

数据库中的过程和函数

用于匹配桥型的函数

```
CREATE FUNCTION matchingBridge(@temp_mainSpan INT, @temp_towerHeight DECIMAL(10,3))
RETURNS TABLE
AS
RETURN
    SELECT TOP 3 bridgeID
    FROM bridge
    ORDER BY ABS(mainSpan-@temp_mainSpan) + ABS(towerHeight - @temp_towerHeight) *
0.25
GO
```

用来匹配相似桥型的表值函数，传入参数为

1. @temp_mainSpan: 用户指定的主跨，INT
2. @temp_towerHeight: 用户指定的桥面以上塔高，DECIMAL(10,3)

返回值为按照相似度排名的已有桥梁的编号列

测试例子

```
SELECT * FROM dbo.matchingBridge(800, 250)
```

用于查询某用户已经创建的模型数量

如果模型编号按照 `userID + modelNum` 的格式，则这个更有用，但是我感觉 `userID` 前缀其实是不必要的。

```
CREATE PROC modelNum
@temp_userID VARCHAR(10),
@temp_modelNum INT OUT
AS
    SELECT @temp_modelNum = COUNT(modelID)
    FROM bridgeModel
    WHERE userID = @temp_userID
GO
```

这里是用来返回某一个用户已存入 `bridgeModel` 表中的行数，传入参数为用户编号 `userID`。返回值为一个 INT。

测试例子

```
DECLARE @temp_userID VARCHAR(10)
DECLARE @temp_modelNum INT
SET @temp_userID = 'NanxiChen'
EXEC modelNum @temp_userID, @temp_modelNum OUT
PRINT @temp_modelNum
```

用于查询所有用户创建的模型个数

直接按照形如 m001,m002... 给模型编号，这个更有用，就相当于直接 `SELECT COUNT(*) FROM bridgeModel`，但是据说存储过程更安全（更容易得分T_T）。

```
CREATE PROC modelNumAll
@temp_modelNum INT OUT
AS
    SELECT @temp_modelNum = COUNT(modelID)
    FROM bridgeModel
GO
```

不传入参数，返回值也为 INT

测试例子

```
DECLARE @temp_modelNum INT
EXEC modelNumALL @temp_modelNum OUT
PRINT @temp_modelNum
```