```
CASE WHEN sex = '1' THEN '男'
          WHEN sex = '2' THEN '女'
ELSE '其他' END
```

```
CASE WHEN columnName is null THEN 0 ELSE columnName END
```

http://www.cnblogs.com/yazdao/archive/2009/12/09/1620482.html

Case具有两种格式。简单Case函数和Case搜索函数。

```
--简单Case函数
CASE sex
WHEN '1' THEN '男'
WHEN '2' THEN '女'
ELSE '其他' END
--Case搜索函数
CASE WHEN sex = '1' THEN '男'
WHEN sex = '2' THEN '女'
ELSE '其他' END
```

这两种方式，可以实现相同的功能。简单Case函数的写法相对比较简洁，但是和Case搜索函数相比，功能方面会有些限制，比如写判断式。
还有一个需要注意的问题，Case函数只返回第一个符合条件的值，剩下的Case部分将会被自动忽略。

```
--比如说，下面这段SQL，你永远无法得到“第二类”这个结果
CASE WHEN col_1 IN ( 'a', 'b') THEN '第一类'
WHEN col_1 IN ('a')       THEN '第二类'
ELSE'其他' END
```

下面我们来看一下，使用Case函数都能做些什么事情。

一，已知数据按照另外一种方式进行分组，分析。

有如下数据:(为了看得更清楚，我并没有使用国家代码，而是直接用国家名作为Primary Key)

国家（country）	人口（population）
中国	600
美国	100
加拿大	100
英国	200
法国	300
日本	250
德国	200
墨西哥	50
印度	250

根据这个国家人口数据，统计亚洲和北美洲的人口数量。应该得到下面这个结果。

洲	人口
亚洲	1100
北美洲	250
其他	700

想要解决这个问题，你会怎么做？生成一个带有洲Code的View，是一个解决方法，但是这样很难动态的改变统计的方式。

如果使用Case函数，SQL代码如下:

```
SELECT  SUM(population),
CASE country
WHEN '中国'      THEN '亚洲'
WHEN '印度'      THEN '亚洲'
WHEN '日本'      THEN '亚洲'
WHEN '美国'      THEN '北美洲'
```

```
WHEN '加拿大'  THEN '北美洲'
WHEN '墨西哥'  THEN '北美洲'
ELSE '其他' END
FROM    Table_A
GROUP BY CASE country
WHEN '中国'      THEN '亚洲'
WHEN '印度'      THEN '亚洲'
WHEN '日本'      THEN '亚洲'
WHEN '美国'      THEN '北美洲'
WHEN '加拿大'  THEN '北美洲'
WHEN '墨西哥'  THEN '北美洲'
ELSE '其他' END;
```

同样的，我们也可以用这个方法来判断工资的等级，并统计每一等级的人数。SQL代码如下；

```
SELECT
CASE WHEN salary <= 500 THEN '1'
WHEN salary > 500 AND salary <= 600  THEN '2'
WHEN salary > 600 AND salary <= 800  THEN '3'
WHEN salary > 800 AND salary <= 1000 THEN '4'
ELSE NULL END salary_class,
COUNT(*)
FROM    Table_A
GROUP BY
CASE WHEN salary <= 500 THEN '1'
WHEN salary > 500 AND salary <= 600  THEN '2'
WHEN salary > 600 AND salary <= 800  THEN '3'
WHEN salary > 800 AND salary <= 1000 THEN '4'
ELSE NULL END;
```

二，用一个SQL语句完成不同条件的分组。

有如下数据

国家（country）	性别（sex）	人口（population）

中国	1	340
中国	2	260
美国	1	45
美国	2	55
加拿大	1	51
加拿大	2	49
英国	1	40
英国	2	60

按照国家和性别进行分组，得出结果如下

国家	男	女
中国	340	260
美国	45	55
加拿大	51	49
英国	40	60

普通情况下，用UNION也可以实现用一条语句进行查询。但是那样增加消耗(两个Select部分)，而且SQL语句会比较长。

下面是一个是用Case函数来完成这个功能的例子

```
SELECT country,
SUM( CASE WHEN sex = '1' THEN
population ELSE 0 END),  --男性人口
SUM( CASE WHEN sex = '2' THEN
population ELSE 0 END)   --女性人口
FROM  Table_A
GROUP BY country;
```

这样我们使用Select，完成对二维表的输出形式，充分显示了Case函数的强大。

三，在Check中使用Case函数。

在Check中使用Case函数在很多情况下都是非常不错的解决方法。可能有很多人根本就不用Check，那么我建议你在看过下面的例子之后也尝试一下在SQL中使用Check。
下面我们来举个例子
公司A，这个公司有个规定，女职员的工资必须高于1000块。如果用Check和Case来表现的话，如下所示

```
CONSTRAINT check_salary CHECK
( CASE WHEN sex = '2'
THEN CASE WHEN salary > 1000
THEN 1 ELSE 0 END
ELSE 1 END = 1 )
```

如果单纯使用Check，如下所示

```
CONSTRAINT check_salary CHECK
( sex = '2' AND salary > 1000 )
```

女职员的条件倒是符合了，男职员就无法输入了。