**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ**

**ИМ. Б.Л. РОЗИНГА (ФИЛИАЛ) СПбГУТ**

**(АКТ (ф) СПбГУТ)**

**Отчеты по лабораторным и практическим работам**

**МДК 11.01**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИСПП-35 | |  |  | *Волков Н.А.* |
|  | (Группа) | | (Подпись) | (Дата) | (И.О. Фамилия) |
| Преподаватель | | |  |  | *Маломан Ю. С.* |
|  | |  | (Подпись) | (Дата) | (И.О. Фамилия) |

Архангельск 2024

# **Лабораторная работа №19**

**Выполнение SQL-команд и SQL-подпрограмм**

1. **Цель работы** 
   1. Научиться выполнять SQL-команды и вызывать хранимые процедуры и функции пользователя средствами EF Core.
2. **Контрольные вопросы**

Вопрос: Как выполнить команду на выборку данных в EF Core?

Для выполнения команды на выборку данных в EF Core можно использовать следующие методы:

1. LINQ-запросы:

var users = dbContext.Users.Include(u => u.Address).Where(u => u.Age > 18);

2. Raw SQL-запросы:

var users = dbContext.Users.FromSqlRaw("SELECT \* FROM Users WHERE Age > 18");

3. Stored Procedure:

var users = dbContext.Users.FromSqlProcedure("GetUsersOver18");

4. Table-valued function:

var users = dbContext.Users.FromSqlFunction("GetActiveUsers");

Вопрос: Как выполнить команду на модификацию данных в EF Core?

Для модификации данных в EF Core используются следующие методы:

1. Add:

dbContext.Users.Add(new User { Name = "John Doe" });

2. Update:

var user = dbContext.Users.Find(1);

user.Name = "Jane Doe";

dbContext.Entry(user).Property(e => e.Name).IsModified = true;

3. Remove:

dbContext.Users.Remove(userToDelete);

4. Bulk operations:

dbContext.Users.AddRange(new List<User> { /\* ... \*/ });

5. Execute raw SQL:

dbContext.Database.ExecuteSqlRaw("UPDATE Users SET Name = 'Updated' WHERE Id = 1");

Вопрос: Как объявить и передать параметр в SQL-команду в EF Core?

Для передачи параметров в SQL-команду в EF Core можно использовать следующие способы:

1. Using parameterized queries:

var userId = 1;

var name = "John Doe";

var query = $"SELECT \* FROM Users WHERE Id = {userId} AND Name = '{name}'";

var result = dbContext.Users.FromSqlRaw(query);

2. Using DbParameter:

var userIdParam = new SqlParameter("@UserId", 1);

var nameParam = new SqlParameter("@Name", "'John Doe'");

var query = "SELECT \* FROM Users WHERE Id = @UserId AND Name = @Name";

var result = dbContext.Users.FromSqlRaw(query, userIdParam, nameParam);

3. Using FromSqlRaw with named parameters:

var query = "SELECT \* FROM Users WHERE Id = @UserId AND Name = @Name";

var result = dbContext.Users.FromSqlRaw(query, new { UserId = 1, Name = "John Doe" });

Вопрос: Как вызвать табличную функцию в EF Core?

Табличные функции в EF Core вызываются с помощью метода FromSqlFunction [1]:

var result = dbContext.Users.FromSqlFunction("dbo.GetActiveUsers");

Вопрос: Как вызвать хранимую процедуру в EF Core?

Хранимые процедуры в EF Core вызываются с помощью метода FromSqlProcedure [1]:

var result = dbContext.Users.FromSqlProcedure("sp\_GetUserDetails");

Вопрос: Как объявить и передать выходной параметр из хранимой процедуры в EF Core?

Для работы с выходными параметрами в хранившихся процедурах в EF Core можно использовать следующие подходы:

1. Using DbParameter:

var outputParam = new SqlParameter("@OutputParam", SqlDbType.NVarChar, 50);

outputParam.Direction = ParameterDirection.Output;

var result = dbContext.Database.ExecuteSqlRaw("EXEC sp\_GetUserDetails @OutputParam OUTPUT", outputParam);

var outputValue = outputParam.Value.ToString();

2. Using FromSqlRaw with named parameters:

var query = "EXEC sp\_GetUserDetails @OutputParam OUTPUT";

var result = dbContext.Database.ExecuteSqlRaw(query, new { OutputParam = "" });

var outputValue = result.OutputParam;

1. **Вывод**
   1. В ходе лабораторной работы мы научились выполнять SQL-команды и вызывать хранимые процедуры и функции пользователя средствами EF Core.