

· 信息应用 ·

doi:10.3969/j. issn. 1674 - 6732. 2010. 01. 009

SQL CLR 聚合函数在环境质量业务数据计算中的应用

魏房忠,卞 莉

(江苏省环境信息中心,江苏 南京 210036)

摘要:结合声环境质量业务数据处理的实践,阐述了开发部署 SQL CLR 聚合函数的意义和关键步骤,并通过实际应用场景展现了 SQL CLR 聚合函数在复杂数据计算中具备的高效性、高可复用性,展望了该项技术在其他环境业务数据处理中的应用前景。

关键词:SQL CLR; 数据聚合; 环境质量数据; 功能区噪声

中图分类号:TP312

文献标识码:B

文章编号:1674 - 6732(2010) - 01 - 0032 - 06

Application of SQL CLR Aggregate Function to Environmental Quality Data Processing

WEI Fang-zhong, BIAN Li

(Jiangsu Provincial Environmental Information Center, Nanjing, Jiangsu 210036, China)

ABSTRACT: The purpose and procedures for development of SQL CLR aggregate function data processing was presented and the efficiency and the reproducibility of the technique in complex data processing environment was showcased through a case study. Finally, the potential application of the technique to more environmental data processing business was projected.

KEY WORDS: SQL CLR; data aggregation; environmental quality data; noise in functional zones

在环境质量业务数据处理过程中,经常涉及对各类原始监测数据进行分组统计。使用数据库 SQL 查询或其他统计软件,能够完成平均值、最小值、最大值、计数等简单分组统计。然而在实际业务数据处理过程中,往往会遇到更复杂的数据分组计算要求。借助专业数据统计软件进行手工方式处理较为费时费力,而普通的程序实现,则往往存在代码量较大、通用性低、复用性差、使用不便等问题。SQL Server 2005 带来的 SQL CLR 编程技术,通过开发部署 SQL CLR 聚合函数,实现自定义聚合逻辑计算方法,将 SQL 的灵活分组查询能力和 CLR 的高效计算能力有机整合,实现分组统计的灵活拓展,是一种非常有效的解决方法。

1 功能区环境噪声评价

在所有环境质量综合分析数据处理中,功能区环境噪声评价不仅计算过程复杂,而且要求逐级分组汇总,具有较好的代表性。

1.1 功能区噪声平均等效声级的算法

功能区环境噪声的评价根据噪声平均等效声级进行判定,噪声平均等效声级的计算过程非常复杂,不是简单的算术平均计算,而是将各监测点位

一天 24 h 的等效声级原始监测数据,经过能量转换求和、算术平均、计算自然对数,最后算出各监测点位一天白天、夜间和昼夜的平均等效声级。

功能区环境噪声昼间、夜间和昼夜平均等效声级的计算公式如下:

$$L_d = 10 \lg \left[\frac{1}{16} \sum_{i=1}^{16} 10^{0.1Leq_i} \right]$$

$$L_n = 10 \lg \left[\frac{1}{8} \sum_{j=1}^8 10^{0.1Leq_j} \right]$$

$$L_{dn} = 10 \lg \left\{ \frac{1}{24} \left[\sum_{i=1}^{16} 10^{0.1Leq_i} + \sum_{j=1}^8 10^{(Leq_j+10)} \right] \right\}$$

式中: Leq_i ——白天(06:00 ~ 21:00)16个h中第*i*个小时的等效声级;

Leq_j ——夜间(22:00 ~ 05:00)8个h中第*j*个小时的等效声级。

1.2 功能区环境噪声评价计算步骤

在对各市功能区环境噪声进行综合评价计算的过程中,为保证数据的代表性与准确性,需要对

收稿日期:2008-12-01;修订日期:2009-03-09

基金项目:国家十一五水体污染控制与治理科技重大专项项目(2009ZX07527-004-004)。

作者简介:魏房忠(1983—),男,助工,本科,主要从事环境信息技术工作。

功能区噪声原始监测数据进行逐级分组计算,具体步骤如下:

(1)根据公式分别计算出各功能区噪声测点每天等效声级监测值的能量加权平均值(L_d 、 L_n 、 L_{dn})。

(2)在各测点能量日平均等效声级的基础上,计算所有测点在统计时段内(季度、半年、年度等)的等效声级算术平均值。

(3)根据测点各自的功能区规划,计算各市4类功能区等效声级的算术平均值,即功能区噪声平均等效声级。

1.3 传统计算方法

为达到能量加权平均和逐级分组的计算要求,功能区平均噪声等效声级的计算步骤较多,过程较为复杂。例如,使用 Excel 数据透视表进行计算,主要操作步骤有:

(1)计算所有等效声级的能量值;

(2)按各点位每天昼夜进行分组,计算能量平均值;

(3)根据能量平均值,计算各点位每天昼夜等效声级;

(4)计算所有测点各时段内等效声级的算术平均值;

(5)计算各市各类功能区等效声级的算术平均值。

由于计算复杂、梯级聚合的特点,采用手工方式处理效率较低、差错率较高,而 SQL Server 2005 自定义聚合函数的开发应用正好能有效解决问题,有力提升环境质量综合分析自动化计算能力。

2 SQL CLR 聚合函数简介

在 SQL Server 数据库中函数有标量值函数、表值函数、聚合函数,并允许用户自定义标量值函数和表值函数,但无法使用 Transact – SQL 创建自定义聚合函数。SQL Server 2005 引入了通用运行语言(Common Language Runtime,CLR),通过建立标量函数、表值函数(TVFs)、程序、触发器、自定义聚合函数以及用户定义的数据类型(UDTs),有效地扩展 SQL Server 数据库引擎的功能^[1]。

与系统聚合函数一样,用户定义聚合函数对一组数值执行计算,返回单个数值。其灵活之处在于允许程序开发人员结合具体业务需求,开发适用于各类业务数据处理的逻辑聚合算法,通过把问题整体分解,实现代码复用。

3 开发 SQL CLR 聚合函数

3.1 前提准备

为能运行 SQL CLR,需要在 SQL Server 数据库中做相应设置准备。首先,设置数据库兼容性级别。将 SQL Server 数据库的兼容性级别设置为“SQL Server 2005(90)”或更高。其次,启用 CLR 集成功能。在“SQL Server 外围应用配置器”中,启用 CLR 集成功能,或在 SQL Server Management Studio 中执行如下 SQL 语句进行设置:

```
EXEC sp_configure 'clr enabled',1
```

```
RECONFIGURE
```

```
GO
```

3.2 程序开发

3.2.1 开发步骤

以 Microsoft Visual Studio 2005/2008 作为开发环境,使用一种 Microsoft .NET Framework 语言,创建聚合类,编写代码实现聚合逻辑,并编译生成 CLR 程序集。以噪声等效声级能量加权平均聚合函数为例,主要开发步骤有:

(1)首先新建一个“SQL Server 项目”。将该项目命名为“SqlServerProject”。

(2)添加 SQL Server 数据库引用。设置目标数据库服务器、用户名、密码等。

(3)在项目“SqlServerProject”中添加一个“聚合”类,此处命名为“LeqAvg”,编写 LeqAvg 类,实现聚合逻辑。

(4)编译生成 CLR 程序集,即 dll 文件。

(5)将项目部署到 SQL Server 中。对项目右键点击“部署”自动添加程序集到数据库中,并创建新的聚合函数。(手工部署方法参考 3.3 节)

3.2.2 C# 代码编写

聚合逻辑的实现依赖于该类中提供了几个模板方法:Init()、Accumulate()、Merge()、Terminate(),在新建聚合类之后将自动生成,开发人员在其中编写相关代码,实现自定义聚合逻辑。这几个方法的作用分别是:

(1)Init():分组计算之前初始化各变量。在 LeqAvg 类中对有效个数变量_intCount、能量总和变量_dobTotal、返回值变量_dobResult 进行数值初始化。

(2)Accumulate():在每个分组中扫描每一条记录时执行,对组内数据进行累积计算。LeqAvg 类中定义了将“能量总和”累积到变量_dobTotal

中,将“有效个数”累积到变量_intCount 中。

(3) Merge():数据合并。

(4) Terminate():扫描结束,返回结果。将“能量总和”除以“有效个数”计算能量平均值,并返回作为该分组的计算结果,如果有效个数为0,则返回NULL值。

本实例中选用C#语言编写LeqAvg聚合类,完整源码如下:

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Data.SqlTypes;
using Microsoft.SqlServer.Server;
[Serializable]
[Microsoft.SqlServer.Server.SqlUserDefinedAggregate(Format.Native)]
public struct LeqAvg
{
    private double _dobResult; //等效声级平均值,函数返回值
    private double _dobTotal; //能量总和
    private int _intCount; //有效个数

    public void Init()
    { //初始化各个变量
        _dobResult = -1;
        _dobTotal = 0;
        _intCount = 0;
    }

    public void Accumulate(SqlDouble Value)
    { //累积计算
        if (Value.IsNull())
        { //聚合函数忽略NULL,跳过
            return;
        }
        else
        {
            _dobTotal = _dobTotal + Math.Pow(10, 0.1 * Convert.ToDouble(Value.Value));
            _intCount++;
        }
    }

    public void Merge(LeqAvg Group)
    { //合并
        _dobTotal = _dobTotal + Math.Pow(10, 0.1 * Group._dobTotal);
        _intCount = _intCount + Group._intCount;
    }

    public SqlDouble Terminate()
    { //结束,返回计算结果
    }
}
```

```
if (_intCount > 0)
{
    _dobResult = Math.Round(10 * Math.Log10(_dobTotal / _intCount), 1); //保留一位小数
    return new SqlDouble(_dobResult);
}
else
{ //如果有效个数为0,则返回NULL
    return SqlDouble.Null;
}

#endregion
```

3.3 部署程序集

3.3.1 注册程序集

为能够在SQL Server实例中使用SQL CLR用户自定义聚合函数,必须首先将聚合类编译成程序集并注册到SQL Server实例中。可使用“CREATE ASSEMBLY”语句进行注册:

```
CREATE ASSEMBLY [Hjjcpro.SqlServerProject] FROM 'D:\HJJCPRO\SqlServerProject\HjjcPro.SqlServerProject.dll'
WITH PERMISSION_SET = SAFE;
GO
```

其中[Hjjcpro.SqlServerProject]为程序集名称,“D:\HJJCPRO\SqlServerProject\HjjcPro.SqlServerProject.dll”为程序集dll文件所在的路径。

3.3.2 创建用户定义聚合函数

为了调用.NET方法,需要引用已注册的程序集,新建一个SQL Server自定义函数来访问指定的.NET方法。使用“CREATE AGGREGATE”语句创建引用已注册程序集的用户定义聚合函数,并在其内使用“EXTERNAL NAME”来通知SQL Server使用CLR功能,并指定程序集和方法名称。SQL创建代码如下:

```
CREATE AGGREGATE [dbo].[LeqAvg]
(@Value [float])
RETURNS [float]
EXTERNAL NAME [HjjcPro.SqlServerProject].[LeqAvg]
GO
```

4 计算实例

4.1 自定义聚合函数的基本用法

自定义聚合函数的使用方法和系统聚合函数完全一样,与SQL语法无缝结合。可使用以下SQL语句计算所有功能区环境噪声监测点位2009年1月份昼间等效声级日平均值:

```
SELECT 测点代码,测点名称,年,月,日,dbo. LeqAvg( lega )
AS 昼间平均等效声级
```

FROM 功能区环境噪声监测数据表

GROUP BY 测点代码,测点名称,年,月,日

WHERE 年 = 2009 AND 月 = 1 AND 时 BETWEEN 7 AND 22

在上述SQL语句中, GROUP BY语句实现灵活分组,而复杂的等效声级能量平均的计算过程则由自定义聚合函数 dbo. LeqAvg() 来完成。通过与SQL语法紧密结合,有机集成SQL灵活分组查询能力和CLR高效计算能力,将算法过程与分组查询调用相互分离,使自定义聚合函数一次编写,多处使用。

4.2 昼间、夜间等效声级计算

根据实际业务特点,为了方便分组查询,需要在SQL Server中建立辅助查询的标量值函数,用以增强查询功能,同时保持更高的可读性和简便性。

4.2.1 昼间夜间判定函数

建立标量值函数 dbo. F_Ldn(),用以根据小时判定昼间或者夜间,快速地对小时进行昼间和夜间的分组查询。SQL 创建语句为:

```
CREATE FUNCTION dbo. F_Ldn
(
    @intHour INT
)
RETURNS NVARCHAR(10)
AS
```

```
BEGIN
DECLARE @Rst NVARCHAR(10)
SELECT @Rst = CASE
WHEN @intHour BETWEEN 7 AND 22 THEN '昼间'
ELSE '夜间'
END
RETURN @Rst
END
```

备注:江苏省境内以6点和22点作为昼间与夜间的分界点。

4.2.2 季度判定函数

建立标量值函数 dbo. F_GetSeasonByMonth(),用以根据月份判定季度,快速地对月份进行季度分组查询。SQL 创建语句为:

```
CREATE FUNCTION dbo. F_GetSeasonByMonth
(
    @MON INT
)
RETURNS INT
AS
BEGIN
DECLARE @Season INT
SELECT @Season = CASE
WHEN (@MON <= 3 AND @MON >= 1) THEN 1
WHEN (@MON <= 6 AND @MON >= 4) THEN 2
WHEN (@MON <= 9 AND @MON >= 7) THEN 3
WHEN (@MON <= 12 AND @MON >= 10) THEN 4
END
RETURN @Season
END
```

备注:四个季度分别用数字1,2,3,4表示。

4.2.3 昼间夜间分组计算

原始数据表格式见表1。

表1 功能区环境噪声原始监测数据

STCODE (城市代码)	YE (年)	PONAME (测点名称)	POCODE (测点代码)	NDISC(功能区 代码1,2,3,4)	MON (月)	DA (日)	HOR (时)	LEQA (等效声级)
320100	2008	xxx 测点	100	4	2	13	1	66.2
320100	2008	xxx 测点	100	4	2	13	2	63.3
320100	2008	xxx 测点	100	4	2	13	3	60.4
.....								
320100	2008	xxx 测点	100	4	2	13	24	68.5
.....								

为实现按城市、季度、功能区计算噪声昼间、夜

间平均等效声级,须通过子查询逐级分组查询进行

汇总。噪声等效声级在一天范围内的昼间(夜间、昼夜)平均值应采用能量加权平均算法,其他级别汇总采取算术平均算法。SQL查询语句如下^[2]:

```
SELECT stcode, ye, season, dn, [1] AS [1类区], [2] AS [2类区], [3] AS [3类区], [4] AS [4类区] -- 行列转置
```

```
FROM
```

```
(
```

```
-- 计算各个功能区的平均值, 算术平均
```

```
SELECT stcode, ye, dbo.F_GetSeasonByMonth(mon) AS season, dn, ndisc, AVG(leqa) AS leqa
```

```
FROM
```

```
-- 计算各点位每天昼间、夜间等效声级能量平均值
```

```
SELECT stcode, ye, mon, da, pocode, poname, ndisc, dbo.F_Ldn(hor) AS dn, dbo.LeqAvg(leqa) AS leqa
```

```
FROM 功能区环境噪声监测数据表
```

```
WHERE ye = 2008
```

```
GROUP BY stcode, ye, mon, da, pocode, poname, ndisc, dbo.F_Ldn(hor)
```

```
) AS 各测点昼间夜间噪声日平均等效声级
```

```
GROUP BY stcode, ye, dbo.F_GetSeasonByMonth(mon),
```

```
dn, ndisc
```

```
) AS 各城市每季度昼间夜间平均
```

```
PIVOT
```

```
(
```

```
AVG(leqa) FOR ndisc IN([1], [2], [3], [4])
```

```
)
```

```
AS pvt
```

其中关键步骤为:

(1) 根据原始数据表“功能区环境噪声监测数据表”生成子查询“各测点昼间夜间噪声日平均等效声级”。

(2) 在子查询“各测点昼间夜间噪声日平均等效声级”的基础上,对城市、季度分组,生成子查询“各城市每季度昼间夜间平均”。

(3) 在子查询“各城市每季度昼间夜间平均”的基础上,使用 pivot 进行数据透视,生成所需要的数据格式。

最终将查询得到表 2。

表 2 各城市功能区环境噪声季度平均等效声级

STCODE (行政区代码)	YE (年)	SEASON (季度)	DN (昼、夜)	1类区	2类区	3类区	4类区
320100	2008	1	夜间	40.7	46.6	44.8	62.3
320100	2008	1	昼间	48.7	57.1	56.7	67.2
320100	2008	2	夜间	45.9	50.0	50.9	63.7
320100	2008	2	昼间	50.9	57.7	59.6	66.4
320100	2008	3	夜间	42.8	50.1	56.8	61.7
320100	2008	3	昼间	50.9	57.2	59.1	67.3
320200	2008	1	夜间	43.2	51.2	46.5	63.3
320200	2008	1	昼间	54.4	59.6	53.2	68.9
320200	2008	2	夜间	43.3	42.4	42.2	64.5
320200	2008	2	昼间	54.0	53.7	51.8	68.7
320200	2008	3	夜间	44.6	49.2	57.7	60.6
320200	2008	3	昼间	49.3	53.8	64.2	66.0

4.4 昼夜等效声级计算

昼夜平均等效声级计算过程与昼间、夜间平均等效声级略有不同,要求在计算之前将夜间原始监测值加上 10,再参与计算。SQL 查询语法为:

```
SELECT stcode, ye, '昼夜' dn, [1] AS [1类区], [2] AS [2类区], [3] AS [3类区], [4] AS [4类区] -- 行列转置
```

```
FROM
```

```
(
```

```
SELECT stcode, ye, dbo.F_GetSeasonByMonth(mon) AS sea-
```

```
son, ndisc, avg(leqa) AS leqa FROM
```

```
(
```

```
-- 2008 年各点位昼夜
```

```
SELECT stcode, pocode, ye, mon, da, ndisc, dbo.LeqAvg(leqa) AS leqa
```

```
FROM -- 各个功能区各个测点的月平均值, 能量平均
```

```
( -- 处理原始监测值, 夜间监测值 + 10
```

```
SELECT stcode, ye, mon, da, pocode, ndisc,
```

```
CASE(dbo.F_Ldn(hor))
```

```
WHEN '昼间' THEN leqa
```

```

WHEN '夜间' THEN leqa + 10
END AS leqa
FROM 功能区环境噪声监测数据表
WHERE ye = 2008 and dbo.F_GetSeasonByMonth(mon) = 1
) AS 处理之后的原始监测数据
GROUP BY stcode, ye, mon, da, pocode, ndisc
) AS 昼夜等效声级日平均
GROUP BY stcode, ye, dbo.F_GetSeasonByMonth(mon), ndisc
) 各城市每季度昼夜平均
pivot
(
    AVG(leqa) FOR ndisc IN([1],[2],[3],[4])
)
AS pvt

```

4.5 使用总结

昼间、夜间等效声级与昼夜等效声级拥有类似的逻辑计算过程,结合各自业务实际只须对SQL查询语句稍作调整,两者计算过程中繁琐的能量加权平均则统一使用自定义聚合函数dbo.LeqAvg()来完成。

SQL CLR 聚合函数封装自定义聚合逻辑,并能像 AVG、MIN、MAX 等系统聚合函数一样,结合 SQL 语法可以进行灵活的分组计算。通过对功能区噪声等效声级的实际应用计算,不难发现,开发使用 SQL CLR 聚合函数非常简便、高效、准确。并且真正做到了“一次编码,到处运行”,实现代码复用。

5 性能分析

在数据访问方面,Transact-SQL 比托管代码在性能方面通常会更好,然而在过程逻辑、复杂数据计算、字符串处理等方面,相比之下托管代码在性能方面具有决定性的优势^[3]。因此,可用 CLR 编写计算和逻辑密集的代码,以弥补 Transact-SQL 数据计算处理能力的不足。实践证明,在功能区噪声等效声级的 SQL CLR 聚合计算中,所有查询计算过程都在毫秒级时间内完成。

6 其他应用场景

借助 SQL CLR 编程技术,结合各类实际业务可以自由编写自定义聚合逻辑,因此 SQL CLR 聚

合函数的适用范围非常广。在环境质量业务数据处理中,还可以应用在以下场景中:

(1) 城市区域声环境质量评价:建立包含测点网格面积与测点等效声级监测值的用户自定义类型(UDTs),参数传递测点网格面积与测点等效声级监测值,累积网格总面积以及测点网格面积与监测值的乘积,返回网格面积加权平均等效声级。

(2) 道路交通声环境质量评价:建立包含监测路段长度与监测路段等效声级监测值的用户自定义类型,参数传递监测路段长度与监测路段等效声级监测值,累积路段总长度以及路段长度与监测值的乘积,返回监测路段长度加权平均等效声级。

(3) 酸雨 pH 雨量加权平均计算:建立包含降雨量与 pH 值的用户自定义类型,参数传递降雨量与 pH 值,累积氢离子(H⁺)含量和降雨量,返回氢离子(H⁺)加权平均浓度对应的 pH 值。

(4) 酸雨发生率统计:参数传递 pH 值,累积 pH 值总个数与 pH 值小于 5.6 的个数,返回 pH 值小于 5.6 所占比例,及酸雨发生率。

(5) 超标率统计:参数传递是否超标,累积测点总个数与超标测点个数,返回超标率。

7 结语

使用 SQL Server 2005 带来的 SQL CLR 聚合函数,能够突破 SQL Server 系统聚合函数的局限性,开发适用于各类环境质量业务数据聚合处理的计算方法。SQL CLR 聚合函数集成 SQL 数据快速访问、灵活分组能力,以及 CLR 复杂逻辑聚合高效计算能力,将大大提升数据分析自动化处理水平。

[参考文献]

- [1] GERT D,杨岩. CLR 核对清单——在你使用 CLR 以前需要考虑 7 件事[J]. Windows & Net Magazine:国际中文版,2005(6):71.
- [2] ITZIK B G,DEJAN S, ROGER W. Microsoft SQL Server 2005 技术内幕:T-SQL 程序设计[M]. 赵立东,译. 北京:电子工业出版社, 2007.
- [3] 谭恒松,程光华. SQL Server 2005 中的 CLR 应用研究[J]. 浙江工商职业技术学院学报, 2007,6(2):46-48.