

**Частное учреждение профессиональная образовательная организация
«СТОЛИЧНЫЙ БИЗНЕС КОЛЛЕДЖ»**

ЛАБОРАТОРНАЯ РАБОТА

Пример реализации запросов и подготовленный запрос.

специальность 09.02.03 Программирование в компьютерных системах

Выполнил студент:

Д. Д. Суравенков

(подпись)

Группы 1404ПК.

Руководитель :

(подпись)

Л.А. Киселёв

Москва

2022

Пример реализации SQL запроса и подготовленного запроса в MySql.Data

```
string userInput = Console.ReadLine();  
MySqlCommand command = new MySqlCommand("INSERT INTO table (pol)  
VALUES (@param)", conn);  
command.Parameters.AddWithValue("@param", userInput);  
command.ExecuteNonQuery();
```

Данный код считывает строку ввода пользователя с помощью метода Console.ReadLine() и сохраняет ее в переменной b типа string.

Затем создается новый объект MySqlCommand с заданным SQL-запросом в качестве параметра и связывается с открытым объектом MySqlConnection conn. Конкретно, запрос "INSERT INTO table (pol) VALUES (@b)" вставляет значение переменной b в столбец "pol" таблицы "table". Знак "@" перед параметром "b" указывает на то, что это параметр, который будет заменен на конкретное значение при выполнении запроса.

Далее, метод command.Parameters.AddWithValue("@b", b) добавляет значение переменной b в параметры запроса, чтобы оно могло быть использовано при выполнении запроса. Этот метод заменит знак "@"

перед параметром "b" на значение переменной b, когда будет выполнен объект MySqlCommand.

Наконец, метод command.ExecuteNonQuery() выполняет SQL-запрос на сервере базы данных MySQL, связанный с объектом MySqlConnection conn, и возвращает количество строк, которые были затронуты этим запросом. В данном случае, этот метод вставляет значение переменной b в таблицу "table" и возвращает количество затронутых строк (которое должно быть 1, если операция выполнена успешно).

Таким образом, данный код выполняет SQL-запрос на сервере базы данных MySQL, который вставляет значение переменной `b` в таблицу "table" в столбец "pol".

Соответственно, SQL запрос на добавление в базу данных какой-либо информации можно реализовать в виде:

```
MySqlCommand {имя команды} = new MySqlCommand("INSERT INTO {имя  
таблицы}({имя столбца}) VALUES ({значения})", conn);  
{имя команды}.Parameters.AddWithValue("{параметр запроса}", {имя  
переменной, в которой хранится значение});  
{имя команды}.ExecuteNonQuery();
```

Также данный запрос является примером “подготовленного” запроса. Подготовленный запрос - это концепция, которая используется в базах данных для повышения производительности и защиты от SQL-инъекций. Подготовленный запрос - это запрос SQL, который предварительно компилируется и сохраняется на сервере базы данных. Когда клиентский код отправляет подготовленный запрос на сервер базы данных, сервер уже знает, как выполнять этот запрос. Это уменьшает накладные расходы на парсинг запроса и оптимизацию его выполнения, что приводит к более быстрому выполнению запроса.

Кроме того, использование подготовленных запросов может предотвратить SQL-инъекции, так как значения параметров запроса могут быть переданы на сервер отдельно от самого запроса. Это позволяет защитить базу данных от вредоносного кода, который может быть внедрен в запрос, если значения параметров не будут правильно очищены.

В языке программирования C#, подготовленные запросы реализуются с помощью класса `MySqlCommand` и метода `Prepare()`. При использовании

подготовленных запросов в C# коде, каждый запрос может быть параметризован, то есть значения параметров запроса могут быть переданы в запрос отдельно, что обеспечивает безопасность и повышает производительность выполнения запросов в базе данных.

Пример, как можно реализовать SQL запрос на получение информации из базы данных:

```
MySqlCommand commandShow = new MySqlCommand("SELECT pol FROM  
table", conn);  
MySqlDataReader reader = commandShow.ExecuteReader();  
while (reader.Read())  
{  
    Console.WriteLine(reader.GetString(0));  
}
```

MySqlDataReader связывается с выполненным объектом MySqlCommand commandShow.

Когда выполняется объект MySqlCommand, который содержит SQL-запрос SELECT, он возвращает результат в виде объекта MySqlDataReader.

Метод ExecuteReader() вызывается для выполненного объекта

MySqlCommand commandShow, чтобы получить доступ к результатам запроса SELECT.

Объект MySqlDataReader предоставляет последовательный доступ к строкам результата запроса и предоставляет методы для чтения каждого столбца в текущей строке.

Таким образом, объект MySqlDataReader представляет собой потоковый набор данных, который может быть прочитан построчно, чтобы получить результаты выполненного SQL-запроса.

Цикл `while (reader.Read())` выполняется, пока метод `reader.Read()` возвращает `true`. Этот метод перемещает указатель на следующую строку в объекте `MySqlDataReader` и возвращает `true`, если строка существует, и `false`, если строки закончились.

Внутри цикла, метод `reader.GetString(0)` читает значение из первого столбца текущей строки в виде строки типа `string`. Число `0` в методе `GetString(0)` указывает на то, что мы читаем первый столбец строки, поскольку индексы столбцов в объекте `MySqlDataReader` начинаются с `0`. Затем строка, которая была прочитана, выводится в консоль с помощью метода `Console.WriteLine()`.

Таким образом, данный код читает каждую строку, которая была получена из таблицы "table" при выполнении объекта `MySqlCommand`, и выводит значение первого столбца каждой строки в консоль.