

Запросы в СУБД MySQL

Волков Виталий Александрович

Мордовский государственный университет им. Н. П. Огарева

студент

Аннотация

В данной статье рассматриваются простые и сложные запросы, выполняемые в MySQL. Раскрываются их подвиды. Предоставляются примеры применения их на практике. Обращается внимание на возникновение возможных трудностей при написании запросов.

Ключевые слова: СУБД; MySQL; SQL; SELECT; INSERT; UPDATE; DELETE; DROP; запросы; простые; сложные; FROM; ORDRE BY; DESC; LIMIT; WHERE; база данных; БД; таблица; поле.

The queries in the MySQL DBMS

Volkov Vitaliy Alexandrovich

Ogarev Mordovian State University

student

Abstract

This article focuses on simple and complex queries executed in the MySQL. Disclosed their subspecies. Provided examples of their application in practice. Attention is drawn to the emergence of possible difficulties when writing queries.

Keywords: DBMS; MySQL; SQL; SELECT; INSERT; UPDATE; DELETE; DROP; requests; simple; complex; FROM; ORDRE BY; DESC; LIMIT; Where CLAUSE; database; DB; table; box.

MySQL – это свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle. Запрос – это средство выбора необходимой информации из базы данных. Вопрос, сформированный по отношению к базе данных, и есть запрос. MySQL запрос – это обращение к базе данных MySQL, с помощью которого мы можем реализовать: получение, изменение, удаление, сортировку, добавление, и другие манипуляции с данными базы.

Все MySQL запросы можно разделить на простые и сложные. Простые MySQL запросы – это запросы, в которых участвует одна таблица базы данных. Сложные – запросы, в которых могут участвовать две и более таблиц базы данных. Также можно выделить категорию очень простых запросов, структура которых очень понятна и не требует дополнительных разъяснений [1-3].

К очень простым мы отнесём такие запросы как `showdatabases` - вывести список всех баз, и `showtablesинbase_name` - показать список всех таблиц в базе данных.

К простым запросам относятся:

1. `SELECT` (выбрать);
2. `INSERT` (новая запись);
3. `UPDATE` (перезаписать, дописать);
4. `DELETE` (удалить запись);
5. `DROP` (удалить таблицу);

SELECT – это запрос, который выбирает уже существующие данные из базы данных. Для выбора можно указывать определённые параметры выбора [4, 5].

- Выбрать все данные из таблице `tbl_name`.
 - `SELECT * FROM tbl_name;`
- Вывести N записей из таблице `tbl_name`.
 - `SELECT count(*) FROM tbl_name;`
- Выбрать из таблицы `tbl_name` лимит в 3 записи, начиная со 2.
 - `SELECT * FROM tbl_name LIMIT 2,3;`
- Выбрать все записи из таблицы `tbl_name` и отсортировать их по `id` в порядке возрастания.
 - `SELECT * FROM tbl_name ORDER BY id;`
- Выбрать все записи из таблицы `tbl_name` и отсортировать их по полю `id` в в порядке убывания.
 - `SELECT * FROM tbl_name ORDER BY id DESC;`
- Выбрать первые 5 записей из таблицы `users` и отсортировать их по полю `id` в порядке возрастания.
 - `SELECT * FROM users ORDER BY id LIMIT 5;`
- Выбрать все записи из таблицы `users`, где поле `fname` соответствует значению `Ivan`.
 - `SELECT * FROM users WHERE fname='Ivan';`
- Выбрать все записи из таблицы `users`, где значение поля `fname` начинается с `Iv`.
 - `SELECT * FROM users WHERE fname LIKE 'Iv%';`
- Выбрать все записи из таблицы `users`, где `fname` заканчивается на `an`, и упорядочить их в порядке возрастания значения `id`.
 - `SELECT * FROM users WHERE fname LIKE '%an' ORDER BY id;`
- Выбрать все данные из колонок `fname`, `lname` из таблице `users`.
 - `SELECT fname, lname FROM users;`
- Выбрать все данные строк из таблицы `users`, где `age` имеет значения 18,19 и 21.
 - `SELECT * FROM users WHERE age IN (18,19,21);`
- Выбрать максимальное (минимальное) значение `age` в таблице `users`.
 - `SELECT max/min (age) FROM users;`

INSERT – это запрос, который позволяет первоначально вставить запись в базу данных. То есть создаёт новую запись в базе.

```
INSERT INTO users (name, age) VALUES ('Антон', '21');
```

Данный запрос добавляет новую запись в таблицу users. В поле name добавиться Антон, а в поле age - 21. Таким образом, в таблицу дописывается новая строки с данными значениями.

UPDATE – запрос, который позволяет перезаписать значения полей или дописать что-то в уже существующей строке в базе данных.

```
UPDATE users SET age = '18' WHERE id = '3';
```

Данный запрос перезаписывает, в таблице user, значение поля age на 18, где значение поля id равно 3.

DELETE – запрос, который удаляет строку из таблицы.

Удаление строки из таблицы usersc полем id равным 10.

```
DELETE FROM users WHERE id = '10';
```

DROP – запрос, который удаляет таблицу.

Удаление таблицы tbl_name.

```
DROP TABLE tbl_name;
```

Теперь рассмотрим сложные запросыMySQL. Как уже упоминалось раньше, они работают более чем с одной таблицей базы данных. Данные MySQL запросы, мы будем рассматривать более в индивидуальном порядке, так как они сложные и их будет немного.

```
SELECT DISTINCT last_name FROM person, address WHERE person.adress_no = address.address_no AND city LIKE 'L%';
```

Выводит все уникальные фамилии людей (last_name), которые живут в городе с названием на букву L.

```
SELECT id,name,country FROM users,admins WHERE TO_DAYS(NOW()) - TO_DAYS(registration_date) <= 14 AND activation != '0' ORDER BY registration_date DESC;
```

Данныйсложныйзапросвыбираетколонки id, name, country в таблицах users, admins, где registration_date (дата) не старше 14 дней и activation не равно 0, с сортировкой registration_date поубыванию.

```
UPDATE users SET age = '18+' WHERE age = ( SELECT age FROM users WHERE male = 'man');
```

Выше указан пример так называемого запроса в запросе. Обновить возраст среди пользователей на 18+, где пол - мужской. Подобные варианты запроса не рекомендуются. Лучше создать несколько отдельных запросов, так как они будут прорабатываться быстрее.

```
SELECT heroes.char_name, heroes.count, char_templates.ClassName FROM char_templates, heroes WHERE char_templates.ClassId = heroes.class_id Order by char_templates.ClassName;
```

Берем из таблицы heroes поле char_name и поле count, из таблицы char_templates поле ClassName, где char_templates. ClassId и heroes.class_id имеют общий идентификатор и сортируем запрос по имени класса героев.

Таким же образом, можно создавать сложные MySQL запросы с помощью update, insert, delete и др.

Выше мы рассмотрели виды запросов в СУБД MySQL. По сути, запросы не очень то и сложные, нужно только указать имя таблицы базы данных, а потом, через точку указать поле таблицы. Умение быстро и грамотно создавать запросы может прийти только с большим количеством практики и, конечно, теорией.

Библиографический список

1. Аббакумов А.А., Акимов В.Л., Егунова А.И., Лещанкин К.А., Таланов В.М. Базы данных (MS ACCESS, MYSQL). Саранск: Изд-во Средневолжского математического общества, 2011. 112 с.
2. Аббакумов А.А., Егунова А.И., Таланов В. М. Базы данных (MS SQL Server). Учеб.пособие. Саранск: Изд-во СВМО, 2015. 66 с.
3. Таланов В. М., Федосин С. А. Проектирование информационных систем и баз данных. Учеб.пособие. Саранск: Изд-во СВМО, 2013. 72 с.
4. Кевбрин В.А., Волков В.А., Трунин Д.С. Работа с базами данных на С# на примере MicrosoftAccess // APRIORI. Серия: Естественные и технические науки. 2016. №3 [Электронный ресурс]. Режим доступа: <http://www.apriori-journal.ru/seria2/3-2016/Kevbrin-Volkov-Trunin.pdf>
5. Волков В.А., Трунин Д.С., Митьков С.А. Анализ особенностей использования систем управления базами данных MSAccess, MySQL // APRIORI. Серия: Естественные и технические науки. 2015. №6 [Электронный ресурс]. Режим доступа: <http://www.apriori-journal.ru/seria2/6-2015/Volkov-Trunin-Mitikov.pdf>