[**SQL 关于apply的两种形式cross apply 和 outer apply**](http://www.cnblogs.com/shuangnet/archive/2013/04/02/2995798.html)

**SQL 关于apply的两种形式cross apply 和 outer apply**

**apply有两种形式： cross apply 和 outer apply**

**先看看语法：**

<left\_table\_expression>  {cross|outer} apply <right\_table\_expression>

**再让我们了解一下apply运算涉及的两个步骤：**

1. A1:把右表表达式(<right\_table\_expression>)应用到左表(<left\_table\_expression>)输入的行；
2. A2:添加外部行；

　　使用apply就像是先计算左输入，让后为左输入中的每一行计算一次右输入。（**这一句很重要，可能会不理解，但要先记住，后面会有详细的说明**）

**最后结合以上两个步骤说明cross apply和outer apply的区别：**

　　cross apply和outer apply 总是包含步骤A1，只有outer apply包含步骤A2，如果cross apply左行应用右表表达式时返回空积，则不返回该行。而outer apply返回改行，并且改行的右表表达式的属性为null。

**看到上面的解释或步骤大家可能还是一头的雾水，不知所云。下面用例子来说明：**

先建表一（[dbo].[Customers]  字段说明：customerid -- 消费者id ， city -- 所在城市）：







CREATE TABLE [dbo].[Customers](

 [customerid] [char](5) COLLATE Chinese\_PRC\_CI\_AS NOT NULL,

 [city] [varchar](10) COLLATE Chinese\_PRC\_CI\_AS NOT NULL,

PRIMARY KEY CLUSTERED

(

 [customerid] ASC

)WITH (IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]







向表一插入数据：

insert into dbo.Customers values('FISSA','Madrid');

insert into dbo.Customers values('FRNDO','Madrid');

insert into dbo.Customers values('KRLOS','Madrid');

insert into dbo.Customers values('MRPHS','Zion');

查询所插入的数据：

select \* from dbo.Customers

结果如图：



再建表二（[dbo].[Orders]  字段说明：orderid -- 订单id  ， customerid -- 消费者id）：







CREATE TABLE [dbo].[Orders](

 [orderid] [int] NOT NULL,

 [customerid] [char](5) COLLATE Chinese\_PRC\_CI\_AS NULL,

PRIMARY KEY CLUSTERED

(

 [orderid] ASC

)WITH (IGNORE\_DUP\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]







向表二插入数据：







insert into dbo.Orders values(1,'FRNDO');

insert into dbo.Orders values(2,'FRNDO');

insert into dbo.Orders values(3,'KRLOS');

insert into dbo.Orders values(4,'KRLOS');

insert into dbo.Orders values(5,'KRLOS');

insert into dbo.Orders values(6,'MRPHS');

insert into dbo.Orders values(7,null);







查询插入的数据：

select \* from dbo.orders

结果如图：



**例子：题目：得到每个消费者最新的两个订单：**

**用cross apply**

**sql：**







select \*

from dbo.Customers as C

 cross apply

 (select top 2 \*

 from dbo.Orders as O

 where C.customerid=O.customerid

 order by orderid desc) as CA







**结果如图：**



**过程分析：**

　　它是先得出左表【dbo.Customers】里的数据，然后把此数据一条一条的放入右表表式中，分别得出结果集，最后把结果集整合到一起就是最终的返回结果集了（T1的数据 像for循环一样 一条一条的进入到T2中 然后返回一个集合  最后把所有的集合整合到一块  就是最终的结果），最后我们再理解一下上面让记着的话（**使用apply就像是先计算左输入，让后为左输入中的每一行计算一次右输入**）是不是有所明白了。

**实验：用outer apply 试试看看的到的结果：**

**sql语句：**







select \*

from dbo.Customers as C

 outer apply

 (select top 2 \*

 from dbo.Orders as O

 where C.customerid=O.customerid

 order by orderid desc) as CA







**结果如图：**



**结果分析：**

　　发现outer apply得到的结果比cross多了一行，我们结合上面所写的区别（**cross apply和outer apply 总是包含步骤A1，只有outer apply包含步骤A2，如果cross apply左行应用右表表达式时返回空积，则不返回该行。而outer apply返回改行，并且改行的右表表达式的属性为null**）就会知道了。