**数据库命名规范**

文档信息

|  |  |
| --- | --- |
| 文档名称 | “山东省人力资源市场数据采集系统”  数据库命名规范 |
| 描述 | 该文档描述“山东省人力资源市场数据采集系统”的数据库设计命名规范。所有设计人员、开发人员、测试人员以及其他团队成员都应该一概文档作为产品的功能定义，并衍生出其他文档 |
| 负责人 | 吴志达 |
| 状态 | 1.0版 |

审核结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 审核人 | 审核时间 | 意见 | 签名档 | 备注 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1 概述 2](#_Toc446233211)

[1.1 基本原则 2](#_Toc446233212)

[1.2 基本规范 2](#_Toc446233213)

[2 命名规范 3](#_Toc446233214)

[2.1 对象命名 3](#_Toc446233215)

[2.1.1 数据库 3](#_Toc446233216)

[2.1.2 数据库文件 3](#_Toc446233217)

[2.1.3 关系型数据仓库 3](#_Toc446233218)

[2.1.4 数据架构 3](#_Toc446233219)

[2.1.5 数据表 4](#_Toc446233220)

[2.1.6 数据视图 4](#_Toc446233221)

[2.1.7 数据列 4](#_Toc446233222)

[2.1.8 存储过程 4](#_Toc446233223)

[2.1.9 函数 4](#_Toc446233224)

[2.1.10 用户定义数据类型 5](#_Toc446233225)

[2.1.11 DML触发器 5](#_Toc446233226)

[2.1.12 DDL触发器 5](#_Toc446233227)

[2.2 参数命名 5](#_Toc446233228)

[2.2.1 数据列参数 5](#_Toc446233229)

[2.2.2 非数据列参数 6](#_Toc446233230)

[2.3 常见命名 2.3.1 常用字段命名 6](#_Toc446233231)

[3 SQL编写 6](#_Toc446233232)

[3.1 大小写 6](#_Toc446233233)

[3.2 使用“;” 6](#_Toc446233234)

[3.3 存储格式 7](#_Toc446233235)

[3.4 类型选择 7](#_Toc446233236)

[3.5 默认值 7](#_Toc446233237)

[3.6 字段长度 7](#_Toc446233238)

[3.7 使用“'” 7](#_Toc446233239)

[3.8 语句缩进 7](#_Toc446233240)

[3.9 语句换行 8](#_Toc446233241)

[3.10 语句分割 8](#_Toc446233242)

[3.11 使用“\*” 8](#_Toc446233243)

[3.12 表名别名 8](#_Toc446233244)

[3.13 类型转换 8](#_Toc446233245)

[3.14 数值比较 8](#_Toc446233246)

[3.15 排序 9](#_Toc446233247)

[3.16 Unicode字符串 9](#_Toc446233248)

[3.17 BEGIN...END 块 9](#_Toc446233249)

[3.18 TRY块 9](#_Toc446233250)

[3.19 TOP子句 9](#_Toc446233251)

[3.20 TRANSACTION编写 10](#_Toc446233252)

[3.21 存储过程 10](#_Toc446233253)

[4 代码注释 10](#_Toc446233254)

[4.1 代码头部注释 10](#_Toc446233255)

[4.2 TRANSACTION注释 11](#_Toc446233256)

[5 附录A 命名规则 11](#_Toc446233257)

[5.1 Pascal 大小写 11](#_Toc446233258)

[5.2 Camel 大小写 11](#_Toc446233259)

[5.3 匈牙利命名法 12](#_Toc446233260)

[6 附录B 参考资源 12](#_Toc446233261)

[6.1 Microsoft SQL Server 2005 联机丛书 12](#_Toc446233262)

[6.2 SQL Server 2005示例数据库 12](#_Toc446233263)

[6.3 编写可移植的 Transact-SQL 代码 12](#_Toc446233264)

[6.4 T-SQL 编码标准 12](#_Toc446233265)

# 1 概述

## 1.1 基本原则

以大小写敏感编写SQL语句。

尽量使用Unicode 数据类型。

优先使用连接代替子查询或嵌套查询。

尽量使用参数化SQL查询代替语句拼接SQL查询。

禁止使用[拼音]+[英语]的方式来命名SQL对象或变量。

尽量使用存储过程代替SQL语句。

## 1.2 基本规范

建议采用Pascal样式或Camel样式命名数据库对象。

大写T-SQL语言的所有关键字，谓词和系统函数。

# 2 命名规范

在一般情况下，采用Pascal样式或Camel样式命名数据库对象，使在开发基于数据库应用程序的时候通过ORM工具生成的数据访问代码不需要调整就符合程序开发语言（比如C#）命名规范。另外，关系型数据库同Xml结合得越来越紧密,规范的命名越来越重要。

在实际数据库开发过程中，如果需求方已经提供数据库设计方案，建议以提供的方案为准;在原有数据库上进行升级开发时，在可行的情况下可适当做出设计调整以符合编程规范。

## 2.1 对象命名

### 2.1.1 数据库

第一种方式，采用Pascal样式命名，命名格式为[项目英文名称]。

示例：AdventureWorks

第二种方式，采用Pascal样式命名，命名格式为[项目英文名称] + Db。

示例：AdventureWorksDb

  BizTalkRuleEngineDb

建议采用第一种方式。

### 2.1.2 数据库文件

数据文件：[数据库名称] + \_Data.mdf

日志文件：[数据库名称] + \_Log.ldf

示例：AdventureWorks\_Data.mdf

      AdventureWorks\_Log.ldf

### 2.1.3 关系型数据仓库

采用Pascal样式命名，命名格式为[项目英文名称] + DW。

示例：AdventureWorksDW

### 2.1.4 数据架构

除SQL Server 系统定义的数据架构外，新建架构采用Pascal样式命名，命名格式为[架构名]。

示例：HumanResources

      Production

对数据库对象 Table，View，Procedure，Function等使用数据架构进行归类。在SQL Server 2000中dbo为默认架构。

### 2.1.5 数据表

采用Pascal样式命名，命名格式为[表名]。

示例：Employee

      Product

表名以英文单数命名，主要是参考SQL Server 2005示例数据库，个人理解不采用复数是为了更好的使用ORM工具生成符合编程规范的代码（比如C#）。

示例：使用Product

  而不是Products

### 2.1.6 数据视图

视图名称采用Pascal样式命名，命名格式为v + [视图名称]。

示例：vEmployee

      vSalesPerson

### 2.1.7 数据列

列名称命名采用英文单词或缩写，英文单词只来自于具体业务定义，尽量表达清楚含义。采用Pascal样式命名，命名格式为[列名称]。

示例：AddressID

      PostalCode

尽量避免使用拼音命名，如果不可避免，对于比较短的列名，采用拼音全写，如果拼音列名比较复杂，可以采用首个字用全拼，其它字用首字母大写表示。

示例：福州 Fuzhou

经营方式 JingYFS

### 2.1.8 存储过程

建议采用匈牙利样式命名，命名格式为[存储过程名称]。

示例：uspGetUser

     uspAddUser

备注：在SQL Server 2005示例数据库中使用Camel样式命名。

### 2.1.9 函数

自定义函数采用匈牙利样式命名，命名格式为[函数名]。

示例：

ufnGetDate()

### 2.1.10 用户定义数据类型

采用Pascal样式命名，命名格式为[自定义数据类型名称]。

示例：Flag

      NameStyle

### 2.1.11 DML触发器

DML触发器是当数据库服务器中发生数据操作语言 (DML) 事件时要执行的操作。DML 事件包括对表或视图发出的 UPDATE、INSERT 或 DELETE 语句。根据事件不同命名规则使用前缀进行区分,格式为 [u|i|d] + [表名|视图名]

示例：uEmployee

      iEmployee

      dEmployee

另外一种方式为，

AFTER 触发器：TR\_表名\_[后面插入加I,修改加U,删除加D]。

INSTEAD OF 触发器：TR\_表名或视图名\_OF[后面插入加I,修改加U,删除加D]

### 2.1.12 DDL触发器

响应各种数据定义语言 (DDL) 事件而激发。这些事件主要与以关键字 CREATE、ALTER 和 DROP 开头的 Transact-SQL 语句对应。执行 DDL 式操作的系统存储过程也可以激发 DDL 触发器。

采用Camel样式命名，命名单词能够描述DDL触发器功能。

示例：

CREATE TRIGGER safety

ON DATABASE

FOR DROP\_TABLE, ALTER\_TABLE   AS

   PRINT 'You must disable Trigger "safety" to drop or alter tables!'

   ROLLBACK

另外一种方式为添加ddl前缀，

示例：

CREATE TRIGGER [ddlDatabaseTriggerLog]

ON DATABASE

FOR DDL\_DATABASE\_LEVEL\_EVENTS   AS

## **2.2 参数命名**

### 2.2.1 数据列参数

命名格式为 @ + [列名称]。

示例：@EmployeeID

在列名不符合Pascal样式时（早期遗留系统），例如使用全部大写的列名称，或使用“\_”进行连接的字段名称，参数名称定义使用 @ + [列名称]，这里的列名称尽量符合Pascal样式命名。

### 2.2.2 非数据列参数

在参数无法跟列名称进行关联时，使用能够反映该参数功能的英文单词或单词组合, 采用Pascal样式命名。

示例：@ErrorID

      @Flag

## 2.3 常见命名 2.3.1 常用字段命名

这里的常用字段是指在建表时频繁使用的表名或列名，下表对常用字段进行建议性定义,

列名称          数据类型             说明

CreatedDate     datetime                纪录创建日期，一般使用GETDATE()自动生成

ModifiedDate        datetime                纪录最后修改日期，首次使用GETDATE()

DeletedDate     datetime                记录删除（标记删除）日期

StartDate       datetime                开始日期

EndDate         datetime                结束日期

StartTime       datetime                开始时间

EndTime         datetime                结束时间

rowguid         uniqueidentifier        唯一标识行的ROWGUIDCOL号，用于支持合并复制

ID              int                 使用ID代替Id或id。一般为自增长主键列

ParentID            int                 父ID

Status          int                 状态

# 3 SQL编写

## 3.1 大小写

大写T-SQL 语言的所有关键字，谓词和系统函数。变量名称及游标名称使用Pascal样式。数据类型定义使用全部小写。

示例：DECLARE @LastName nvarchar(32);

## 3.2 使用“;”

使用“;”作为 Transact-SQL 语句终止符。虽然分号不是必需的，但使用它是一种好的习惯。

示例：

USE AdventureWorks;  GO

DECLARE @find varchar(30);

SET @find = 'Man%';

SELECT LastName, FirstName, Phone

FROM Person.Contact

WHERE LastName LIKE @find;

## 3.3 存储格式

尽量采用Unicode数据存储格式，提高可移植性和兼容性，实际应用中尽量使用nchar、nvarchar、ntext代替char、varchar、text。

## 3.4 类型选择

如果字符具有明确的长度，使用nchar代替nvarchar;char代替varchar。

在只有两个可能数值时，使用bit代替int或smallint。

在SQL Server 2005中，使用nvarchar(MAX)代替ntext;varchar(MAX)代替text;varbinary(MAX)代替image。

在特殊的数据表结构中可考虑xml数据类型，达到事半工倍的效果。

## 3.5 默认值

在建立数据表时，尽量使用默认值代替NULL值。比如设置CreatedDate列默认值为GETDATE()。在可行的情况下设置字段为不允许空。

## 3.6 字段长度

始终指定字符数据类型的长度，并确保允许用户可能需要的最大字符数，避免超出最大长度时出现字符丢失现象。对于字符型数据，建议采用2的n次方来定义数据长度。

示例：nvarchar(32)

  varchar(64)

## 3.7 使用“'”

**在 T-SQL 代码中为字符常量使用单引号,避免使用双引号。**

## 3.8 语句缩进

一个嵌套代码块中的语句使用四个空格的缩进。使用Microsoft SQL Server Management Studio ,选择“工具”菜单，打开“选项”菜单，在选项对话框中选择文本编辑器->纯文本->制表符,选中“插入空格单选框”,设置“制表符大小”为4，缩进大小为“4”。

## 3.9 语句换行

建议SQL代码每行以关键字或“'”开头。

示例：

SELECT [ShiftID]

      ,[Name]

      ,[StartTime]

      ,[EndTime]

      ,[ModifiedDate]

  FROM [AdventureWorks].[HumanResources].[Shift]

## 3.10 语句分割

使用一个（而不是两个）空行分隔 T-SQL 代码的逻辑块。

## 3.11 使用“\*”

尽量避免在任何代码中使用 “SELECT \***”。**

## 3.12 表名别名

表名别名要简短，但意义要尽量明确。通常使用大写的表名作为别名，使用 AS 关键字指定表或字段的别名。

## 3.13 类型转换

不要依赖任何隐式的数据类型转换，不要假定 T-SQL 会进行必要的转换。例如，把数字变量赋予字符值。相反，在为变量赋值或比较值之前，应使用适当的 CONVERT 函数使数据类型相匹配。

## 3.14 数值比较

不要将空的变量值直接与比较运算符（符号）比较。如果变量可能为空，应使用 IS NULL 或 IS NOT NULL 进行比较，或者使用 ISNULL 函数。

## 3.15 排序

决不要依赖 SELECT 语句会按任何特定顺序返回行，除非在 ORDER BY 子句中指定了顺序。通常，应将 ORDER BY 子句与 SELECT 语句一起使用。可预知的顺序（即使不是最方便的）比不可预知的顺序强，尤其是在开发或调试过程中。在返回行的顺序无关紧要的情况下，可以忽略 ORDER BY ，减少资源开销。

## 3.16 Unicode字符串

在Unicode字符前面使用N前缀，避免引起数据的不一致。

示例：

-- Assumes the default code page is not Greek

CREATE TABLE #t1 (c1 nchar(1))

INSERT #t1 VALUES(N'Ω')

INSERT #t1 VALUES('Ω')

SELECT \* FROM #t1

输出结果：

c1     ----

## 3.17 BEGIN...END 块

在SQL代码快中尽量使用BEGIN...END 语句块，提高代码可阅读性。

## 3.18 TRY块

在SQL Server 2005中对一些可能执行失败的语句尽量使用TRY块。Transact-SQL 语句组可以包含在 TRY 块中，如果 TRY 块内部发生错误，则会将控制传递给 CATCH 块中包含的另一个语句组。

示例：

BEGIN TRY

    SQL 语句组1

END TRY

BEGIN CATCH

    SQL 语句组2

END CATCH;

## 3.19 TOP子句

在SQL Server 2005中加强了TOP的使用，尽量使用TOP(变量)来减少SQL拼串现象。

## 3.20 TRANSACTION编写

只要在例程中使用多个数据库修改语句，包括在一个循环中多次执行一个语句，就应考虑声明显式事务。在SQL SERVER 2005 中，增加了TRY块可进行很好的应用。

实例：

    BEGIN TRY

        BEGIN TRANSACTION;

        UPDATE [HumanResources].[Employee]

        SET [Title] = @Title

            ,[HireDate] = @HireDate

            ,[CurrentFlag] = @CurrentFlag

        WHERE [EmployeeID] = @EmployeeID;

        INSERT INTO [HumanResources].[EmployeePayHistory]

            ([EmployeeID]

            ,[RateChangeDate]

            ,[Rate]

            ,[PayFrequency])

        VALUES (@EmployeeID, @RateChangeDate, @Rate, @PayFrequency);

        COMMIT TRANSACTION;

    END TRY

    BEGIN CATCH

        -- Rollback any active or uncommittable transactions before

        -- inserting information in the ErrorLog

        IF @@TRANCOUNT > 0

        BEGIN

            ROLLBACK TRANSACTION;

        END

        EXECUTE [dbo].[uspLogError];

    END CATCH;

## 3.21 存储过程

在编写存储过程时，使用PROCEDURE 代替 PROC 简写。

示例：CREATE PROCEDURE [dbo].[存储过程名字]

# 4 代码注释

## 4.1 代码头部注释

在SQL代码块（sql文件或存储过程）的头部进行注释，标注创建人(Author)、创始日期(Create date)、修改信息(Modify [n])。

格式：

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

-- Author:      <Author,,Name>

-- Create date: <Create Date,,>

-- Description: <Description,,>

-- Modify [n]:  < Modifier,Date, Description >

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

示例：

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

-- Author:      Author

-- Create date: yyyy-MM-dd 

-- Description: \*\*\*\*\*\*\*

-- Modify [1]:  UserName, yyyy-MM-dd, \*\*\*\*\*\*\*

-- Modify [2]:  UserName,  yyyy-MM-dd, \*\*\*\*\*\*\*

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

注：日期格式使用 yyyy-MM-dd。Modify [n] n代表修改序号，从1开始，每次修改加1。

## 4.2 TRANSACTION注释

建议在每个事务的开头进行注释，说明该事务的功能。

-- < Modifier,Date, Description >

BEGIN TRANSACTION;

# 5 附录A 命名规则

常见命名规则有四种样式：完全大写、完全小写、Pascal 大小写和 Camel 大小写。

## 5.1 Pascal 大小写

组成标识符的每个单词的首字母大写，其余字母小写的书写约定。对于缩写的双字母单词，要求全部大写。

例如：ApplicationException

      ID

## 5.2 Camel 大小写

标识符的首字母小写，每个后面连接的单词的首字母大写，其余字母小写的书写约定。对于缩写的双字母单词，要求它们出现在标识符首部时全部小写，否则全部大写。

例如：applicationException

      id

## 5.3 匈牙利命名法

匈牙利命名法由匈牙利程序员发明，他在微软工作了多年，此命名法就是通过微软的各种产品和文档传出来。多数有经验的程序员，不管他们用的是哪门语言，都或多或少在使用它。

基本原则：变量名 = 属性 ＋ 类型 ＋ 对象描述

即一个变量名是由三部分信息组成，这样，程序员很容易理解变量的类型、用途，而且便于记忆。

# 6 附录B 参考资源

## 6.1 Microsoft SQL Server 2005 联机丛书

http://www.microsoft.com/downloads/details.aspx?displaylang=zh-cn&FamilyID=BE6A2C5D-00DF-4220-B133-29C1E0B6585F

## 6.2 SQL Server 2005示例数据库

AdventureWorks

AdventureWorksDW

## 6.3 编写可移植的 Transact-SQL 代码

http://www.microsoft.com/china/msdn/library/data/sqlserver/USsqldnsqldevdev\_06112004L.mspx

## 6.4 T-SQL 编码标准

http://www.microsoft.com/china/msdn/library/data/sqlserver/sp04l9.mspx