Temporal table using EF in .Net 6

What is use of temporal table?

Temporal feature introduced in EF core 6 on .net 6 supported.

It’s used to storing records of data changes and enabling analysis of the table data in different point in time.

History table recording changes to data in the original table along with their UTC timestamps,

1. Record is Update / Delete in the main table before that values are copied and stored in the history table

Install Dotnet Cli in the project directory path from command on your terminal   
  
dotnet tool update --global dotnet-ef

Create a LoginUser model class

public class LoginUser : BaseEntity, ISoftDelete

{

public string FirstName { get; set; }

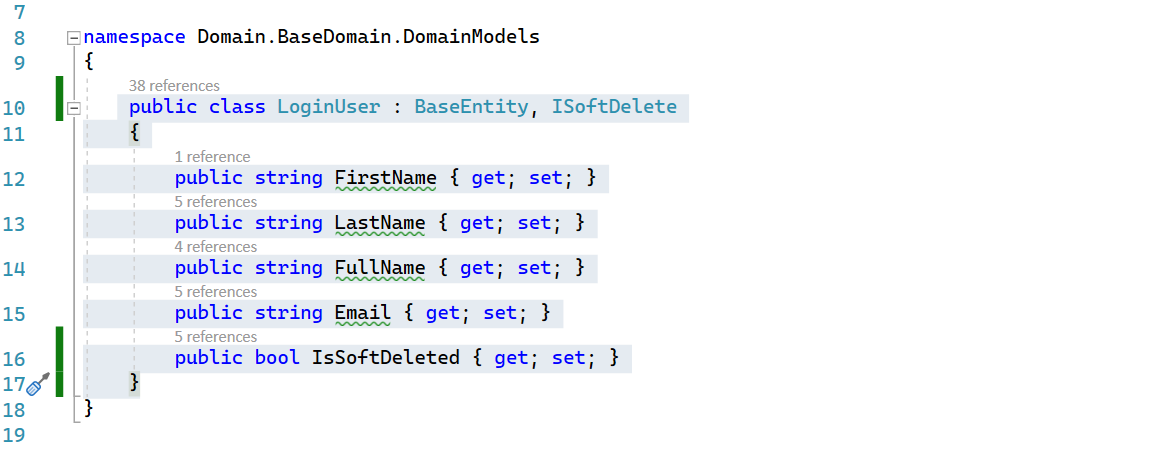
public string LastName { get; set; }

public string FullName { get; set; }

public string Email { get; set; }

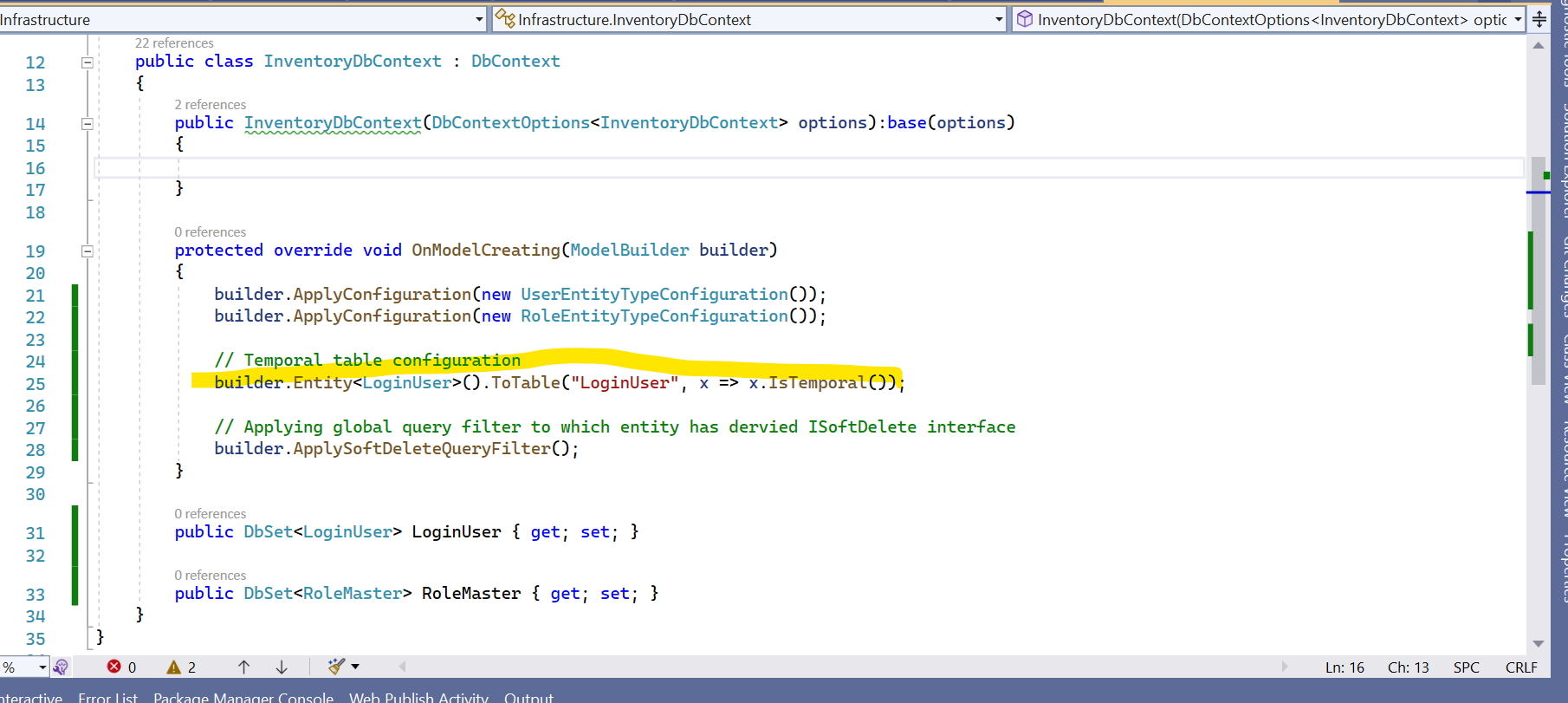
public bool IsSoftDeleted { get; set; }

}

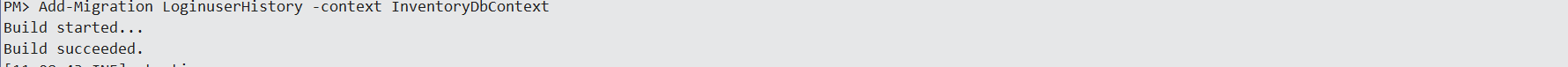


Create a InventoryDbContext class and inherit DbContext base class from EntityFramework library.

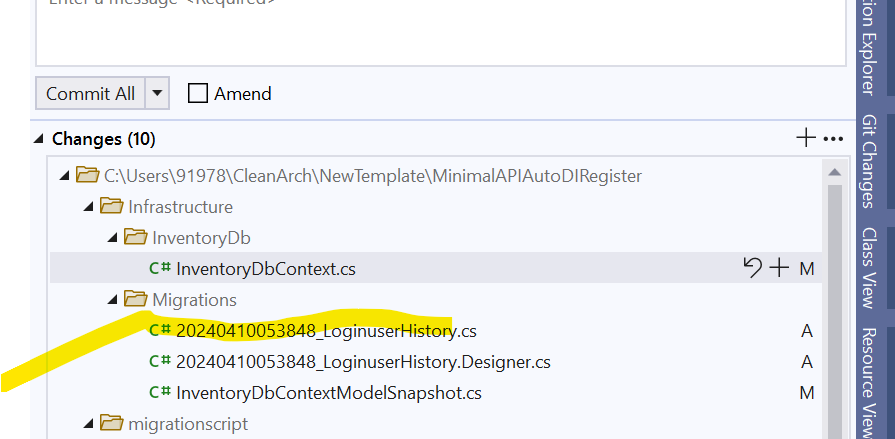
**OnModelCreating**() method we specify that **LoginUser** table should be created as temporal table



Add migration using PMC

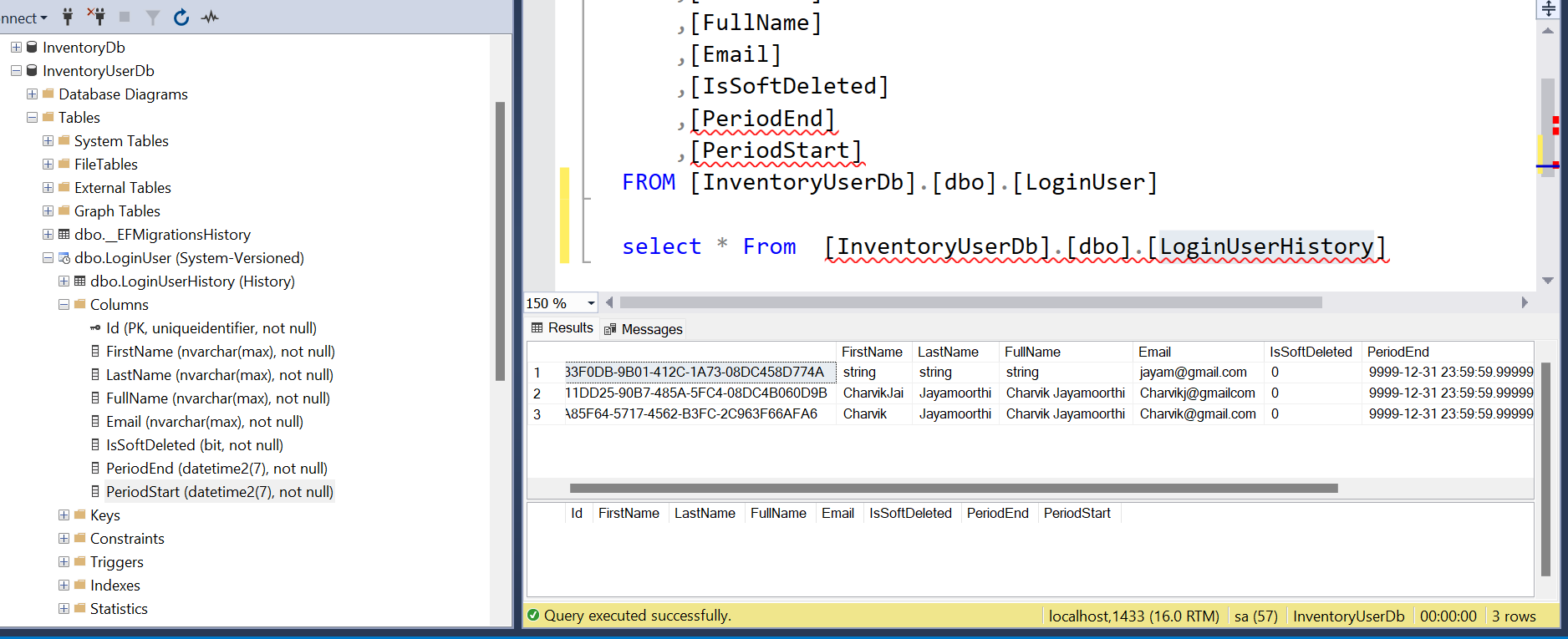


migration class file generated has successfully



Now, go to db and refresh,

[PeriodEnd] and [PeriodStart] field were added in the main table (LoginUser) and history LoginUserHistory table has been created.



How do read data from history table?

You can use various EF LINQ extension methods to query the historical data.

**TemporalAsOf** returns elements that were present in database at a given point in time.

**TemporalAll**returns all historical versions of the entities as well as their current state.

**TemporalFromTo** returns elements that were present in the database between two given points in time. Elements created at the starting point and removed at the end point are not included.

**TemporalBetween** returns elements that were present in the database between two given points in time. Elements created at the starting point are not included, elements removed at the end point are included.

**TemporalContainedIn** returns elements that were present in the database between two given points in time. Elements created at the starting point and removed at the end point are included.

Add new method for Getting history table

public interface IUserRepository: IRepository<LoginUser>

{

Task<List<LoginUser>?> GetTemporalAllUsersQueryAsync(Guid id);

}