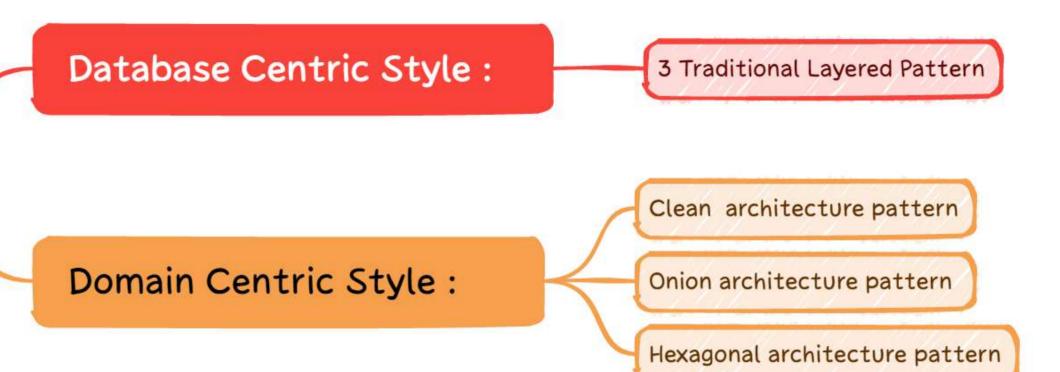
## System Design

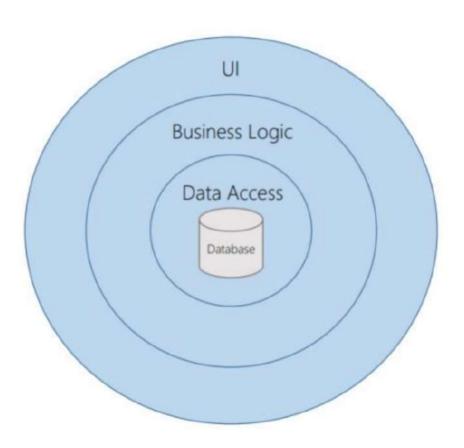
Part-I software architecture and design

Ch-2 Software Architecture Styles and Pattern

# Software Development Architecture Style:



#### **Database Centric**



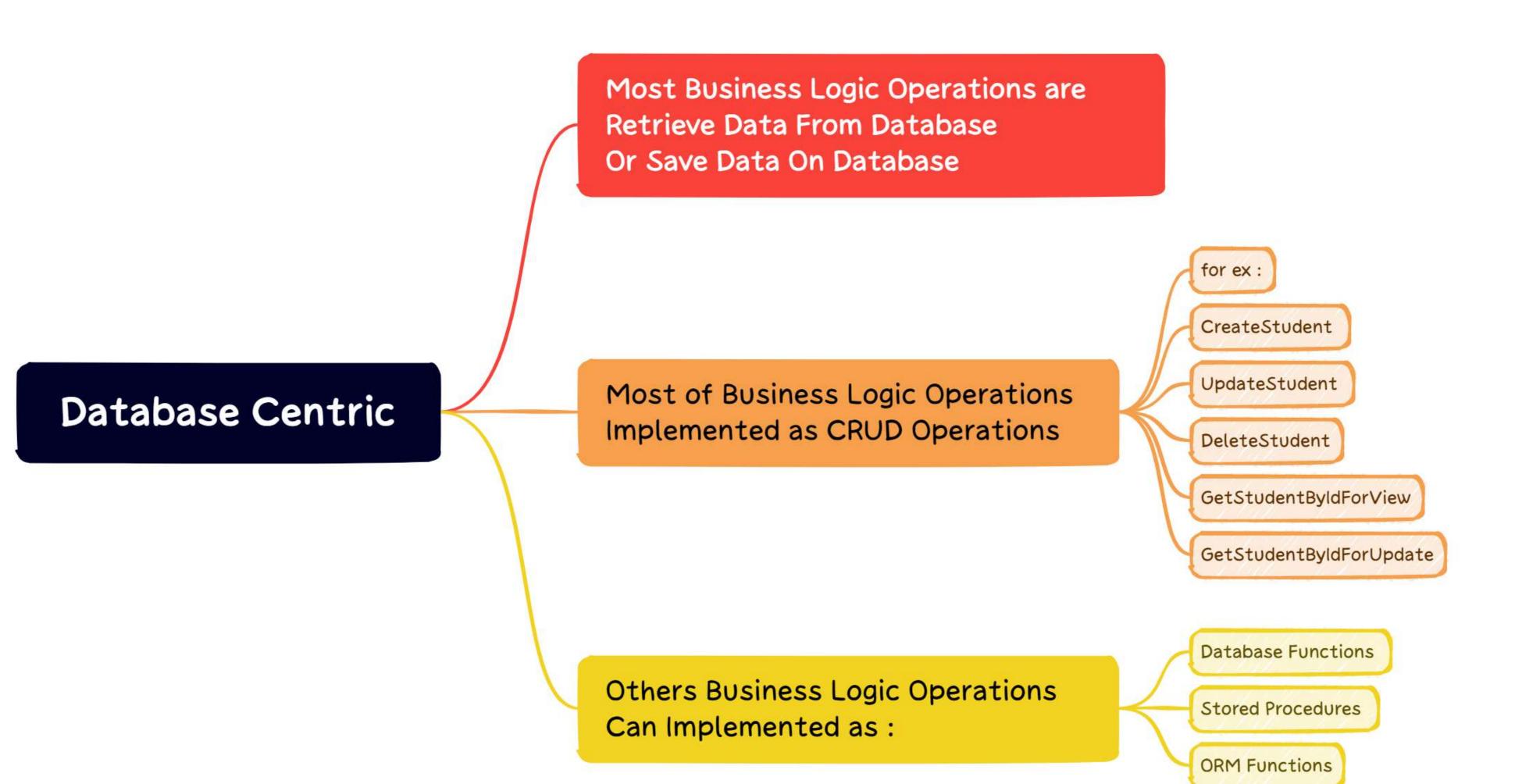
### Database is The Core Of The System

### Database Centric

Usually Database Model is Design First Then

Domain Model or The Application Code.

Business Logic and Application Code are Dependent on How Database Work?



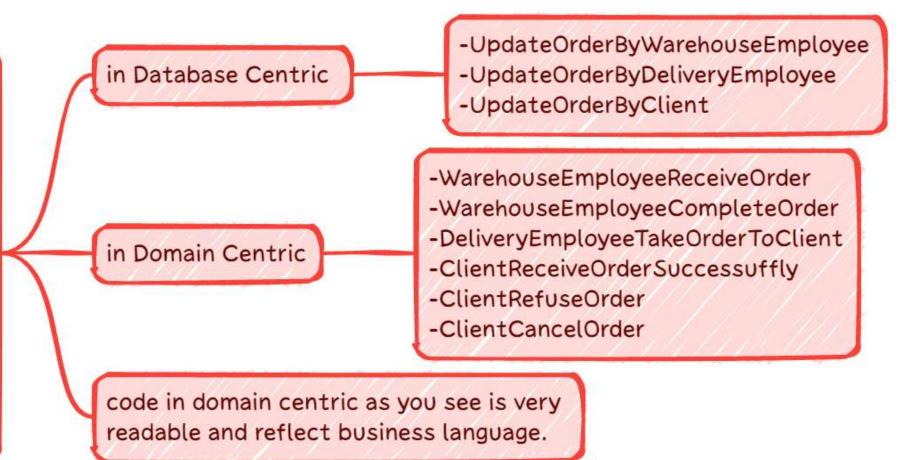
## Disadvantages of Database Centric

Application Code and Business Logic are Coupled and Depend On Database.

With The Growth of Data On Your System, You May Need To Change Your Database or Shift Some Part of Your Database To Another Type of Database. The Application and Business Logic is Very Coupled on Database and ORM Frameworks, So Changing Current Database or Current ORM to another new Technology Can Cause a lot of Refactoring for ex: SQL Server To MongoDB Ado.net Framework To Entity Framework Core

## Disadvantages of Database Centric

When The Business Operation
Has a lot Of Different Steps,
Different Scenarios and Used
By Different Users, Writing
Business Logic as CRUD
Operations is Non-Readable
and Very Very Complex To
Understand By The Developers



Most Business Logic Operations are Retrieve Data From Database Or Save Data On Database

## When To Use Database Centric?

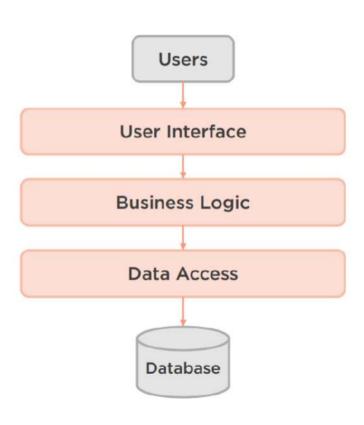
Business Logic is Simple and Straightforward

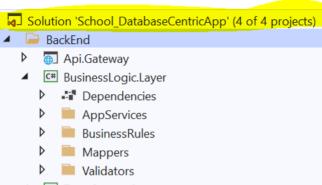
No Complex Business Logic or Different Scenarios Per Single Business Logic Operation Or Application Use Cases

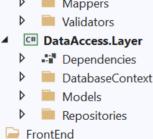
There are no a lot of third parties, integration services, external services or extra infrastructure.

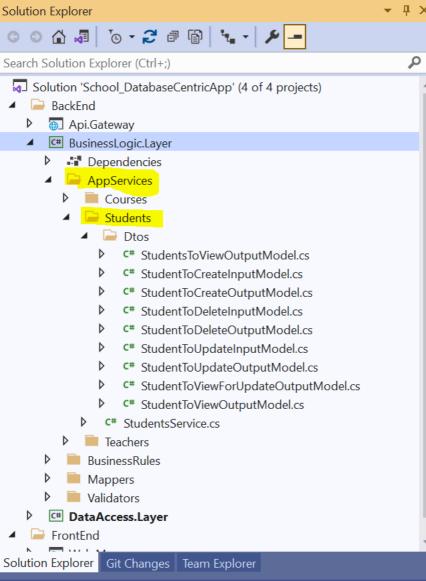
### Database Centric Pattern Example:

#### Traditional 3 Layers Architecture



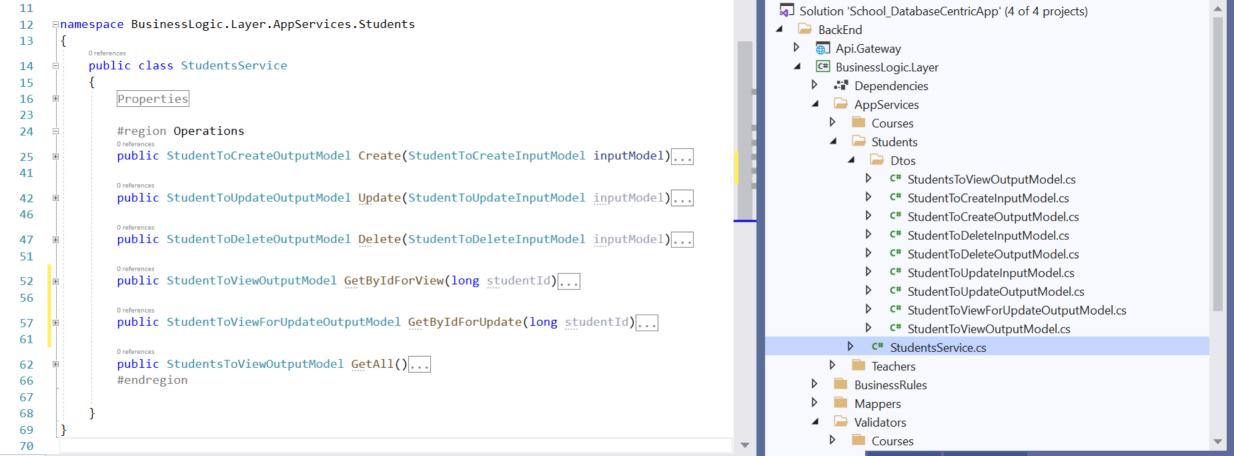






#### Solution Explorer Search Solution Explorer (Ctrl+:) Solution 'School DatabaseCentricApp' (4 of 4 projects) BackEnd Api.Gateway C# BusinessLogic.Layer Dependencies **AppServices** BusinessRules D Courses Students D C# StudentAgeShouldBeGreaterThan7YearsBusinessRule.cs C# StudentNameShouldNotContainAnvNumbersBusinessRule.cs C# StudentNationalIdShouldBeLegalBusinessRule.cs C# StudentPhoneNumberShouldBeUniqueBusinessRule.cs C# StudentPhoneNumberShouldContainNumbersOnlyBusineeRule.cs Teacher Mappers Validators C# DataAccess.Layer FrontEnd Web Myc

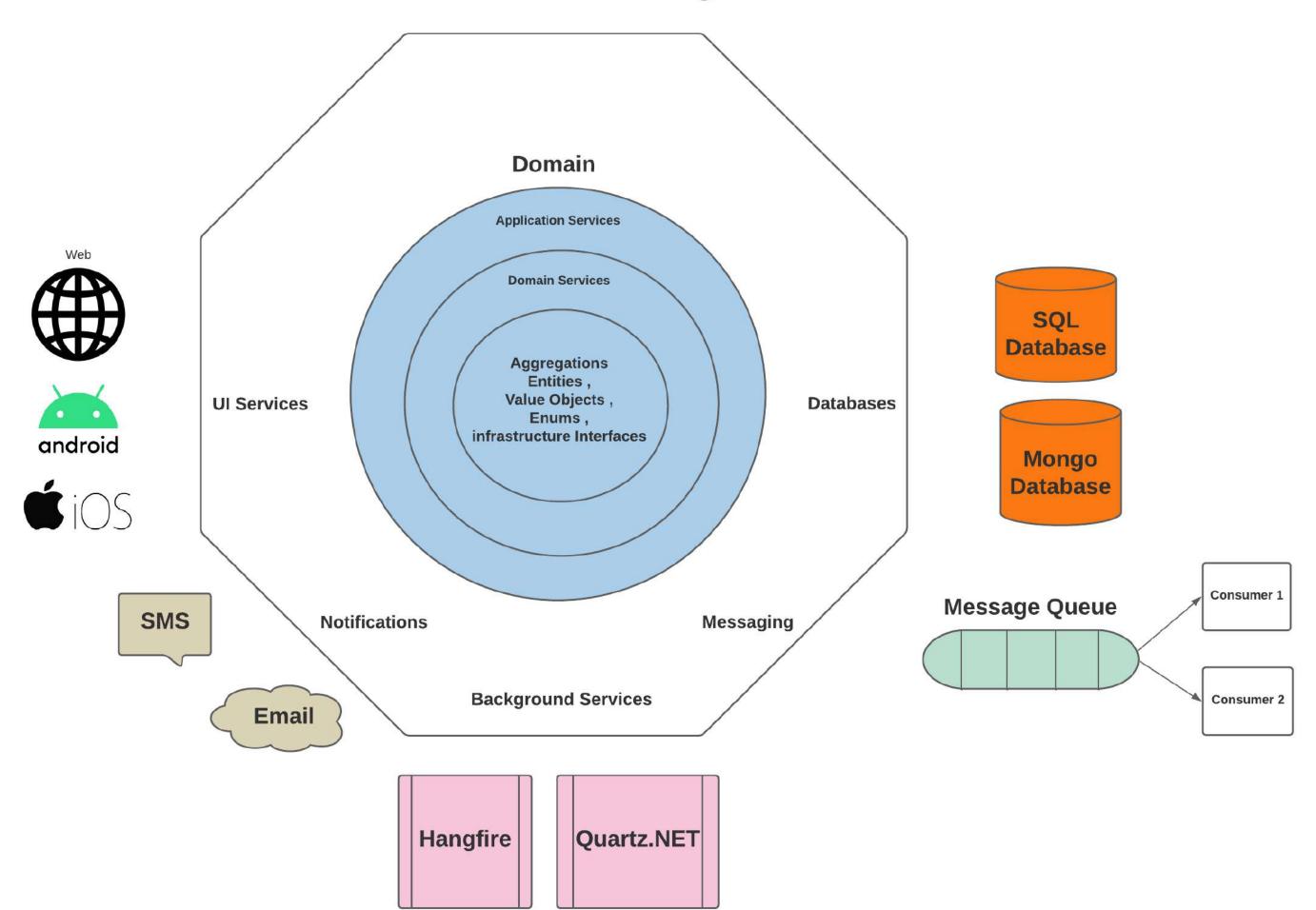
```
Solution 'School_DatabaseCentricApp' (4 of 4 projects)
      BackEnd
  D
      Api.Gateway
         BusinessLogic.Layer
         Dependencies
             AppServices
     D
             BusinessRules
     D
             Mappers
             Validators
                Courses
                Students
                C# StudentToCreateValidator.cs
                C# StudentToDeleteValidator cs
                C# StudentToUpdateValidator.cs
                C# StudentToViewValidator.cs
        D
                Teachers
                GenericValidatorResult.cs
      C# DataAccess.Layer
      FrontEnd
      Web.Mvc
```



```
Search Solution Explorer (Ctrl+;)
     □namespace BusinessLogic.Layer.AppServices.Students
                                                                                                                           Solution 'School DatabaseCentricApp' (4 of 4 projects)
         public class StudentsService
                                                                                                                                 BackEnd
                                                                                                                                 Api.Gateway
             #region Properties
                                                                                                                                 C# BusinessLogic.Layer
             StudentToCreateValidator StudentToCreateValidator:
             StudentToUpdateValidator StudentToUpdateValidator;
                                                                                                                                    Dependencies
             StudentToDeleteValidator StudentToDeleteValidator;
                                                                                                                                    AppServices
             StudentsMapper StudentsMapper;
                                                                                                                                       Courses
             StudentsRepositoy StudentsRepositoy;
             #endregion
                                                                                                                                       Students
                                                                                                                                         Dtos
             #region Operations
                                                                                                                                             C# StudentsToViewOutputModel.cs
             public StudentToCreateOutputModel Create(StudentToCreateInputModel inputModel)
25
                                                                                                                                             C# StudentToCreateInputModel.cs
26
                                                                                                                                             C# StudentToCreateOutputModel.cs
                  var ValidatorResult = StudentToCreateValidator.Check(inputModel);
                                                                                                                                             C# StudentToDeleteInputModel.cs
                  if (ValidatorResult.Success)
                                                                                                                                             C# StudentToDeleteOutputModel.cs
                                                                                                                                             C# StudentToUpdateInputModel.cs
                     Student Student = StudentsMapper.Map(inputModel);
                                                                                                                                             C* StudentToUpdateOutputModel.cs
                     StudentsRepositoy.Create(Student);
                     StudentsRepositoy.SaveOnDatabase();
                                                                                                                                             C# StudentToViewForUpdateOutputModel.cs
                     return StudentToCreateOutputModel.BuildSuccessModel(Student);
                                                                                                                                             C# StudentToViewOutputModel.cs
                                                                                                                                             StudentsService.cs
                 else
36
                                                                                                                                          Teachers
                     return StudentToCreateOutputModel.BuildErrorModel(ValidatorResult.Errors);
                                                                                                                                       BusinessRules
                                                                                                                                       Mappers
```

```
Search Solution Explorer (Ctrl+;)
         anamespace BusinessLogic.Layer.Validators
    10
                                                                                                                                                               C# StudentToCreateInputModel.cs
              1 reference
              public class StudentToCreateValidator
                                                                                                                                                                C# StudentToCreateOutputModel.cs
                                                                                                                                                                   StudentToDeleteInputModel.cs
                  GenericValidatorResult Result = new GenericValidatorResult():
    13
                                                                                                                                                                C# StudentToDeleteOutputModel.cs
                  StudentToCreateInputModel InputModel;
    14
                  StudentAgeShouldBeGreaterThan7YearsBusinessRule StudentAgeShouldBeGreaterThan7YearsBusinessRule;
                                                                                                                                                                   StudentToUpdateInputModel.cs
                  StudentNameShouldNotContainAnvNumbersBusinessRule StudentNameShouldNotContainAnvNumbersBusinessRule:
    16
                                                                                                                                                                C# StudentToUpdateOutputModel.cs
                  StudentNationalIdShouldBeLegalBusinessRule StudentNationalIdShouldBeLegalBusinessRule:
                                                                                                                                                                  StudentToViewForUpdateOutputModel.cs
                  StudentPhoneNumberShouldBeUniqueBusinessRule StudentPhoneNumberShouldBeUniqueBusinessRule:
    18
                  StudentPhoneNumberShouldContainNumbersOnlyBusineeRule StudentPhoneNumberShouldContainNumbersOnlyBusineeRule;
                                                                                                                                                                C# StudentToViewOutputModel.cs
    19
                                                                                                                                                                Students Service cs
                  public GenericValidatorResult Check(StudentToCreateInputModel inputModel)
    20
    21
                                                                                                                                                             Teachers
                      InputModel = inputModel;
                                                                                                                                                          BusinessRules
                                                                                                                                                          Mappers
                      CheckStudentAgeShouldBeGreaterThan7YearsBusinessRule();
                      CheckStudentNameShouldNotContainAnyNumbersBusinessRule();
                                                                                                                                                         Validators
                      CheckStudentNationalIdShouldBeLegalBusinessRule();
    26
                                                                                                                                                            Courses
                      CheckStudentPhoneNumberShouldBeUniqueBusinessRule();
                                                                                                                                                            Students
                      CheckStudentPhoneNumberShouldContainNumbersOnlyBusineeRule();
    28
                                                                                                                                                            C# StudentToCreateValidator.cs
    29
                      if(Result.Errors.Count > 0)
                          Result.Success = false:
    30
                                                                                                                                                            C# StudentToDeleteValidator.cs
                      else
                                                                                                                                                             C# StudentToUpdateValidator.cs
                          Result.Success = true;
                                                                                                                                                                StudentToViewValidator.cs
                      return Result;
    33
    34
                                                                                                                                                             Teachers
                                                                                                                                                            GenericValidatorResult.cs
    35
                  void CheckStudentAgeShouldBeGreaterThan7YearsBusinessRule()
    36
                                                                                                                                                   C# DataAccess.Layer
                      if (StudentAgeShouldBeGreaterThan7YearsBusinessRule.Check(InputModel.BirthDay))
    37
                                                                                                                                                    FrontEnd
                           Result.Errors.Add("Student Age Should Be Greater Than 7 Years");
    38
                                                                                                                                                   Web.Mvc
    39
                                                                                                                                           Solution Explorer Git Changes Team Explorer
60 %
             No issues found
                                                                                                                    Ch: 10
                                                                                                                             SPC
                                                                                                                                   CRLF
                                                                                                             Ln: 41
```

### **Domain Centric Style**

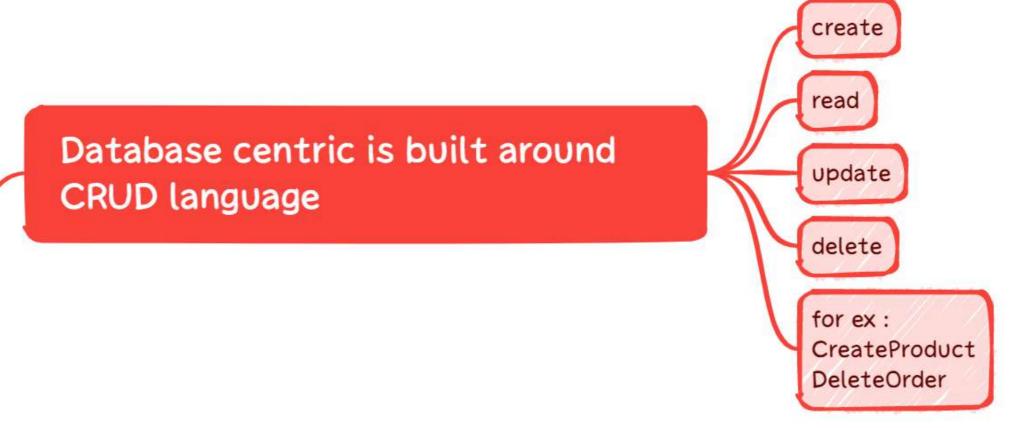


### Domain-Centric Style

The domain or business logic is the core of the system.

the system is independent on any framework, user interface, database or any other infrastructure component.

All Business Logic or Rules are Implemented on the application code not in the database.



Domain-Centric Style

Domain centric is built around Ubiquitous Language

Developers write code by using the same terms, words and concepts that used by domain experts or real users of the system in real life or real world.

stock employee dose not say "CreateProduct "
But "AddProductToStock"

users do not say "DeleteOrder"

But " CancelOrder "

## Domain-Centric Style

Domain centric is used to build systems that have complex business logic

Domain centric help developers to build complex systems by :

make the design of system is flexible to meet the changes of the requirements (agile)

make the development of system is simple and code is very readable