

ном порядке (`lo_export/lo_read`). Все операции с большими объектами должны происходить внутри транзакции.

Следует отметить, что необходимость интерфейса больших объектов на текущий момент не так уж и очевидна. Стандартными средствами в *PostgreSQL* можно сохранять бинарные данные размером вплоть до 1 Гб, что вполне может соперничать с максимальным размером для большого объекта (2 Гб).

## ЕСРPG

Чтобы не отставать от коммерческих баз данных, *PostgreSQL* имеет свой собственный вариант «встроенного SQL». Эта технология позволяет смешивать обычный язык C с SQL-структурами, примерно следующим образом:

```
// файл test.pgc
#include <stdio.h>
#include <stdlib.h>
// структура для обработки ошибок
EXEC SQL include sqlca;
// реакция в случае ошибки/предупреждения
EXEC SQL whenever sqlwarning sqlprint;
EXEC SQL whenever sqlerror do ExitForError();
void ExitForError() {
    fprintf(stderr, "Все, конец - это фатально.\n");
    sqlprint();
    exit(1);
}

int main(int argc, char **argv)
{
    // определение переменных, чтобы их можно было использовать
    // инструкциях ЕСРPG
    EXEC SQL BEGIN DECLARE SECTION;
    const char *dbname = "test";
    const char *user = "baldin";
    VARCHAR FIO[128];
    VARCHAR NUMBER[128];
    EXEC SQL END DECLARE SECTION;
    // соединение с базой данных
    // внешние переменные предваряются двоеточием
    EXEC SQL CONNECT TO :dbname USER :user;
    // определение курсора через SELECT
    EXEC SQL DECLARE mycursor CURSOR FOR
    SELECT fio, number FROM fiodata, phonedata
    WHERE fiodata.id=phonedata.id;
    EXEC SQL open mycursor;
    // чтение данных из курсора
    EXEC SQL FETCH NEXT FROM mycursor INTO :FIO,:NUMBER;
    while (sqlca.sqlcode == 0) { // не 0, если данные больше нет
        printf("ФИО: %s номер: %s\n", FIO.arr, NUMBER.arr);
        EXEC SQL FETCH NEXT FROM mycursor INTO :FIO, :NUMBER;
    }
    // разъединение с базой данных
    EXEC SQL DISCONNECT;
}
```

Все SQL-команды начинаются с метки **EXEC SQL**. Она используется препроцессором `espg` как маркер для конструкций, подлежащих обработке. Внутри SQL-команд можно использовать переменные C. Для этого перед ними необходимо поставить двоеточие «:».

Для компиляции выше процитированного файла (`test1.pgc`) необходимо выполнить следующие действия:

```
> # установить espg
> sudo apt-get install libecpg-dev
> # запустить препроцессор
> espg test1.pgc
```

## PostgreSQL в лицах: Сергей Копосов



### Визитка LXF:

Дипломированный физик.  
Профессиональный астроном. Участник программы Google Summer of Code 2006.  
Домашняя страница: <http://lnfm1.sai.msu.ru/~math/>.

**Евгений М. Балдин (ЕМБ):** Что привлекло вас к разработке именно *PostgreSQL*?

**Сергей Е. Копосов (СЕК):** Вначале я познакомился с *PostgreSQL* как пользователь. Причины были следующие: необходимо было работать с большим астрономическим каталогом (2MASS), который содержал более 200 миллионов звезд. Здесь без базы данных не обойдешься. То, что выбран был именно *PostgreSQL*, связано с тем, что рядом были люди\*, которые его уже использовали.

А дальше все пошло по нарастающей. Начав работать с *PostgreSQL*, я вдруг обнаружил, что в нем не существует способов, с помощью которых можно было бы быстро искать объекты на сфере. В итоге это сподвигло меня на создание проекта Q3C (*QuadTreeCube* <http://q3c.sf.net>). Процесс работы над ним потребовал выйти за рамки стандартного *PostgreSQL*. Например, мне были необходимы битмар-индексы, а Том Лэйн [Tom Lane] их тогда только реализовывал. Из-за этого мне пришлось перейти на работу с CVS-версией *PostgreSQL*. Во время разработки Q3C мне удалось заметить несколько проблем со скоростью работы битмар'ов, которые позже были исправлены Томом.

После интенсивной разработки Q3C я сильно «подсел» на списки рассылки *pg-hackers*, *pg-patches*, и я, в частности, написал несколько маленьких патчей для тех проблем в клиенте *pSQL*, которые мне не нравились. Сначала автодополнение **DROP-FUNCTION**, а потом улучшение работы *pSQL* с многострочными запросами. За последнее меня уже официально включили в список «Official contributors *PostgreSQL*» – чем я несказанно горд [улыбка].

**ЕМБ:** За что вас выбрали для участия в программе Summer of Code 2006, и чем эта эпопея закончилась?

**СЕК:** Тут сложно сказать. На самом деле, предложенных на Summer of Code проектов было не так уж и много, а мой проект не был ни чрезвычайно сложным, ни очень простым. Была общая подсказка от Тома, в каком направлении реализовывать проект. Сам проект был вполне естественным улучшением уже существующей функциональности *PostgreSQL*, реализующей к тому же одну из маленьких недостающих частей стандарта SQL:2003. К тому же я уже был хотя бы минимально, но известен в сообществе *pg-hackers*.

А закончилось все тем, что сейчас мой код по агрегатным функциям включен в CVS и будет в версии 8.2 *PostgreSQL*. За что, конечно, спасибо Тому Лэйну, который фиксировал и слегка корректировал код.

**ЕМБ:** Вы – профессиональный астроном, и, судя по списку проектов, вы активно работаете по основному профилю. *PostgreSQL* не отвлекает от вашей деятельности?

**СЕК:** Отвлекает. Очень. Поэтому пока я взял паузу в кодировании для *PostgreSQL*. А дальше, поживем – увидим...

\* Множество «эти люди» включало Олега Бартунова.