

Применение внешних ключей на таблицы базы данных MySQL

Кочитов Михаил Евгеньевич

Приамурский государственный университет им. Шолом-Алейхема

студент

Аннотация

В данной статье рассматриваются внешние ключи, которые применяются для образования связей между реляционными таблицами в базе данных с целью сохранения целостности в них данных. Также будет разработан собственный пример на языке программирования PHP и на структурированном языке запросов MySQL, где будет создана тестовая база данных с таблицами, в которых будут применены внешние ключи для образования в них связей.

Ключевые слова: база данных, внешние ключи, связь, таблицы, MySQL

Applying Foreign Keys to MySQL Database Tables

Kochitov Mikhail Evgenevich

Sholom-Aleichem Priamursky State University

student

Abstract

This article discusses foreign keys that are used to form relationships between relational tables in a database in order to preserve the integrity of their data. We will also develop our own example in the PHP programming language and in the structured query language MySQL, where a test database with tables will be created, in which foreign keys will be used to form relationships in them.

Keywords: database, foreign keys, link, tables, MySQL

Базы данных всегда содержат очень много таблиц с разным количеством данных, однако чтобы определенные таблицы зависели от других, то устанавливаются связи с применением внешних ключей. Это дает возможность большое количество данных вмещать не в одну большую таблицу, что будет сложно для чтения, а на несколько маленьких таблиц с определенными данными. Внешние ключи также помогают обеспечить связанным таблицам целостность данных, чтобы в них всегда содержались актуальные данные.

В статье А.Н. Ряшенцевой рассматривается разработка базы данных для интернет-магазина на примере иммунохимических анализаторов [1]. Рассматривая статью Р.Ю. Катренко, А.А.Рыбанова можно увидеть разработку алгоритма получения количественных метрик хранимых процедур баз данных MySQL [2]. П.А. Шарыпов рассмотрел сравнительный анализ пространственных функций баз данных [3]. В статье Е.В. Болдырева

рассматривается организация и хранение геоточек в базе данных MySQL [4]. Рассматривая статью А.В. Лукьянченко, В.Г. Першенковой можно увидеть оптимизацию схем баз данных и запросов в СУБД MySQL [5].

Целью данной статьи является применение внешних ключей на таблицы базы данных MySQL, которые позволяют создавать связи между реляционными таблицами, обеспечивая в них целостность данных. Также будет разработан собственный пример с использованием языка программирования PHP и языка запросов MySQL, благодаря которому будет создана тестовая база данных с таблицами с примененными в ней внешними ключами.

Для того, чтобы база данных работала стабильно, то необходим локальный сервер. В качестве примера будет использовать локальный веб-сервер OpenServer [6] на Windows, который предоставляется пользователям в свободном доступе. На этом сервере будет создаваться база данных на основе PHP файлов с запросами MySQL. Для начала необходимо создать базу данных под названием «test» и в ней создать три таблицы «customers», «products» и «orders». Для осуществления этой цели необходимо на сервере создать PHP файл «create.php» и написать в него следующий код

```
<?php
$link = new mysqli("localhost", "root", "root", "test");

$query_text = "
    CREATE TABLE customers
    (
        customer_id INT PRIMARY KEY AUTO_INCREMENT,
        name VARCHAR(50) NOT NULL,
        surname VARCHAR(50) NOT NULL
    );
";

mysqli_query( $link, $query_text );

$query_text = "
    CREATE TABLE products
    (
        product_id INT PRIMARY KEY AUTO_INCREMENT,
        name VARCHAR(50) NOT NULL
    );
";

mysqli_query( $link, $query_text );

$query_text = "
    CREATE TABLE orders
    (
        order_id INT PRIMARY KEY AUTO_INCREMENT,
        customer_id INT NOT NULL,
        product_id INT NOT NULL,
        FOREIGN KEY (customer_id) REFERENCES customers (customer_id)
            ON DELETE CASCADE,
        FOREIGN KEY (product_id) REFERENCES products (product_id)
            ON DELETE CASCADE,
        date_arrival DATE NOT NULL
    );
";

mysqli_query( $link, $query_text );
```

Рисунок 1 – PHP код создания трех таблиц: «customers», «products» и «orders»

На рисунке 1 показан PHP код из файла «create.php», в котором с помощью запросов MySQL создаются три таблицы. Теперь рассмотрим данный код более подробно. Для начала происходит подключение к базе данных «test», передавая логин и пароль «root», после этого в переменные \$query_text формируется запрос создания таблиц с указанными в них столбцами и функция «mysqli_query» выполняет сам запрос и создает собственно сами таблицы. Далее рассмотрим, какие таблицы, с какими столбцами будут созданы после выполнения этих запросов. Первая таблица под названием «customers» предназначена для внесения информации о клиентах, в ней имеются столбцы: customer_id – идентификатор клиента, name – имя клиента и surname – фамилия клиента. Дальше вторая таблица «products» будет использоваться для хранения в ней продуктов, в ней всего лишь два столбца: product_id – идентификатор продукта, name – наименование продукта. Последняя третья таблица «orders» будет содержать заказы с созданием четырех столбцов, из которых два будут содержать внешние ключи со связями предыдущих двух таблиц. Структура таблицы «orders»: order_id – идентификатор заказа, customer_id – идентификатор клиента (внешний ключ связывается с таблицей «customers» и первым в ней столбцом customer_id), product_id – идентификатор продукта (внешний ключ связывается с таблицей «products» и первым в ней столбцом product_id), date_arrival – дата прибытия заказа. Внешние ключи создаются в таблице заказов «orders», которые будут связаны с таблицами клиентов «customers» и продуктов «products».

Далее необходимо в браузере открыть данный PHP файл «create.php», чтобы запросы выполнились и в базе данных были созданы три таблицы «customers», «products» и «orders». Веб-страница сама будет пустой, так как в самом PHP файле нет отображаемого на страницу контента, кроме как выполнения MySQL запросов, происходящих на сервере.

После того, как таблицы были созданы в базе данных, то необходимо добавить в них содержимое данных клиентов, продуктов и заказов, для этого на сервере создадим новый PHP файл «insert.php» и впишем в него следующий код

```
$link = new mysqli("localhost", "root", "root", "test");

$query_text = "
INSERT INTO `customers` (`customer_id`, `name`, `surname`) VALUES
(1, 'Николай', 'Иванов'),
(2, 'Михаил', 'Асеров'),
(3, 'Светлана', 'Гончарова'),
(4, 'Дмитрий', 'Линаев'),
(5, 'Мария', 'Попова'),
(6, 'Евгений', 'Казанов'),
(7, 'Анастасия', 'Морозова'),
(8, 'Андрей', 'Кольцов'),
(9, 'Павел', 'Никифоров'),
(10, 'Надежда', 'Носова'),
(11, 'Геннадий', 'Шилов'),
(12, 'Руслан', 'Воронов'),
(13, 'Владимир', 'Енотов'),
(14, 'Галина', 'Лотова'),
(15, 'Денис', 'Ростов');
";

mysqli_query($link, $query_text);
```

Рисунок 2 – Добавление данных клиентов в таблицу «customers»

```
$query_text = "
INSERT INTO `products` (`product_id`, `name`) VALUES
(1, 'Молоко'),
(2, 'Хлеб'),
(3, 'Сыр'),
(4, 'Йогурт'),
(5, 'Орехи'),
(6, 'Яблоки'),
(7, 'Помидор'),
(8, 'Огурец'),
(9, 'Кефир'),
(10, 'Груша'),
(11, 'Арбуз'),
(12, 'Булка'),
(13, 'Мясо'),
(14, 'Тесто'),
(15, 'Яйца'),
(16, 'Персик'),
(17, 'Мандарин'),
(18, 'Апельсин'),
(19, 'Сок'),
(20, 'Изюм');
";

mysqli_query($link, $query_text);
```

Рисунок 3 – Добавление данных продуктов в таблицу «products»

```
$query_text = "
INSERT INTO `orders` (`order_id`, `customer_id`, `product_id`, `date_arrival`) VALUES
(1, 5, 13, '2020-07-21'),
(2, 12, 10, '2020-06-04'),
(3, 3, 1, '2020-05-23'),
(4, 11, 3, '2020-06-17'),
(5, 5, 10, '2020-04-08'),
(6, 1, 9, '2020-02-15'),
(7, 9, 4, '2020-01-29'),
(8, 15, 2, '2020-06-16'),
(9, 7, 8, '2020-08-08'),
(10, 14, 17, '2020-02-15'),
(11, 13, 5, '2020-07-11'),
(12, 12, 11, '2020-04-04'),
(13, 6, 7, '2020-02-18'),
(14, 2, 13, '2020-03-10'),
(15, 6, 8, '2020-08-10'),
(16, 11, 1, '2020-07-12'),
(17, 14, 16, '2020-06-14'),
(18, 12, 5, '2020-03-19'),
(19, 11, 3, '2020-01-16'),
(20, 7, 19, '2020-02-09');
";

mysqli_query( $link, $query_text );
```

Рисунок 4 – Добавление данных заказов в таблицу «orders»

На рисунках 2, 3 и 4 изображены запросы по добавлению данных клиентов, продуктов и заказов в соответствующие для них таблицы «customers», «products» и «orders». Теперь необходимо открыть данный PHP файл «insert.php» в браузере, чтобы содержимое данных добавилось во все три таблицы. После этого требуется создать PHP файл «view.php», который на веб-странице будет отображать собственно содержимое всех трех таблиц, для этого напишем следующий в этом файле код

```
<?php

$link = new mysqli("localhost", "root", "root", "test");

$query_text = "
SELECT * FROM customers;
";

$result = mysqli_query( $link, $query_text );
echo '
<h3>Таблица "customers"</h3>
<table border=1>
<tr>
.....
<th>ID клиента</th>
<th>Имя</th>
<th>Фамилия</th>
</tr>
';
while ($row = mysqli_fetch_row($result)) {
echo '
<tr>
.....
<td>'.$row[0].'</td>
<td>'.$row[1].'</td>
<td>'.$row[2].'</td>
</tr>
';
}
echo '</table>';
```

Рисунок 5 – PHP код отображения на странице таблицы «customers»

```
$query_text = "
    SELECT * FROM products;
";

$result = mysqli_query( $link, $query_text );
echo '
    <h3>Таблица "products"</h3>
    <table border=1>
    <tr>
        <th>ID продукта</th>
        <th>Наименование</th>
    </tr>
';
while ($row = mysqli_fetch_row($result)) {
    echo '
        <tr>
            <td>'.$row[0].'</td>
            <td>'.$row[1].'</td>
        </tr>
    ';
}
echo '</table>';
```

Рисунок 6 – PHP код отображения на странице таблицы «products»

```
$query_text = "
    SELECT * FROM orders;
";

$result = mysqli_query( $link, $query_text );
echo '
    <h3>Таблица "orders"</h3>
    <table border=1>
    <tr>
        <th>ID заказа</th>
        <th>ID клиента</th>
        <th>ID продукта</th>
        <th>Дата прибытия</th>
    </tr>
';
while ($row = mysqli_fetch_row($result)) {
    echo '
        <tr>
            <td>'.$row[0].'</td>
            <td>'.$row[1].'</td>
            <td>'.$row[2].'</td>
            <td>'.$row[3].'</td>
        </tr>
    ';
}
echo '</table>';
```

Рисунок 7 – PHP код отображения на странице таблицы «orders»

На рисунках 5, 6 и 7 изображен PHP код, в котором на веб-страницу с помощью HTML тегов строятся все три таблицы со всем их содержанием клиентов, продуктов и заказов. Далее рассмотрим результат отображения таблиц на самой веб-странице.

Таблица "customers"

ИД клиента	Имя	Фамилия
1	Николай	Иванов
2	Михаил	Асеров
3	Светлана	Гончарова
4	Дмитрий	Линаев
5	Мария	Попова
6	Евгений	Казанов
7	Анастасия	Морозова
8	Андрей	Кольцов
9	Павел	Никифоров
10	Надежда	Носова
11	Геннадий	Шилов
12	Руслан	Воронов
13	Владимир	Енотов
14	Галина	Лотова
15	Денис	Ростов

Рисунок 8 – Отображенная таблица «customers» на веб-странице

Таблица "products"

ИД продукта	Наименование
1	Молоко
2	Хлеб
3	Сыр
4	Йогурт
5	Орехи
6	Яблоки
7	Помидор
8	Огурец
9	Кефир
10	Груша
11	Арбуз
12	Булка
13	Мясо
14	Тесто
15	Яйца
16	Персик
17	Мандарин
18	Апельсин
19	Сок
20	Изюм

Рисунок 9 – Отображенная таблица «products» на веб-странице

Таблица "orders"

ID заказа	ID клиента	ID продукта	Дата прибытия
1	5	13	2020-07-21
2	12	10	2020-06-04
3	3	1	2020-05-23
4	11	3	2020-06-17
5	5	10	2020-04-08
6	1	9	2020-02-15
7	9	4	2020-01-29
8	15	2	2020-06-16
9	7	8	2020-08-08
10	14	17	2020-02-15
11	13	5	2020-07-11
12	12	11	2020-04-04
13	6	7	2020-02-18
14	2	13	2020-03-10
15	6	8	2020-08-10
16	11	1	2020-07-12
17	14	16	2020-06-14
18	12	5	2020-03-19
19	11	3	2020-01-16
20	7	19	2020-02-09

Рисунок 10 – Отображенная таблица «orders» на веб-странице

Как можно увидеть на рисунках 8, 9 и 10 показан результат отображения трех таблиц клиентов, продуктов и заказов. Теперь перейдем к особенностям внешних ключей.

Внешние ключи позволяют связывать таблицы как родительскую таблицу и дочернюю таблицу, в которой между ними зависимые данные будут контролироваться через обновление или удаление самих строк таблиц. Оператор обновления строк ON UPDATE и оператор удаления строк ON DELETE позволяет внести определенные ограничения к внешним ключам, которые будут рассмотрены ниже:

CASCADE – данное ограничение позволяет автоматически обновлять или удалять строки в дочерней таблице, если в родительской таблице были также удалены или обновлены зависимые строки.

SET NULL – данное ограничение позволяет в дочерней таблице оставлять пустые строки NULL, если в родительской таблице были удалены зависимые строки.

RESTRICT – данное ограничение запрещает обновлять или удалять зависимые строки из родительской таблицы. Разрешается удалять только с таким ограничением строки из дочерней таблицы.

После того как рассмотрели какие бывают виды ограничений внешних ключей, то сначала определим какие отношения будут иметь созданные три таблицы «customers», «products» и «orders». Таблицы клиентов и продуктов будут родительскими, а таблица заказов будет дочерней. Именно в дочерней таблице создаются столбцы внешних ключей, а родительские таблицы не имеют подобных столбцов и сами подключаются к дочерним таблицам.

В данном примере будет рассматриваться ограничение CASCADE на удаление зависимых строк, используя оператор ON DELETE. Теперь для выполнения поставленной задачи необходимо создать на сервере новый PHP файл «change.php» и вписать в него следующий код

```
<?php
$link = new mysqli("localhost", "root", "root", "test");

$query_text = '
DELETE FROM customers WHERE name = "Дмитрий";
DELETE FROM customers WHERE name = "Руслан";
DELETE FROM customers WHERE surname = "Носова";
DELETE FROM customers WHERE surname = "Лотова";
';

mysqli_query( $link, $query_text );

$query_text = "
DELETE FROM products WHERE name = 'Орехи';
DELETE FROM products WHERE name = 'Помидор';
DELETE FROM products WHERE name = 'Изюм';
DELETE FROM products WHERE name = 'Сыр';
";

mysqli_query( $link, $query_text );
```

Рисунок 11 – PHP код удаление строк из таблицы «customers» и «products»

На рисунке 11 продемонстрирован PHP код, в котором формируется запрос на удаление строк родительских таблиц клиентов «customers» и продуктов «products». После выполнения данного PHP файла в веб-браузере, то необходимо открыть файл отображения таблиц «view.php», чтобы посмотреть изменение содержимого трех таблиц на веб-странице

Таблица "customers"

ИД клиента	Имя	Фамилия
1	Николай	Иванов
2	Михаил	Асеров
3	Светлана	Гончарова
5	Мария	Попова
6	Евгений	Казанов
7	Анастасия	Морозова
8	Андрей	Кольцов
9	Павел	Никифоров
11	Геннадий	Шилов
13	Владимир	Енотов
15	Денис	Ростов

Рисунок 12 – Удаленные клиенты с номерами 4, 10, 12 и 14 в родительской таблице «customers»

Таблица "products"

ID продукта	Наименование
1	Молоко
2	Хлеб
4	Йогурт
6	Яблоки
8	Огурец
9	Кефир
10	Груша
11	Арбуз
12	Булка
13	Мясо
14	Тесто
15	Яйца
16	Персик
17	Мандарин
18	Апельсин
19	Сок

Рисунок 13 – Удаленные продукты с номерами 3, 5, 7 и 20 в родительской таблице «products»

На рисунках 12 и 13 изображены родительские таблицы «customers» и «products» с измененным содержимым, где были удалены четыре клиента и продукта.

Далее нужно проверить работу внешних ключей у дочерней таблицы заказов «orders», ведь в ней должны автоматически удалиться заказы с удаленными клиентами и продуктами, так как использовалось для оператора ON DELETE ограничение CASCADE.

Таблица "orders"

ID заказа	ID клиента	ID продукта	Дата прибытия
1	5	13	2020-07-21
3	3	1	2020-05-23
5	5	10	2020-04-08
6	1	9	2020-02-15
7	9	4	2020-01-29
8	15	2	2020-06-16
9	7	8	2020-08-08
14	2	13	2020-03-10
15	6	8	2020-08-10
16	11	1	2020-07-12
20	7	19	2020-02-09

Рисунок 14 – Таблица «orders» с автоматически удаленными заказами номеров 2, 4, 10, 11, 12, 13 и 17

На рисунке 14 показан окончательный результат измененной дочерней таблицы «orders», в ней были удалены заказы, в которых содержались

удаленные клиенты и продукты. Внешние ключи у дочерней таблицы полностью выполнили задачу и удалили зависимые строки, связанные с удаленными вручную строками двух родительских таблиц.

Таким образом, были рассмотрены внешние ключи, позволяющие создавать связи между реляционными таблицами базы данных MySQL. Также был создан собственный пример, в котором была создана база данных с тремя таблицами и к ним были применены внешние ключи для образования связей и зависимостей между их столбцами.

Библиографический список

1. Ряшенцева А.Н. Разработка базы данных для интернет-магазина на примере иммунохимических анализаторов // В сборнике: Виртуальное моделирование, прототипирование и промышленный дизайн. 2018. С. 49-51.
2. Катренко Р.Ю., Рыбанов А.А. Разработка алгоритма получения количественных метрик хранимых процедур баз данных MySQL // Постулат. 2017. № 5-1 (19). С. 107.
3. Шарыпов П.А. Сравнительный анализ пространственных функций баз данных // В сборнике: Альманах научных работ молодых ученых Университета ИТМО. Материалы XLVI научной и учебно-методической конференции. 2017. С. 264-267.
4. Болдырев Е.В. Организация и хранение геоточек в базе данных MySQL // В сборнике: Научное сообщество студентов. Междисциплинарные исследования. Электронный сборник статей по материалам XLVIII студенческой международной научно-практической конференции. 2018. С. 18-22.
5. Лукьянченко А.В., Першенкова В.Г. Оптимизация схем баз данных и запросов в СУБД MySQL // Молодежный научно-технический вестник. 2016. № 5. С. 20.
6. Open Server Panel – Локальный веб-сервер для Windows URL: <https://ospanel.io/> (дата обращения 24.08.2020)
7. MySQL Внешние ключи FOREIGN KEY URL: <https://metanit.com/sql/mysql/2.5.php> (дата обращения 24.08.2020)
8. Использование внешних ключей в MySQL URL: <http://denis.in.ua/foreign-keys-in-mysql.htm> (дата обращения 24.08.2020)