Using Database Views with EF Core 2.0

Julie Lerman
MOST TRUSTED AUTHORITY ON ENTITY FRAMEWORK
@julielerman thedatafarm.com

Working with views is
limited in EF Core 2.0,
but mapping to views
will be supported
in EF Core 2.1.

It’s still possible, in 2.0, to query against database views and capture their results.

Module Overview

- Understand limited use of views in 2.0
- Defining classes to receive view results
- Querying against views
- Force migrations to create a view
- Some “gotchas” to watch out for
- Prepare for improved support in 2.1

Identifying the Scope of Working with Views in EF Core 2.0

How EF Core Sees Database Views

EF Core doesn't know that views are different than tables

Views are composable (filter, sort, etc.)

Use LINQ to query database views

Migrations doesn't comprehend views

Scaffold doesn't comprehend views

How EF Core Sees Database Views

EF Core doesn't know that views are different than tables

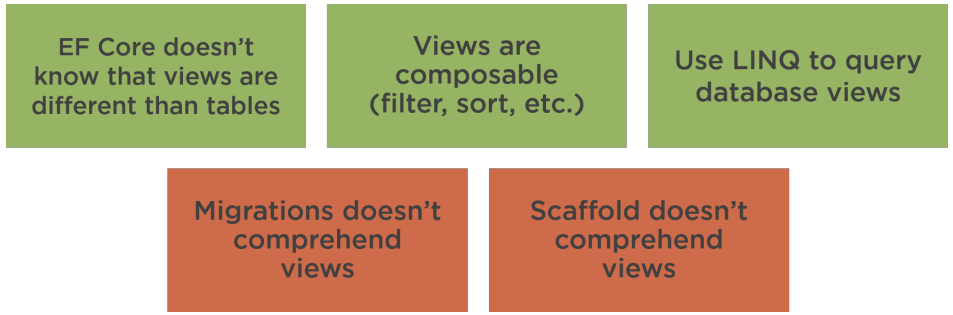
Views are composable (filter, sort, etc.)

Use LINQ to query database views

Migrations doesn't comprehend views

Scaffold doesn't comprehend views

Your Responsible for:



✓Create the database view

✓Create the class to capture view results

Recipe for Querying Views

Database view

Entity to capture results

Key property in the entity

DbSet of the entity

Creating the Entity and DbSet for Querying

It is possible to
query a view,
then use a stored procedure
to update.

No Key? Create a Composite Key

Note: It's best to use non-nullable properties

```
public class SamuraiStat
{
  public string Name { get; private set; }
  public int NumberOfBattles { get; private set; }
  public string EarliestBattle { get; private set; }
}
```

```
modelBuilder.Entity<SamuraiStat>()
            .HasKey(s => new {
                              s.Name,
                              s.EarliestBattle,
                              s.NumberOfBattles
                             });
```

```
CREATE VIEW
 dbo.SamuraiBattleStats
```

```
public DbSet<SamuraiStat>
 SamuraiStatsView {get; set;}
```

```
modelBuilder
.Entity<SamuraiStat>()
.ToTable("SamuraiBattleStats");
```

- View name: SamuraiBattleStats
- DbSet name: SamuraiStatsView
- Use ToTable mapping for the db's view name

Querying the Database Views

AsNoTracking Queries Against Views

Avoids possible exception if change tracker detects duplicate keys

Performance benefit when not creating and managing state information

Preparing for Some Potential Problems

A Peek at View Mappings in EF Core 2.1

EF Core 2.1 Query Type

Define non-entities in DbContext

Classes do not need key properties

Don’t map to database tables

Objects are not tracked

Defining a DbQuery and Query Type for a View

```
public DbSet<Samurai> Samurais { get; set; }
public DbSet<Quote> Quotes { get; set; }
public DbSet<Battle> Battles { get; set; }
public DbQuery<SamuraiStat> SamuraiStats { get; set; }
```

```
modelBuilder.Query<SamuraiStat>()
            .ToView("SamuraiBattleStats");
```

```
context.SamuraiStats.ToList();
```

More to come for view support in migrations and scaffolding post-2.1

Query Types Rock!

(in EF Core 2.1)

Review

Database views are not officially supported in 2.0

You can still query against them

You are responsible for creating the entity and the view

Pay attention to key properties, migrations and special DbContext code

EF Core 2.1 query types will provide better support

Resources

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

EF Core Roadmap bit.ly/efcoreroadmap

EF Core Documentation docs.microsoft.com/ef

GitHub conversation about views in 2.0 and 2.1:
https://github.com/aspnet/EntityFrameworkCore/issues/9290

EF Core 2.1 Query Types Docs docs.microsoft.com/en-us/ef/core/modeling/query-types

Entity Framework Core 2: Mappings

Julie Lerman
MOST TRUSTED AUTHORITY ON ENTITY FRAMEWORK
@julielerman thedatafarm.com