

# sql实际应用-递归查询 - zhangjf\_兔子 - 博客园

## 1、既然要谈到sql，数据库表是必须的

树形结构 ( TuziTree )		
编号	int	<pk>
名称	nvarchar(64)	
父级编号	int	
节点路径	nvarchar(1028)	

## 2、数据结构

	Id	Name	ParentId
1	1000	软件开发语言	0
2	1001	JAVA	1000
3	1002	Hibernate	1001
4	1003	Struts	1001
5	1004	Spring	1001
6	1005	DotNet	1000
7	1006	ASP.NET	1005
8	1007	MVC	1005
9	1008	MYSQL	1002

## 3、获取某个节点的所有子节点

传统的写法 (sql2000) 很麻烦，暂且就不写了

来看看CTE的写法



```
CREATE PROC sp_getTreeById(@TreeIdint)ASBEGINWITH cteTree AS (SELECT*FROM TuziTree WHERE Id
=@TreeId--第一个查询作为递归的基点 (锚点) UNION ALL SELECT TuziTree.*--第二个查询作为递归成员， 下
属成员的结果为空时，此递归结束。FROM
cteTree INNERJOIN TuziTree ON cteTree.Id = TuziTree.ParentId) SELECT*FROM cteTree END
```



测试一下啊

```
exec sp_getTreeById @TreeId=1001
```

结果

	Id	Name	ParentId
1	1001	JAVA	1000
2	1002	Hibernate	1001
3	1003	Struts	1001
4	1004	Spring	1001
5	1008	MYSQL	1002

## 4、使用节点路径来做 (每个节点路径都保存自身的路径和所有父节点的路径=自己和所有父节点的关联)

	Id	Name	ParentId	NodePath
1	1000	软件开发语言	0	/1000/
2	1001	JAVA	1000	/1000/1001/
3	1002	Hibernate	1001	/1000/1001/1002/
4	1003	Struts	1001	/1000/1001/1003/
5	1004	Spring	1001	/1000/1001/1004/
6	1005	DotNet	1000	/1000/1005/
7	1006	ASP.NET	1005	/1000/1005/1006/
8	1007	MVC	1005	/1000/1005/1007/
9	1008	MYSQL	1002	/1000/1001/1002/1008/

## 5、既然有个路径

那么查询其所有子节点 只需要 where nodePath like '/1001/%'了

这样就会简单很多，加上索引。

总结：

如果在性能的需要上，我们可以采用**按需加载**，点击节点时候 才会加载其所有子节点。

如果在变化不大的情况下，可以采用**缓存**。这样的处理 可以满足很多业务需求。

良好的表设计会给后期的开发以及需求变化 带来更多的便利。

下次继续总结sql方面的知识，案例一切以实际工作演变而来。