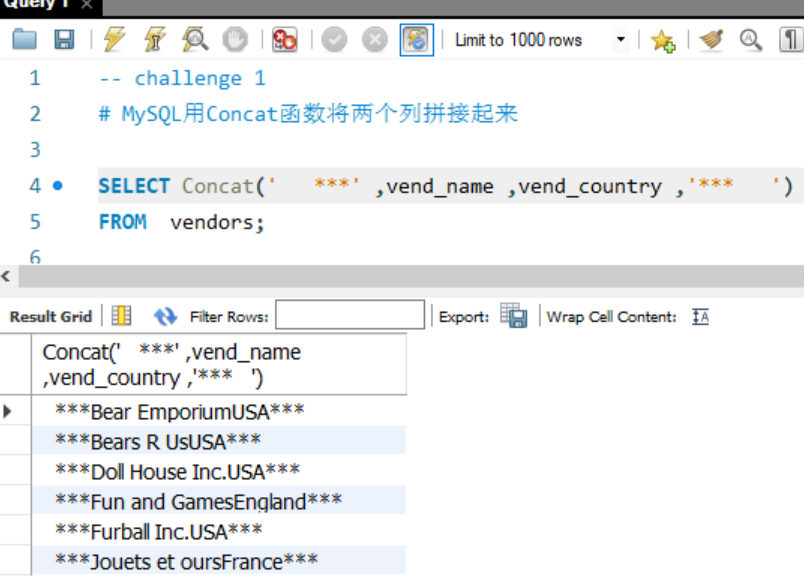
（3）仔细观察通配符所在的位置

# 第七章 创建计算机字段

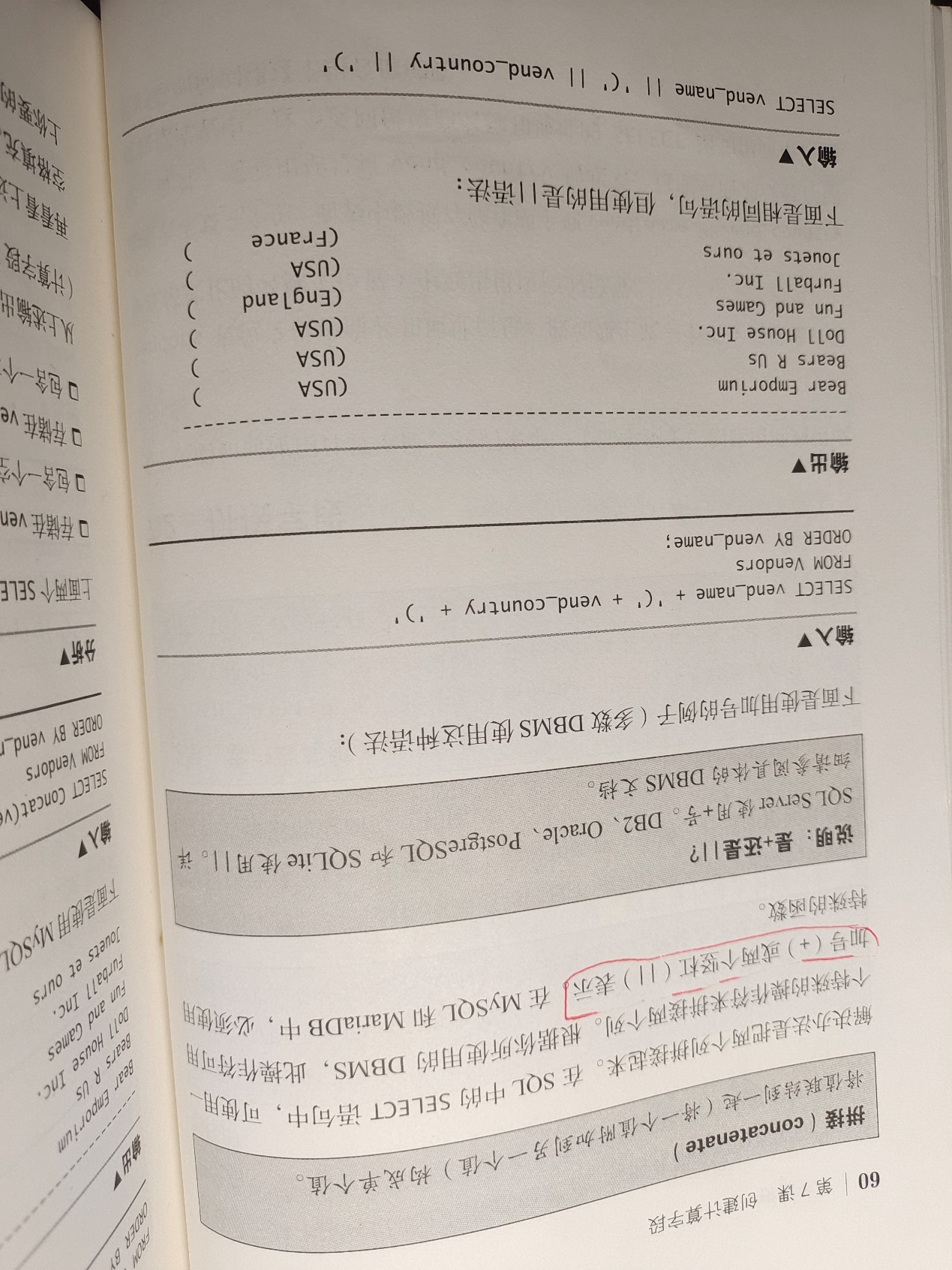
1.拼接字段。MySQL利用Concat函数来拼接两个列。注意用逗号隔开列名和字符串。

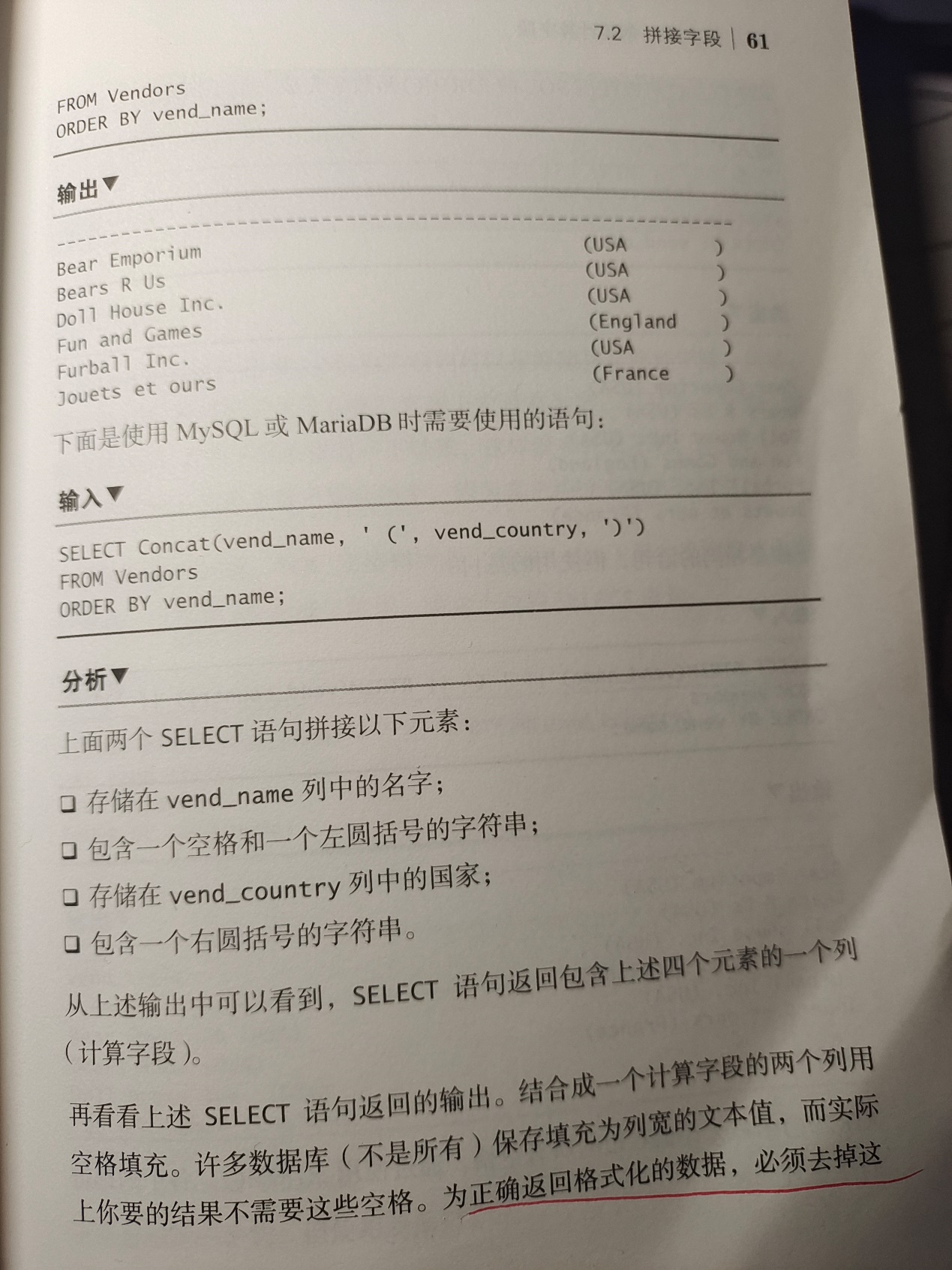


格式为：Concat(字符串，列名，列名)，字符串可以根据你的需要，放在任意位置，如下图我可以在前面Concat括号 开头放个字符串。



2.对于其他的一些数据库的拼接字段。





3.TRIM函数，去掉字符串左右两边的空格

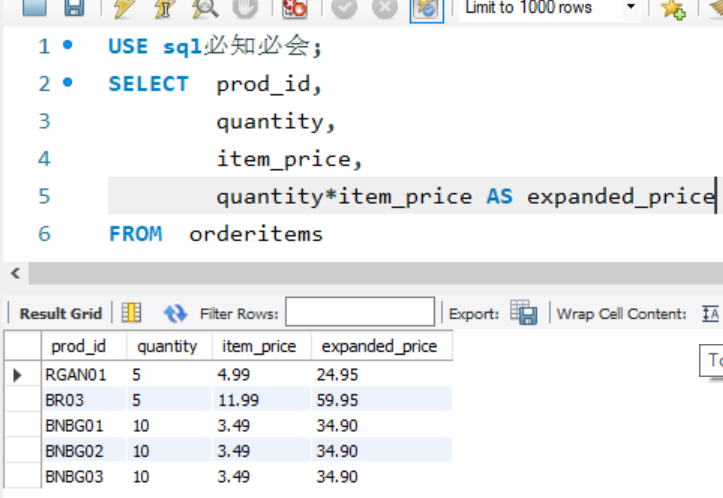
LTRIM（vend\_name）当列vend\_name中值的左边出现了空格，则把空格去掉。RTRIM也可以据此推理。

4.使用别名 。可用AS给列名 起一个别名。当我们用拼接字段时，它们的列名也拼接在一起，但用 AS 起一个别名，可以更简洁的表示。



5.计算字段，字段在数据库中也叫做列。

可以对检测出的数据进行算数计算（即+-\*/）。



# 第八章 使用函数处理数据

1.文本函数：处理文本字符串

2.数值函数：在数值数据上进行算术运算。

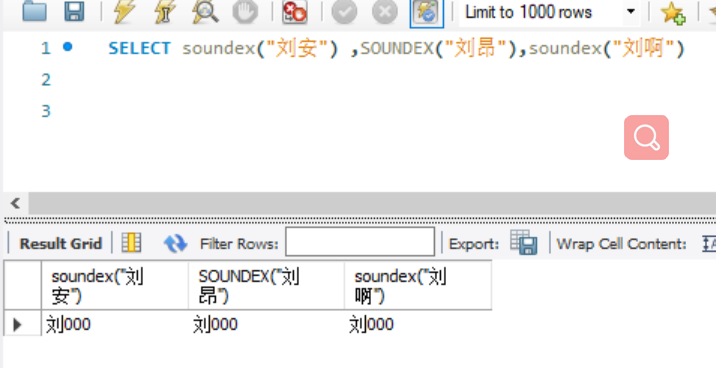
3.日期和时间函数：

4.格式化函数 （生成好看的输出内容）

5.系统函数：返回DBMS正使用的特殊信息（如返回用户登录信息）

例：（1）文本处理函数

Soundex() Soundex()是一个将任何文本串转换为描述其语音表示的字母数字模式的算法。Soundex()考虑了类似的发音字符和音节，使得对字符串进行发音比较而不是字母比较。虽然，soundex()不是SQL概念，但是多数DBMS都提供对soundex()的支持。



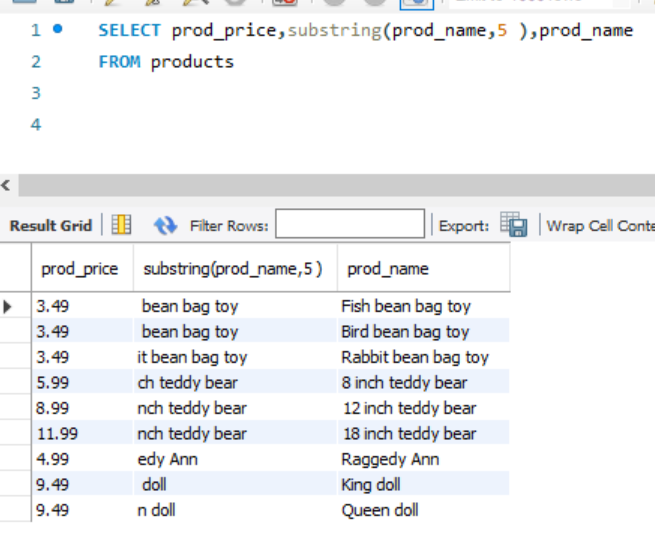
Left(字符串，字符串长度) ，根据长度提取字符串左边的字符。

Right(字符串，字符串长度),同上，不过是右边

Lower()Upper() 将字符串转换成大小写

TRIM() 去掉字符串中的空格

Sunstring(字符串，从字符串的什么位置开始)

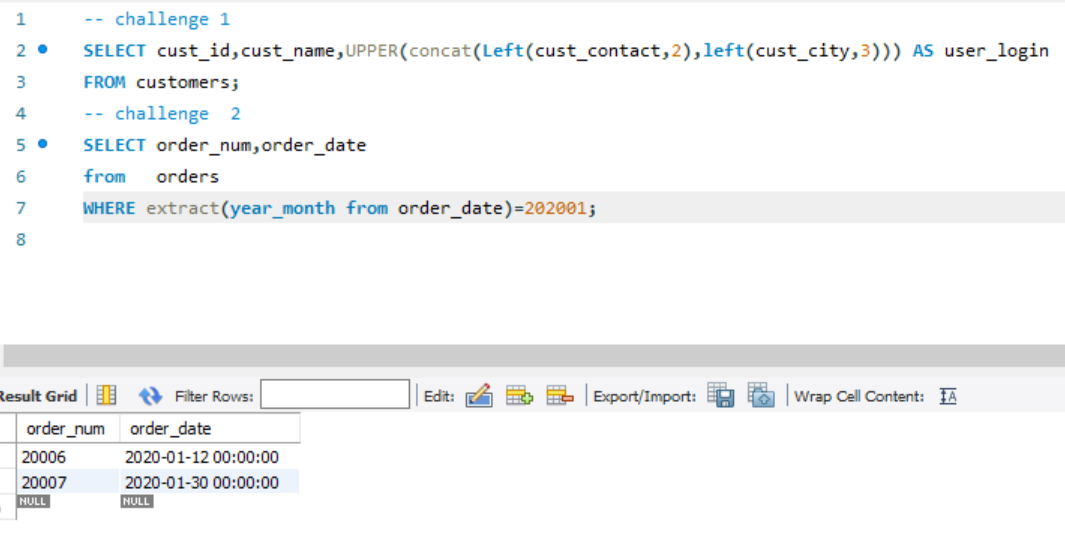


（2）日期处理函数,数值处理函数，用到再补充吧。

MySQL文档自己去查自己去看

<a href=<https://dev.mysql.com/doc/>>

下面的图是我搜寻出来的一个extract（）函数



# 第九章 汇总数据

## 9.1聚集函数

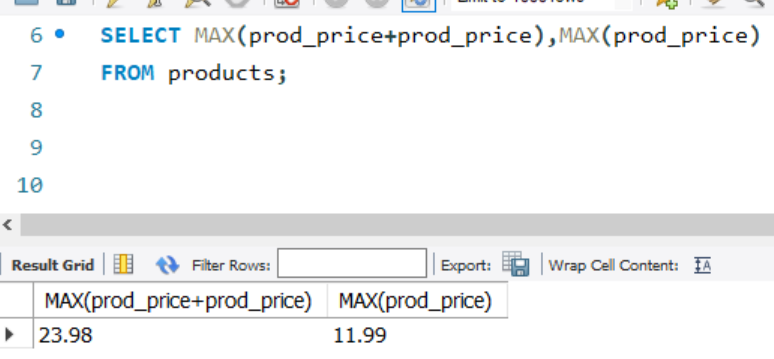
1.聚集函数：我们经常需要汇总数据而不需要实际把他们检索出来，像确定表中行数，获得表中某些行的和等等，都需要汇总出表中数据，而不需要查出数据本身，因为返回实际表很浪费时间和资源。

2.SQL聚集函数： AVG() COUNT() MAX() MIN() SUM()

3.AVG()函数只用于单个列，AVG(列名，列名)这种形式是错误的。

但是利用标准的算术运算符，所有聚集函数都可用来执行多个列的计算。

如下图：



暂停记笔记，开始写作业。

2022.11.8,continue

4.avg()函数忽略列值为NULL的行

5.count()函数进行计数。确定表中行的数目或符合特定条件的行的个数。

两种使用方式：

1. Count(\*),对表中行的数目进行计数，不管是NULL，还是非空值。
2. Count（column），对特定列中具有值的行进行计数，忽略NULL。

6.MAS（）函数，返回指定列中的最大值。

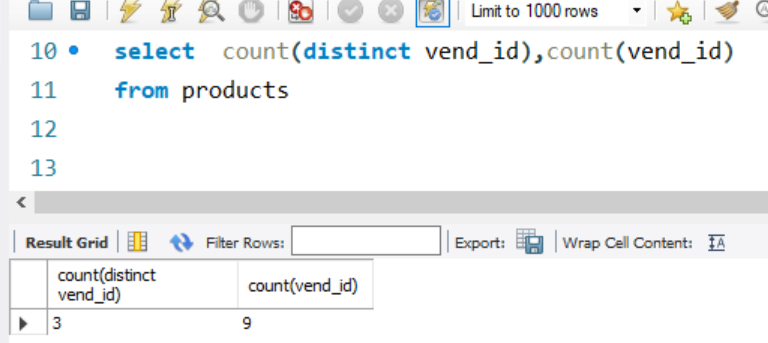
7.MIN（）函数，返回指定列的最小值

8.sum() 函数，返回指定列值的和

## 9.2 聚集不同值

1.只包含不同的值，用DISTINCT参数。

对所有行执行计算，用all参数，或者不用。因为默认为all



2.distinct 不能用于count(\*)

3.distinct 用于max（）和min（）,没有什么意义

# 第十章 分组数据

1.使用分组可以将数据分为多个逻辑组，分别对每个组进行聚集计算。

2.关于group子句的一些规定，我不是很清楚。

3.Group by 列名 根据这个列来来分组。

3.过滤分组：除了能用group by 进行分组数据外，SQL还允许过滤分组，规定包括哪些分组，排除哪些分组。因此要用having子句

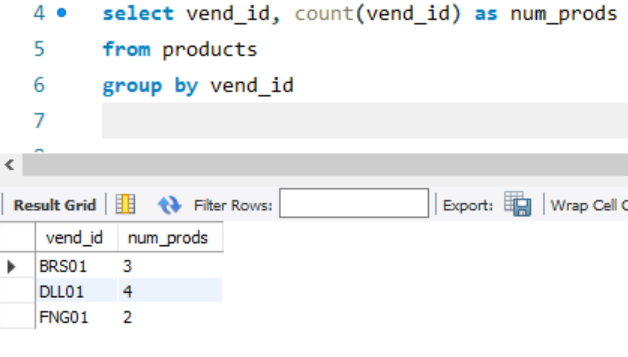
4.having非常类似于where。但where用于过滤行，having用于过滤分组。

5.where子句的条件，像带多个操作符的语句，通配符条件，都可用having语句。

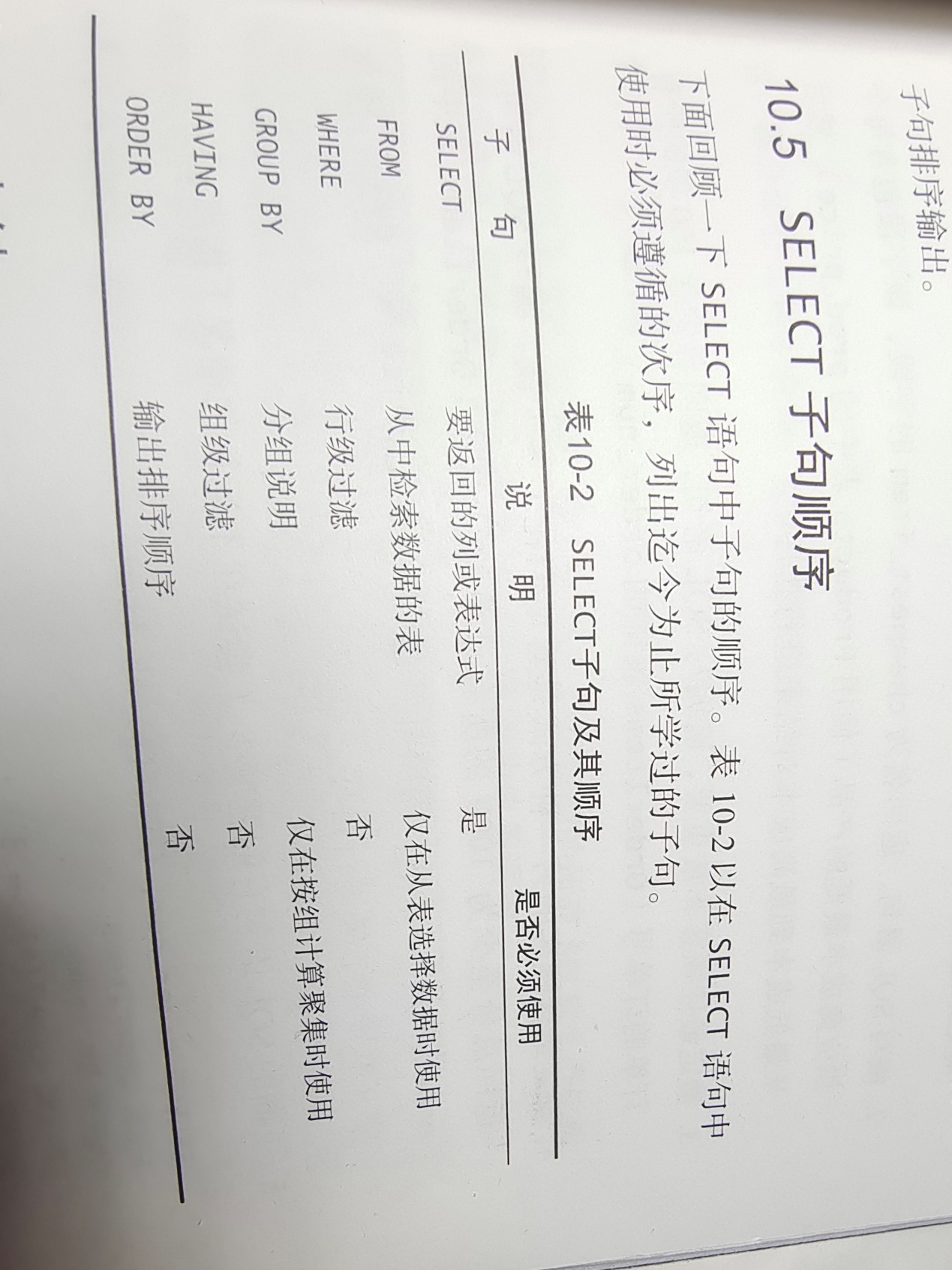
6.where子句在数据分组前进行过滤，having在数据分组后进行，可以用这一特点，先用where过滤一些不想要的行，然后再用group by与 having的结合过滤分组。

7.分析一个例子：

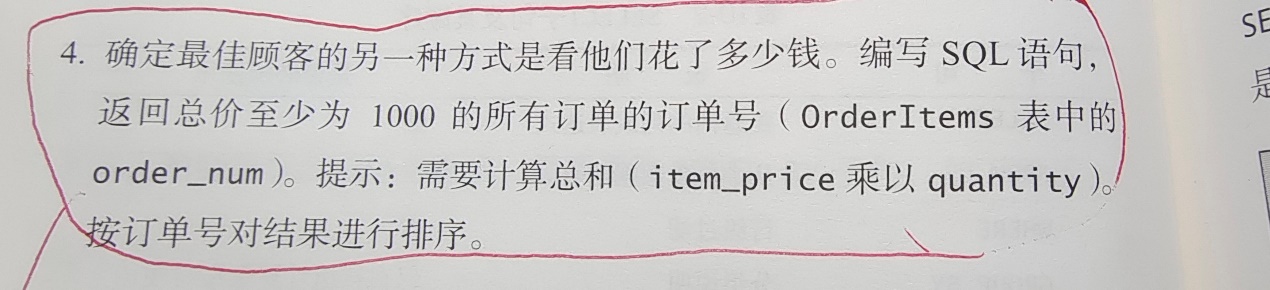
如下图，分组后，便在select语句中，便会对根据vend\_id分组的每个组中的行数进行计算。

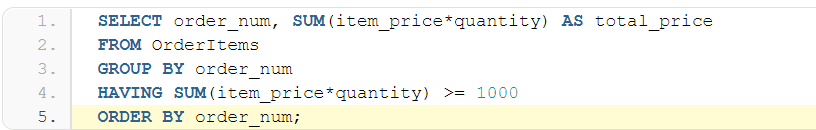


8.



10.这章有个课后习题，有点意思 有点顿悟。





# 十一章 使用子查询

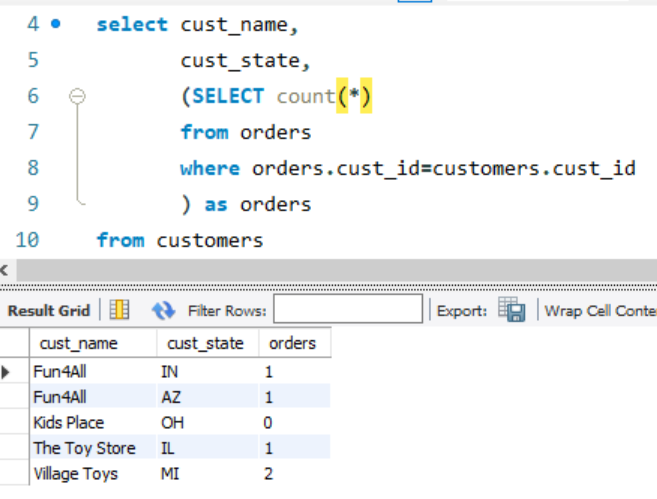
11.1 用子查询进行过滤

1.子查询

（1）可以作为计算字段使用子查询，有限制。

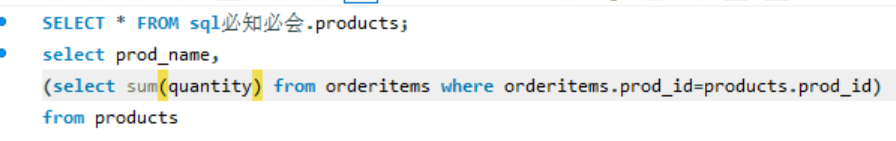
用一个 . 来分隔表名和列名，这被称作完全限定列名。

分析下表：子查询中的where子句告诉SQL，比较orders表中cust\_id与当前正在从customers表中检索的cust\_id 进行比较。



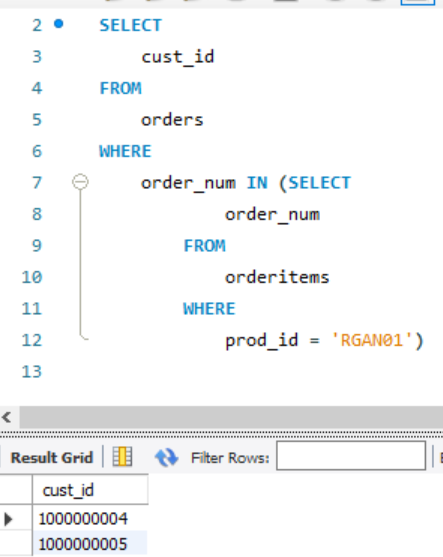
做课后习题时，发现的限制，如下图：去掉子查询的sum后就会报错，让我们来分析查询过程，（1）先查询prod\_name的第一个名字，其对应的prod\_id=‘BNBG01’（2）然后进入子查询，此时子查询where子句中的products.prod\_id是第一个名字对应的‘BNBG01’（3）然后子查询把orderitems的prod\_id遍历一遍，

求出ordertims.prod\_id=‘BNBG01’这个条件下的orderitem表的quantity的和。

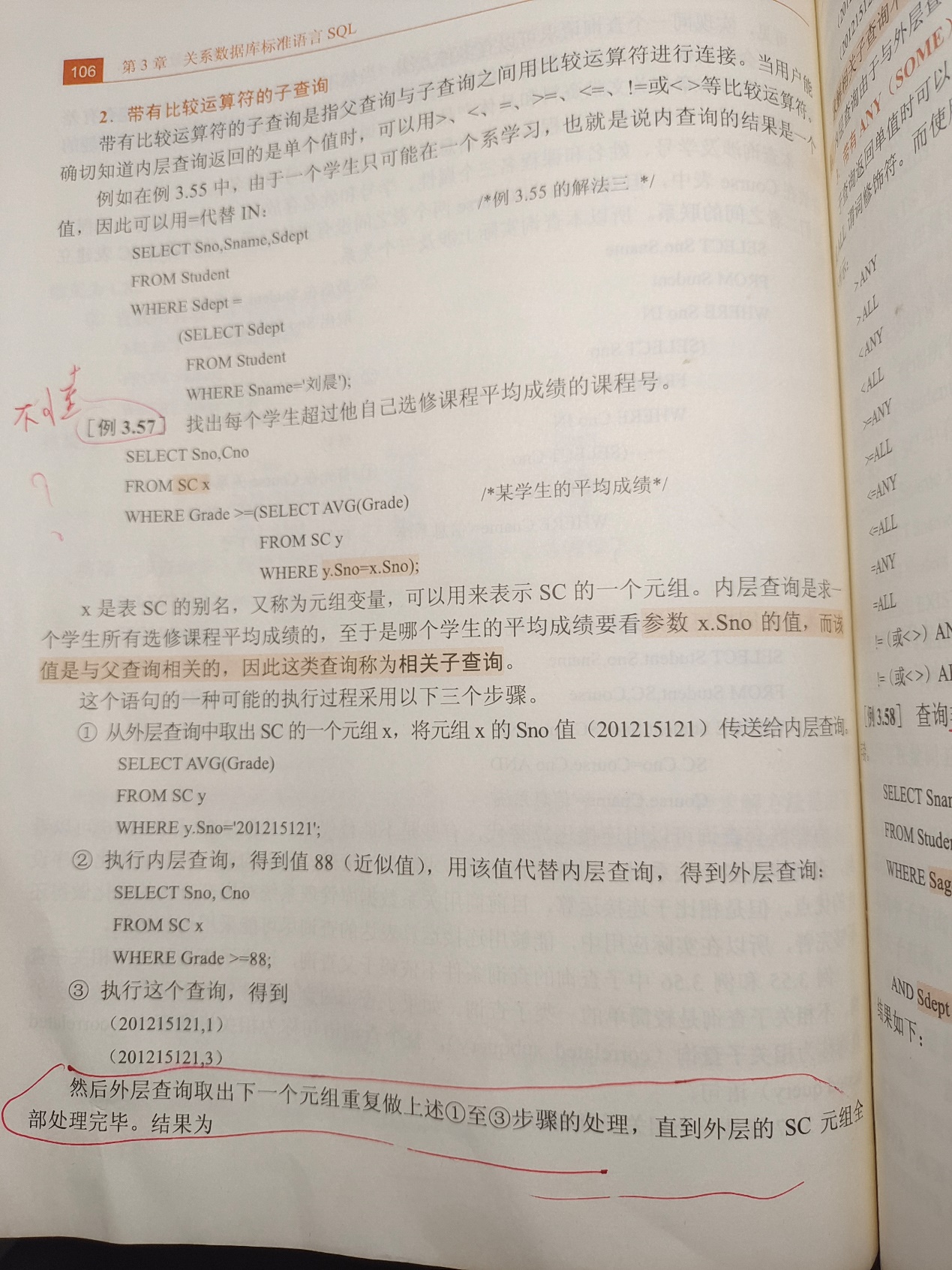


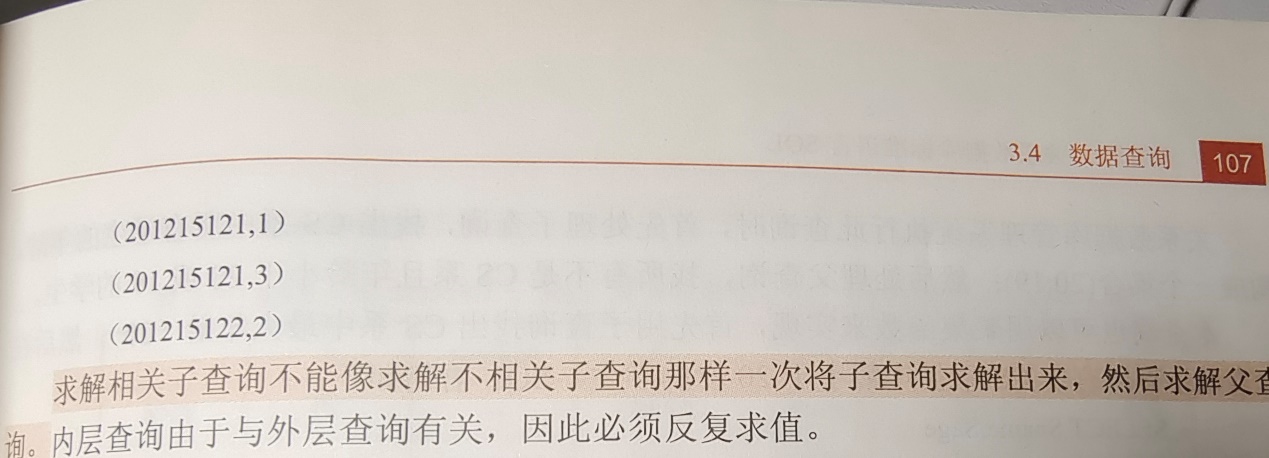
（2）利用子查询进行过滤，即where子句中含有子查询，常与IN进行联用。

Pay attention to ：作为子查询的select语句只能查询单个列。企图检索多个列将返回错误。



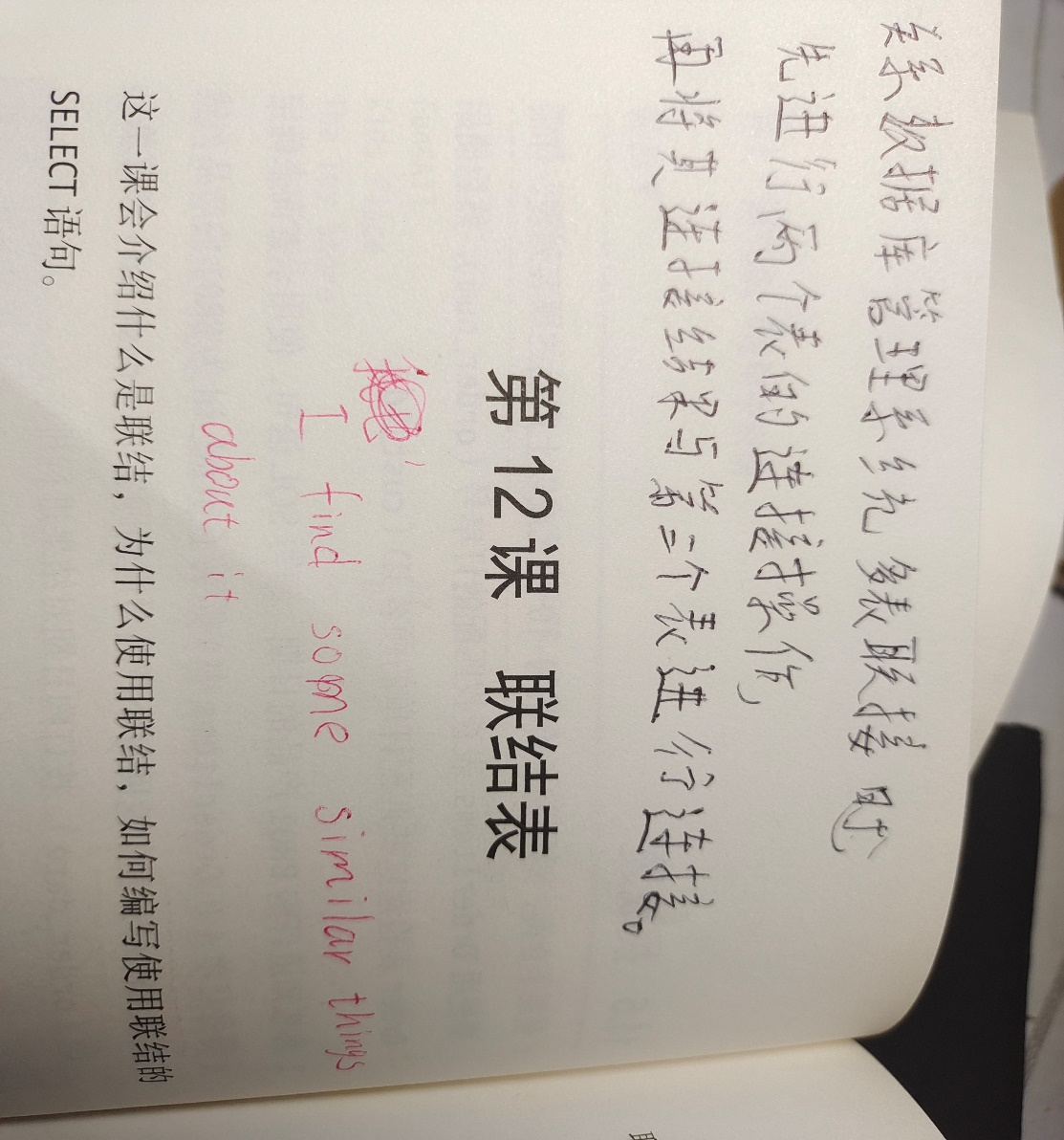
在做相关习题时，遇到相关子查询和非相关子查询，这部分知识也得了解一下。





# 第十二章 联结

补充：



1.对于联结，首先要知道笛卡尔积的概念，我推荐中国大学mooc上北京师范大学数据库原理 这门课程，大家可以去看看。

<a href=” [中国大学MOOC(慕课)\_国家精品课程在线学习平台 (icourse163.org)](https://www.icourse163.org/)”>中国大学慕课</a>

2.对于等值联结，也叫做内联结，有两种语法。

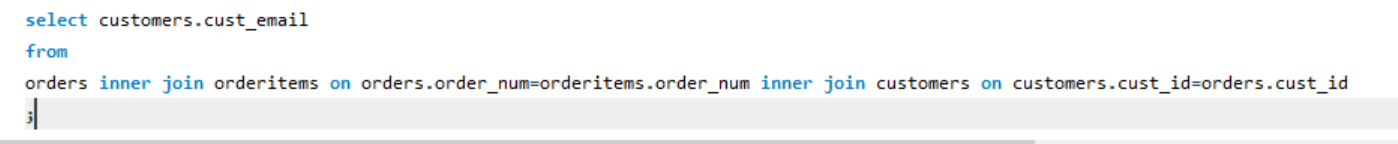
（1）简单语法:from子句后跟要联结的表名，where子句给出条件

（2）标准语法：在from子句中写，表名 inner join 表名 on 条件。 不需要where子句了



3.多表联结。我认为《SQL必知必会》并没有讲的很清楚。可参考本章开始的补充内容。

从标准语法也更容易理解：



这个图中的代码里有两个 inner join ，意思是：orders表与orderitems表联结的结果与customers表进行联结。

4.联结时，条件中的列名可以不一样。Vendors.vend\_id 与products.vend\_id,两个表的列名均为vend\_id，实际上在取名时，可以不一样。

# 第十三章 创建高级联结

1. 除了可以用as 给列起别名，其实也可以用as给表起别名。

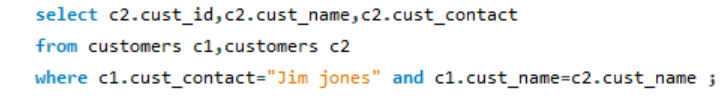
除了方便外，还有一个作用：当select语句中不止一次引用相同的表。

但是，有些数据库管理系统并不支持as,直接在表名后面写上别名就行，记得用空格隔开

1. 除了上一章的等值联结（内联结），还有 自联结（self-join）,

自然联结（natural-join） ,外联结（outer-join）

1. 自联结，同一个表多次进行联结，来最终达到查询的结果。虽然子查询也可以达到同样的效果，但自联结的效率更高。



如上图，便是自联结。但要注意:select语句应该用c2表的列，因为在where子句中 已经令c1.cust\_contact=”Jim jones” ,所以c1表的列cust\_contact的值全为”Jim jones” 。

1. 自然连接，定义：自然连接是关系R和S在所有公共属性（common attribute）上的等接（Equijoin）。但在得到的结果中公共属性只保留一次，其余删除。

数据库管理系统不会把重复的列给删去，这需要你自己在select语句后写列名时，自己写下来，保证自己写下来的列没有重复。

5.外联结：在左外连接和右外连接时都会以一张表为[基表](https://baike.baidu.com/item/%E5%9F%BA%E8%A1%A8?fromModule=lemma_inlink)，该表的内容会全部显示，然后加上两张表匹配的内容。如果基表的数据在另一张表没有记录。那么在相关联的[结果集](https://baike.baidu.com/item/%E7%BB%93%E6%9E%9C%E9%9B%86?fromModule=lemma_inlink)行中列显示为[空值](https://baike.baidu.com/item/%E7%A9%BA%E5%80%BC?fromModule=lemma_inlink)（NULL）

跟内联结的标准语法很像，语法如下：

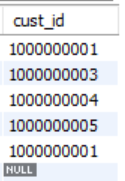
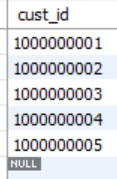
表名 left/out/full outer join 表名。

Mysql不支持full。

Left关键字指出的是outer join左边的表中的所有行，即使没有与右边的表相匹配的属性值，但是仍会显示出来左边表中的全部行的名字，右边显示NULL。

例子：

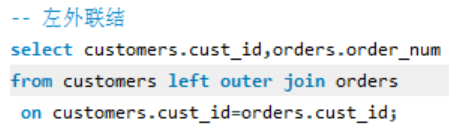
Orders表： customers表：

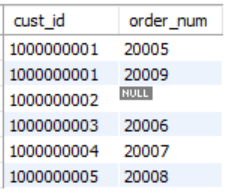
 

很明显，customers.cust\_id并不能与orders.cust\_id中全部匹配。

因为，orders.cust\_id没有100000002，customers.cust\_id有。

下面，我们进行 左外联结。Customers表在left outer join左边，所以要把customers所有的cust\_id列的所有值显示出来。如果orders.cust\_id里面没有对应的值，便让orders表里的列值置为NULL。



很明显，由于orders里表没有与customers表中对应的cust\_id，因此把orders表中的order\_num置为NULL。

6.有些DBMS(数据库管理系统) ，并没有右联结，那怎么办，调换表的位置就可以解决了。

7.进行联结后的表，也可以利用 聚合函数和分组。

*在这里，与各位再共同勉励一次，我们要做聪明的人，做有智慧的人有慈爱又肯诚实对人对己的勇者。就算天大的事情来了，也不逃避它，心平气和地为自己争取最合理的解决之道，不可以做一个弱者，凡是不顺心便跌倒的人是要被社会淘汰的，做一个有弹性的人，当是我们一生追求的目标*。

# 第十四章 组合查询

本章讲述怎样利用union将多条SELECT语句组合成一个结果集。

1.先前我们的查询只有一个select，但现在可以用union组合多条select语句。

2.使用union的两种情况

（1）在一个查询中从不同的表返回数据

（2）对一个表执行多个查询，按一个查询返回数据

3.union自动把查询结果中重复的行去掉。

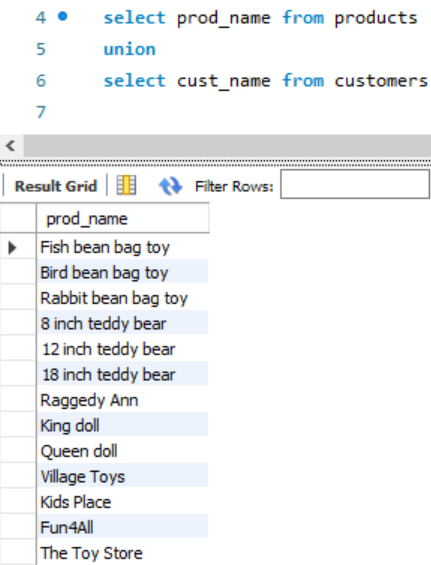
当我们不想把重复的结果去掉时，要用union all

4．要想对union的结果排序，应该写在最后一个select语句的后面。

5.数据库管理系统有其他的联合多个select语句的关键字，像

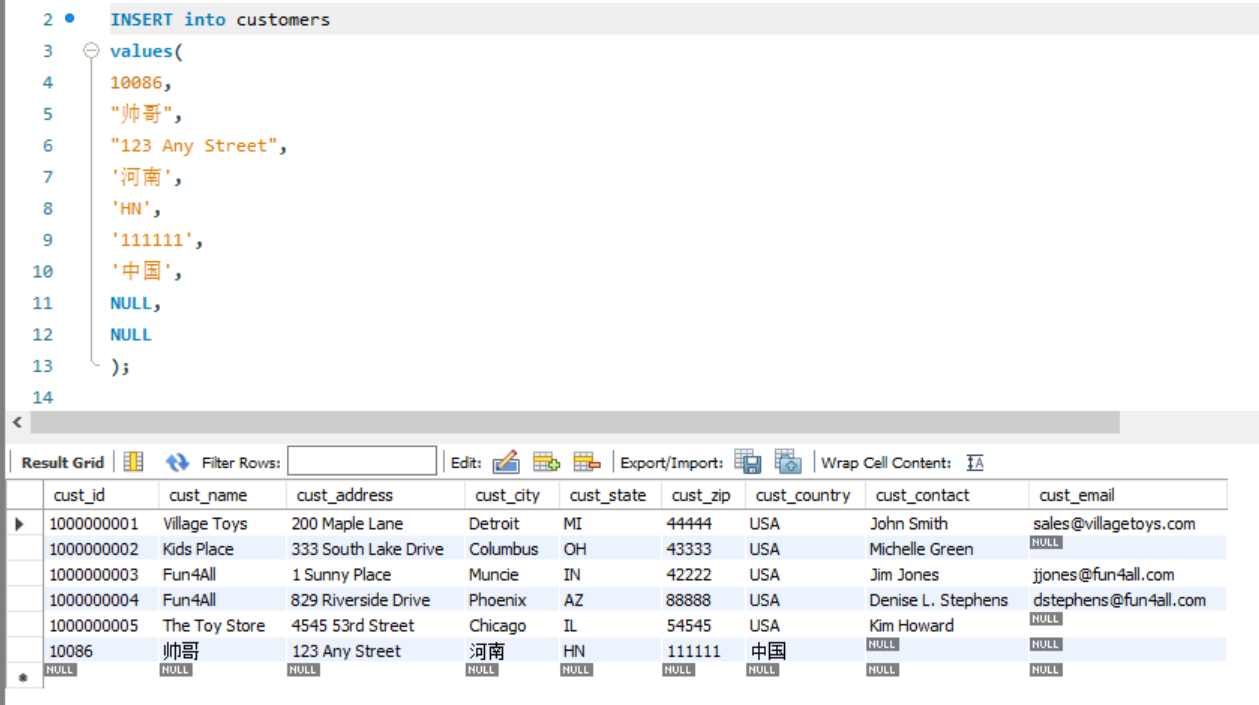
交，差，除 ……（就看DBMS支持哪一个了）

4.union也可对 列名不相同 的进行操作。不过结果的列名会变成第一个select语句对应的列名。



# 第十五章 插入数据

1. 前面我们一直讲的SELECT语句，现在来介绍能往表里插入数据的INSERT语句。
2. insert into 表名 values(每一列对应的数据)



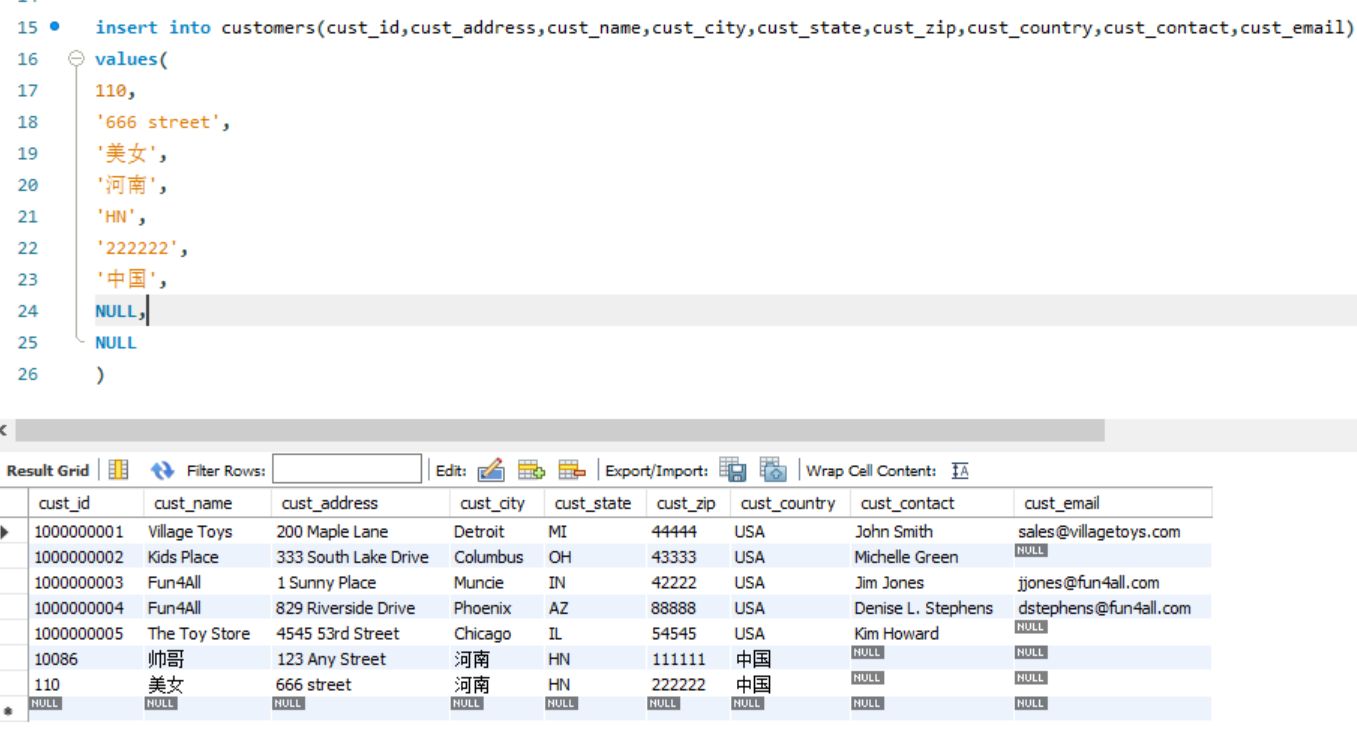
1. into这个关键字可以省略，

但为了DBMS之间的可移植性，最好加上去。

4.insert into 表名（列名，列名）

Values(列名对应的数据，……)

如果表名后不跟列名的话，按照表中列的顺序去在values()里面写数据。但用 表名（列名，列名） 后，按照 列名在括号里面的顺序填入数据。



最好是用 insert into 表名（列名，……）这种形式，即使表的结构发生改变，不会发生重大错误。

1. 所插入的值，一个不漏的写出表中的所有列对应的值，你可以在insert into 表名（列名，列名） 在表名后的括号里 写出 有限的列名，根据这些列名写values(),而为未写出的列名会被置为NULL或是你预先设定的默认值。

//像之前的两个表里，cust\_contact 和cust\_email均为NULL。

1. 插入检索出的数据。

利用insert into 表名（列名）

select语句

这时候就不需要values（）了。

检索出来的列依次填入表里面，跟select语句检索出的列名无关

1. 把一个表复制到另一个表里面。

custcopy是创建的新表，里面的数据跟customers表一样。



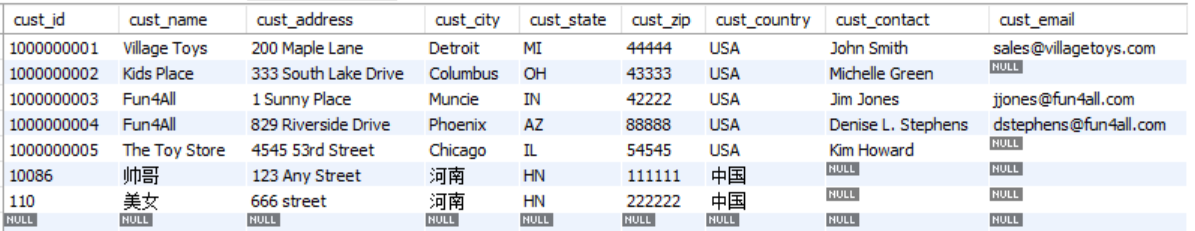
也可以只把customers表中的一部分列复制到新表custcopy,如下图。



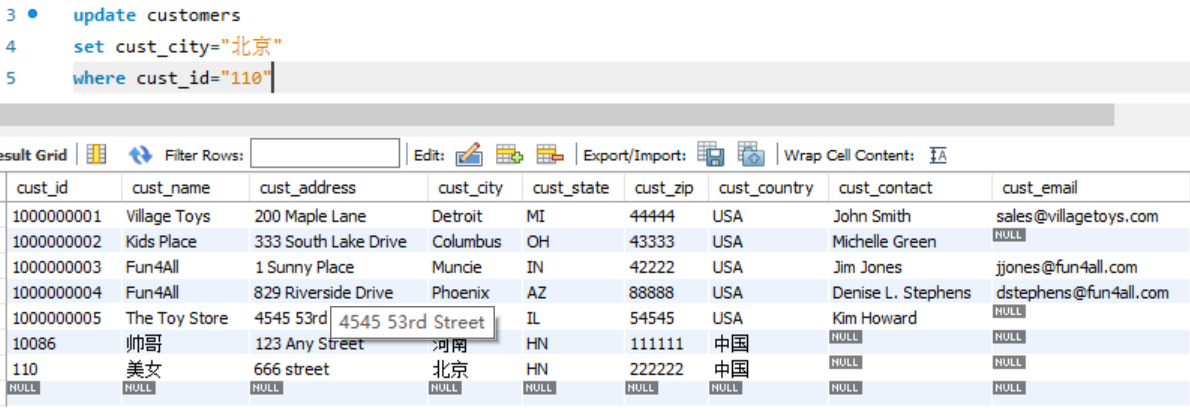
# 第十六章 更新和删除数据

1. 利用update 更新特定行或所有行的值。
2. 现在来利用update来将cust\_id=110这一行的cust\_city改为北京。

下图为修改前：



下图为修改后：



1. update 的语法为:

update 表名

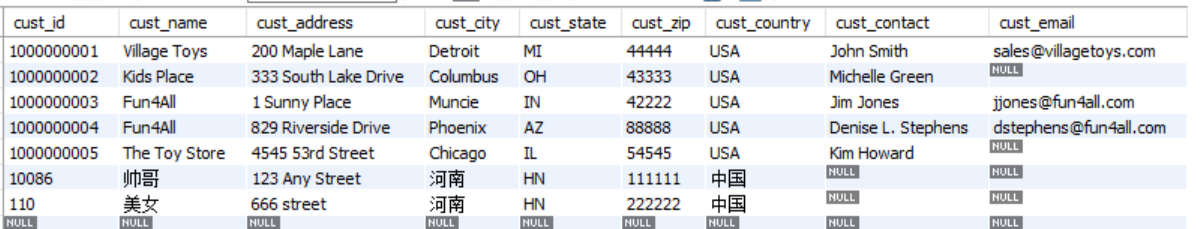
set 列名= 值 //所要更改的行的属性值

where 字句 //过滤条件

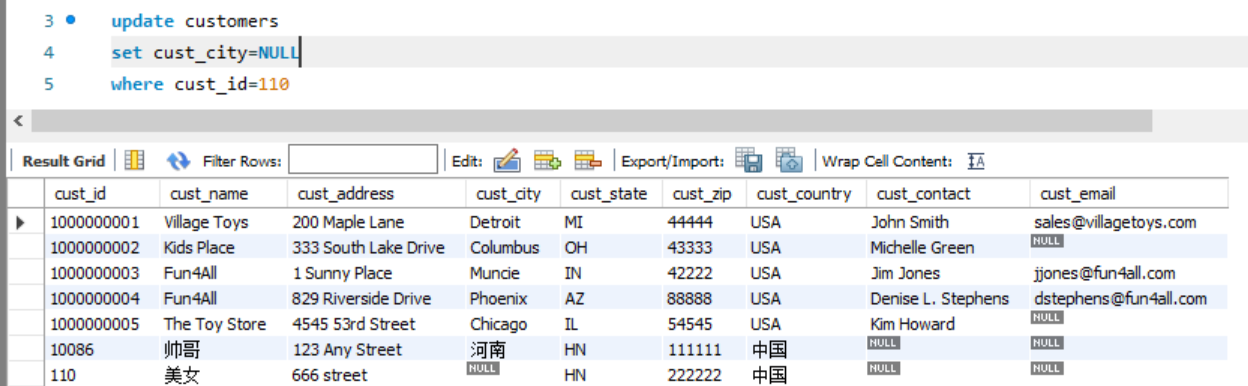
1. 也可以利用update来删除某个列的值

下面把cust\_id=110的行的cust\_city值删除

下图为删除前：



下图为删除后：



！！！！！用delete无法删除某个列的值，delete只能删除特定行或所有行，不能在特定的列上进行删除。（下面我们来了解一下delete）

1. delete 语法：

delete from 表名 //所要删除的表

where子句 //来确定删除条件

# 第十七章 创建和操纵表

1.创建表：

Create table 表名

（ 列名 数据类型 ,

列名 数据类型,

……………………

）

在创建表时，也可以设置 NULL，default,主键等

2.添加列：

alter 表名

add column 所要添加的列的名字 列的数据类型

3.删除列：

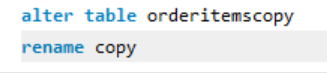
alter 表名

drop column 表中存在的列名

4.删除表：

drop table 表名

5.修改表的名字：



（不同的DBMS用法不同）

# 第十八章 视图

1.当原表的内容发生改变后，视图也改变。

2.视图内容的改变会引起原表内容的改变。

3.对于视图的修改，也可以像表那样使用delete,update,insert

4.创建视图

Create view 视图名

As select语句

5.对于视图的操作，例如select跟表的操作一样，不过一些数据库管理系统不允许视图在进行查询时使用order by子句，