

УДК 004.4

В.А. Павлій, асистент
Донецкий национальный технический университет, г. Донецк, Украина
pavliy@cs.dgtu.donetsk.ua

Создание регионального data-центра как основы информационной системы экологического мониторинга в областях Украины

Предложена информационная модель регионального data-центра для хранения и обобщения данных субъектов экологического мониторинга атмосферы. Рассмотрены практические аспекты его реализации: структура БД, механизм репликации, модули ввода-вывода.

Информационная модель, data-центр, PCRE шаблоны, база данных, репликация данных, SQL, протоколы передачи данных, АСОИЗА, автоматизированный пост мониторинга атмосферного воздуха

Введение

Оценка действующих сетей мониторинга атмосферы, проводимая регулярно независимыми экспертами ООН, показывает, что в ряде стран Восточной Европы их качество не соответствует международным стандартам. В рамках совещаний Рабочей группы по мониторингу окружающей среды ЕЭК ООН были разработаны рекомендации [1-4] по модернизации действующих в этих странах сетей мониторинга. Анализ данных рекомендаций, выполненный в работе автора [5], показал, что наиболее значимыми из них с технической точки зрения являются внедрение автоматизированных систем контроля и создание единых data-центров, предназначенных для сбора, хранения и обобщения данных наблюдений.

На Украине модернизация действующей сети мониторинга атмосферы с учетом данных рекомендаций продвигается достаточно слабо. Связано это, в первую очередь, с ограничением финансирования, и, как следствие, практически полным отсутствием программно-технических решений автоматизированных систем контроля. Несмотря на это, прототипы подобных систем разрабатываются, и на сегодняшний день станции автоматизированного сбора данных действуют в г. Донецке, г. Запорожье и г. Кривой Рог. Для приема и систематизации данных, поступающих от этих станций в режиме реального времени, необходимо создание региональных data-центров.

Выполненный в работах автора [5-7] анализ действующих систем мониторинга и методики сбора данных о состоянии атмосферы показал, что в большинстве случаев эти данные хранятся на бумажных носителях у субъектов мониторинга, либо распространяются в виде файлов формата АСОИЗА, описание которого рассмотрено в руководящем документе [8]. По мнению многих авторов [9-10] в настоящее время этот формат устарел и не может использоваться в дальнейшем.

Однако, поскольку на сегодняшний день имеется достаточно большое количество информации, собираемой субъектами мониторинга на Украине с 2000 года (а в ряде стран еще раньше), и вся она хранится именно в формате АСОИЗА, поддержка данного формата при проектировании единого data-центра становится очень желательной.

Одной из сопутствующих рекомендаций ЕЭК ООН для систем мониторинга регионального уровня, является их возможность моделирования распространения примеси [5]. В дополнение к этому на государственном уровне также рассматриваются законодательно закрепленные типовые задачи мониторинга, основными из которых являются поддержка принятия решений, информирование населения и т.п. Исходя из этого, в составе типовой системы мониторинга регионального уровня обязательно должна быть подсистема моделирования и прогнозирования, которая решает задачу прогнозирования в полном объеме и в значительной степени облегчает принятие административных решений. Создание подобной подсистемы может быть выполнено только после анализа накопленной информации и поиска закономерностей рассеивания примесей в атмосфере. Это подтверждает необходимость обобщения данных субъектов мониторинга и представления их в виде регионального data-центра.

В соответствии с международными рекомендациями к построению информационной системы экологического мониторинга [1-4] можно выделить следующие стадии автоматизированной обработки экологической информации:

- прием и программный разбор (парсинг) данных наблюдений субъектов мониторинга, поступающих в региональный центр данных по одному из трех наиболее распространенных способов: импорт данных из файлов в формате АСОИЗА, получение данных в реальном времени от автоматизированных постов, ручной ввод;

- обобщение, систематизация, сохранение данных в базе данных предложенной структуры;
- прогноз распространения примеси в соответствии с алгоритмами, полученными на этапе анализа пространственно-временных закономерностей с использованием опытных данных субъектов мониторинга;
- визуализация и публикация в режиме on-line результатов прогноза на экологических сайтах или общественных информационных порталах;
- репликация экологической информации на сервера верхнего уровня с использованием произвольных XML-схем передачи данных.

Постановка задачи

Основной задачей данной работы является создание информационной модели регионального data-центра, который может стать основой для разработки региональной интегрированной информационной системы экологического мониторинга атмосферы.

Информационная модель регионального data-центра

Информационная модель регионального data-центра построена в соответствии с архитектурой "клиент-сервер" и использует протокол HTTP для обмена данными. Серверная часть включает СУРБД и прикладное ПО сервера, предназначенное для приема данных от клиента и представленное в виде набора программных модулей. Помимо этого, серверная часть включает модули, предназначенные для репликации данных на сервера государственного уровня, и модули экспорта данных.

На рис. 1 показана структурная схема информационной модели регионального data-центра. Необходимым минимумом для создания подобного data-центра является СУРБД, а также базовые модули ввода-вывода, обеспечивающие интеграцию с основными форматами исходных данных.

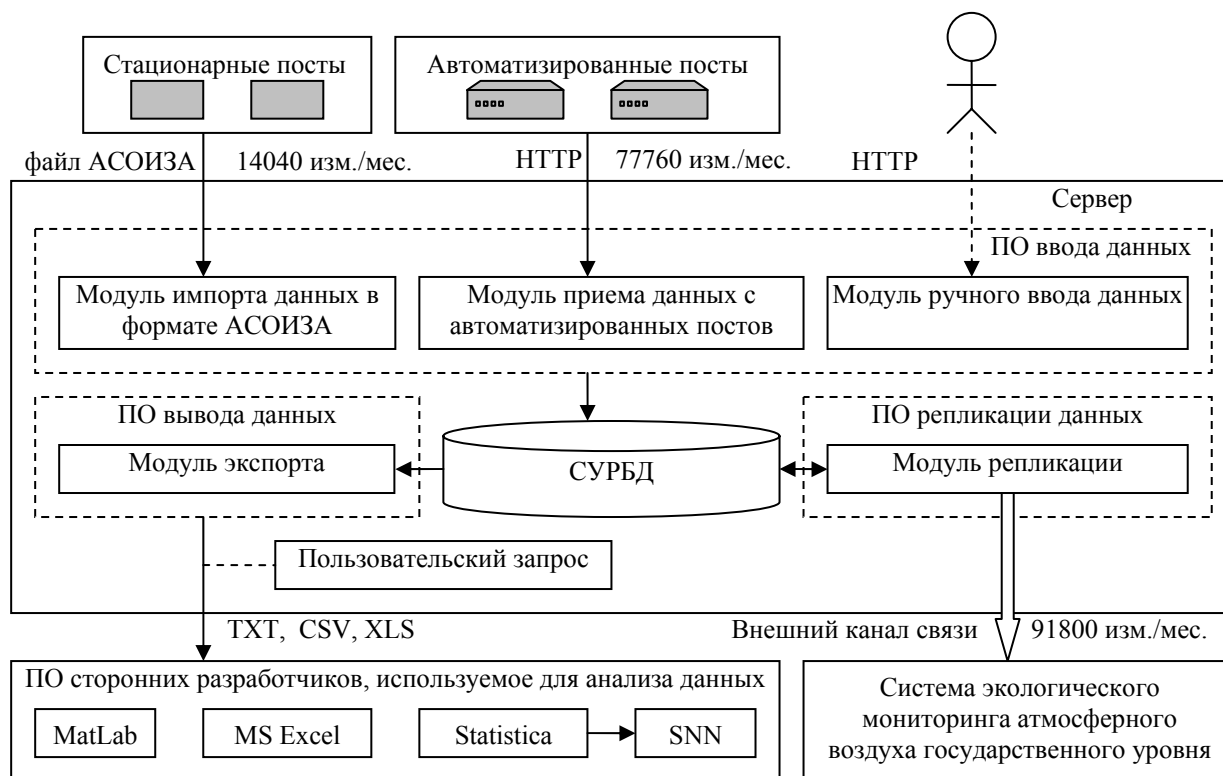


Рисунок 1 – Структурная схема информационной модели регионального data-центра

Способы приема данных наблюдений

Прием данных наблюдений осуществляется сервером по одному из трех способов передачи данных, каждый из которых основан на использовании протокола HTTP.

1) Передача данных от стационарных постов осуществляется путем импорта файлов в

формате АСОИЗА непосредственно на сервер при помощи web-интерфейса. В этом случае клиентом является браузер пользователя, сама передача осуществляется при помощи POST запроса.

Файл в формате АСОИЗА (см. рис. 2) представляет собой текстовый файл, в котором хранятся данные наблюдений, измеряемые с 6-ти часовым интервалом, общая продолжительность наблюдений составляет 1 месяц. Указанные

данные представлены в виде набора дискретных временных рядов и могут содержать как значения концентрации примесей, так и метеорологические параметры. Обычно, каждый файл АСОИЗА содержит наблюдения всех стационарных постов административно-территориальной единицы (города, района и т.д.) Файлы формата АСОИЗА формируются одноименной программой, которая была создана в начале 80-х годов и получила распространение среди субъектов экологического мониторинга в странах бывшего СССР. Структура файла АСОИЗА включает нескольких разделов, каждый из которых состоит из секции заголовка

(т.н. призначный блок) и секции данных. В секции заголовка содержится служебная информация о номере поста, его координатах, высоте над уровнем моря и т.п. Помимо этого, в призначном блоке хранится год и месяц наблюдений. В секции данных хранятся номер вещества в соответствии с общепринятым классификатором, временная метка, измеренные субъектами мониторинга значения метеорологических показателей и концентрации загрязнителей. Более подробное описание формата АСОИЗА приведено в руководящем документе [8].

```
##1100 010240 481378 070624 150502
02175 07000 01000 02037 04003 05005 19004 :: -0550 28047 ::
02172 13000 19004 22000 :: ::
02176 19000 01000 02076 04002 05010 19002 22000 :: -0530 18067 ::
02182 01000 19002 22000 :: ::
02185 07000 01000 02020 04004 05002 19002 :: -0160 19050 ::
02182 13000 19004 22000 :: ::
```

— призначный блок
} секция данных

Рисунок 2 – Пример файла в формате АСОИЗА

Для импорта файлов АСОИЗА в модели используется одноименный программный модуль, основное назначение которого заключается в синтаксическом анализе формата, программном разборе (парсинге) данных и занесении их в базу данных. Ввиду того, что описание формата является общедоступным и несложным, для синтаксического анализа и программного разбора в равной степени могут применяться как традиционные функции обработки строк, так и более сложные приемы, например, регулярные выражения [11]. Использование регулярных выражений позволяет существенно упростить исходный код модуля. В последнем случае для выделения всех призначных блоков (см. рис. 2) может использоваться следующий PCRE-шаблон:

```
/\#\#11 # признак начала блока
([0-9]{2}) # год (2 последние цифры)
\s # пробельный символ
([0-9]{2}) # месяц (2 цифры)
[0-9]{4} # высота станции
\s # пробельный символ
[0-9]{6} # номер метеостанции
\s # пробельный символ
([0-9]{2}) # общее число постов
([0-9]{2}) # общее число примесей
([0-9]) # код типа поста
[0-9] # знак координат поста
\s # пробельный символ
[0-9]{4} # координаты поста
([0-9]{2}) # номер поста
(.*) # секция данных
/x
```

В терминологии языка PCRE выражения в круглых скобках соответствует т.н. карманам, в которые записываются найденные подстроки, число в фигурных скобках задает количество повторений ранее идущей последовательности символов, а квадратные скобки представляют собой оператор альтернативы, который позволяет

выбирать любые значения из указанного набора. В представленном выше шаблоне, далеко не все выражения заключены в круглые скобки, что позволяет отсеять часть служебной информации призначного блока, которая впоследствии при занесении данных в БД использоваться не будет. Подобные выражения используются и для разбора секции данных.

2) Передача данных автоматизированных постов осуществляется путем подачи GET или POST запроса к серверу. В этом случае клиентом является ПО автоматизированного поста. Каждый пост представляет собой программно-аппаратную систему, которая включает автоматический газоанализатор и метеостанцию, миниЭВМ, а также необходимое ПО. Сопряжение миниЭВМ и измерительного оборудования осуществляется с использованием стандартных интерфейсов COM и USB. Основной задачей ПО автоматизированного поста является опрос датчиков газоанализатора и метеостанции, начальная обработка принимаемых данных, формирование HTTP запроса и отправка данных на сервер. В случае отсутствия доступа к сети Интернет по техническим причинам программное обеспечение может накапливать наблюдаемые значения и осуществлять передачу, как только доступ будет восстановлен.

Поскольку передача данных по протоколу HTTP является открытой, в целях повышения безопасности вместе с данными должны передаваться также данные авторизации клиента. Программное обеспечение сервера со своей стороны должно идентифицировать клиента и проверить корректность данных перед занесением их в базу. Для выполнения указанных операций в модели используется модуль приема данных с автоматизированных постов (см. рис. 1).

Таким образом, для передачи данных от автоматизированного поста по протоколу HTTP может использоваться GET запрос следующего вида:

```
http://akiam.org.ua/writedata.php?  
user=_&password=_&idstation=_  
&date=_&value[]=_&value[]=_
```

В данном запросе обозначены следующие параметры:

user	– имя пользователя;
password	– пароль для авторизации;
idstation	– идентификатор станции;
date	– дата и время измерений;
value[]	– идентификатор загрязнителя (метеопказателя) и значение измерения, разделенные пробелом.

ПО автоматизированного поста заменяет символы '_' на реальные значения передаваемых параметров. Все параметры приведенного выше запроса соответствуют общепринятой форме, за исключением последнего, в имени которого присутствуют символы '['', '''. Интерпретаторы некоторых серверных языков при разборе строки запроса (QUERY STRING) трактуют подобное выражение как элемент автомассива, ключи которого проставляются автоматически. Таким образом, существует возможность передавать в одном запросе сразу несколько измерений.

В случае длительного отсутствия доступа к Интернет автоматизированный пост, учитывая высокую частоту проводимых измерений, может накопить существенный объем данных. При возобновлении связи для передачи данных на сервер вместо GET предпочтительнее применить POST запрос, который позволяет передавать на сервер данные в бинарном виде, не имеет ограничений на общую длину запроса, в то время как длина GET запроса в некоторых web-серверах (например, Apache) определяется конфигурацией сервера и может быть ограничена [12].

3) Ввод данных в ручном режиме при помощи модуля ручного ввода. Данный модуль также имеет web-интерфейс и является частью панели администрирования системы. Поскольку ручной ввод данных требует существенных трудозатрат, используется он достаточно редко, что отмечено на рис. 1 штрихпунктирной линией.

Проектирование структуры БД регионального data-центра

Для хранения данных наблюдений за состоянием атмосферы может использоваться любая БД с поддержкой пространственных расширений. Это необходимо для сохранения месторасположения постов, картографических объектов, а также для анализа их взаимного

расположения с использованием SQL запросов. На рис. 3 показана структура базы данных регионального data-центра.

Предлагаемая структура базы данных включает 9 таблиц, которые представляют собой необходимый минимум для сохранения экологической информации. Из указанных таблиц только три являются основными таблицами первого уровня, остальные таблицы являются справочными, и связаны с основными таблицами реляционными отношениями типа $[1 \rightarrow \infty]$. В случае дальнейшего развития информационной системы экологического мониторинга указанная структура может быть дополнена рядом таблиц, предназначенных для сохранения промежуточных результатов анализа экологической информации или параметров системы.

Таблица *air_dimension* является основной таблицей БД и может быть использована как для сохранения данных наблюдений за состоянием атмосферного воздуха, так и метеорологических показателей. Структура таблицы включает 5 полей и хранит временную метку и значение наблюдения. Оставшиеся два поля (исключая первичный ключ) образуют реляционные связи со справочными таблицами, предназначенными для хранения информации о постах и ингредиентах. Помимо первичного ключа данная таблица содержит уникальный индекс по полям *stationid*, *substanceid*, *date*, обычный индекс по полю *stationid*, а также обычный индекс по полю *substanceid*. Наличие уникального индекса в таблице *air_dimension* не позволяет создавать дубликаты записей (например, при случайном повторном импорте данных), а наличие обычных индексов ускоряет выполнение многих SQL запросов, например, группировки данных по типу ингредиента. Для создания уникального и обычных индексов используются следующие SQL запросы:

```
CREATE UNIQUE INDEX unique_data_ind  
ON air_dimension  
(stationid, substanceid, date)  
  
CREATE INDEX stationid_ind  
ON air_dimension  
(stationid)  
  
CREATE INDEX substanceid_ind  
ON air_dimension  
(substanceid)
```

Предлагаемая структура базы данных включает 9 таблиц, которые представляют собой необходимый минимум для сохранения экологической информации. Из указанных таблиц только три являются основными таблицами первого уровня, остальные таблицы являются справочными, и связаны с основными таблицами реляционными отношениями типа $[1 \rightarrow \infty]$. В

случае дальнейшего развития информационной системы экологического мониторинга указанная структура может быть дополнена рядом таблиц,

предназначенных для сохранения промежуточных результатов анализа экологической информации или параметров системы.

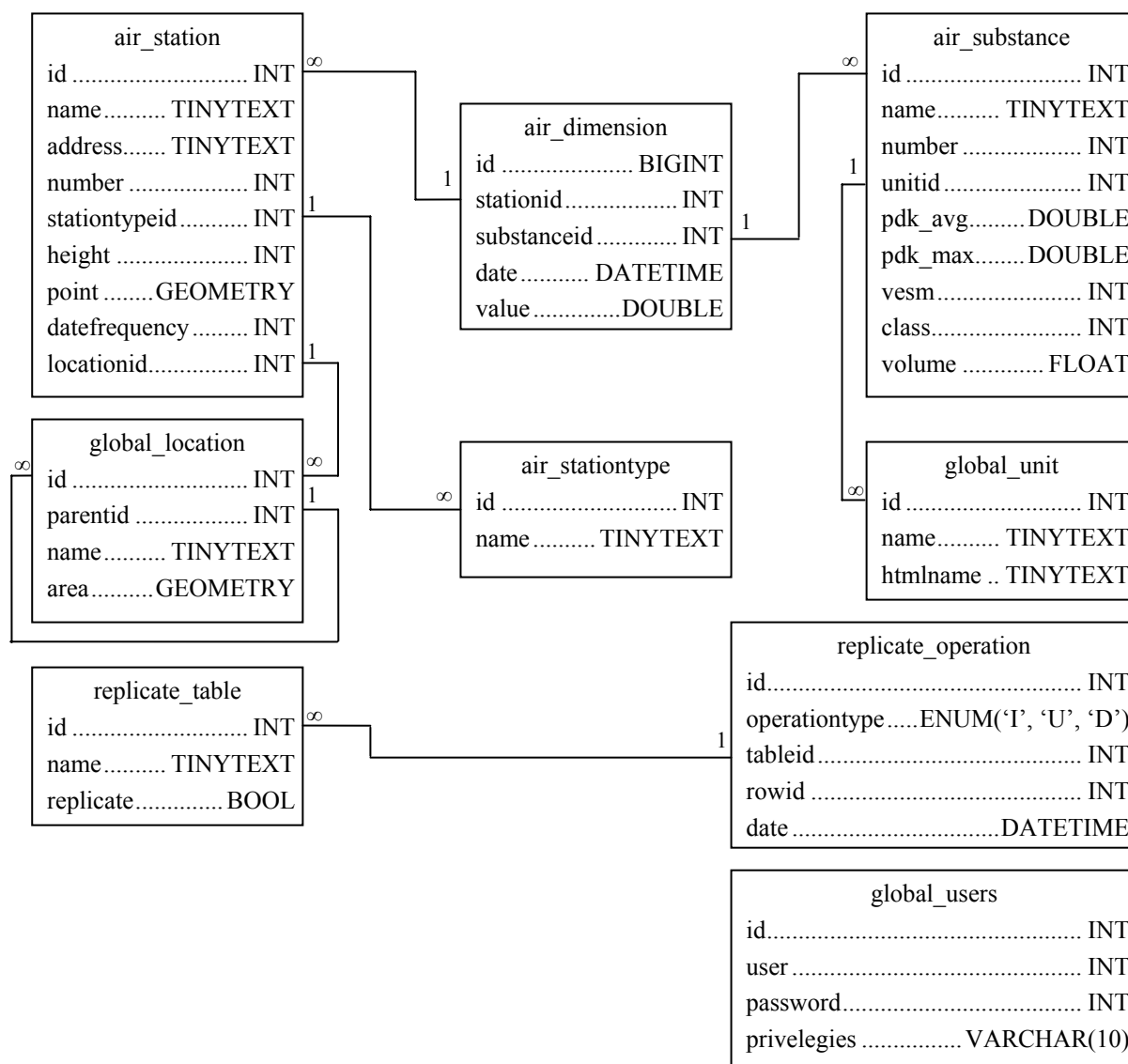


Рисунок 3 – Структура БД регионального data-центра

Таблица *air_dimension* является основной таблицей БД и может быть использована как для сохранения данных наблюдений за состоянием атмосферного воздуха, так и метеорологических показателей. Структура таблицы включает 5 полей и хранит временную метку и значение наблюдения. Оставшиеся два поля (исключая первичный ключ) образуют реляционные связи со справочными таблицами, предназначенными для хранения информации о постах и ингредиентах. Помимо первичного ключа данная таблица содержит уникальный индекс по полям *stationid*, *substanceid*, *date*, обычный индекс по полю *stationid*, а также обычный индекс по полю *substanceid*. Наличие уникального индекса в таблице *air_dimension* не позволяет создавать

дубликаты записей (например, при случайном повторном импорте данных), а наличие обычных индексов ускоряет выполнение многих SQL запросов, например, группировки данных по типу ингредиента. Для создания уникального и обычных индексов используются следующие SQL запросы:

```

CREATE UNIQUE INDEX unique_data_ind
ON air_dimension
(stationid, substanceid, date)
CREATE INDEX stationid_ind
ON air_dimension
(stationid)
CREATE INDEX substanceid_ind
ON air_dimension
(substanceid)
  
```

В системах мониторинга регионального уровня, для которых число автоматизированных постов лежит в пределах от 2 до 6, ежегодный прирост данных может составлять от миллиона и более измерений. Поскольку каждое измерение фактически представляет собой запись в таблице *air_dimension*, тип данных для поля, хранящего идентификатор записи (*id*), выбран достаточно большой (BIGINT). Ввиду того, что одна запись имеет размер 32 байта (типы данных BIGINT, DOUBLE, DATETIME занимают по 8 байт, а INT – 4 байта), физический эквивалент ежегодного прироста данных может составлять более 30 Мб без учета дополнительного объема, необходимого для хранения индексов.

Таблица *air_station* является справочной таблицей, которая предназначена для хранения информации о постах мониторинга. Данная информация включает тип поста, его имя и адрес, частоту поставки данных, а также координаты поста мониторинга, переведенные в декартово представление относительно начальной точки. Для задания координат используется точечный объект пространственного расширения. В случае, если сервер БД поддерживает пространственные расширения в соответствии со спецификацией OpenGIS, допустимо использовать абстрактный тип данных GEOMETRY, который предназначен для хранения произвольных геометрических объектов, в том числе и точечных [13]. Кроме того, поля этой таблицы образуют реляционные связи с двумя таблицами, которые хранят информацию о типе поста и локации, в которой он расположен.

Таблица *air_substance* также является справочной таблицей и предназначена для хранения информации о наблюдаемой примеси. Данная таблица содержит 9 полей, в которых хранится атрибутивная информация примеси: название, номер по классификатору АСОИЗА, максимально разовое и среднесуточное значение ПДК, точность измерения, класс опасности, а также минимальный объем пробы воздуха, необходимого для проведения измерения. Кроме того, имеется одна реляционная связь с таблицей нижнего уровня, которая хранит названия единиц измерения.

Таблицы *air_stationtype* и *global_unit* предназначены для хранения названий типа поста и единиц измерения. Дополнительных индексов (за исключением первичного ключа) указанные таблицы не имеют.

Таблица *global_location* используется для хранения картографических объектов, задаваемых с использованием пространственных расширений БД и представленных в виде геометрических примитивов. Данные объекты описываются в виде многоуровневой древовидной структуры. Взаимосвязь между элементами разных уровней

задается при помощи обратной реляционной связи, образованной полями *id* и *parentid*. Ввиду отсутствия родительских элементов в элементах самого верхнего уровня поле *parentid* имеет значение *null*. Поле *area*, имеющее тип GEOMETRY, необходимо для пространственного представления объекта. Применение древовидной структуры обусловлено необходимостью учета административно-территориальных отношений для объектов. Это необходимо, например, при оценке соответствия количества постов некоторой территории международным нормам. Так, для определения количества постов на территории региона, административно разделенного на ряд городов, можно использовать вложенный запрос с применением оператора IN:

```
SELECT count(*) AS count
FROM `air_station`
WHERE locationid IN
      (SELECT id FROM `global_location`
       WHERE parentid=m
      )
ORDER BY name
```

где: m – числовое значение, определяющее идентификатор города.

Следует отметить, однако, что подобная организация пространственных данных не позволяет решать некоторые задачи. Например, в случае визуализации пространственных прогнозов и векторной карты существует множество объектов, которые не подчиняются подобной иерархии (например, водоем может относиться к нескольким районам). Исходя из этого, данная таблица должна включать минимальное число объектов, необходимых только для описания административно-территориальных отношений, а для решения других задач следует использовать другую организацию пространственных данных или готовые решения, например, Google Maps, Google Earth и т.д. [14].

Таблица *global_users* необходима для хранения учетных записей пользователей системы, которые создаются как для реальных пользователей, так и для автоматизированных постов. Данная таблица используется при входе пользователей в систему, а также при отправке данных с автоматизированных постов для проверки подлинности клиента. Поле *privileges* задает набор прав доступа для конкретного пользователя.

Модель репликации данных

В работе автора [5] отмечается, что в соответствии с международными нормами в системах мониторинга регионального уровня должна поддерживаться возможность передачи наблюдаемой информации на вышестоящие уровни, т.е. осуществлять ее репликацию. Сегодня

практически все СУРБД имеют встроенные средства репликации, однако в данном случае эти средства не могут быть использованы по ряду причин, перечисленных в порядке значимости:

- несовместимость структуры таблиц базы данных регионального и вышестоящих уровней;
- несовместимость СУРБД регионального и вышестоящих уровней, а также их версий;
- невозможность управления процессом репликации; репликация данных с применением встроенных средств обычно выполняется либо в режиме реального времени, либо автоматически с заданным интервалом, а не по запросу пользователя;

- в некоторых СУРБД для организации процесса репликации данных часто требуются повышенные права доступа (в частности, в MySQL пользователь должен иметь категорию SUPER).

Исходя из этого, региональный data-центр должен включать собственные программные решения для передачи данных на сервер верхнего уровня и средства для ведения журнала операций в базе данных.

Для репликации данных в структуре базы данных предусмотрены две таблицы *replicate_operation* и *replicate_table*. Таблица *replicate_operation* содержит актуальный список (журнал) операций, которые проводились на сервере, но еще не были переданы на верхний уровень. В таблице протоколируются только три типа операций (INSERT, UPDATE, DELETE), которые приводят к изменению данных в базе. Поле *rowid* содержит идентификатор изменяемой (удаленной или вставленной) строки таблицы, а поле *tableid* образует реляционную связь с таблицей *replicate_table*, в которой помимо имени таблицы может также храниться служебная информация.

Для занесения данных в таблицу *replicate_operation* используется хранимая процедура *replicate*, вызов которой выполняется в одном из триггеров. Для каждой таблицы, подлежащей репликации, создается три триггера (для всех операций). Таким образом, заполнение и поддержание журнала операций в актуальном состоянии осуществляется самой СУРБД, что является достаточно удобным. Текст основной процедуры репликации показан ниже:

```
CREATE PROCEDURE replicate
  (IN _operationtype CHAR(1),
  IN _tablename VARCHAR(64),
  IN _rowid INT
  )
BEGIN
  SELECT id INTO @tableid
  FROM replicate_table
  WHERE name=_tablename;
  INSERT INTO replicate_operation
  (operationtype, tableid, rowid, date)
  VALUES (_operationtype, @tableid,
```

```
_rowid, NOW());
END //
```

Перед созданием хранимой процедуры необходимо переопределить признак конца команды при помощи команды DELIMITER //. Текст триггера приведен ниже:

```
CREATE TRIGGER t_global_unit_insert
AFTER INSERT ON global_unit
FOR EACH ROW
BEGIN

  CALL replicate('I', 'global_unit',
                NEW.id);

END;
//
```

Указанный триггер срабатывает сразу после добавления (AFTER INSERT) новой строки в таблицу с именем *global_unit*. Префикс *NEW* содержит ссылку на новую (только что добавленную) запись. В триггере, привязанном к событию удаления записи, вместо префикса *NEW* используется префикс *OLD*.

Репликация данных производится модулем репликации, который входит в состав системы (см. рис. 1). Основная задача репликации может быть разбита на ряд подзадач, основными из которых являются оптимизация журнала операций и формирование пакета данных в соответствии с протоколом обмена между подсистемами мониторинга, функционирующих на разных уровнях. Исходные последовательности операций, относящиеся к каждой строке реплицируемой таблицы, могут быть получены в модуле репликации из таблицы *replicate_operation* путем выполнения запроса:

```
SELECT group_concat(operationtype
  SEPARATOR '') AS sequence
FROM
  replicate_operation, replicate_table
WHERE
  replicate_operation.tableid =
  replicate_table.id
AND
  replicate_table.replicate = true
GROUP BY
  tableid, rowid
ORDER BY
  date
```

Оптимизация журнала операций должна выполняться для каждой строки реплицируемой таблицы. Суть оптимизации заключается в определении результирующей команды (R), которая должна передаваться на сервер верхнего уровня. Например, если запись сначала была вставлена (I), потом модифицирована (U), потом удалена (D), то данная последовательность команд фактически является пустой (E) и не должна приводить к изменениям базы данных на сервере вышестоящего уровня. С другой стороны, последовательность IDU вообще лишена смысла

(E_r), поскольку третья команда (U) будет пытаться модифицировать несуществующую (удаленную перед этим командой D) запись. Исходя из этого, логическим путем была получена таблица истинности (см. табл. 1) для оптимизации последовательности операций репликации.

Таблица 1 – Таблица истинности для оптимизации операций репликации

T ₁	T ₂	T	T ₁	T ₂	T
I	I	E _r	D	U	E _r
I	U	I	D	D	E _r
I	D	E	E	I	I
U	I	E _r	E	U	E _r
U	U	U	E	D	E _r
U	D	D	E	E	E
D	I	I	E _r	*	E _r

Следует отметить, что последовательности команд подчинены мультипликативному закону, т.е.:

$$R = (T_1 T_2) T_3 = T_1 (T_2 T_3) \quad (1)$$

Рассмотренная ранее таблица истинности (см. таблицу 1) совместно с уравнением (1) используется при репликации для вычисления результирующей команды (R), эквивалентной исходной последовательности всех команд. После этого запрос, соответствующий результирующей команде R, добавляется в общий пакет данных репликации. Структура данного пакета зависит от используемого протокола обмена между сервером регионального и вышестоящего уровня. В пакет добавляются только не лишние смысла операции (I, U или D). Сформированный пакет данных отправляется на сервер вышестоящего уровня. Если импорт данных выполнен успешно, этот сервер посылает подтверждение, после чего таблица *replicate_operation* может быть очищена. Для повышения надежности репликации перечень необходимых SQL команд (как на сервере регионального, так и вышестоящего уровней) следует поместить в транзакцию.

Модуль экспорта данных используется на начальном этапе для извлечения данных из базы и представления их в удобном виде. В качестве такого модуля может использоваться любое

средство управления базой данных: PhpMyAdmin [15], PostgreSQL Lite и им подобное. Выбор данного средства зависит, в первую очередь, от выбранной базы данных.

Ранее отмечалось, что информационная система мониторинга атмосферы регионального уровня, в основе которой лежит предлагаемая модель регионального data-центра, обязательно должна включать модуль моделирования и прогнозирования, который реализует одну или несколько аналитических моделей рассеивания примеси, как в пространстве, так и во времени. Все эти модели в качестве исходных данных используют данные субъектов мониторинга, представленные в виде рассмотренной ранее БД. Однако для разработки подобных моделей необходим анализ данных при помощи одного из статистических, аналитических, регрессионных и других методов.

Ввиду того, что сервер любой базы данных обладает ограниченными возможностями и не позволяет проводить полноценный анализ данных с применением подобных методов, для анализа данных на начальном этапе следует использовать ПО сторонних разработчиков (Statistica, MatLab). Исходная база данных (полностью или частично) экспортируется в один из распространенных форматов (TXT, XLS и др.), которые поддерживаются указанным ПО. Эти данные являются основой для поиска закономерностей и построения прогнозных моделей, которые, в свою очередь, являются основой построения модуля моделирования и прогнозирования, а также других составляющих информационной системы.

Выводы

Предложенная в работе информационная модель регионального data-центра, включающая структуру БД, программные модули ввода-вывода и механизм репликации данных позволит обобщить и систематизировать данные субъектов мониторинга, которые поступают из различных источников.

Практическая значимость предложенной информационной модели заключается в том, что эта модель может стать базовой основой для разработки программно-аналитических модулей интегрированной региональной информационной системы экологического мониторинга атмосферы.

Дальнейшее направление исследований будет связано с поиском закономерностей по данным субъектов мониторинга, что позволит в полном объеме решать задачи моделирования распространения примеси в пространственно-временных координатах.

Список литературы

1. Рабочее совещание по информационным технологиям для проведения экологического мониторинга и оценки [Текст] : СЕР/АС.10/2002/13. – ЕЭК ООН, Обнинск. – 13 марта 2002. – 5 с.
2. Развитие экологических сетей и информационных систем в странах ВЕКЦА [Электронный ресурс] / А. Свирчевский // Доклад. – Обнинск, 2003.
Режим доступа к докл.:
<http://unese.org/env/europe/monitoring/Obninsk.Jun.2003/NETWORKING.final.rev.R.pdf>
3. Адаптация сетей мониторинга в странах ВЕКЦА: Мониторинг качества воздуха [Текст] : ЕСЕ/СЕР/АС.10/2006/3. – ЕЭК ООН, Женева. – 3 апреля 2006. – 27 с.
4. Рекомендации по совершенствованию национальных систем мониторинга и информации по ОС для стран ВЕКЦА [Текст] : ЕСЕ/СЕР/109. – 21 мая, 2003. – 6 с.
5. Павлий, В. А. Направления развития информационных систем мониторинга атмосферного воздуха. [Текст] : научн. журн. / В. А. Павлий, А. А. Бабакина // Наук. праці ДонНТУ. – [під ред. Є. О. Башкова]. – Сер. "Системний аналіз та інформаційні технології у науках про природу та суспільство". – (САІТ-2012). – Вип. 2(134) – Донецьк: ДонНТУ, 2012. – С. 71 – 81.
6. Павлий, В. А. Алгоритмы моделирования полей загрязнения атмосферы по экспериментальным данным [Текст] : научн. журнал / В. А. Павлий, Г. В. Аверин // Вісник Донецького університету. – [гол. редак. В. П. Шевченко]. – Сер. А: Природничі науки, №2, 2007. – С. 338 – 346.
7. Павлий, В. А. Построение информационной модели расчета распространения примеси в приземном слое атмосферы над территориально-распределенными объектами [Текст] : научн. журн. / Виталий Александрович Павлий // Наук. праці Донецького Національного технічного університету. – [під ред. Є. О. Башкова]. – Сер. "Системний аналіз та інформаційні технології у науках про природу та суспільство". – (САІТ-2011). – Вип. 1(133) – Донецьк: ДонНТУ, 2011. – С. 71 – 81.
8. Руководство по контролю загрязнения атмосферы [Текст] : РД 52.04.186-89. – М.: Госкомгидромет СССР. – 693 с.
9. Модернизация и совершенствование сети мониторинга загрязнения атмосферного воздуха в Республике Узбекистан [Электронный ресурс] / Гранкина, Г.Н. // Доклад. – Узгидромет, 2007. –
Режим доступа к докл.:
<http://www.unese.org/fileadmin/DAM/env/europe/monitoring/workshops/wksp%207.06.07.Geneva/presentation/Uzbekistan.ppt>
10. Рабочее совещание по взаимодействию между мониторингом качества атмосферного воздуха и стратегиями охраны атмосферного воздуха в странах Восточной Европы, Кавказа и Центральной Азии [Электронный ресурс] / С. Чичерин // Доклад. – Женева, 2007.
Режим доступа к докл.:
<http://unese.org/fileadmin/DAM/env/europe/monitoring/workshops/wksp%207.06.07.Geneva/Russia.pdf>
11. Фридл, Дж. Регулярные выражения [Текст] / Дж. Фридл. – СПб.: Питер, 2003. – 464 с.: ил. – (Серия "Библиотека программиста"). – ISBN 5-272-00331-4.
12. Core – Apache HTTP Server [Electronic resource] / The Apache Software Foundation, 2009. – Requirements: Pentium-500; 128 Мб RAM; Windows 98/2000/NT/XP; MSIE 6.0; Adobe Acrobat Reader 6.0. –
Access mode:
<http://httpd.apache.org/docs/2.0/mod/core.html#limitrequestline>
13. OpenGIS Geography Markup Language (GML) Implementation Specification [Electronic resource] / Cox S., Cuthbert A., Lake R., Martell R. – OGC, 2002. – 66 p. – Requirements: Pentium-500; 128 Мб RAM; Windows 98/2000/NT/XP; MSIE 6.0; Adobe Acrobat Reader 6.0. –
Access mode:
http://portal.opengeospatial.org/files/?artifact_id=11339
14. Google Maps [Electronic resource] / Google, AND, Europe Technologies, 2009. – Requirements: Pentium-500; 128 Мб RAM; Windows 98/2000/NT/XP; MSIE 6.0; Adobe Acrobat Reader 6.0. –
Access mode:
<http://maps.google.com>
15. PhpMyAdmin [Electronic resource] / phpMyAdmin devel team, 2003-2012. – Requirements: Pentium-500; 128 Мб RAM; Windows 98/2000/NT/XP; MSIE 6.0; Adobe Acrobat Reader 6.0. –
Access mode:
http://www.phpmyadmin.net/home_page/index.php

В.О. ПАВЛІЙ

Донецький національний технічний університет

V.A. PAVLIY

Donetsk National Technical University

Створення регіонального data-центру як основи інформаційної системи екологічного моніторингу в областях України

Запропоновано інформаційну модель регіонального data-центру для зберігання й узагальнення даних суб'єктів екологічного моніторингу атмосфери. Розглянуто практичні аспекти його реалізації: структуру БД, механізм реплікації, модулі введення-виведення.

Інформаційна модель, data-центр, PCRE шаблони, база даних, реплікація даних, SQL, протоколи передачі даних, ACOIZA, автоматизований пост моніторингу атмосферного повітря

The development of regional datacenter as a basis for information system of ecological monitoring in the regions of Ukraine

The paper suggests information model of regional datacenter for data storage and summarization obtained from entities of ecological monitoring of atmosphere. Practical aspects of its development are examined: database scheme, replication mechanism, input/output modules.

Information model, datacenter, PCRE templates, database, data replication, SQL, data transfer protocols, ASOIZA, automated station for atmospheric air monitoring