

从现在开始我们开始研究不同表之间的关系。

一. 基本概念

1. **关系表**：举个简单的例子：比方说我们要存放同一个生产商生产的多个不同的商品（注意在这里我们还要存放生产商的信息）。如果把它们都放在同一个表中，那么会有如下缺点：①表中会有大量**重复信息**，这既浪费时间又浪费空间。②如果生产商的信息**改变**，那么我们不得不一个一个地去改变每一个对应商品的信息。所以我们要把这两种信息分开存储，放在不同的表格当中。

2. **主键**：每一行都应该有表示那一行的独特的信息。这些信息我们把它叫做主键。（由此可以看出在表中每一列的地位并不都是均等的，其它列的存在**依赖于主键**）

3. **外键**：外键就是另一张表中的主键，他专门用来联结不同的表。

（还有一个概念：可伸缩性：能够不断增加工作量而不失败，我们称为这种东西**可伸缩性好**）

4. **内部联结**：基本上由于主键和外键的存在，我们在联结表时使用的都是**相等**的判断。这种就是内部联结。

二. 创建联结的两种方法

1. 使用WHERE和完全限定列名

1. 示例：`SELECT vend_name, prod_name, prod_price FROM vendors, products WHERE vendors.vend_id = products.vend_id ;`

2. 注意事项：

①在这里我们从FROM后面跟了多个表名。

②我们使用了WHERE，而且使用了完全限定列名（如果不使用WHERE，我们将会得到笛卡儿积（就是返回的行数是对应函数相乘））（另外，如果列名**没有歧义**，可以不适用完全限定列名）

③WHERE后面的条件我们使用的是**主键和外键**的连接。

2. 使用FROM...INNER JOIN和ON

1. 示例：`SELECT vend_name, prod_name, prod_price FROM vendors INNER JOIN product ON vendors.vend_id = products.vend_id;`

2. 注意事项：与WHERE相比，我们用INNER JOIN 取代了逗号（, ），同时使用ON取代了WHERE,但是我们还是使用了完全限定列名。

（为什么我们要用这种方法呢？因为这是ANSI SQL规范首选的）

三. 联结多个表

在联结多个表这方面我们只能使用WHERE了，首先列出所有表，然后定义表之间的关系：

示例：(这个就不示例了，太长了)

用这个性质我们可以**将子查询和联结**相互转化。

不过从性能上考虑联结是相当消耗内存的，所以在使用联结时一定要谨慎，不要意外联结了不必要的表。

四. 高级联结的使用

1. 自联结：

应用背景：比如一个商家生产的A有问题，那么我们就有必要列出该商家所有的产品。在相关的表中找到对应的商家后，我们需要**返回到原来的表**，去查看他所对应的数据。

2. 自然联结：

这个书上说得很含糊，不过我们迄今为止建立的每个内部联结都是自然联结。而且，我们基本永远都用不到不是自然联结的内部联结。

3. 外部链接：

有些时候要用到包含**没有关联行的那些行**。

比如：对每个客户下了多少订单进行计数，包括那些没有人订购的产品。

使用关键字：FROM LEFT/RIGHT OUTER JOIN .. ON ..

LEFT 代表OUTER JOIN左边的行，RIGHT代表OUTER JOIN右边的行。

五. 其它技巧

使用表别名：因为我们涉及到了不同的表，所以我们经常要使用完全限定列名，这意味着我们要一遍又一遍地输入相同的表名。其实我们可以用FROM .. AS来给表取一个别名。这样在下面的使用中我们就不用重复工作了。