

# 2021 Fall CS307 Project 2

---

Contributors:

Zhu Yueming , Yu Tiancheng, Lu Hongyi, Wang Ziqin, Wang Weiyu, He Yirui, Yang Xiaosu, Chen Junfeng, Li Xin, Leng Ziyang

**DEADLINE: Friday, December 31st, 2021, 18:30**

## Overview

---

This is a group project for **no more than three people**. Each group should finish the project by yourselves and submit your source code and an SQL file. The presentation will be hosted on December 31st, 2021, i.e., the last lab class. The details about the presentation will be announced later.

---

You should submit a report before the deadline, **the Topscore will be 100**; for the report submitted after the deadline and before the resubmission date, **the Topscore will be 70**; for the report submitted after the resubmission date, the score will be 0.

---

**Please be honest.** DO NOT copy ANY words, figures, and others from the Internet and others. If you use open-sourced code from the Internet, you are required to state it in comments.

---

DBMS can help us to manage data conveniently and improve the efficiency of data.

Your work of project Part 2 is mainly divided into the following parts below:

- Implement the service and factory interfaces to pass the base test cases.
- Design your (PostgreSQL) database to satisfy the requirements of interfaces.
- Profile your implementation and find ways to speed it up.
- Make sure your result of code will be Correct(Most significant) and Effective.
- (Optional) Find other ways to implement similar functionalities as our interfaces and compare (some of) them, are they better, worse or have different use cases.

## Framework Source Code

---

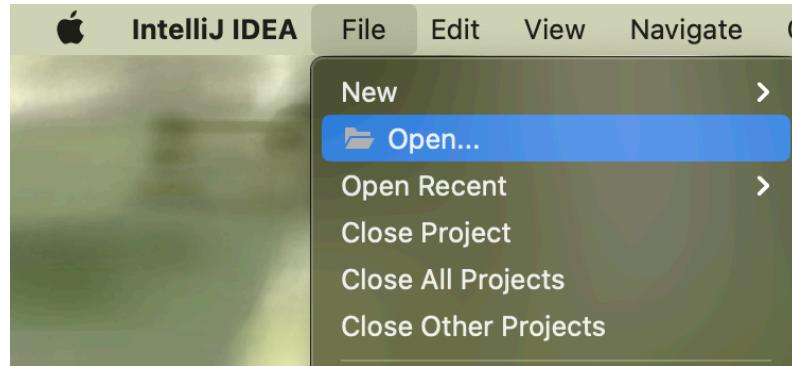
Only Java is supported.

<https://mirrors.sustech.edu.cn/git/sustech-2021fall-db/sustech-sql-project>

**Read the README.md and this document carefully!**

# A Brief User Manual

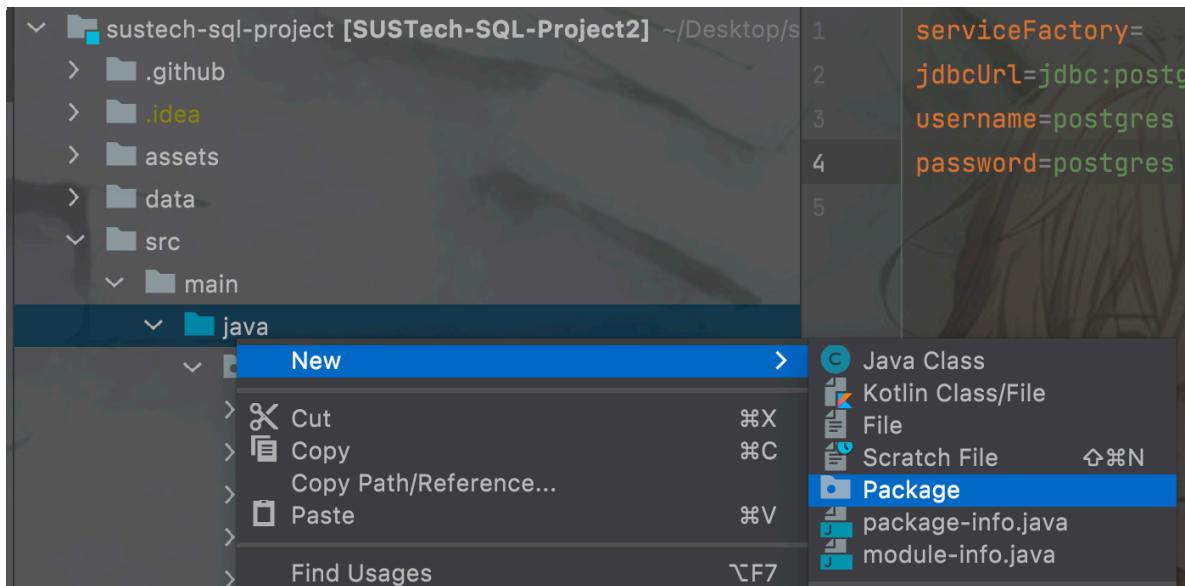
**Step 1:** Open Project in IDEA.



**Step 2:** Design your database.

Please make sure your database can **be rebuilt by the sql file and work properly.**

**Step 3:** Create your package and classes that implement interfaces in `cn.edu.sustech.cs307.service`.



The screenshot shows a Java file structure on the left and its corresponding code on the right. The file structure includes a main package with sub-packages like benchmark, config, database, dto, exception, factory, service, util, and test. Under service, there are interfaces CourseService, DepartmentService, InstructorService, MajorService, SemesterService, StudentService, and UserService. Implementations for these are in me.yuki.impl. A specific implementation, MyCourseService, is selected. The code for MyCourseService implements the CourseService interface, providing implementations for methods like addCourse, addCourseSection, addCourseSectionClass, removeCourse, and removeCourseSection.

```

6 import cn.edu.sustech.cs307.dto.Student;
7 import cn.edu.sustech.cs307.dto.prerequisite.Prerequisite;
8 import cn.edu.sustech.cs307.service.CourseService;
9
10 import javax.annotation.Nullable;
11 import java.time.DayOfWeek;
12 import java.util.List;
13 import java.util.Set;
14
15 public class MyCourseService implements CourseService {
16     @Override
17     public void addCourse(String courseId, String courseName, int credits) {
18     }
19
20     @Override
21     public int addCourseSection(String courseId, int semesterId, String sectionName) {
22     }
23
24     @Override
25     public int addCourseSectionClass(int sectionId, int instructorId, String className) {
26     }
27
28     @Override
29     public void removeCourse(String courseId) {
30     }
31
32     @Override
33     public void removeCourseSection(int sectionId) {
34     }
35
36     @Override
37     public void removeCourseSectionClass(int sectionId) {
38     }

```

**Step 4:** Create your Factory that extends from `cn.edu.sustech.cs307.factory.ServiceFactory` and register your service implements in the constructor:

Take CourseService as an example:

The screenshot shows a Java file structure on the left and its corresponding code on the right. The file structure includes a main package with sub-packages like assets, data, and src. Under src/main/java, there is a cn.edu.sustech.cs307 package and a me.yuki.impl package containing MyCourseService and MyFactory. The code for MyFactory extends ServiceFactory and registers MyCourseService as a service implementation.

```

2
3     import cn.edu.sustech.cs307.factory.ServiceFactory;
4     import cn.edu.sustech.cs307.service.CourseService;
5     import me.yuki.impl.services.MyCourseService;
6
7     public class MyFactory extends ServiceFactory {
8         public MyFactory() {
9             super();
10            registerService(CourseService.class, new MyCourseService());
11        }
12    }
13

```

After registering all implementation:

The screenshot shows the complete code for MyServiceFactory, which extends ServiceFactory and registers multiple service implementations: CourseService, DepartmentService, MajorService, StudentService, SemesterService, InstructorService, and UserService.

```

public class MyServiceFactory extends ServiceFactory {
    public MyServiceFactory() {
        super();
        registerService(CourseService.class, new MyCourseService());
        registerService(DepartmentService.class, new MyDepartmentService());
        registerService(MajorService.class, new MyMajorService());
        registerService(UserService.class, new MyUserService());
        registerService(SemesterService.class, new MySemesterService());
        registerService(InstructorService.class, new MyInstructorService());
        registerService(StudentService.class, new MyStudentService());
    }
}

```

For better judgment, please override `getUIDs` in your `ServiceFactory` implementation. These functions should return your group members' student ids:

```
public class MyFactory extends ServiceFactory {
    public MyFactory() {
        super();
        registerService(CourseService.class, new MyCourseService());
    }

    @Override
    public List<String> getUIDs() {
        return List.of("12010000", "12019999", "114514");
    }
}
```

**Step 5:** Modify the setting files (`config.properties`) as follows:

```
serviceFactory=me.yuki.impl.MyFactory
jdbcUrl=jdbc:postgresql://localhost:5432/project3
username=yuki
password=
```

`serviceFactory` is the path of your `ServiceFactory`.

**Step 6:** Run the benchmark, and debug by yourselves.

You can refer to reference implementation result in `data/sampleResult.txt`:

```
Import departments
Import majors
Import users
Import semesters
Import courses
Import sections
Import classes
Import major courses
Import time usage: 2.46s
Test search course 1: 1000
Test search course 1 time: 0.53s
Test enroll course 1: 1000
Test enroll course 1 time: 0.30s
Test drop enrolled course 1: 813
Test drop enrolled course 1 time: 0.03s
Import student courses
Import student courses time: 14.19s
```

```
Test drop course: 416637
Test drop course time: 4.75s
Test course table 2: 1000
Test course table 2 time: 0.72s
Test search course 2: 1000
Test search course 2 time: 0.48s
Test enroll course 2: 1000
Test enroll course 2 time: 0.17s
```

The integer after each test case is the number of your correct result. Your program should give out exactly the same result as the reference implementation.

## What to submit

1. The source code of your project.
2. One SQL file, we will use this file to rebuild your database in the server.

### Please read Tip.3 about this SQL file

Please make sure your database can **be rebuilt by the SQL file and work properly**.

3. A group report.

Make a zip (**NOT RAR**) file of 1 and 2, and submit this zip file and a pdf report.

Only one copy of each group should be submitted on BlackBoard.

## Tips

1. How to check the locale of a database:

Connect to the database in Postgres shell, run command `show all;`, then find the keyword `locale`:



```
psql
 jit_debugging_support | off          | Register JIT compiled function with debugger.
 jit_dump_bitcode     | off          | Write out LLVM bitcode to facilitate JIT debugging.
 jit_expressions      | on           | Allow JIT compilation of expressions.
 jit_inline_above_cost | 500000      | Perform JIT inlining if query is more expensive.
 jit_optimize_above_cost | 500000    | Optimize JITed functions if query is more expensive.
 jit_profiling_support | off          | Register JIT compiled function with perf profiler.
 jit_provider          | llvmjit     | JIT provider to use.
 jit_tuple_deforming   | on           | Allow JIT compilation of tuple deforming.
 joinCollapse_limit    | 8            | Sets the FROM-List size beyond which JOIN constructs are not flattened.
 krb_caseins_users     | off          | Sets whether Kerberos and GSSAPI user names should be treated as case-insensitive.
 krb_server_keyfile    | FILE:/usr/local/etc/postgresql/krb5.keytab | Sets the location of the Kerberos server key file.
 lc_collate            | C            | Shows the collation order locale.
 lc_ctype               | C            | Shows the character classification and case conversion locale.
 lc_messages            | C            | Sets the language in which messages are displayed.
 lc_monetary             | C            | Sets the locale for formatting monetary amounts.
 lc_numeric              | C            | Sets the locale for formatting numbers.
 lc_time                | C            | Sets the locale for formatting date and time values.
 listen_addresses       | localhost   | Sets the host name or IP address(es) to listen to.
 lo_compat_privileges  | off          | Enables backward compatibility mode for privilege checks on large objects.
 local_preload_libraries |           | Lists unprivileged shared libraries to preload into each backend.
 lock_timeout            | 0            | Sets the maximum allowed duration of any wait for a lock.
 log_autovacuum_min_duration | -1          | Sets the minimum execution time above which autovacuum actions will be logged.
 log_checkpoints         | off          | Logs each checkpoint.
 log_connections         | off          | Logs each successful connection.
 log_destination         | stderr      | Sets the destination for server log output.
 log_directory           | log          | Sets the destination directory for log files.
 log_disconnections      | off          | Logs end of a session, including duration.
 log_duration             | off          | Logs the duration of each completed SQL statement.
 log_error_verbosity     | default     | Sets the verbosity of logged messages.
 log_executor_stats      | off          | Writes executor performance statistics to the server log.
 log_file_mode            | 0600        | Sets the file permissions for log files.
 log_filename             | postgresql-%Y-%m-%d_%H%M%S.log | Sets the file name pattern for log files.
```

## 2. How should I know where I am wrong?

**Read the interface documents carefully.**

You are free to **modify any class** in `cn.edu.sustech.cs307`, we will replace them when testing your code in the server.

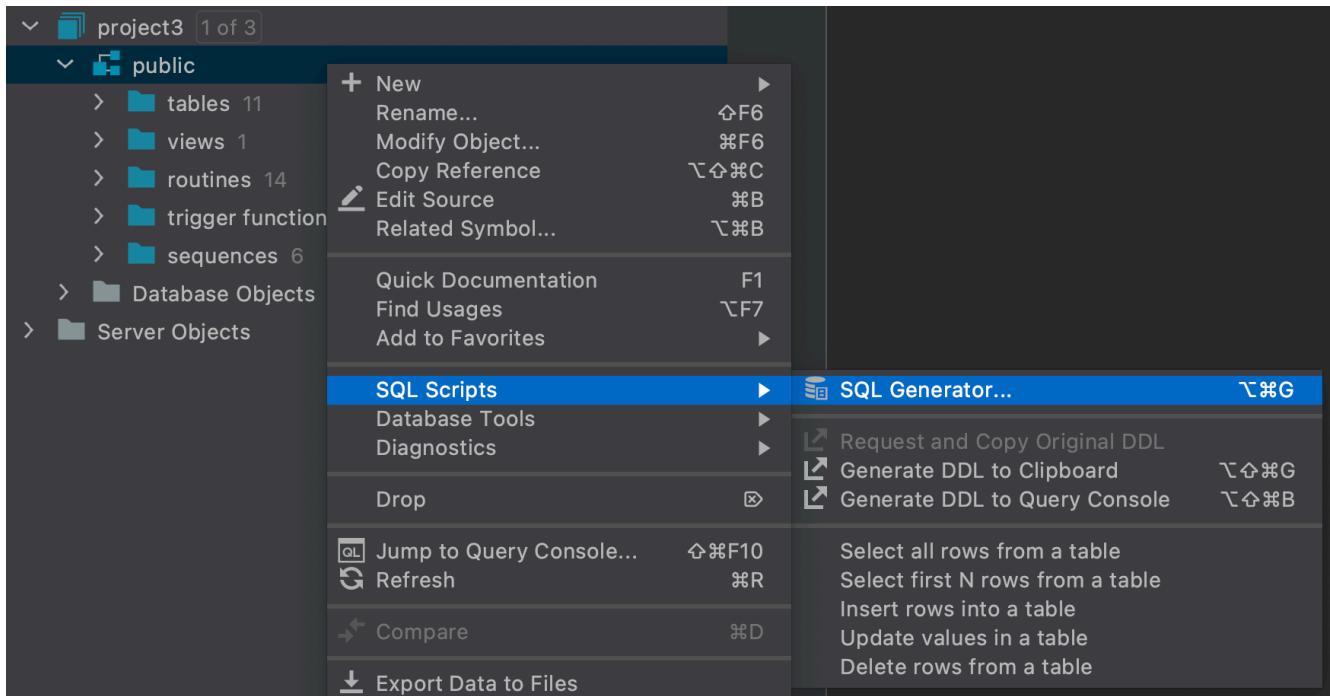
For example, you can modify the `ProjectJudge` and `dto` classes to print your wrong test cases:

```
testCourseTables courseTable2 failed for [11710430, 18145] 2019-09-06 FRIDAY 31, expected:  
CourseTable{  
1: [CourseTableEntry{fn:基础有机化学[英文班-实验1班 (实验课每次都计入期末成绩, 且没有补课) ], begin:3, end:4, location:一教301},  
CourseTableEntry{fn:市场营销管理[中文班], begin:9, end:10, location:一教404}],  
2: [CourseTableEntry{fn:乐理与视唱练耳[中文班], begin:7, end:8, location:荔园1栋201},  
CourseTableEntry{fn:基础有机化学[英文班-实验1班 (实验课每次都计入期末成绩, 且没有补课) ], begin:9, end:10, location:二教201机房},  
CourseTableEntry{fn:常微分方程[中文班], begin:9, end:10, location:一教108},  
CourseTableEntry{fn:计算机程序设计基础A[英文班-实验2班], begin:3, end:4, location:一教311},  
CourseTableEntry{fn:金融数据分析与数据挖掘[英文班], begin:5, end:6, location:一教408},  
CourseTableEntry{fn:高等数学 (下) A[英文3班], begin:9, end:11, location:一教111}],  
3: [CourseTableEntry{fn:中国区域民歌[中文班], begin:7, end:8, location:荔园1栋205},  
CourseTableEntry{fn:化学原理[中文班], begin:1, end:2, location:一教405},  
CourseTableEntry{fn:计算机科学与技术前沿讲座 IV[英文班], begin:1, end:2, location:荔园2栋302教室}],  
4: [CourseTableEntry{fn:艺术造型初步[中文班], begin:5, end:6, location:荔园1栋203}],  
5: [CourseTableEntry{fn:传热学[英文班], begin:5, end:8, location:荔园1栋503生物实验室},  
CourseTableEntry{fn:数学分析精讲[中文班], begin:7, end:8, location:一教506},  
CourseTableEntry{fn:数学建模[英文班], begin:5, end:6, location:一教308},  
CourseTableEntry{fn:财务会计[英文班], begin:1, end:4, location:创园1栋1楼青创空间},  
CourseTableEntry{fn:金融数据分析与数据挖掘[英文班], begin:1, end:2, location:一教408}],  
6: [],  
7: [],  
}  
Getting:  
CourseTable{  
1: [],  
2: [],  
3: [],  
4: [],  
5: [],  
6: [],  
7: [],  
}
```

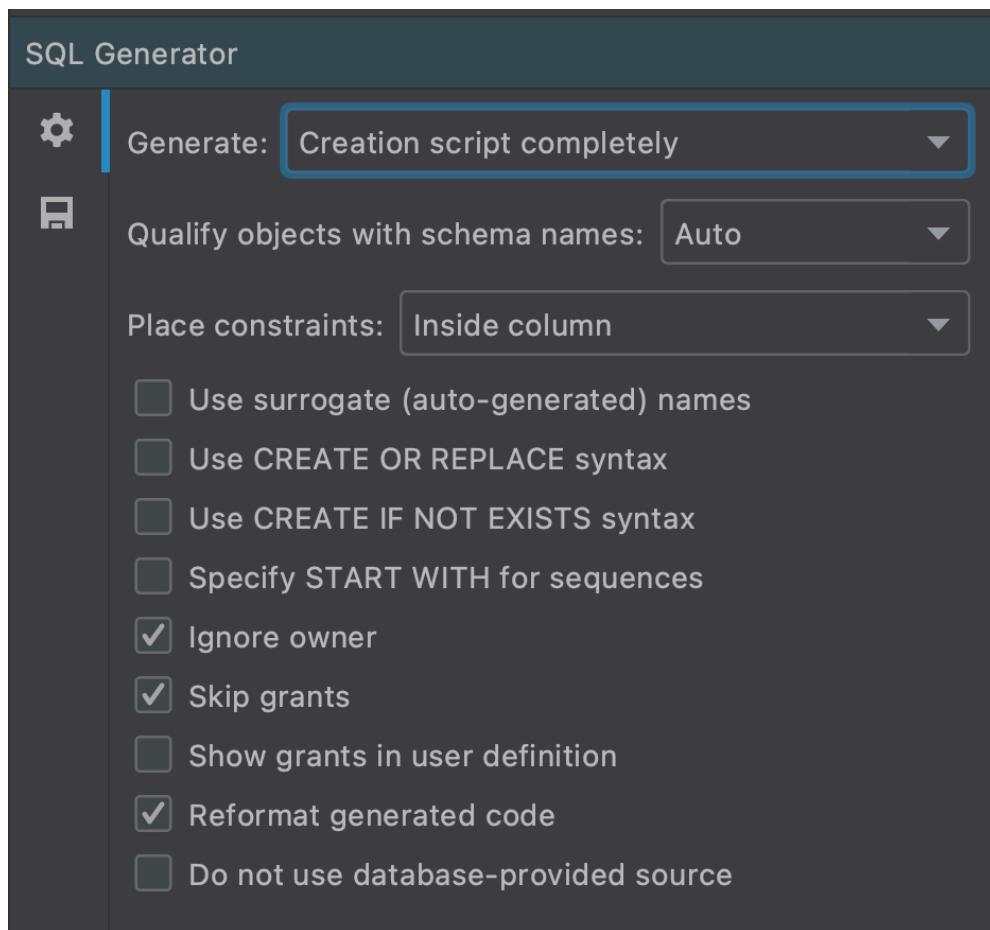
However, DO NOT TRY TO modify the test cases or the judge to make your code "correct". We are not idiots.

## 3. How to generate the SQL file that ensure database is fully rebuildable?

Right click the schema of your project database, click `SQL Generator` in `SQL Scripts`.



Please make sure your settings are consistent with the following image:



Then you can copy the generated SQL, which contains all tables, indexes, and functions.

4. It is recommended to use git to manage group collaboration.
5. How to calculate the number of weeks during two days?

It's recommended to use the following SQL code:

```
SELECT (floor((day_end - day_start) / 7.0)::integer + 1) AS weekoffset;
```