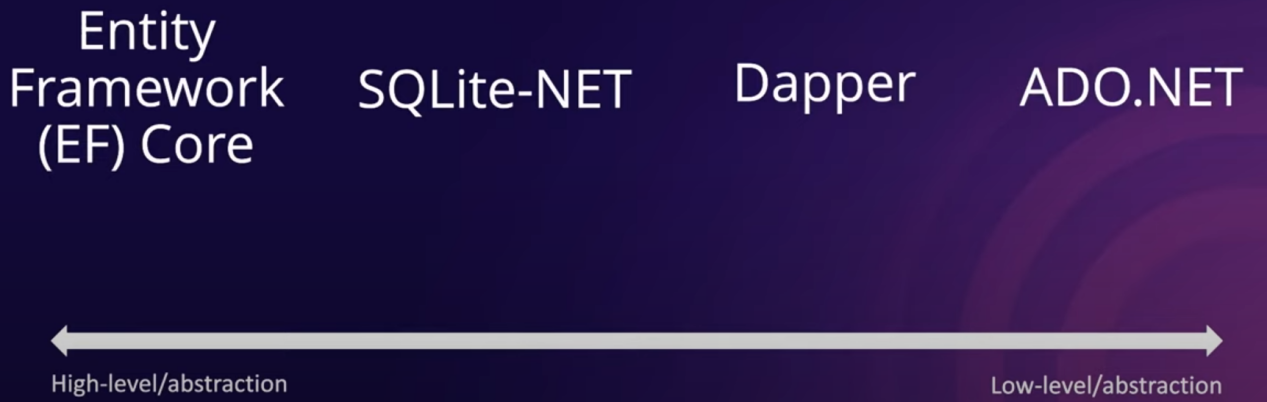
# DB and ORM

## Seeding

## Level of abstraction

Level of abstraction to use c# with SQLite database



Refer <https://github.com/BalajiBaskaran24/CSharp-AccessSQL-Demo> repo for the sample code for all above layers

### EF Core

LINQ instead of SQL

Data as objects

### SQLite-NET

Mobile, LINQ instead of SQL, Data as objects

### Dapper

Data as objects, abstracted database connection, lightweight

### ADO.Net

No abstraction, works for all scenarios

## Database

### Relational Database

#### Join

Combines data or rows from two or more tables based on a common field between them. Related to each other using foreign key constraints.

***SELECT*** *column\_list*

***FROM*** *table1*

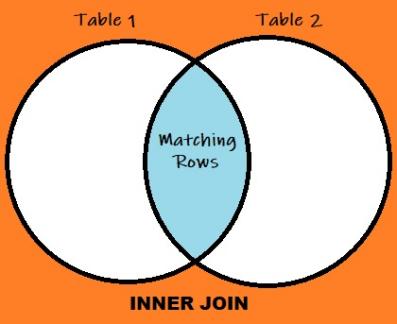
***INNER JOIN/LEFT JOIN/RIGHT JOIN/FULL JOIN*** *table2*

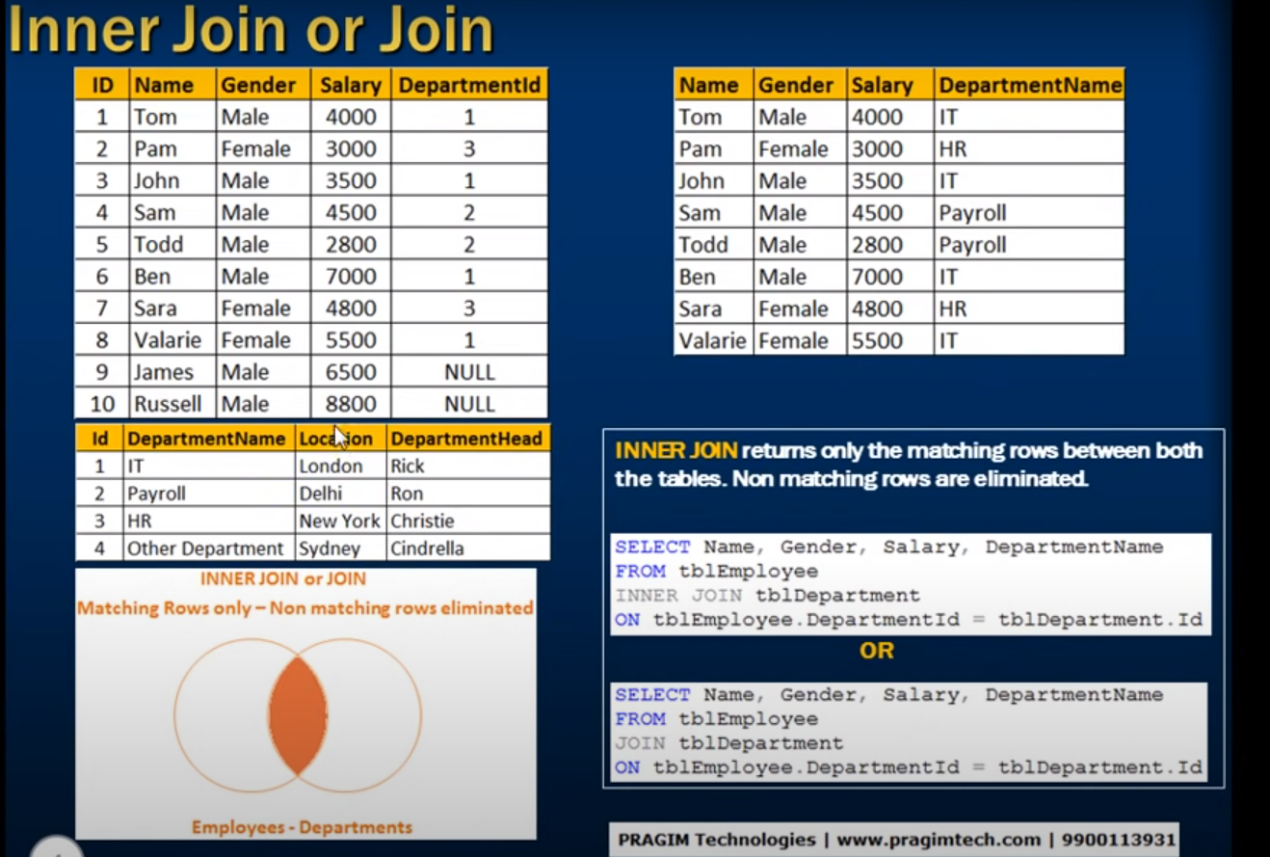
***ON*** *table1.column\_name = table2.column\_name;*

*Use* ***where*** *keyword to include additional conditions like* ***null or not null***

##### Inner Join

Matching rows only, non matching rows are eliminated

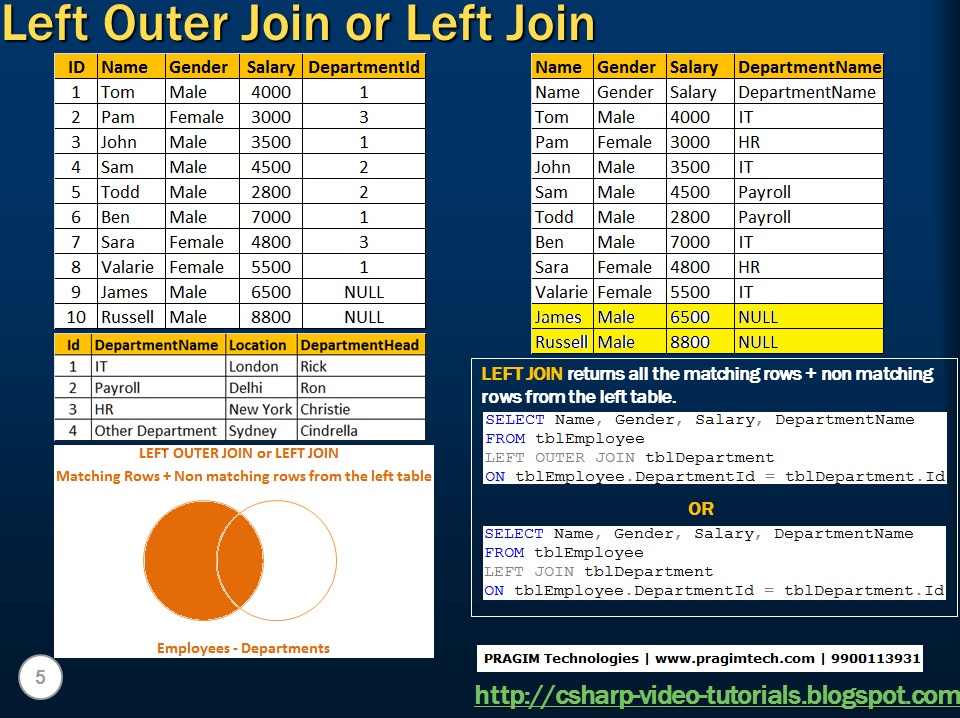




##### Outer Join

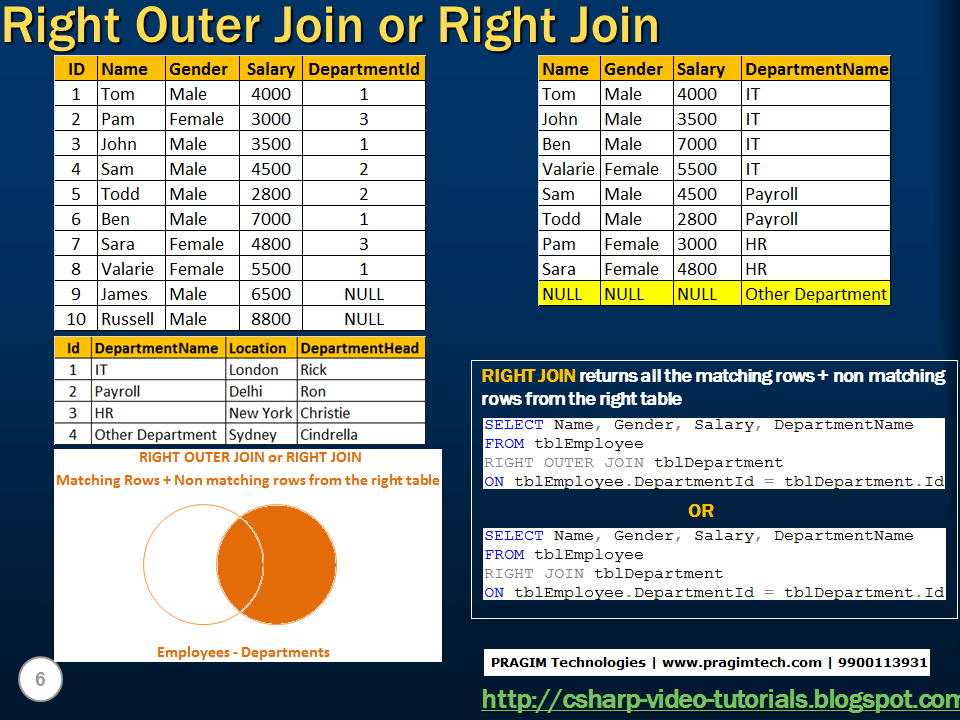
###### Left

All the matching rows + non matching rows from the left table



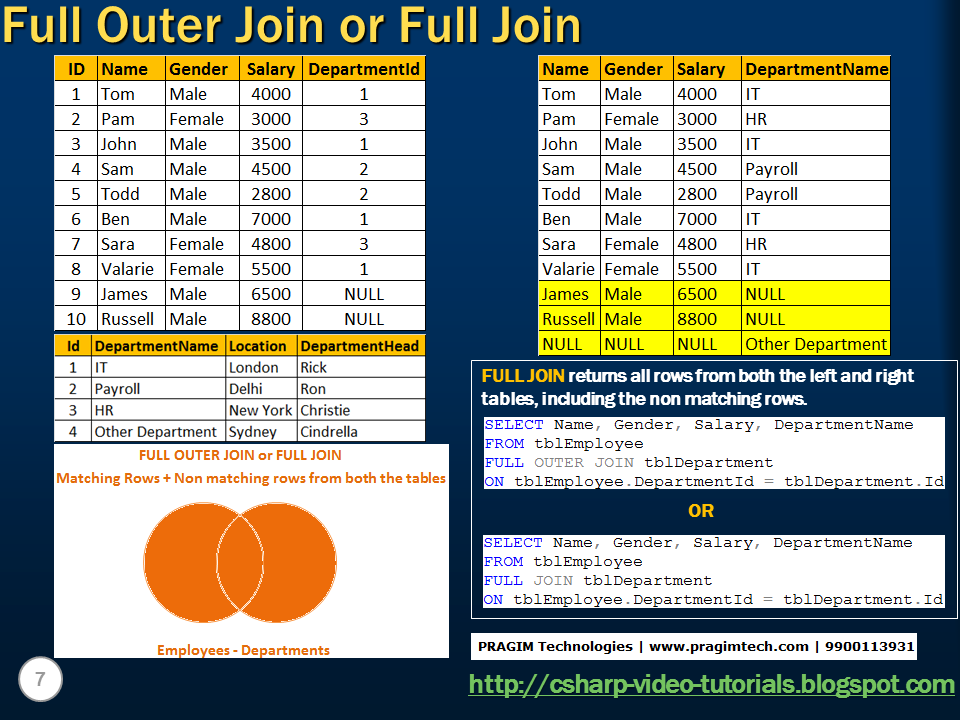
###### Right

All the matching rows + non matching rows from the right table



###### Full

All rows from left and right including non matching rows



##### Cross Join

Cartesian product of two tables involved in the join



##### Hash Join

### IDbConnection Interface

Represents an open connection to a data source, and is implemented by .NET data providers that access relational databases.

**Properties**

|  |  |
| --- | --- |
| **[ConnectionString](https://learn.microsoft.com/en-us/dotnet/api/system.data.idbconnection.connectionstring?view=net-7.0" \l "system-data-idbconnection-connectionstring)** | Gets or sets the string used to open a database. |
| **[ConnectionTimeout](https://learn.microsoft.com/en-us/dotnet/api/system.data.idbconnection.connectiontimeout?view=net-7.0" \l "system-data-idbconnection-connectiontimeout)** | Gets the time to wait (in seconds) while trying to establish a connection before terminating the attempt and generating an error. |
| **[Database](https://learn.microsoft.com/en-us/dotnet/api/system.data.idbconnection.database?view=net-7.0" \l "system-data-idbconnection-database)** | Gets the name of the current database or the database to be used after a connection is opened. |
| **[State](https://learn.microsoft.com/en-us/dotnet/api/system.data.idbconnection.state?view=net-7.0" \l "system-data-idbconnection-state)** | Gets the current state of the connection. |

**Methods**

|  |  |
| --- | --- |
| **[BeginTransaction()](https://learn.microsoft.com/en-us/dotnet/api/system.data.idbconnection.begintransaction?view=net-7.0" \l "system-data-idbconnection-begintransaction)** | Begins a database transaction. |
| **[BeginTransaction(IsolationLevel)](https://learn.microsoft.com/en-us/dotnet/api/system.data.idbconnection.begintransaction?view=net-7.0" \l "system-data-idbconnection-begintransaction(system-data-isolationlevel))** | Begins a database transaction with the specified [IsolationLevel](https://learn.microsoft.com/en-us/dotnet/api/system.data.isolationlevel?view=net-7.0) value. |
| **[ChangeDatabase(String)](https://learn.microsoft.com/en-us/dotnet/api/system.data.idbconnection.changedatabase?view=net-7.0" \l "system-data-idbconnection-changedatabase(system-string))** | Changes the current database for an open Connection object. |
| **[Close()](https://learn.microsoft.com/en-us/dotnet/api/system.data.idbconnection.close?view=net-7.0" \l "system-data-idbconnection-close)** | Closes the connection to the database. |
| **[CreateCommand()](https://learn.microsoft.com/en-us/dotnet/api/system.data.idbconnection.createcommand?view=net-7.0" \l "system-data-idbconnection-createcommand)** | Creates and returns a Command object associated with the connection. |
| **[Dispose()](https://learn.microsoft.com/en-us/dotnet/api/system.idisposable.dispose?view=net-7.0" \l "system-idisposable-dispose)** | Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.  (Inherited from [IDisposable](https://learn.microsoft.com/en-us/dotnet/api/system.idisposable?view=net-7.0)) |
| **[Open()](https://learn.microsoft.com/en-us/dotnet/api/system.data.idbconnection.open?view=net-7.0" \l "system-data-idbconnection-open)** | Opens a database connection with the settings specified by the ConnectionString property of the provider-specific Connection object. |

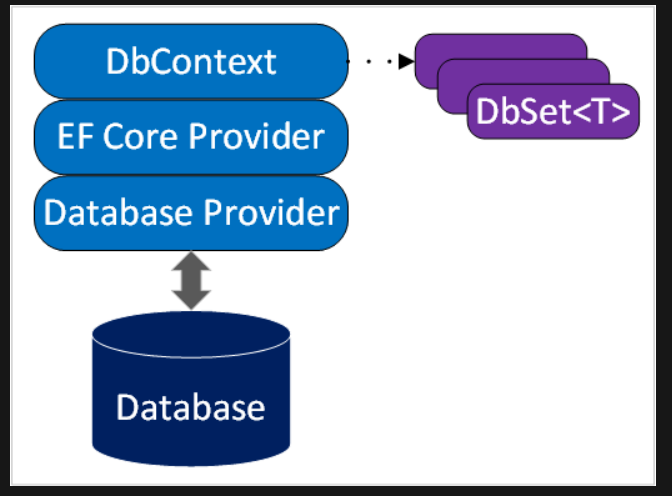
### Connection string

## Dapper - Lightweight

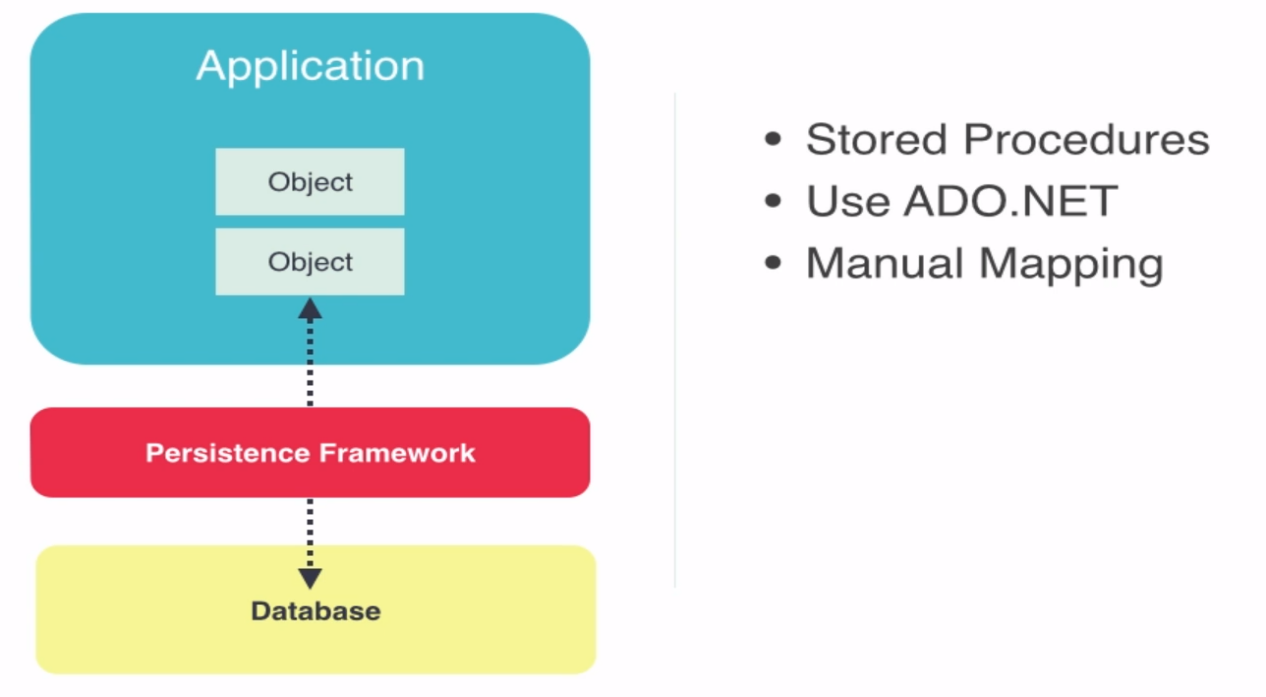
Identity will be used to set the value for particular column in increment seed value provided by user.

Sample code in github

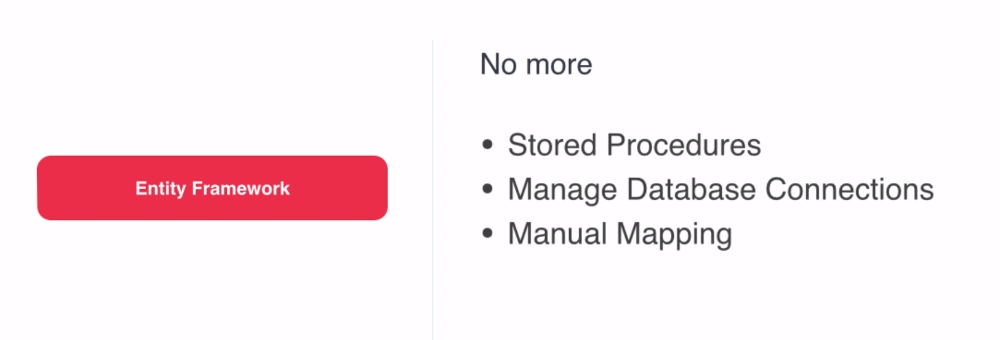
## EF Core Architecture



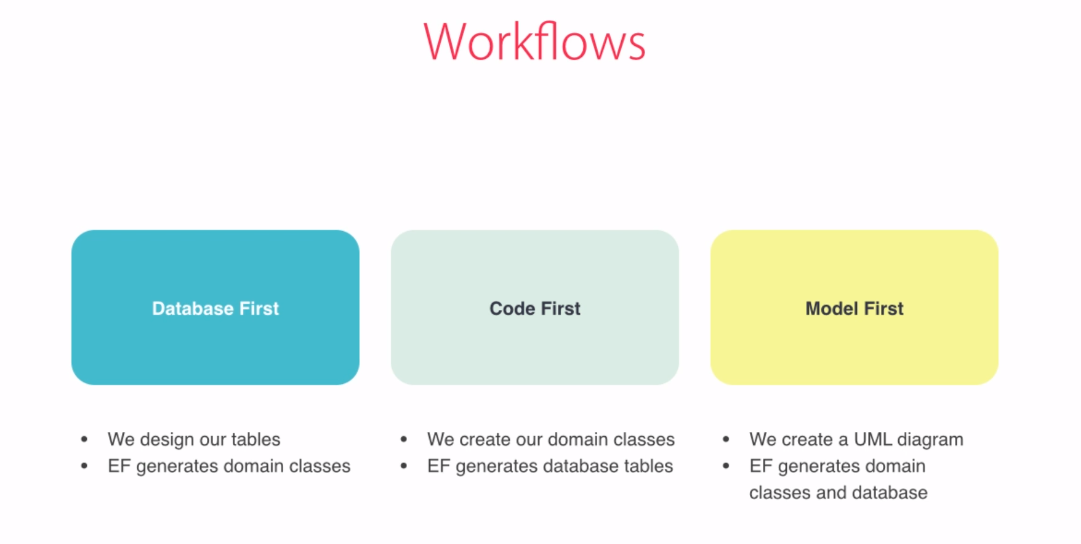
Using SqlConnection, SqlCommand, SqlDataReader



Using EF



Workflows



### Version

EntityFramework 6

EntityFramework Core - No longer developed

### Required Nuget Packages

Microsoft.EntityFrameworkCore

Microsoft.EntityFrameworkCore.SqlServer *(implicitely includes Microsoft.EntityFrameworkCore)*

Microsoft.EntityFrameworkCore.Design

### Migration

To add migration, In powershell, use the following command

*add-migration MigrationName*

To apply migration to database,

*update-database*

Migration Snapshot

*\_EFMigrationHistory*

To remove migration

*remove-migration*

Note: Migration that is added, but not updated to database can be removed. Removing a migration that is already updated to database will create problem. Still if you need to delete the migration make sure that you are very sure about the change involved in that migration.

Removing the table

Removing the DbSet from DbContext class will generate a migration to drop the table

Revert back to old migration

*update-database MigrationName*

Above command will take you to the mentioned migration name

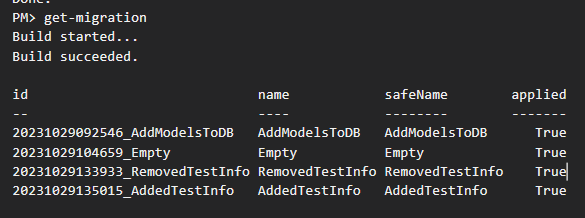
Try,

*Update-database* again to comeback to latest

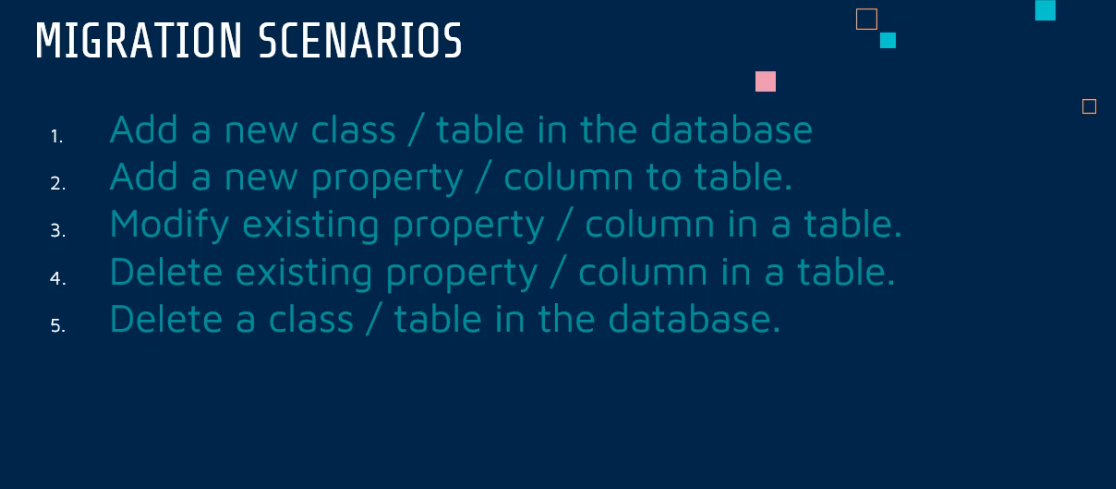
To list all migration and apply status,

*get-migration*

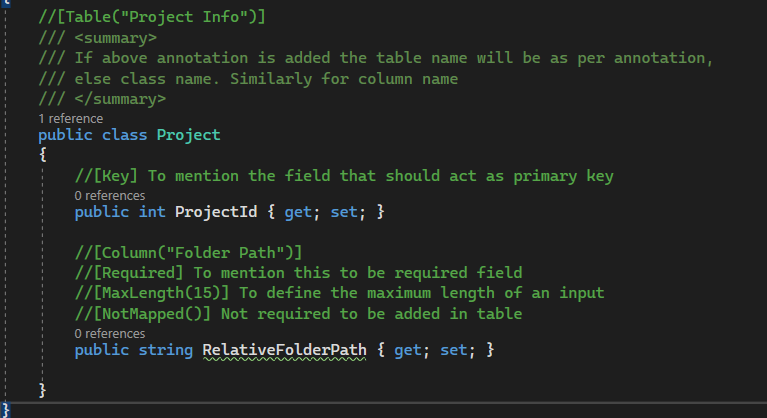
*Drop-database .* Command to drop the database



When migration should be added,

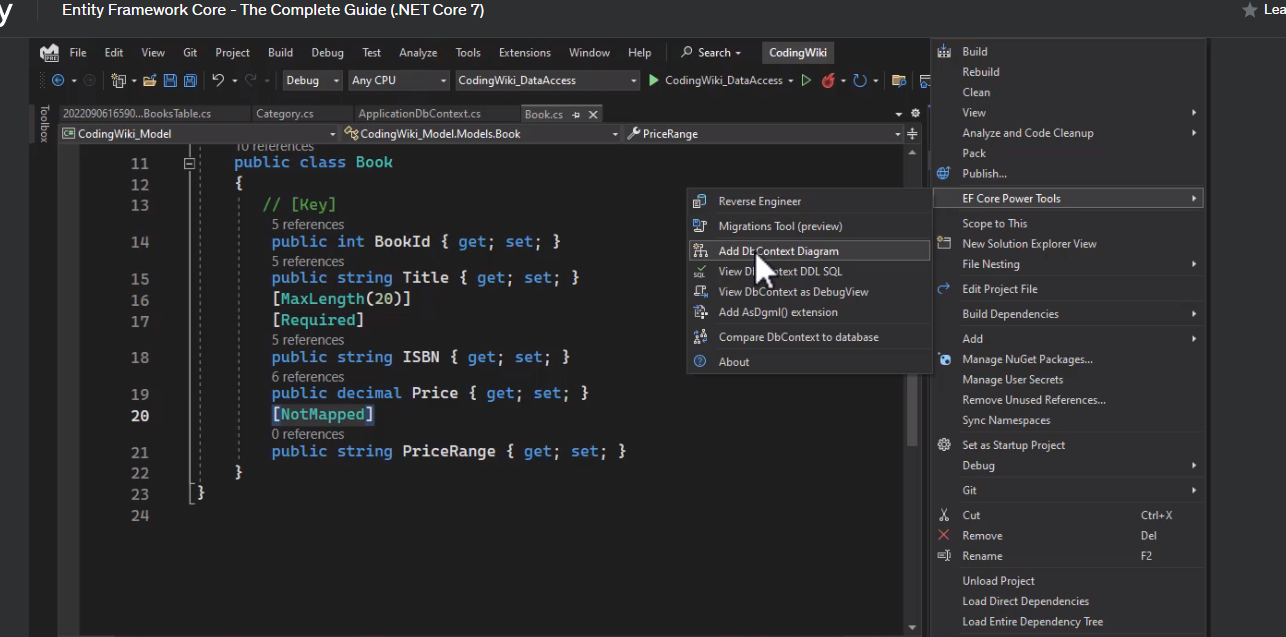


### Annotation



### Database diagram

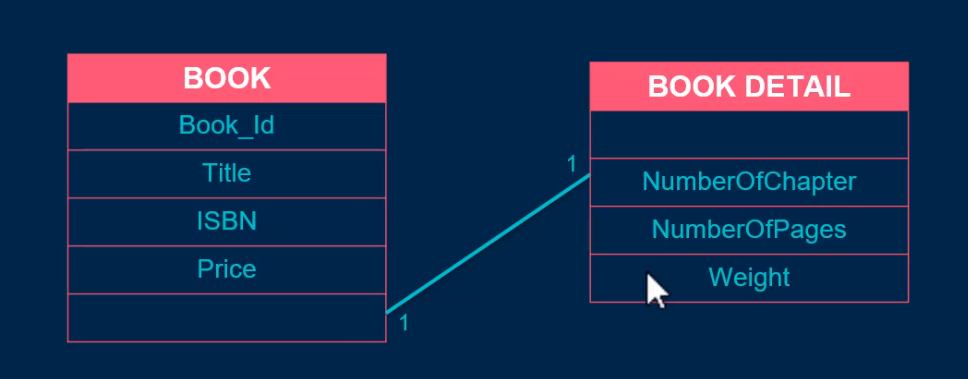
Install EF core power tools in visual studio market place. Then right-click on solution and refer below screenshot.



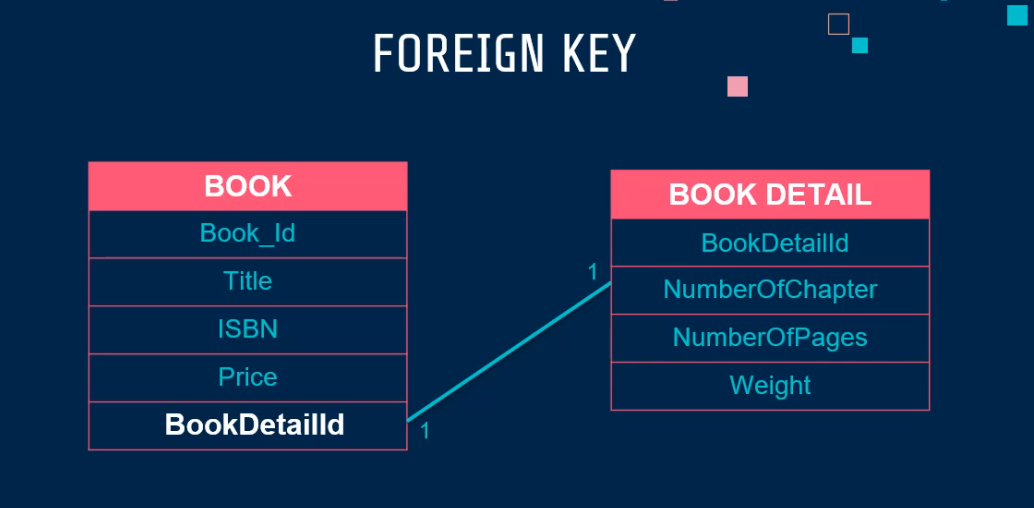
### Relations in EF core

#### One to one

Tables

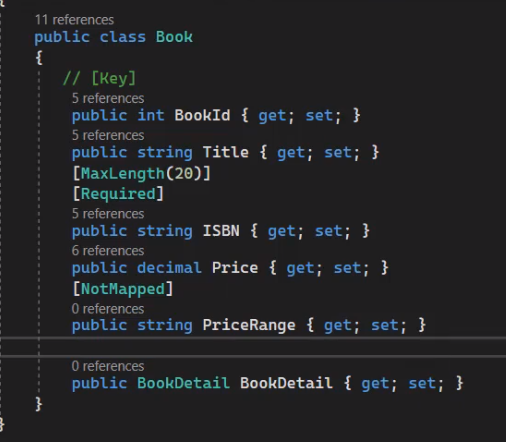


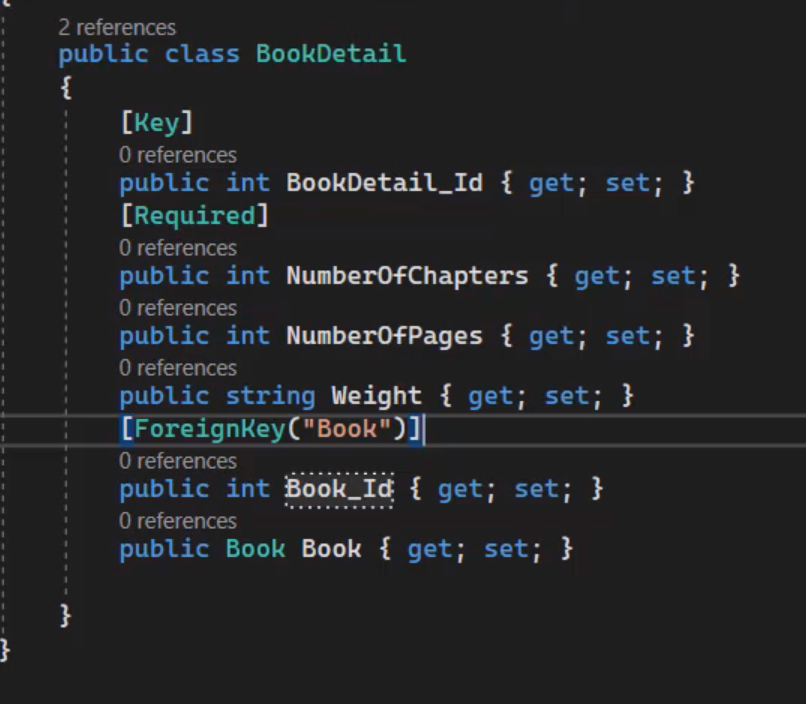
Tables with relationship

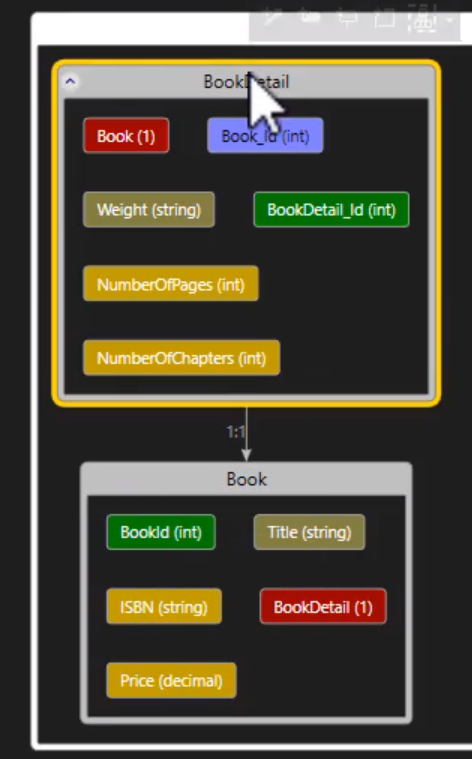


**Code**

1. In Book class, BookDetail property
2. In BookDetail class, property for Book
3. In BookDetail class, property for Book\_Id with [ForeignKey(“Book”)]. Book is proprety name, not class name



**“Book”** - Property name, not class name 



#### One to many

One book can have only one publisher. But one publisher can publish multiple books

