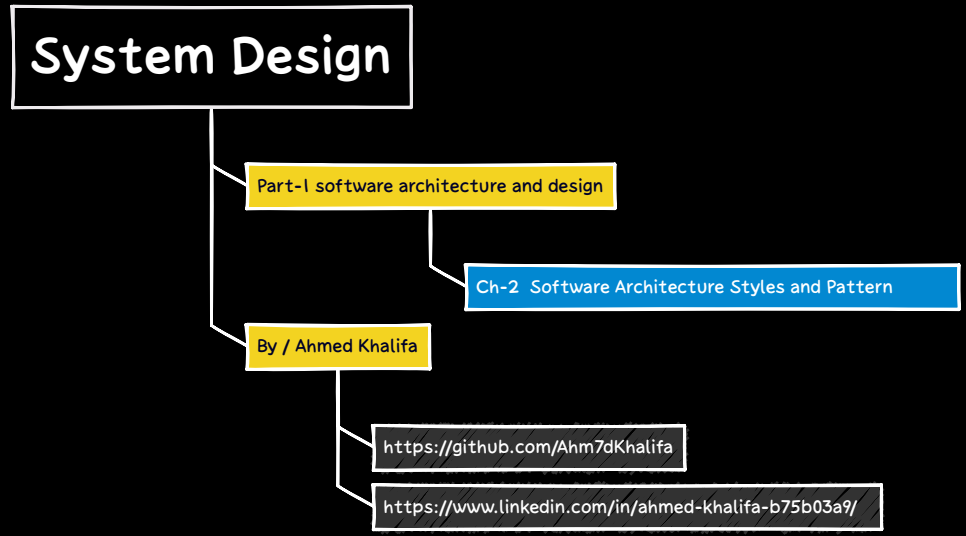
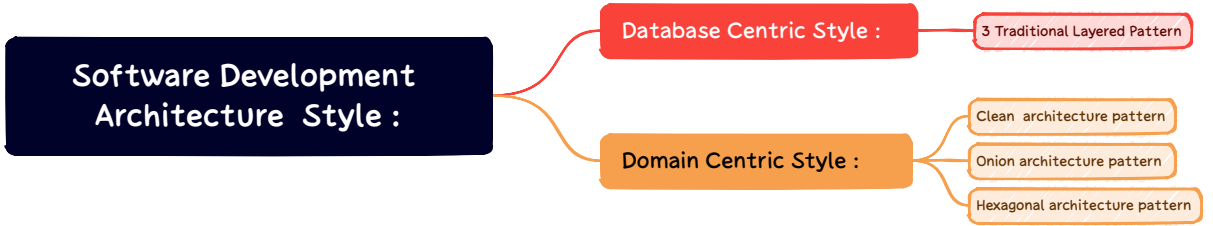
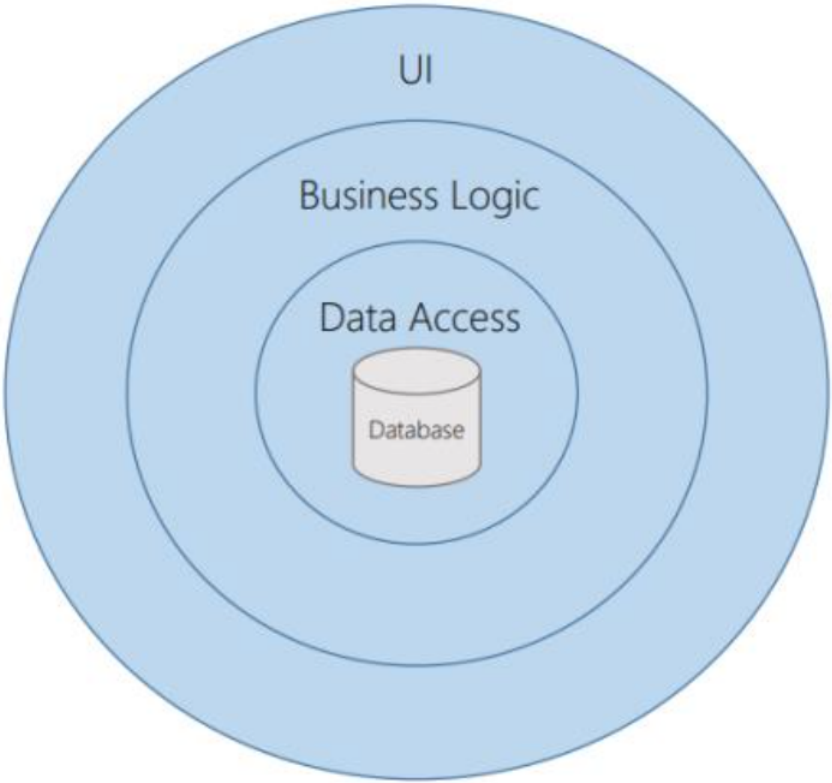
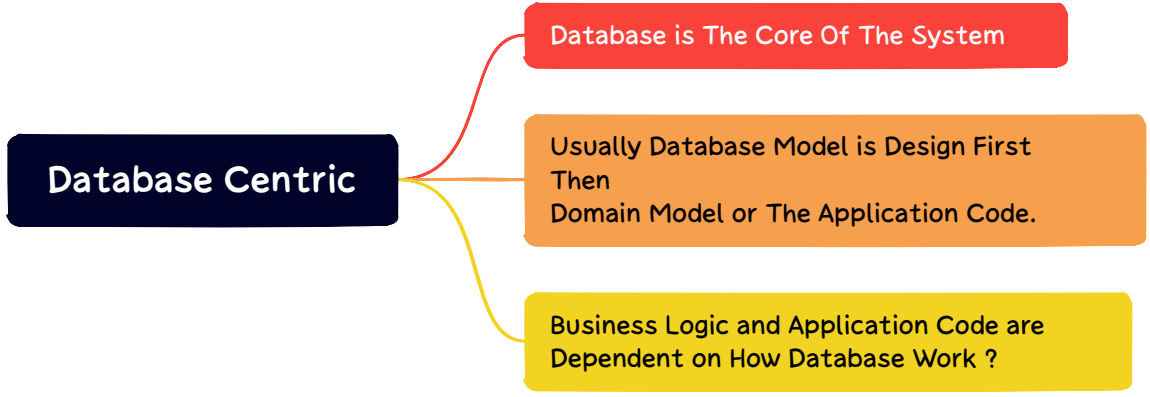
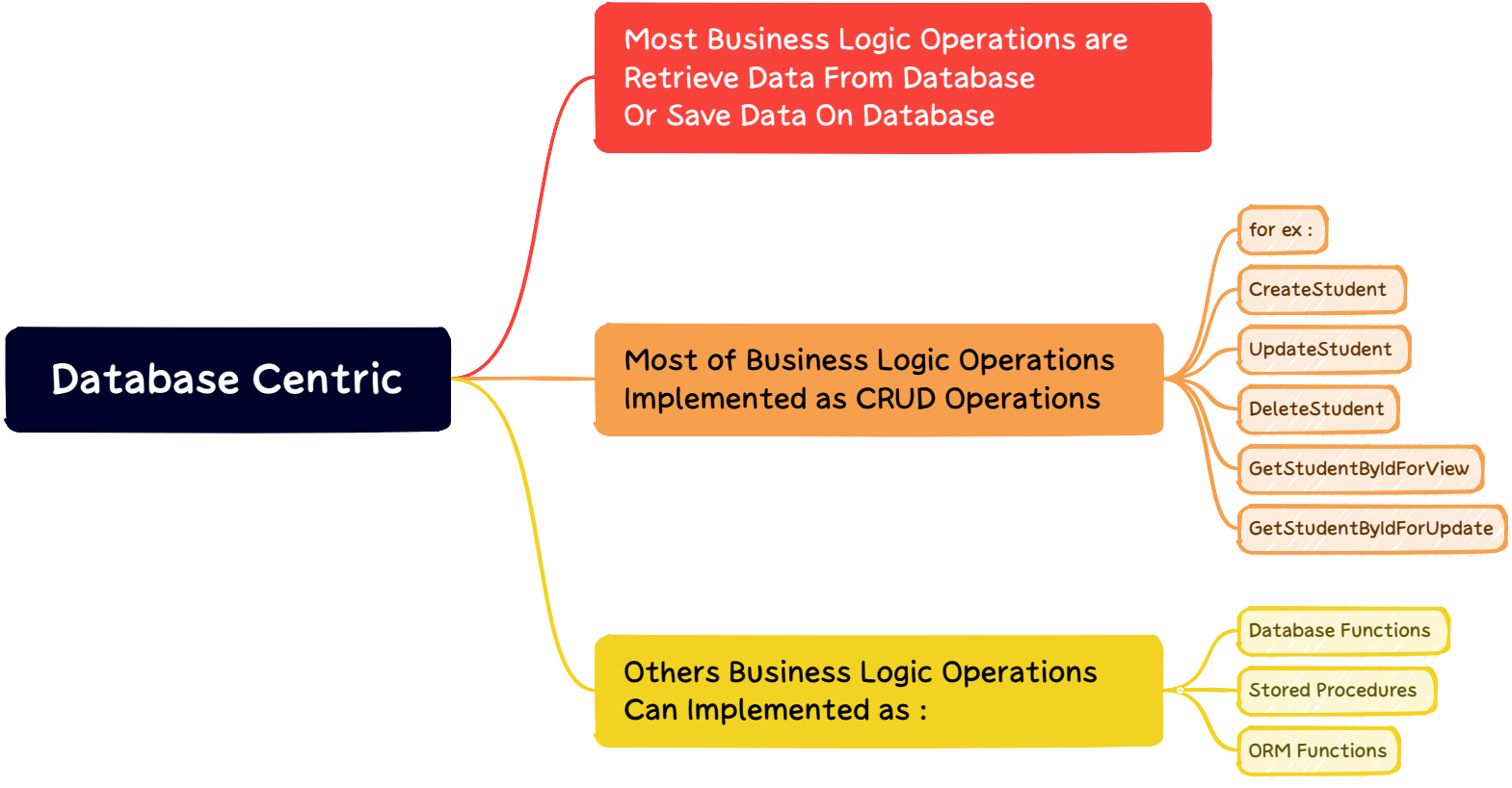


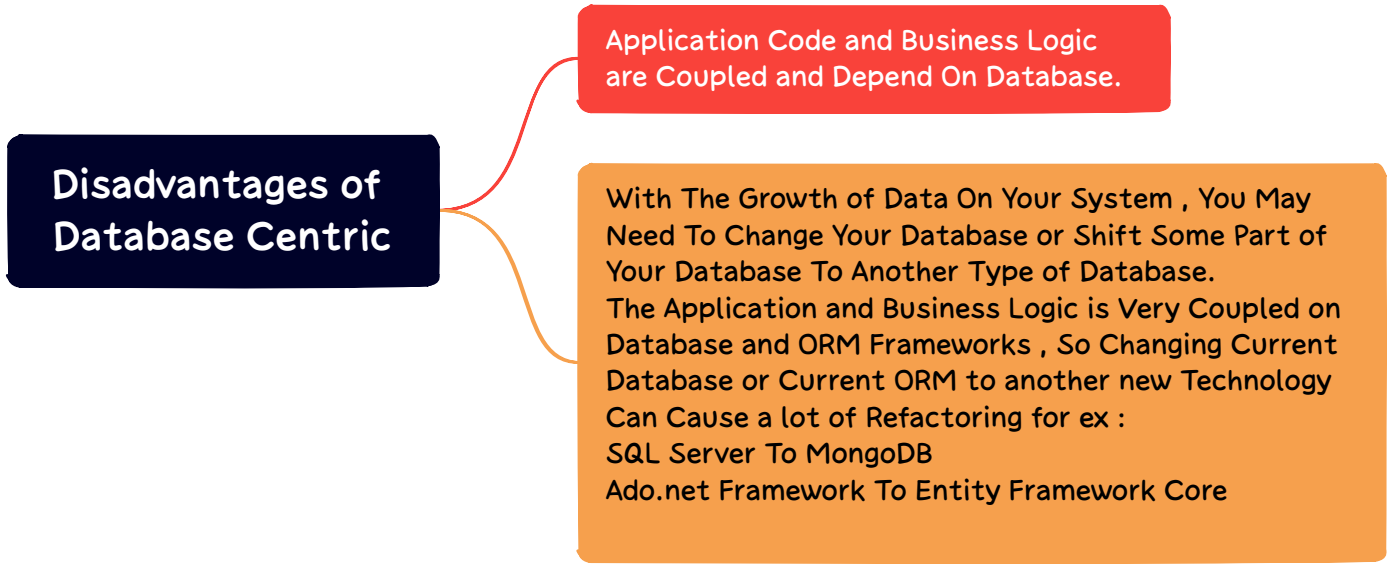


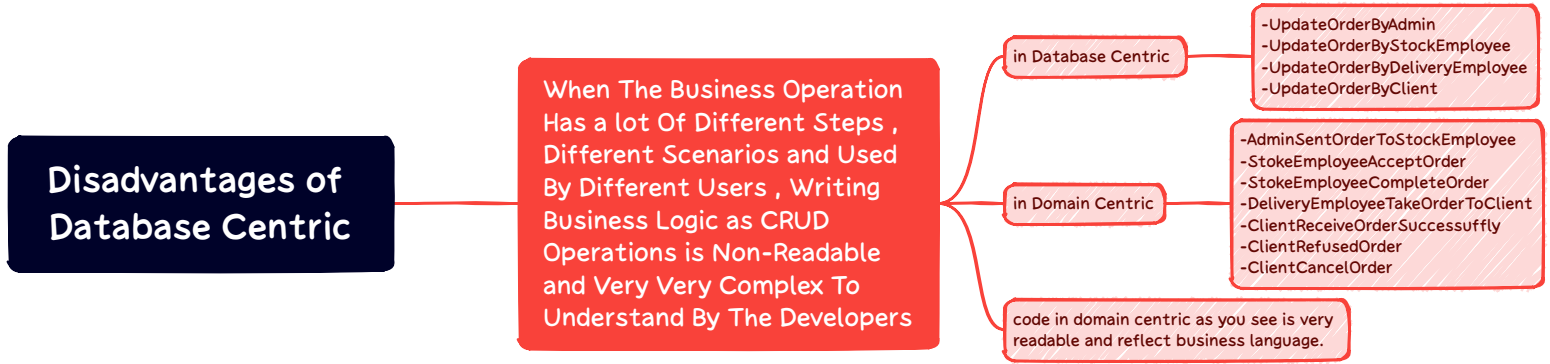
Database Centric

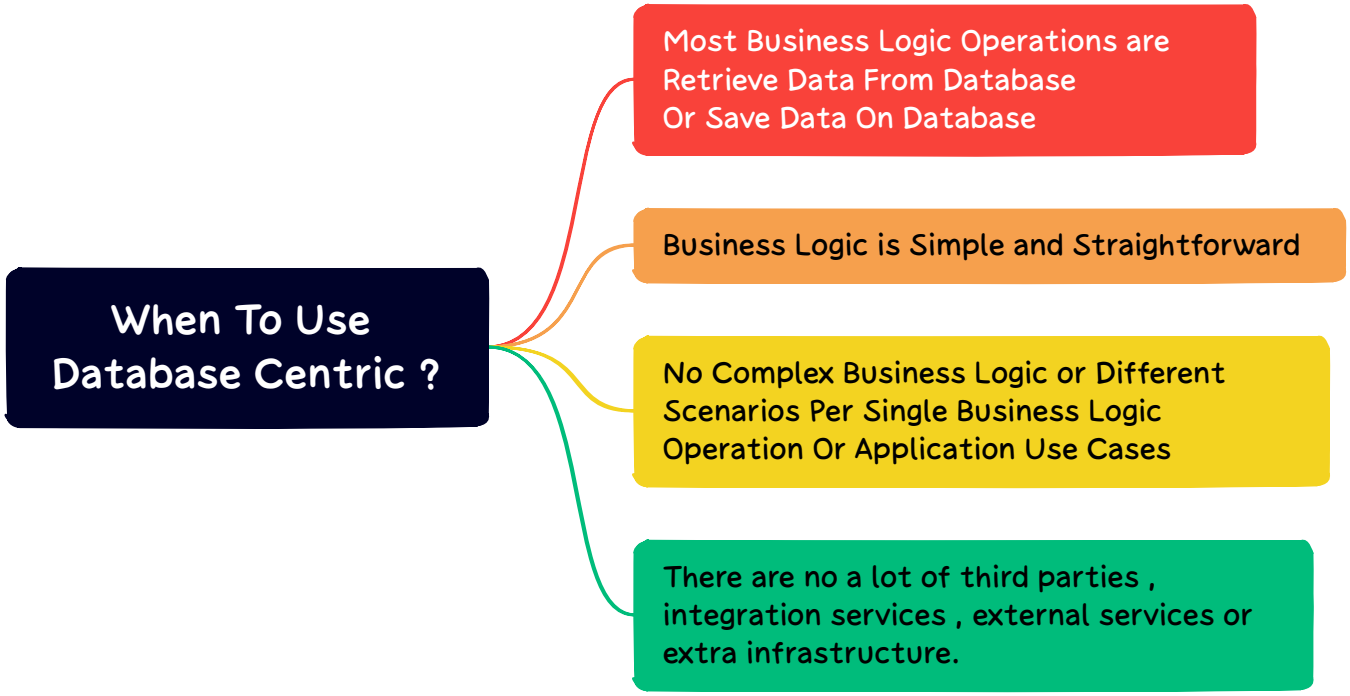






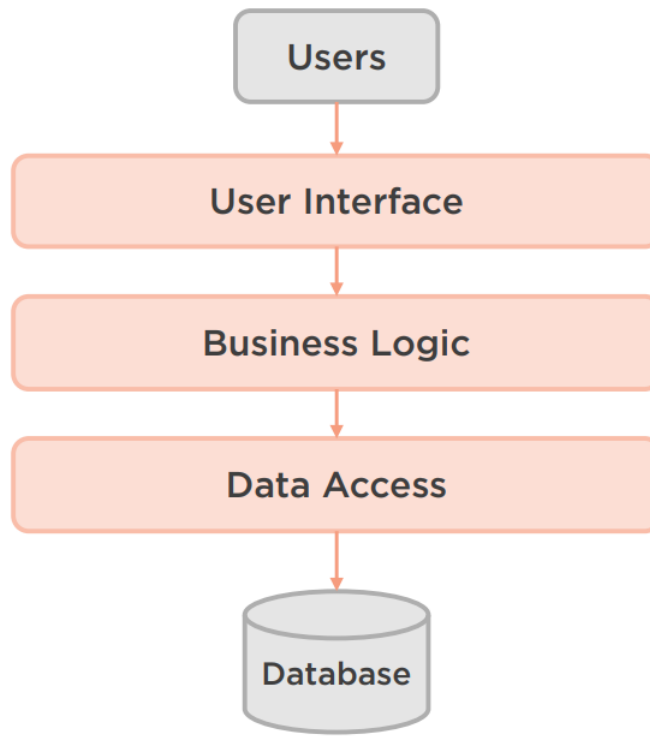


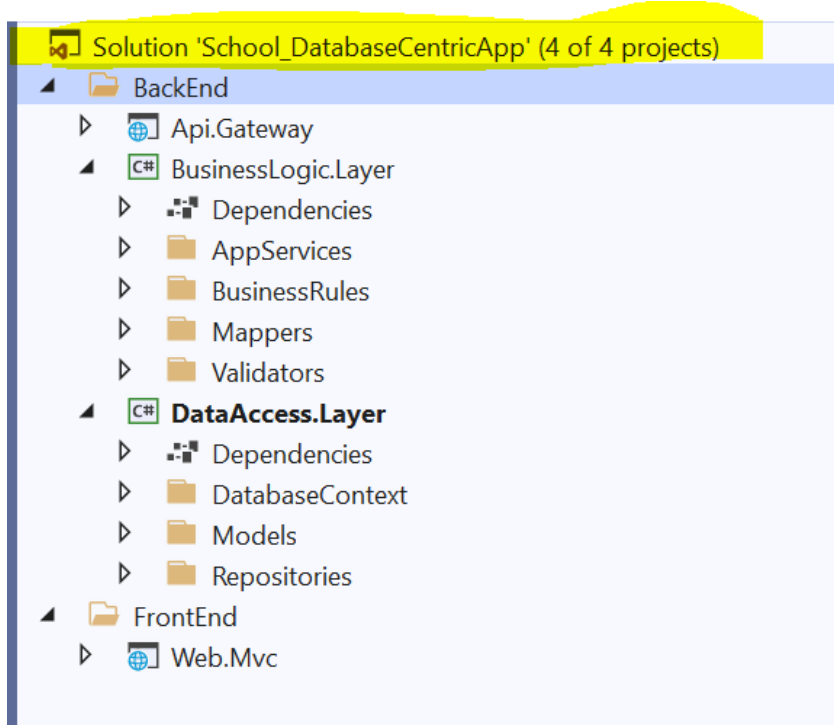


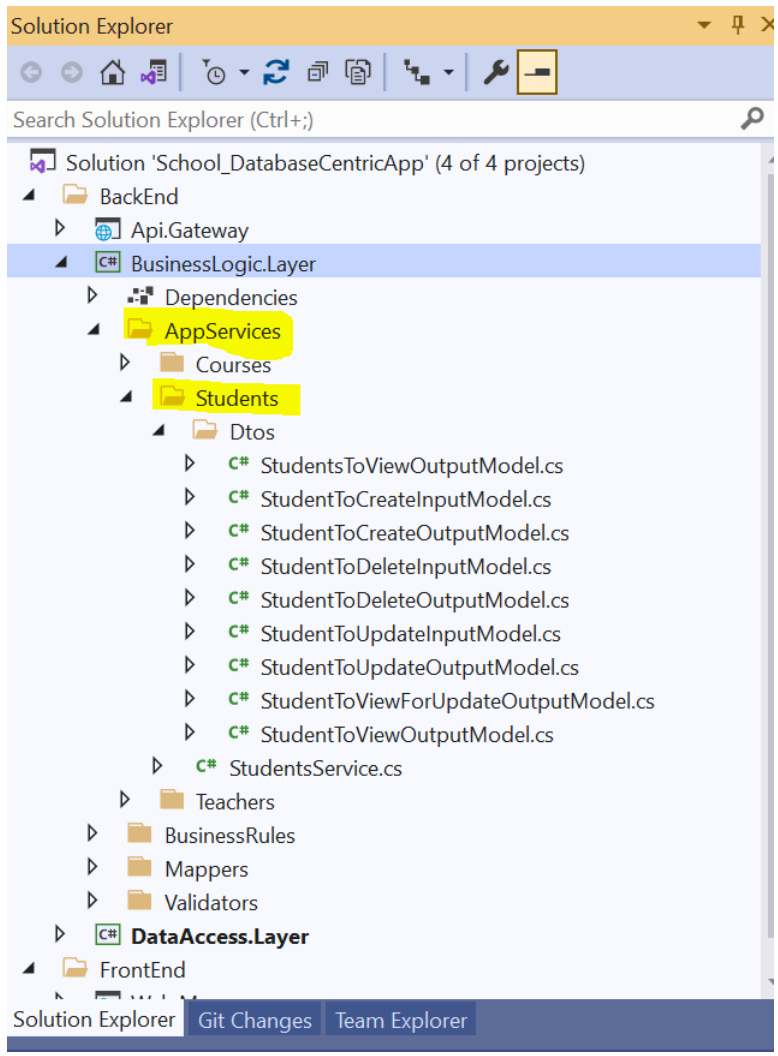


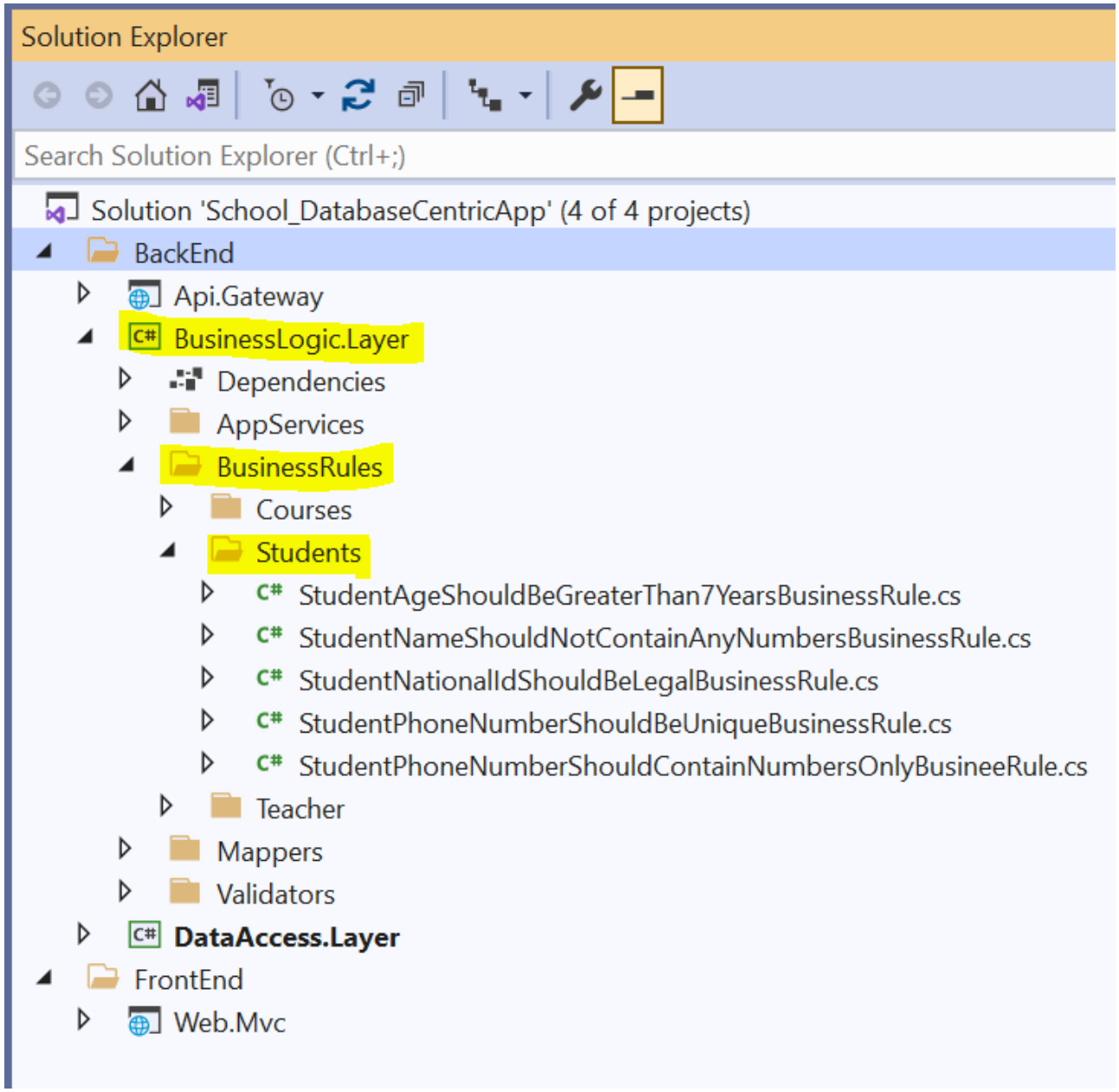
Database Centric Pattern Example :

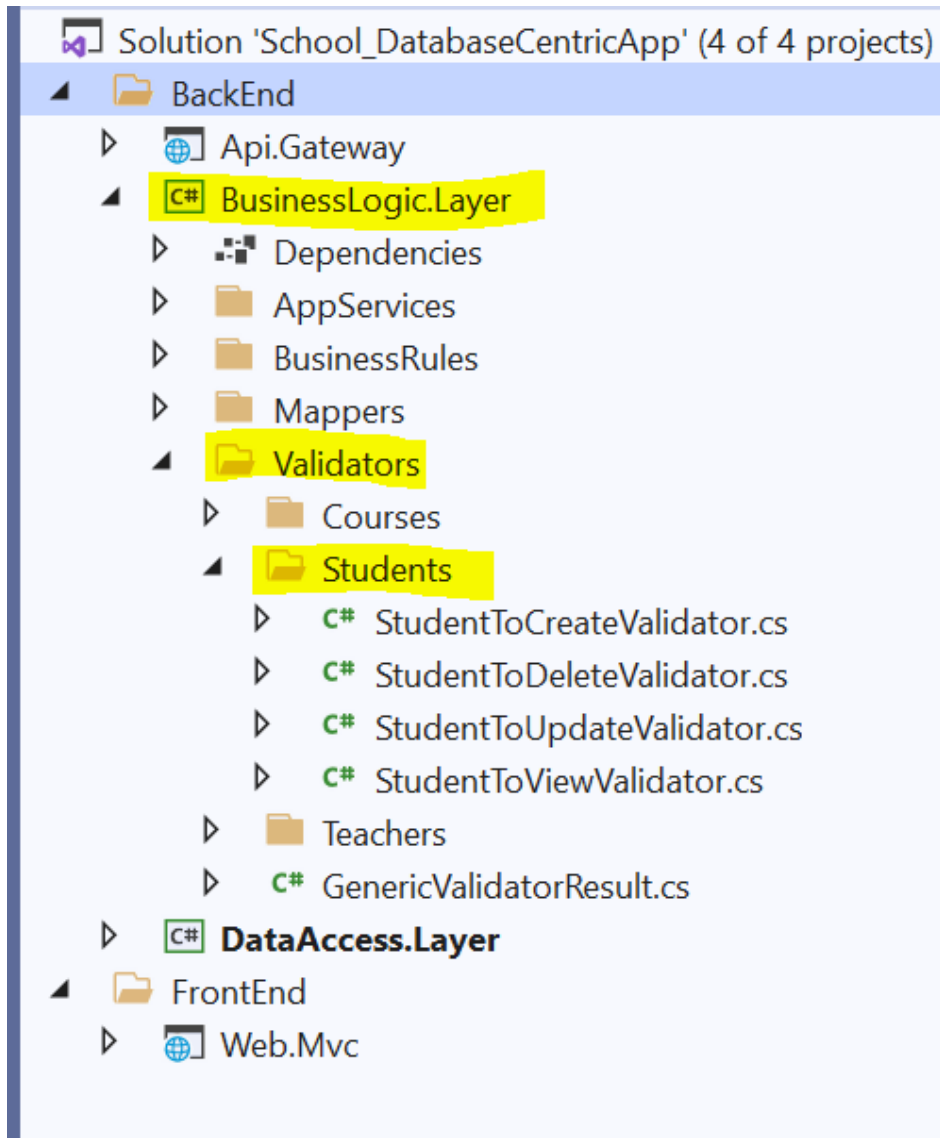
Traditional 3 Layers Architecture




Solution 'School_DatabaseCentricApp' (4 of 4 projects)
BackEnd
Api.Gateway
BusinessLogic.Layer
Dependencies
AppServices
BusinessRules
Mappers
Validators
DataAccess.Layer
Dependencies
DatabaseContext
Models
Repositories
FrontEnd
Web.Mvc


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'School_DatabaseCentricApp' (4 of 4 projects)
BackEnd
Api.Gateway
BusinessLogic.Layer
Dependencies
AppServices
Courses
Students
Dtos
StudentsToViewOutputModel.cs
StudentToCreateInputModel.cs
StudentToCreateOutputModel.cs
StudentToDeleteInputModel.cs
StudentToDeleteOutputModel.cs
StudentToUpdateInputModel.cs
StudentToUpdateOutputModel.cs
StudentToViewForUpdateOutputModel.cs
StudentToViewOutputModel.cs
StudentsService.cs
Teachers
BusinessRules
Mappers
Validators
DataAccess.Layer
FrontEnd
Solution Explorer
Git Changes
Team Explorer


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'School_DatabaseCentricApp' (4 of 4 projects)
BackEnd
Api.Gateway
BusinessLogic.Layer
Dependencies
AppServices
BusinessRules
Courses
Students
StudentAgeShouldBeGreaterThan7YearsBusinessRule.cs
StudentNameShouldNotContainAnyNumbersBusinessRule.cs
StudentNationalIdShouldBeLegalBusinessRule.cs
StudentPhoneNumberShouldBeUniqueBusinessRule.cs
StudentPhoneNumberShouldContainNumbersOnlyBusineeRule.cs
Teacher
Mappers
Validators
DataAccess.Layer
FrontEnd
Web.Mvc


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

```
12  namespace BusinessLogic.Layer.AppServices.Students
13  {
        0 references
14      public class TeachersService
15      {
16          Properties

21

22          #region Operations
                0 references
23          public StudentToCreateOutputModel Create(StudentToCreateInputModel inputModel)...
40
                0 references
41          public StudentToUpdateOutputModel Update(StudentToUpdateInputModel inputModel)...
45
                0 references
46          public StudentToDeleteOutputModel Delete(StudentToDeleteInputModel inputModel)...
50
                0 references
51          public StudentToViewOutputModel GetByIdForView(long studentId)...
                0 references
55          public StudentToViewForUpdateOutputModel GetByIdForUpdate(long studentId)
56          {
57              return new StudentToViewForUpdateOutputModel();
58          }
                0 references
59          public StudentsToViewOutputModel GetAll()...
63          #endregion
64
65      }
66  }
67
```

100 % | No issues found | Ln: 64 | Ch: 1 | SPC | CRLF

Search Solution Explorer (Ctrl+;)

- Solution 'School_DatabaseCentricApp' (4 of 4 projects)
  - BackEnd
    - Api.Gateway
    - BusinessLogic.Layer
      - Dependencies
      - AppServices
        - Courses
        - Students
          - Dtos
            - StudentsToViewOutputModel.cs
            - StudentToCreateInputModel.cs
            - StudentToCreateOutputModel.cs
            - StudentToDeleteInputModel.cs
            - StudentToDeleteOutputModel.cs
            - StudentToUpdateInputModel.cs
            - StudentToUpdateOutputModel.cs
            - StudentToViewForUpdateOutputModel.c
            - StudentToViewOutputModel.cs
          - StudentsService.cs
        - Teachers
      - BusinessRules
      - Mappers
      - Validators
    - **DataAccess.Layer**

Solution Explorer | Git Changes | Team Explorer

```
12  namespace BusinessLogic.Layer.AppServices.Students
13  {
        0 references
14      public class TeachersService
15      {
16          #region Properties
17          StudentToCreateValidator StudentToCreateValidator;
18          StudentToUpdateValidator StudentToUpdateValidator;
19          StudentToDeleteValidator StudentToDeleteValidator;
20          StudentsMapper StudentsMapper;
21          StudentsRepositoy StudentsRepositoy;
22          #endregion
23
24          #region Operations
            0 references
25          public StudentToCreateOutputModel Create(StudentToCreateInputModel inputModel)
26          {
27              var ValidatorResult = StudentToCreateValidator.Check(inputModel);
28
29              if (ValidatorResult.Success)
30              {
31                  Student Student = StudentsMapper.Map(inputModel);
32                  StudentsRepositoy.Create(Student);
33                  StudentsRepositoy.SaveOnDatabase();
34                  return StudentToCreateOutputModel.BuildSuccessModel(Student);
35              }
36              else
37              {
38                  return StudentToCreateOutputModel.BuildErrorModel(ValidatorResult.Errors);
39              }
40          }
41
```

```
namespace BusinessLogic.Layer.Validators
{
    1 reference
    public class StudentToCreateValidator
    {
        GenericValidatorResult Result = new GenericValidatorResult();
        StudentToCreateInputModel InputModel;
        StudentAgeShouldBeGreaterThan7YearsBusinessRule StudentAgeShouldBeGreaterThan7YearsBusinessRule;
        StudentNameShouldNotContainAnyNumbersBusinessRule StudentNameShouldNotContainAnyNumbersBusinessRule;
        StudentNationalIdShouldBeLegalBusinessRule StudentNationalIdShouldBeLegalBusinessRule;
        StudentPhoneNumberShouldBeUniqueBusinessRule StudentPhoneNumberShouldBeUniqueBusinessRule;
        StudentPhoneNumberShouldContainNumbersOnlyBusineeRule StudentPhoneNumberShouldContainNumbersOnlyBusineeRule;
        1 reference
        public GenericValidatorResult Check(StudentToCreateInputModel inputModel)
        {
            InputModel = inputModel;

            CheckStudentAgeShouldBeGreaterThan7YearsBusinessRule();
            CheckStudentNameShouldNotContainAnyNumbersBusinessRule();
            CheckStudentNationalIdShouldBeLegalBusinessRule();
            CheckStudentPhoneNumberShouldBeUniqueBusinessRule();
            CheckStudentPhoneNumberShouldContainNumbersOnlyBusineeRule();
            if(Result.Errors.Count > 0)
                Result.Success = false;
            else
                Result.Success = true;
            return Result;
        }
        1 reference
        void CheckStudentAgeShouldBeGreaterThan7YearsBusinessRule()
        {
            if (StudentAgeShouldBeGreaterThan7YearsBusinessRule.Check(InputModel.BirthDay))
                Result.Errors.Add("Student Age Should Be Greater Than 7 Years");
        }
```