

О. Н. Моргунова  
Сибирский государственный аэрокосмический университет  
имени академика М. Ф. Решетнева, Россия, Красноярск

## ОРГАНИЗАЦИЯ ЖУРНАЛА ОПЕРАЦИЙ С БАЗОЙ ДАННЫХ В СУБД POSTGRESQL

*Предлагается подход к организации журнала операций с базой данных. Основная идея используемой технологии заключается в создании журнальной таблицы для каждой рабочей таблицы базы данных, а также в создании трех правил (CREATE RULE): для операций вставки, обновления и удаления записей из рабочих таблиц. При выполнении операций с рабочей таблицей копии вставляемой, обновляемой или удаляемой записи вставляются в журнальную таблицу.*

При эксплуатации информационных систем возможны не только сбои в работе оборудования, но также и ошибки оператора. В первом случае может потребоваться выполнить восстановление базы данных (БД) с ее резервной копии. Для этого используются стандартные утилиты, входящие в комплект системы управления базами данных (СУБД). Такие утилиты имеются и в составе СУБД PostgreSQL. В случае же ошибок оператора, особенно, если они обнаружены с опозданием, выполнение восстановления базы данных посредством системной утилиты приведет к тому, что будут утрачены и корректные изменения базы данных, выполненные после проведения последнего резервного копирования. Таким образом, даже небольшая ошибка оператора может привести к необходимости повторного выполнения ввода или корректировки значительного объема данных.

Если бы в базе данных сохранялась вся история изменений каждой реляционной таблицы, то администратор имел бы возможность выполнять восстановление не только всей таблицы целиком, но и каждой конкретной строки в таблице.

В докладе предлагается организовать такую систему учета изменений базы данных, которая была бы построена на основе системы правил (rules) СУБД PostgreSQL.

Основные принципы и правила ее построения следующие.

1. В каждую таблицу базы данных вводятся следующие атрибуты, содержащие служебную информацию:

```
who_chg text DEFAULT USER,          -- кем добавлена (изменена) запись
when_chg timestamp
    DEFAULT CURRENT_TIMESTAMP,      -- когда добавлена (изменена) запись
```

2. Для каждой таблицы базы данных, которые назовем рабочими таблицами, должна быть создана таблица, которую будем называть журнальной. Эта таблица содержит все атрибуты рабочей таблицы и дополнительно еще три атрибута:

```
db_operation text,                  -- операция над БД
who_add text DEFAULT USER,         -- кем добавлена запись
when_add timestamp DEFAULT CURRENT_TIMESTAMP -- когда добавлена запись
```

3. Журнальные таблицы создаются без каких-либо ограничений целостности. Ограничения целостности не нужны, поскольку записи в журнальные таблицы добавляются только из рабочих таблиц, а в рабочих таблицах ограничения целостности присутствуют, поэтому данные в них согласованные.

4. Для каждой рабочей таблицы создается три так называемых правила (CREATE RULE). Это расширение языка SQL, поддерживаемое СУБД PostgreSQL. Правила позволяют с каждой операцией, выполняемой над таблицей базы данных, связать некоторые дополнительные операции, выполняемые, возможно, над другими таблицами. В нашей

технологии правила создаются для операций вставки записей в таблицу (INSERT), обновления записи в таблице (UPDATE) и удаления записей из таблицы (DELETE).

Правило для операции вставки записей выглядит так:

```
CREATE RULE Langs_rule_1 AS
  ON INSERT TO Langs
  DO INSERT INTO Langs_2 ( lang_code, lang_fname, lang_sname, comment, who_chg,
when_chg, db_operation ) VALUES ( NEW.lang_code, NEW.lang_fname, NEW.lang_sname,
NEW.comment, NEW.who_chg, NEW.when_chg, 'INSERT' );
```

В этой команде таблица Langs – это основная таблица, а Langs\_2 – журнальная таблица. Атрибут db\_operation будет содержать значение 'INSERT', атрибут who\_add – значение по умолчанию, т. е. имя пользователя, добавившего запись в рабочую таблицу, атрибут when\_add – значение по умолчанию, т. е. временную отметку выполнения операции.

Правило для операции обновления записей выглядит так:

```
CREATE RULE Langs_rule_2 AS
  ON UPDATE TO Langs
  DO INSERT INTO Langs_2 ( lang_code, lang_fname, lang_sname, comment, who_chg,
when_chg, db_operation ) VALUES ( NEW.lang_code, NEW.lang_fname, NEW.lang_sname,
NEW.comment, NEW.who_chg, NEW.when_chg, 'UPDATE' );
```

Правило для операции удаления записей выглядит так:

```
CREATE RULE Langs_rule_3 AS
  ON DELETE TO Langs
  DO INSERT INTO Langs_2 ( lang_code, lang_fname, lang_sname, comment, who_chg,
when_chg, db_operation ) VALUES ( OLD.lang_code, OLD.lang_fname, OLD.lang_sname,
OLD.comment, OLD.who_chg, OLD.when_chg, 'DELETE' );
```

5. В журнальные таблицы записи только добавляются, но не обновляются и не удаляются из них. Поэтому сохраняется вся история изменения каждой записи из рабочей таблицы.

Предложенный подход позволит повысить безопасность выполнения операций с базой данных. В случае ошибки оператора администратор базы данных сможет отыскать запись в журнальной таблице и восстановить то состояние конкретной записи в рабочей таблице, которое требуется.

O. Morgunova

Siberian State Aerospace University named after academician M. F. Reshetnev, Russia,  
Krasnoyarsk

### **DESIGN OF DATABASE LOGGING SYSTEM FOR POSTGRESQL DBMS**

*Design of database logging system for PostgreSQL DBMS is proposed. The main idea of the technology is as follows. For each working table there must be created logging table. Additionally, three rules (CREATE RULE) must be created for database operations, namely – INSERT, UPDATE, DELETE. When inserting, updating or deleting a record a copy of this record is being inserted into a logging table.*