信息化服务事业部

SQL编程规范

|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ] 草稿  [√] 正式发布  [ ] 正在修改 | 文件编号： |  |
| 当前版本： | V1.0 |
| 作 者： | 杜飞 |
| 审 核 人： |  |
| 审核日期： |  |

**目 录**

[1 通用规范 1](#_Toc355994130)

[2 表名规范 1](#_Toc355994131)

[3 字段命名和设计规范 1](#_Toc355994132)

[4 存储过程命名及编码规范 3](#_Toc355994133)

[5 函数命名规范 4](#_Toc355994134)

# 通用规范

1. select 后禁止以\*代替所有字段，应该明确列出字段名称。
2. 所有代码段应该以Begin开始，以End结束。
3. 尽量减少触发器的使用。
4. 所有的关键字必须大写（可适当放宽，推荐大写）。

# 表名规范

**{模块}\_{表名}**

1、表名应该依据表内将要存放主体内容命名，以含义准确的英文单词的名词格式命名，表名禁止缩写。

2、根据表内内容和单词含义之间的关系考虑使用单或复数（如果表示多个具体的信息采用复数；如果表示概念或个体则用单数形式如：Dictionary字典）。

3、作为分类和标识作用的{模块}由于会**大量重复**的出现，应该尽量采用缩写的形式。

例如：

Org\_Organizations 🡪组织机构信息表；

Org\_Users 🡪组织机构用户表；

# 字段命名和设计规范

1、表示存储内容的英文单词或单词的组合；如：LinkMan，大小写采用Pascal命名方式，单词首字母大写。

2、表示存储内容状态的字段，应该尽量集中到一个字段中，如：Status；设计Status思路：

a、当不同状态值是互斥的时候，利用枚举值表达不同地状态。

b、当不同状态可以共生进，利用Bit位来表示不同的状态。

3、ID都使用大写

4、时间字段统一命名XX时间(创建时间)，英文命名XXTime(CreateTime)

**例:组织机构表【Org\_Organizations】**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段** | **字段名称** | **数据类型** | **可为空** | **默认值** | **备注** |
| ID | 组织机构主键Id | Int | No | 自动增长 | 主键，物理索引 |
| Name | 组织机构名称 | Nvarchar(125) | No | 空串 |  |
| TypeID | 机构类型Id | Int |  | -1 | 外键ID🡨Dictionary |
| Status | 机构状态 | Int | No | 0 | 0-刚注册；1—已经审核，2—已经申请企业门户；3—扩展 |
| Flag | 通用指示标识 | Int | No | 0 | 1. 正常状态 2. 临时锁定 |
| LinkMan | 联系人姓名 | Nvarchar(20) | Yes |  |  |
| … | … | … |  | … | … |
| CreateTime | 创建时间 | DateTime | No | Getdate() |  |
| UpdateTime | 最后修改时间 | DateTime | No | Getdate()` |  |

# 存储过程命名及编码规范

**1、SP\_{模块}\_{功能}**

模块和表名规范模块的规范相同，应尽量采用缩写的形式。

功能描述以动词或动宾短语的方式出现；如：UserAdd，书写格式采用Pascal方式，个别情况下会加上查询条件。

如： SP\_Org\_OrgAdd 🡪添加组织机构；

SP\_Org\_ OrgDelete 🡪删除组织机构；

个别情况下命名方式示例：

SP\_Org\_OrginationGetByID

**2、存储过程结构**

-- ======================================================

-- Author: dufei

-- Create date: 2012-05-08

-- Description: 日志批量插入

-- Example: EXE SP\_Log\_LogInsert '参数'

-- ======================================================

IF EXISTS (SELECT \* FROM sysobjects WHERE name = ' SP\_Log\_LogInsert ')

DROP PROC SP\_Log\_LogInsert

GO

CREATE PROCEDURE SP\_Log\_LogInsert

(

@LogXml VARCHAR(max) --日志内容

)

AS

BEGIN

存储过程逻辑

END

--==================修改记录=====================

--dufei 2013-05-10 添加了删除记录的逻辑

--dufei 2013-05-11 修改了删除记录的逻辑

--=============================================

如没有修改记录，不写即可

# 函数命名规范

1、FN\_{模块}\_{功能}

模块和表名规范模块的规范相同，应尽量采用缩写的形式。

功能描述以动词或动宾短语的方式出现；如：UserAdd，书写格式采用Pascal方式，个别情况下会加上查询条件。

如： FN\_Org\_OrgGetName 取得组织机构名称

FN \_Org\_ Org PrivilegedGet  取得有权限的组织机构

个别情况下命名方式示例：

FN \_Org\_OrgGetByUserID

2、函数结构

-- ======================================================

-- Author: dufei

-- Create date: 2012-05-08

-- Description: 日志批量插入

-- Example: SELECT FN \_Org\_OrgNameGetByUserID '参数'

-- ======================================================

IF OBJECT\_ID (N' FN \_Org\_OrgNameGetByUserID ') IS NOT NULL

DROP FUNCTION FN \_Org\_OrgNameGetByUserID

GO

CREATE FUNCTION FN \_Org\_OrgNameGetByUserID

(

@UserID BIGINT --日志内容

)

RETURNS NVARCHAR(max)

AS

BEGIN

函数程逻辑

END

--==================修改记录=====================

--dufei 2013-05-10 添加了删除记录的逻辑

--dufei 2013-05-11 修改了删除记录的逻辑

--=============================================

如没有修改记录，不写即可

# 逻辑编写规范

## 定义变量：

DECLARE @SelectSQL NVARCHAR(MAX) ,

@WhereCondition NVARCHAR(MAX) ,

@ConditionTable NVARCHAR(MAX) ,

## 查询语句

SELECT res.StorageID ,

sto.StorageName ,

res.LogisticsProviderID AS TranCorpID ,

lp.LogisticsProviderName AS TranCorpName ,

res.Amount AS ActualAmount ,

res.AfterTaxAmount ,

0.0 AS ActualAmountPay,

0.0 AS AftertaxAmountPay,

CONVERT(BIT, 0) AS IsPay ,

CONVERT(DATETIME, NULL) AS PayDate ,

res.EffectiveDate AS ShipTime ,

res.StorageReimbursementNo

FROM StorageAdjustAccountApplies res

LEFT JOIN dbo.Storages sto ON res.StorageID = sto.ID

LEFT JOIN dbo.LogisticsProviders lp ON res.LogisticsProviderID = lp.ID

WHERE res.ApproveStatus = 1

AND res.EffectiveDate >= @StartDate

AND res.EffectiveDate <= @EndDate

备注：如果字段较多，可灵活处理select后每行字段个数

## 更新语句

UPDATE r

SET r.IsPay = sr.IsPay ,

r.PayDate = sr.PayDate

FROM #StorageAdjust r

INNER JOIN StorageReimbursements sr ON r.StorageReimbursementNo = sr.ReimbursementNo

WHERE sr.IsPay = 1

AND ISNULL(r.StorageReimbursementNo, '') <> ''

## 删除语句

DELETE i

FROM dbo.FeeItems i

INNER JOIN dbo.FeeRates r ON i.ID = r.FeeItemID

WHERE r.Applier = 'test'