# Exercise: Concurrency and Parallelism

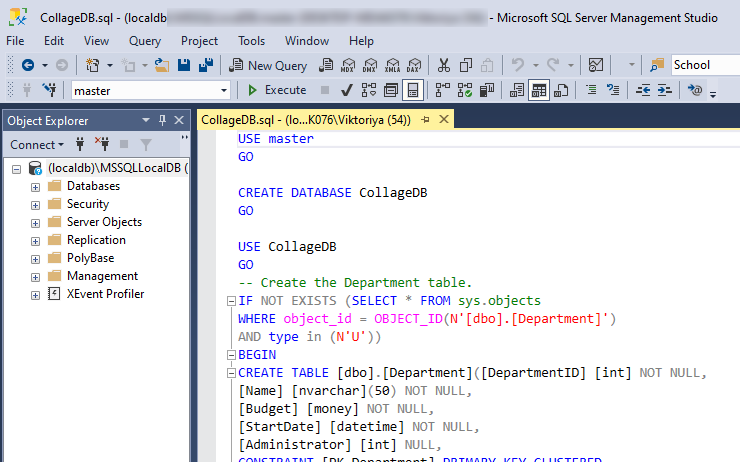
Problems for exercises and homework for the "Concurrent Programming" course from the official "Applied Programmer" curriculum.

**Use the provided skeleton from resources! Do not change its methods, classes and namespaces!**

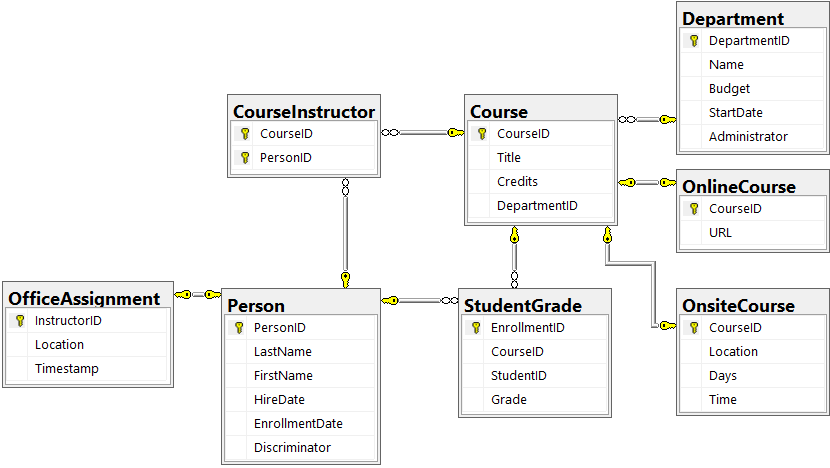
In the StartUp.cs file, **complete** the **code** in the **methods** as indicated **using task parallelism and data parallelism**.

## Import the Collage Database

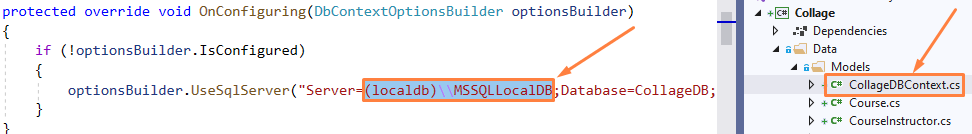
Import the Collage DB into SQL Management Studio by **executing** the provided .sql script.



After **execution** you will **get** the following **database**:



If you do not use SQL Server Express LocalDB, you can change the server from the file CollageDBContext.cs.



## Courses Info

Complete the code in method static void CoursesInfo(CollageDBContext context) so that it **prints** to the **console** for **each** **course** its **name**, **credits**, **department** **name** and **department** **budget** in the following **format**:

"{course title} - Credits: {course credits}, Department: {department name}, Budget { department budget}."

The **budget** value should be **rounded** to the **second** **decimal** **place**.

### Examples

The **result** will be in **random** **order** **each** time the program is **run**, but the **information** **should** **be** **the** **same**.

|  |
| --- |
| **Output** |
| Macroeconomics - Credits: 3, Department: Economics, Budget 200000.00.  Chemistry - Credits: 4, Department: Engineering, Budget 350000.00.  Physics - Credits: 4, Department: Engineering, Budget 350000.00.  Calculus - Credits: 4, Department: Mathematics, Budget 250000.00.  Trigonometry - Credits: 4, Department: Mathematics, Budget 250000.00.  Poetry - Credits: 2, Department: English, Budget 120000.00.  Quantitative - Credits: 2, Department: Economics, Budget 200000.00.  Composition - Credits: 3, Department: English, Budget 120000.00.  Literature - Credits: 4, Department: English, Budget 120000.00.  Microeconomics - Credits: 3, Department: Economics, Budget 200000.00. |

## Department And Credits

Complete the code in the method static void DepartmentAndCredits(CollageDBContext context) so that it takes **all** **grades** **below** **3.50** in **courses** with **id** **2021**, **2030**, **2042** and **prints** the **grade**, **course** **name**, and **course** **credits** to the console in the following **format**:

"Grade: {grade} Course: {cources title} Credits: {cource credits}"

### Examples

The **result** will be in **random** **order** **each** time the program is **run**, but the **information** **should** **be** **the** **same**.

|  |
| --- |
| **Output** |
| Grade: 3.00 Course: Composition Credits: 3  Grade: 3.00 Course: Composition Credits: 3  Grade: 3.00 Course: Literature Credits: 4  Grade: 2.50 Course: Composition Credits: 3 |

## Quick Reference

You need **complete** the **code** in the **two** **methods** below and **run** them **in** **parallel**.

void DepartmentInfo(List<Course> courses, List<Department> departments) – the method should **print** the **information** for **each** **course** in the following **format**:

* "Department id: {department id}, Name: {department name}, Course: {course name}, Credits: {course credits}"

void DepartmentSumCredits(List<Course> courses) – the method should **sum** the **course** **credits** in **each** **department** and **print** the **information** for each department in the following **format**:

* Department id: {department id}, Sum Credits: {sum credit}

Both methods should **sort** the **information** by department id in **ascending** **order**.

### Examples

The **result** will be in **random** **order** **each** time the program is **run**, but the **information** **should** **be** **the** **same**.

|  |
| --- |
| **Output** |
| Department id: 1, Sum Credits: 8  Department id: 1, Name: Engineering, Course: Chemistry, Credits: 4  Department id: 2, Sum Credits: 9  Department id: 4, Sum Credits: 8  Department id: 7, Sum Credits: 8  Department id: 1, Name: Engineering, Course: Physics, Credits: 4  Department id: 2, Name: English, Course: Composition, Credits: 3  Department id: 2, Name: English, Course: Poetry, Credits: 2  Department id: 2, Name: English, Course: Literature, Credits: 4  Department id: 4, Name: Economics, Course: Microeconomics, Credits: 3  Department id: 4, Name: Economics, Course: Macroeconomics, Credits: 3  Department id: 4, Name: Economics, Course: Quantitative, Credits: 2  Department id: 7, Name: Mathematics, Course: Calculus, Credits: 4  Department id: 7, Name: Mathematics, Course: Trigonometry, Credits: 4 |

### Hints

The **key** here is that the **methods** **in** **parallel** should **use** **one** **data** **instance** from the database **instead** of **accessing** the **database** in **each** **method**.