

HJ

中华人民共和国国家环境保护标准

HJ□□□□-201□

环境信息系统数据库访问接口规范

Specification for database access interface of environment information system

(征求意见稿)

201□-□□-□□发布

201□-□□-□□实施

环 境 保 护 部 发 布

目 次

前 言.....	ii
1 适用范围.....	1
2 规范性引用文件.....	1
3 术语和定义.....	1
4 数据库访问接口框架.....	2
5 数据库访问接口方式.....	3
6 数据库访问工作流程.....	4
7 数据库访问条件.....	5
附录A （资料性附录） 触发器方式数据库接口的SQL脚本.....	6
附录B （资料性附录） 时间戳方式数据库接口的SQL脚本.....	11

前 言

为贯彻《中华人民共和国环境保护法》，逐步建立和完善环境信息化标准体系，促进环境信息化工作，明确关系型数据库的访问接口，推动环境信息资源建设，制定本标准。

本标准规定了环境信息系统中关系型数据库的访问接口，确定了数据库访问接口的基本架构以及数据库访问接口方式、数据库访问工作流程。

本标准附录 A、附录 B 为资料性附录。

本标准为首次发布。

本标准由环境保护部科技标准司组织制订。

本标准主要起草单位：环境保护部信息中心、深圳市环境保护宣传教育信息中心。

本标准由环境保护部 2□□□年□□月□□日批准。

本标准自 2□□□年□□月□□日起实施。

本标准由环境保护部解释。

环境信息系统数据库访问接口规范

1 适用范围

本标准规定了环境信息系统中关系型数据库的访问接口，确定了数据库访问接口的基本架构以及数据库访问接口方式、数据库访问工作流程。

本标准适用于同域内（同一局域网内）的数据库访问，以及环境信息系统的关系型数据库访问接口的建设。

2 规范性引用文件

本标准内容引用了下列文件或其中的条款。凡是不注明日期的引用文件，其有效版本适用于本标准。

GB 8567	计算机软件产品开发文件编制指南
GB/T 5271.1	信息技术 词汇 第1部分：基本术语
GB/T 5271.20	信息技术词汇 20部分：系统开发
GB/T 12118	数据处理词汇 21部分：过程计算机系统和技术过程间的接口
GB/T 20001.1	标准编写规则 第1部分：术语
GB/T 20001.2	标准编写规则 第2部分：符号
GB/T 20001.3	标准编写规则 第3部分：信息分类编码
HJ 511	环境信息化标准指南
HJ/T 212	污染源在线自动监控（监测）系统数据传输标准
HJ/T 352	环境污染源自动监控信息传输、交换技术规范（试行）
HJ/T 419	环境数据库设计与运行管理规范

3 术语和定义

下列术语和定义适用于本标准。

3.1 数据库 database

按照概念结构组织的数据的汇集，它描述这些数据的特征及与数据对应的实体间的关系，并支持一个或多个应用领域。

[GB/T 5271.1，定义 01.08.05]

3.2 数据库适配器 adapter

能够按指定要求和标准规范进行数据库访问的功能组件，可以通过对配置文件的解析和处理，完成数据库的访问，实现数据的采集和交换。

3.3 数据库访问接口 database access interface

嵌入在各个环保业务数据库中的，专门针对外部应用访问所做的配置。数据库访问接口与适配器相适应，能够实现外部应用对环保业务数据库内容的访问。

3.4 存储过程 stored procedure

一组为了完成特定功能的 SQL (Structured Query Language) 语句集合，经编译后存储在数据库里。

3.5 触发器 trigger

特殊的存储过程，它的执行不是由程序调用，也不是手工启动，而是由事件来触发，比如当对一个表进行插入、删除、更新操作 (insert, delete, update) 时就会激活它执行。

3.6 时间戳 timestamp

指明时间上某一点（相对于通用时标）的数据项。

4 数据库访问接口框架

在环境信息系统数据管理中，数据库访问接口用于获取同域内的各个业务应用的数据库内容。见图 1：

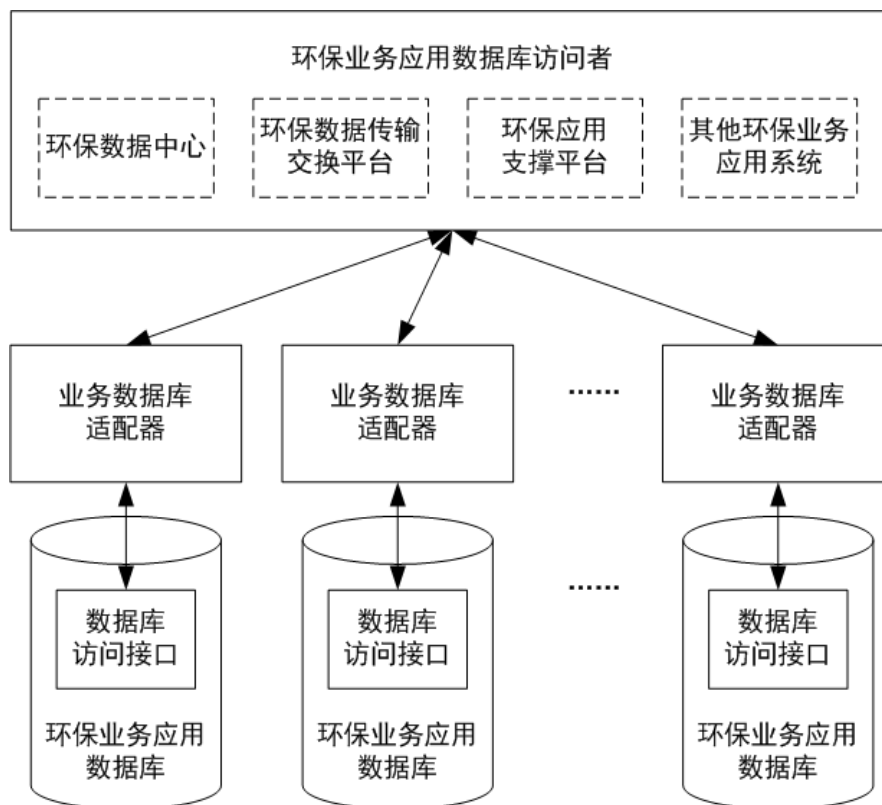


图 1 环境信息系统数据库访问接口框架

环保业务应用数据库的访问者通过各个业务数据库的适配器访问不同的业务应用数据库。业务数据库适配器与嵌入在业务应用数据库中的数据库访问接口进行交互，获取业务应用数据库的内容。

业务数据库适配器是环保业务应用数据库的访问者根据实际业务构建而成的用于访问应用数据库的功能组件。业务适配器在访问环保业务应用数据库时，需要按照指定的要求和标准规范进行数据库访问。

5 数据库访问接口方式

5.1 触发器方式的数据库访问接口

5.1.1 接口方式描述

触发器方式是指，在源数据库的业务应用数据表中添加触发器，监控数据表的新增、更新、删除操作。当操作成功时，触发相应的触发器将该记录的主键等信息插入到指定的访问接口中间表，然后适配器从访问接口中间表中获得业务数据内容的变更情况，从而达到实时的数据库增量访问效果。

5.1.2 接口配置方式

- a) 在被访问的业务应用数据库中建立一张访问接口中间表，用于存放相关触发器和临时交换记录。不同的业务系统临时交换记录都采用统一的中间表格式，具体包括：
 - T_Oth_Msg
- b) 创建一个对访问接口中间表做操作的存储过程，用于执行相关的触发操作，具体包括：
 - P_QueueMsg
- c) 针对要实时访问的数据表建立新增和修改、删除的触发器，用于触发操作，具体包括：
 - TR_AddTrigger_I
 - TR_UpdateTrigger_U
 - TR_DeleteTrigger_D

由于触发器方式只是对业务应用数据表进行操作监控，且只对访问接口中间表进行操作，因此不会对被监控的业务应用数据表造成任何影响。

中间表的格式内容和存储过程参见附录 A。

5.2 时间戳方式的数据库访问接口

5.2.1 接口方式描述

时间戳方式是指，在数据表中配置时间字段，定时轮询数据库是否发生新增或更新的操作，以达到数据增量同步的作用。时间戳方式的适配器将上次执行的时间与当前数据表中的时间字段进行比较，以判别哪些记录在上一个时间周期内发生了新增或更新的操作，从而达到访问数据的目的。

5.2.2 接口配置方式

- a) 在被访问的业务应用数据表中，创建用于比较的时间戳字段：Create_Time、Update_Time。
- b) 每个被访问的业务应用数据表中都必须有两个用于轮询的时间字段作为时间戳。一个用于记录创建时间，另一个用于记录修改时间：Create_Time、Update_Time。时间戳字段的数据类型为业务应用数据库的时间类型，格式必须为：年月日 时分秒（例如：2009-8-20 12: 00: 00）。

每次查询数据表时，对表中所有的数据按时间进行遍历查询，取出最新的记录。这样如果源数据库中数据过多，进行定时轮询时就会对交换的性能造成影响。

填入时间戳的方式有多种，如通过数据库结构化查询语句（Structure Query Language）

填入值、触发器方式自动填入值等。建议采取建立触发器方式，自动获取当前操作时间并填入到对应的字段中。时间戳触发器编写方式参见附录 B。

6 数据库访问工作流程

6.1 数据库触发器访问方式

环保业务应用数据库访问根据触发时间条件分成实时访问和定时访问两种形式。

环保业务应用数据库的实时访问是指当业务应用数据内容发生变化（包括新增、更新、删除等操作）时，实时地触发业务数据库适配器，对业务应用数据库内容进行访问，实时获取业务数据的变化情况。

环保业务应用数据库的实时访问采用触发器方式的数据库访问接口。具体流程见图 2：

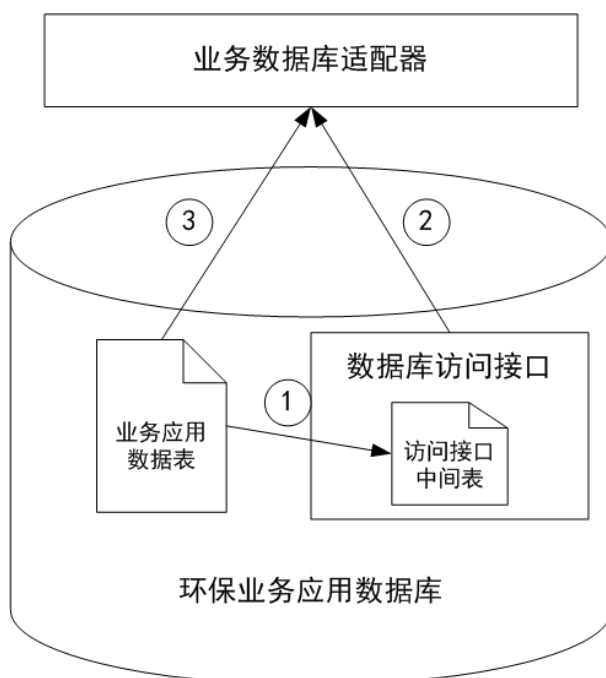


图 2 数据库触发器访问流程

- 业务应用数据表的内容在执行相关操作的时候，通过触发器将受影响的记录直接推送到数据库访问接口的中间表中。如果受影响记录的字段数未超过中间表的字段数，则将受影响记录的所有信息推送。
- 业务数据库适配器从访问接口中间表中获得业务数据内容的变更情况。
- 当受影响的记录字段数超过中间表的字段数时，仅将受影响记录的主键等唯一标识信息推送到中间表。业务数据库适配器可以根据中间表中的数据内容，直接访问业务应用数据表，获得业务应用数据内容。

6.2 数据库时间戳定时访问方式

环保业务应用数据库的定时访问是指业务数据库适配器按照预定的时间周期，定时访问业务应用数据库，并根据业务应用数据表的时间戳，获取上个时间周期里的业务应用数据库所发生的变化内容（包括新增、更新、删除等操作）。

环保业务应用数据库的定时访问通常采用时间戳方式的数据库访问接口。具体流程见图

3:

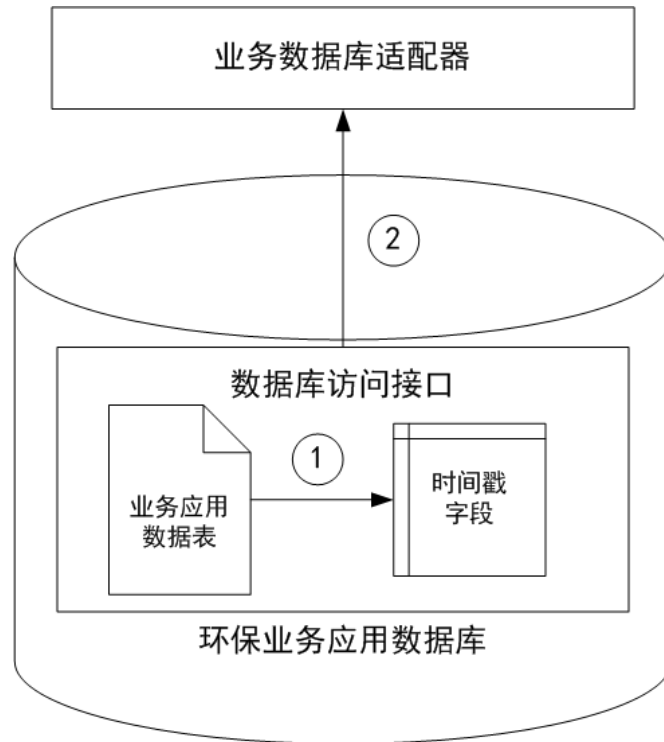


图3 数据库时间戳定时访问流程

- a) 业务应用数据表在新增、更新的同时，更新时间戳字段的内容；
- b) 业务数据库适配器定时访问业务应用数据表，根据时间戳字段所记录的新增和更新时间戳，获取上个时间周期里的业务应用数据库的变化内容。

7 数据库访问条件

对某一数据库进行访问时，在网络连通的情况下，需具备以下条件：

- a) 获取数据库管理系统（DBMS）类型；
- b) 获取被访问的数据库所在服务器的网络地址；
- c) 需要获取数据库的用户名及口令。

附录A

(资料性附录)

触发器方式数据库接口的SQL脚本¹

A.1 触发器方式数据库接口的中间表创建脚本

```
CREATE TABLE [dbo].[T_Oth_Msg](
    [PK_MsgId] [numeric](28, 0) IDENTITY(1,1) NOT NULL,
    [MsgQueuedAt] [datetime] NOT NULL,
    [MsgStatus] [varchar](10) COLLATE Chinese_PRC_CI_AS NOT NULL,
    [MsgDeliveredAt] [datetime] NULL,
    [MsgFailedText] [varchar](255) COLLATE Chinese_PRC_CI_AS NULL,
    [OperationType] [varchar](10) COLLATE Chinese_PRC_CI_AS NULL,
    [MsgParam1] [varchar](255) COLLATE Chinese_PRC_CI_AS NULL,
    [MsgParam2] [varchar](255) COLLATE Chinese_PRC_CI_AS NULL,
    [MsgParam3] [varchar](255) COLLATE Chinese_PRC_CI_AS NULL,
    [MsgParam4] [varchar](255) COLLATE Chinese_PRC_CI_AS NULL,
    [MsgParam5] [varchar](255) COLLATE Chinese_PRC_CI_AS NULL,
    [MsgParam6] [varchar](255) COLLATE Chinese_PRC_CI_AS NULL,
    [MsgParam7] [varchar](255) COLLATE Chinese_PRC_CI_AS NULL,
    [MsgParam8] [varchar](255) COLLATE Chinese_PRC_CI_AS NULL,
    [MsgParam9] [varchar](255) COLLATE Chinese_PRC_CI_AS NULL,
    [MsgParam10] [varchar](255) COLLATE Chinese_PRC_CI_AS NULL,
) ON [PRIMARY]
```

GO

表字段说明:

字段名	字段中文名	应用意义
PK_MsgId	临时记录 id	中间表的主键
MsgQueuedAt	临时记录推送的时间	临时记录从中间表中推送的时间
MsgStatus	临时记录的状态	标识临时记录是否为待推送的信息
MsgDeliveredAt	临时记录推送到目的地的时间	临时记录推送到目的地的时间
MsgFailedText	临时记录推送失败的提示信息	临时记录推送失败的详细信息

¹ SQL 脚本的数据库参考类型为 SqlServer 数据库。¹

OperationType	临时记录的操作类型	临时记录在业务数据表中执行的操作（新增、更新、删除）
MsgParam1	临时记录信息	临时记录的具体信息，来源于存储过程中的参数值
MsgParam2	临时记录信息	临时记录的具体信息，来源于存储过程中的参数值
MsgParam3	临时记录信息	临时记录的具体信息，来源于存储过程中的参数值
MsgParam4	临时记录信息	临时记录的具体信息，来源于存储过程中的参数值
MsgParam5	临时记录信息	临时记录的具体信息，来源于存储过程中的参数值
MsgParam6	临时记录信息	临时记录的具体信息，来源于存储过程中的参数值
MsgParam7	临时记录信息	临时记录的具体信息，来源于存储过程中的参数值
MsgParam8	临时记录信息	临时记录的具体信息，来源于存储过程中的参数值
MsgParam9	临时记录信息	临时记录的具体信息，来源于存储过程中的参数值
MsgParam10	临时记录信息	临时记录的具体信息，来源于存储过程中的参数值

A. 2 触发器方式数据库接口的存储过程脚本

```

create proc [dbo].[P_QueueMsg]
    @OperationType    varchar(10),
    @MsgParam1        varchar(255) = NULL,
    @MsgParam2        varchar(255) = NULL,
    @MsgParam3        varchar(255) = NULL,
    @MsgParam4        varchar(255) = NULL,
    @MsgParam5        varchar(255) = NULL,
    @MsgParam6        varchar(255) = NULL,
    @MsgParam7        varchar(255) = NULL,
    @MsgParam8        varchar(255) = NULL,
    @MsgParam9        varchar(255) = NULL,
    @MsgParam10       varchar(255) = NULL
as

```

```

BEGIN
    declare @ErrorCode int, @ErrorMessage varchar(255), @NumRows int
    select @ErrorCode = 0
    insert T_Oth_Msg ( MsgQueuedAt, MsgStatus, MsgDeliveredAt,
MsgFailedText, OperationType, MsgParam1, MsgParam2, MsgParam3, MsgParam4, MsgParam5,
MsgParam6, MsgParam7, MsgParam8, MsgParam9, MsgParam10 ) select getdate(), 'NEW',
NULL, NULL, @OperationType, @MsgParam1, @MsgParam2, @MsgParam3, @MsgParam4,
@MsgParam5, @MsgParam6, @MsgParam7, @MsgParam8, @MsgParam9, @MsgParam10
    select @ErrorCode = @@error, @NumRows = @@rowcount return @ErrorCode
END

GO

```

存储过程参数设置说明：

参数名	参数含义
MsgQueuedAt	临时记录从中间表中推送的时间
MsgStatus	标识临时记录是否为待推送的信息
MsgDeliveredAt	临时记录推送到目的地的时间
MsgFailedText	临时记录推送失败的详细信息
OperationType	临时记录在业务数据表中执行的操作 (新增、更新、删除)
MsgParam1	业务数据表中受影响记录的具体信息
MsgParam2	业务数据表中受影响记录的具体信息
MsgParam3	业务数据表中受影响记录的具体信息
MsgParam4	业务数据表中受影响记录的具体信息
MsgParam5	业务数据表中受影响记录的具体信息
MsgParam6	业务数据表中受影响记录的具体信息
MsgParam7	业务数据表中受影响记录的具体信息
MsgParam8	业务数据表中受影响记录的具体信息
MsgParam9	业务数据表中受影响记录的具体信息
MsgParam10	业务数据表中受影响记录的具体信息

A.3 触发器方式数据库接口的触发器脚本

```

CREATE TRIGGER TR_AddTrigger_I ON [dbo].[业务应用数据表名] FOR insert AS
BEGIN
    declare [表字段 1, 表字段 2, 表字段 3...]
    declare cursor_insert cursor for select "表主键" from inserted

```

```

open cursor_insert
fetch cursor_insert into @表字段 1, @表字段 2, @表字段 3 ...
while @@fetch_status=0
BEGIN
    exec P_QueueMsg @OperationType='insert', @MsgParam1=@表字段 1, @MsgParam2=@
    @表字段 2, @MsgParam3=@表字段 3 ...
    fetch cursor_insert into @表字段 1, @表字段 2, @表字段 3 ...
END
close cursor_insert
deallocate cursor_insert
END

```

GO

```

CREATE TRIGGER TR_UpdateTrigger_U ON [dbo].[表名] FOR update AS
BEGIN
    declare [表字段 1, 表字段 2, 表字段 3...]
    declare cursor_insert cursor for select "表主键" from inserted
    open cursor_insert
    fetch cursor_insert into @表字段 1, @表字段 2, @表字段 3 ...
    while @@fetch_status=0
    BEGIN
        exec P_QueueMsg @OperationType='update', @MsgParam1=@表字段 1, @MsgParam2=@
        @表字段 2, @MsgParam3=@表字段 3 ...
        fetch cursor_insert into @表字段 1, @表字段 2, @表字段 3 ...
    END
    close cursor_insert
    deallocate cursor_insert
END

```

GO

```

CREATE TRIGGER TR_DeleteTrigger_D ON [dbo].[表名] FOR delete AS
BEGIN
    declare [表字段 1, 表字段 2, 表字段 3...]
    declare cursor_insert cursor for select "表主键" from deleted
    open cursor_insert
    fetch cursor_insert into @表字段 1, @表字段 2, @表字段 3 ...

```

```
while @@fetch_status=0
BEGIN
    exec P_QueueMsg @OperationTypeName = 'delete', @MsgParam1= @表字段 1, @MsgParam2=
@表字段 2, @MsgParam3= @表字段 3 ...
    fetch cursor_insert into @表字段 1, @表字段 2, @表字段 3 ...
END
close cursor_insert
deallocate cursor_insert
END
```

触发器参数设置说明：

触发器中使用到的参数与存储过程、中间表中使用的字段含义一致。

触发器方式的数据库访问接口由中间表、存储过程、触发器之间相互配合完成，具体流程为：

由触发器对业务数据表进行实时监控。当业务数据表有变化时，触发触发器对应的触发器对发生的变化进行捕获，并调用相应的存储过程，将发生的变化信息存储到中间表中。

附录B
(资料性附录)
时间戳方式数据库接口的SQL脚本²

时间戳方式数据库接口的触发器脚本。

```
CREATE TRIGGER 触发器名  
    ON 表名  
    FOR [INSERT | UPDATE ]  
    AS Update 表名 set [create_time | update_time ] =getdate();
```

² SQL 脚本的数据库参考类型为 SqlServer 数据库。²