Working with Views, Stored Procedures and Raw SQL

Julie Lerman

MOST TRUSTED AUTHORITY ON ENTITY FRAMEWORK

@julielerman   thedatafarm.com

Module Overview

Using migrations to add views & stored procedures to a database

Understanding and mapping keyless entities

Mapping and querying with database views

Querying with raw SQL commands and stored procedures
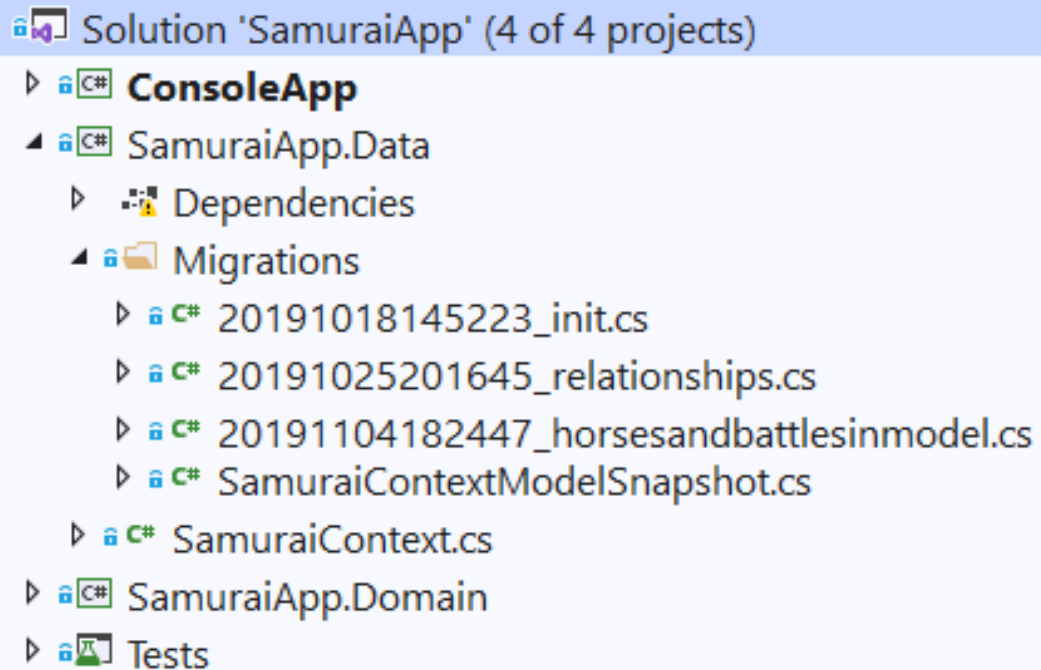
Executing SQL commands on the database

Adding Views and Other Database Objects Using Migrations

Consistent workflows are important whether you are solo or working on a team
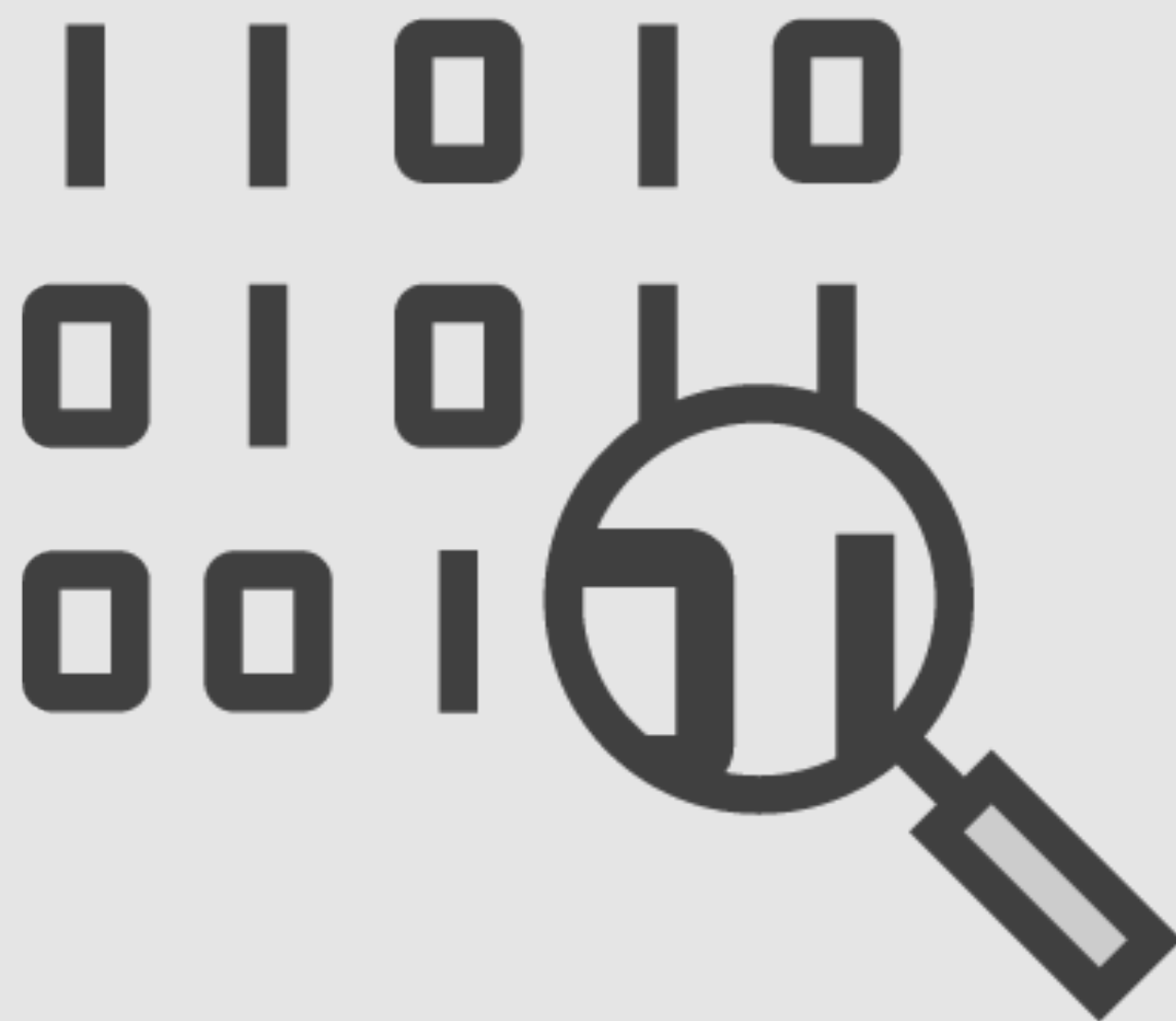
Already Using Migrations for Schema Changes?

Solution 'SamuraiApp' (4 of 4 projects)
- ▷ **ConsoleApp**
- ◢ SamuraiApp.Data
  - ▷ ⚠ Dependencies
  - ◢ Migrations
    - ▷ 20191018145223_init.cs
    - ▷ 20191025201645_relationships.cs
    - ▷ 20191104182447_horsesandbattlesinmodel.cs
    - ▷ SamuraiContextModelSnapshot.cs
  - ▷ SamuraiContext.cs
- ▷ SamuraiApp.Domain
- ▷ Tests

**Include schema changes,
e.g., adding views or creating indexes
in custom migration files**

Using Keyless Entities to Map to Views

Historically,
EF & EF Core only
understood
entities with keys

EF Core 3
can work with
keyless entities

EF Core 2.1/2.2 Query Types
**are now
Keyless Entities**

Keyless Entities != Non-Tracking Queries

Non-Tracking Queries

Entity has a key property

No-tracking is optional

Maps to tables with PK

Keyless Entity

No key property in entity

Will never be tracked

Maps to tables/views with no primary key

Querying the Database Views

Querying with Raw SQL

```
_context.Samurais.FromSQLRaw("some sql string").ToList();

_context.Samurais.FromSQLRawAsync("some sql string").ToList();

_context.Samurais.FromSQLInterpolated($"some sql string {var}").ToList();

_context.Samurais.FromSQLInterpolated($"some sql string {var}").ToList();
```

DbSet Methods to Run Raw SQL

Synchronous and asynchronous options

Special method for interpolated strings

Creates an IQueryable, so you still need an execution method

Use parameters to avoid SQL injection!!

Raw SQL Results Rules (Limitations)

✓ **Must return data for all properties of the entity type**

✓ **Column names in results match mapped column names**

✓ **Query can't contain related data**

✓ **Only query entities and keyless entities known by DbContext**

Breaking Change!

Method names have changed from earlier versions of EF Core!

Interpolated Strings

```
var character="Farmer Manzô";

var someString=$"Happy Birthday to you, {character}";
```

Running Stored Procedure Queries with Raw SQL

Formatted Strings

```
var character="Farmer Manzô";

var age=78;


var s= String.Format("Happy Birthday to you, {0}",
                     character);

var s= String.Format("Happy {0} Birthday to you, {1}",
                     age,character);
```

Executing Non-Query Raw SQL Commands

```
_context.Database.ExecuteSQLRaw("some SQL string");

_context.Database.ExecuteSQLRawAsync("some SQL string");

_context.Database.ExecuteSQLInterpolated($"some SQL string {variable}");

_context.Database.ExecuteSQLInterpolatedAsync($"some SQL string {var}");
```

Run Raw SQL for Non-Query Commands

Only result is number of rows affected

On-the-fly SQL or Stored Procedures

Review

Used customized migrations to add views, stored procedures and more to DB

Learned about keyless entities and mapped them to a database view

Used DbSet to query that database view

More querying using raw SQL and stored procedures including interpolated strings

Execute non-query commands from the DbContext.Database property

...and a handy way to delete entities!

Resources

Entity Framework Core on GitHub github.com/aspnet/entityframework

EF Core Roadmap bit.ly/efcoreroadmap

EF Core Documentation docs.microsoft.com/ef

EF Core 3 Breaking changes: Query Types Consolidated with Entity Types
docs.microsoft.com/en-us/ef/core/what-is-new/ef-core-3.0/breaking-changes#qt

Composing Raw SQL Queries with LINQ
docs.microsoft.com/en-us/ef/core/querying/raw-sql#composing-with-linq

Entity Framework Core 3.0: A Foundation for the Future
codemag.com/Article/1911062/Entity-Framework-Core-3.0-A-Foundation-for-the-Future

"Mapping Database Scalar Functions" module of EF Core 2: Mappings Course
bit.ly/2LppcMj

Up Next!
EF Core in an
ASP.NET Core app