

Оптимизация баз данных в системе управления баз данных MySQL

Козич Полина Александровна

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

Одну из ключевых ролей в информационных системах играет система управления базами данных (СУБД). Данная система чаще всего развернута на специальном сервере, к которой пользователи обращаются, чаще всего не напрямую, а через клиентское программное обеспечение, которое служит оболочкой, интерфейсом для работы с базой данных. При таком взаимодействии, для пользователя главное, чтобы запросы, отправляемые к базе данных, исполнялись как можно быстрее. В данной статье будет рассмотрено что такое хранимые процедуры, какими качествами и недостатками обладает такое решение и как создать хранимые процедуры в MySQL и вызывать их используя язык программирования PHP.

Ключевые слова: Система управления базами данных (СУБД), MySQL, оптимизация, хранимые процедуры, хранимые функции, язык программирования, PHP.

Optimization of databases in management system of databases of MySQL

Kozich Polina Alexandrovna

Sholom-Aleichem Priamursky State University

Student

Abstract

One of key roles in information systems is played by the database management system (DMS). This system is most often torn on the special server which users address, most often not directly, and through the client software which serves as a jacket, the interface for operation with the database. In case of such interaction, for the user the main thing that the requests sent to the database were executed as soon as possible. In this article it will be considered what is stored procedures what qualities and shortcomings such decision has and how to create stored procedures in MySQL and to cause them using the PHP programming language.

Keyword: Database Management System (DMS), MySQL, optimization, stored procedures, storable functions, programming language, PHP.

Одну из ключевых ролей в информационных системах играет система управления базами данных (СУБД). Данная система чаще всего развернута на специальном сервере, к которой пользователи обращаются, чаще всего не напрямую, а через клиентское программное обеспечение, которое служит

оболочкой, интерфейсом для работы с базой данных. При таком взаимодействии, для пользователя главное, чтобы запросы, отправляемые к базе данных, исполнялись как можно быстрее.

Цель данной статьи: рассмотреть что такое хранимые процедуры, какими качествами и недостатками обладает такое решение и как создать хранимые процедуры в MySQL и вызывать их используя язык программирования PHP.

О проектировании баз данных в своих работах пишут В.В. Бойко, В.М. Савинков в своей работе рассматривают проектирование баз данных информационных систем [1], С.М. Диго в своей работе рассматривает проектирование и использование баз данных [2]. В.А. Глаголев в своей работе исследует создание баз данных для оценки и прогноза пожарной опасности растительности по природно-антропогенным условиям [3]. Е.А. Шельмина и И.Г. Боровской в своей статье провели анализ производительности современных настольных и клиент-серверных СУБД [4]. Так же данной проблемой занимаются зарубежные ученые [5].

В качестве решения задачи по уменьшению времени выполнения запросов к базе данных могут быть использованы хранимые процедуры, так же известные как хранимые функции. Хранимая процедура – это объект базы данных, представляющий собой набор SQL-инструкций, который компилируется один раз и хранится на сервере. Таким образом, при повторном вызове процедуры, она не будет компилироваться повторно, а будет извлечена из оперативной памяти. Такой подход имеет свои качества и недостатки.

Качества:

- Инкапсуляция функциональности – качество, которое так же известно, как один из принципов объектно-ориентированного программирования. В данном случае, весь код реализации скрыт в СУБД и программист, в своем коде работает только с вызовом функции. Для него это черный ящик, который возвращает тот или иной результат, или делает те или иные действия. Здесь стоит отметить важность правильного названия процедуры. Название должно быть такое, чтоб программисту не пришлось смотреть исходный код или отвлекать коллегу от рабочего процесса, чтоб разъяснить ситуацию.

- Изоляция доступа к данным – хранимая процедура становится своего рода прослойкой к данным. Весь доступ к ним регулируется в рамках данной процедуры локально. Как было написано выше, данная процедура – черный ящик для использующего ее программиста. С учетом того, что в команде могут работать разработчик баз данных, как отдельный тип программиста, подобный плюс позволяет разделить ответственность за состояние СУБД.

- Повышение скорости работы сервера – тот плюс, ради которого чаще всего и делается выбор в пользу использования хранимых процедур. Происходит ускорение за счет кеширования и объединения запросов.

Недостатки:

- Повышается нагрузка на сервер баз данных. Связано это с тем, что код процедур выполняется на сервере, как в следствии, расходуя его ресурсы.
- Усложняется манипуляция данными, так как часть кода все равно придется писать на клиентской стороне.
- Усложняется процесс переноса базы данных в другую СУБД.

Разберем, как создать хранимую процедуру, используя инструмент phpMyAdmin. Для начала, необходимо в панели справа выбрать необходимую базу данных. Затем в меню сверху выбрать вкладку «Процедуры», внизу нажать кнопку «Новая» (Рисунок 1).

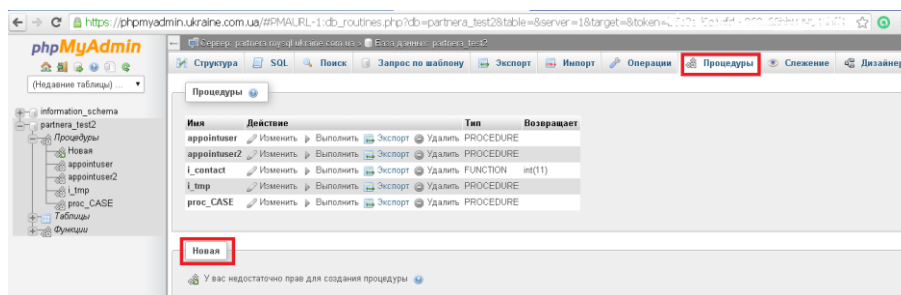


Рисунок 1 – Окно с содержимым вкладки «процедуры»

При создании новой процедуры необходимо ввести имя процедуры, тип, определение – это сам код процедуры, определителя – пользователь от которого будет выполняться процедура и, при необходимости, можно ввести комментарий. Пример на рисунке 2.

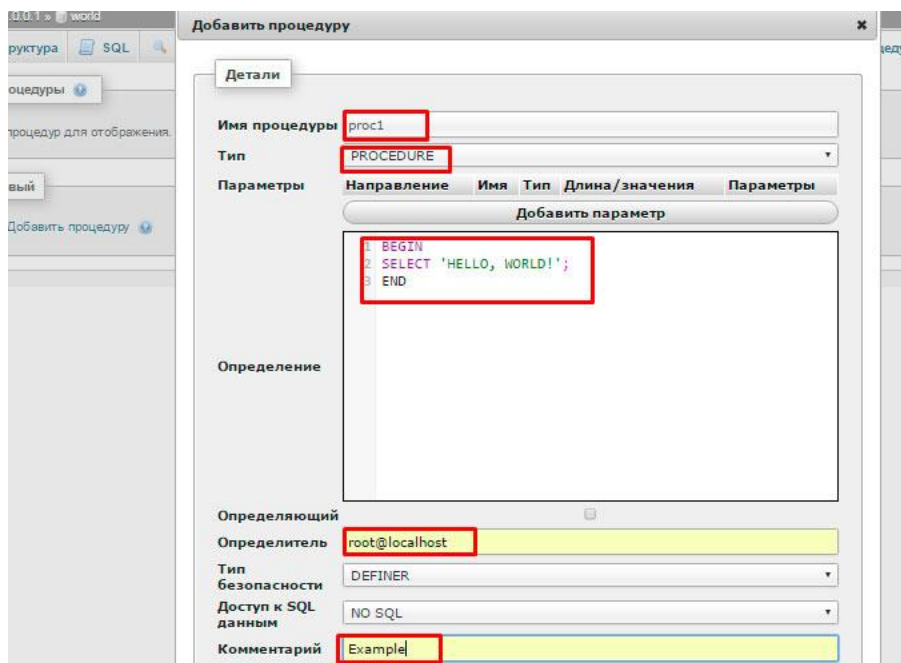


Рисунок 2 – Пример создания новой процедуры

Стоит отметить, что код процедуры обернут в ключевые слова Begin и End. По нажатию кнопки «Ок», процедура будет создана. Стоит заметить, что для того, чтобы хранимая процедура работала, необходимо в конфигурационном файле MySQL с именем «my.ini» добавить строку, изображенную на рисунке 3.

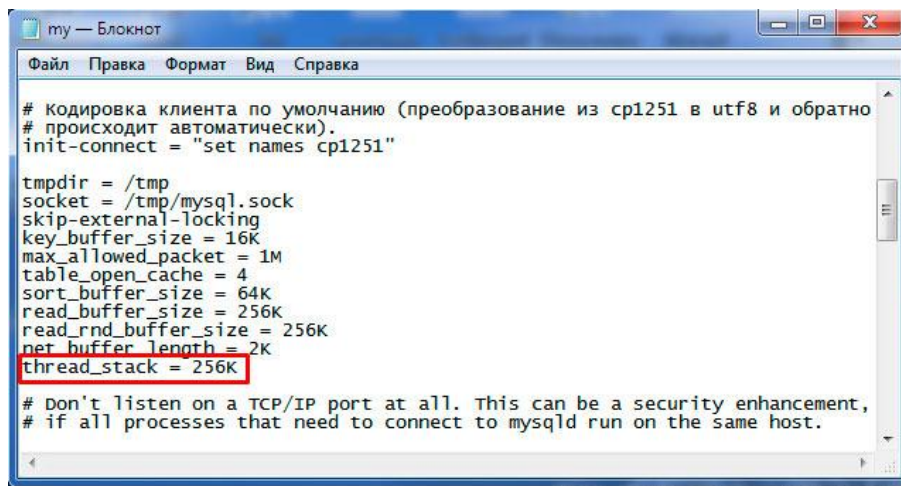


Рисунок 3 – Конфигурационный файл MySQL

Данный параметр отвечает за выделение оперативной памяти одному потоку, выполняющий запрос. По умолчанию в MySQL данный параметр задан 128 kb. Для обычных запросов этого обычно достаточно, но хранимая процедура расходует больше ресурсов памяти.

Для того, чтобы вызвать процедуру необходимо в запрос ввести ключевое слово CALL, а затем название хранимой процедуры. Для того, чтобы сделать это с помощью PHP, можно использовать класс PDOStatement, созданный для работы с базой данных через SQL. Данный класс поддерживается с версии PHP 5.1.0. Пример его использования для вызова хранимой процедуры можно увидеть на рисунке 4.

```
1 <?php
2 $dbms = 'mysql';
3 $host = 'localhost';
4 $db_name = 'world';
5 $user_name = 'root';
6 $pass = '';
7 $str_con = "$dbms:host=$host;dbname=$db_name";
8 $cn=new PDO($str_con, $user_name, $pass);
9 $q=$cn->exec('CALL proc1(@var)');
10 $result=$cn->query('select @var')->fetchColumn();
11 print($result);
12 ?>
```

Рисунок 4 – Пример использования класса PDOStatement для вызова хранимой процедуры

В итоге была создана хранимая процедура, был изменен конфигурационный файл MySQL для выделения большей памяти потоку и был приведен пример, вызова хранимой процедуры. Также стоит учитывать, что использование хранимых процедур повсеместно, не самое лучшее решение.

Библиографический список

1. Бойко В.В., Савинков В.М. Проектирование баз данных информационных систем. М.: Финансы и статистика, 1989.
2. Диго С.М. Базы данных. Проектирование и создание. М.: Издательский центр ЕАОИ, 2008.
3. Глаголев В.А. Создание баз данных для оценки и прогноза пожарной опасности растительности по природно-антропогенным условиям // Региональные проблемы. 2014. Т. 17. № 2. С. 78-82.
4. Шельмина Е.А., Боровской И.Г. Анализ производительности современных настольных и клиент-серверных СУБД // Ползуновский вестник . 2017. №3. С. 81-84.
5. Mensah K. Stored Procedures as Database Programming Model // Oracle Database Programming using Java and Web Services. 2006. Pages 3-21.