

values 妙用

参考：<https://blog.csdn.net/twtwffg/article/details/5733795>

1、 values 虚表

—最简单

```
Select * from (values 1) as a;
```

—这是SQL很有用，可以做一些逻辑判断，如果加上where条件的话。

```
Select * from (values 1) as a where &num>0;
```

其中&num将用数值替换。

—建个多字段的表

```
Select * from (values ('1A@cn.ibm.com',1),('ABC@126.com',2) ) as a(name,id);
```

可以用这种方法指定一些常量，用来left outer join.

2、

```
db2 => values 1
```

```
1
```

```
-----
```

```
1
```

1 条记录已选择。

— 获取系统当前日期

```
select current date from sysibm.sysdummy1;
```

```
select current time from sysibm.sysdummy1;
```

```
select current timestamp from sysibm.sysdummy1;
```

—sysibm.sysdummy1表是一个特殊的内存中的表，用它可以发现如上面演示的 DB2 寄存器的值。您也可以使用关键字 VALUES 来对寄存器或表达式求值。

```
VALUES current date;
```

```
VALUES current time;
```

```
VALUES current timestamp;
```

3、 values 赋值

— 给变量赋值的另一种方法

```
values expr1, expr2, expr3 into a, b, c;
```

— 相当于

```
set a = expr1;
```

```
set b = expr2;
```

```
set c = expr3;
```

4、

在 VALUES 子句中包含多行的内容。

例如，下面的语句将插入三行：INSERT INTO mytable VALUES (1, 'abc'), (2, 'def'), (3, 'ghi')

5、

数据库的视图里有sysibm.tables和syscat.tables，这两个有什么区别？很多资料上不是说sysibm模式的是基表吗，怎么在视图里也有sysibm模式，是干什么用的？

sysibm.tables的表里面的信息比syscat.tables的少，或者说IBM认为你想知道的信息在sysibm.tables里面已经都可以知道了，这个应该是给用户使用的。syscat.tables应该是给系统使用的。

sysibm这个schema下的view，是对很多表的简化，可以直接查view来检索你感兴趣的资料。

