

# Spring 2021 CS307 Project Part 1

---

contributors:

ZHU Yueming , YU Tiancheng, LU Hongyi, WANG Ziqin

- It is a group project with only **2-3** persons. Each group should do it separately and submit one report by group.
- You should submit a report before the deadline. The report submitted after the deadline will be zero score. And finally, in the end of the semester, you still need to complete the Project Part 2 with your original teammates.
- DO NOT copy ANY words, figures and others from Internet and your classmates.

DBMS can help us to manage data conveniently, and also can improve the efficiency of data retrieval greatly, so that your work of the project Part 1 is mainly divided into three parts below:

1. Designing a relational database using postgres according to the given data.
2. Finding ways to import whole data into your database
3. Comparing data retrieval implemented by yourself using C/C++, Java, python or any other language with DBMS to find the differences.

## Task Requirements:

---

### Task 1: Database design

- Design a database by **PostgreSQL** allowing to manage all information mentioned in **course\_info.json** and **select\_course.csv** in this document.
- Your design needs to follow the requirements of the three normal forms
- Use primary key and foreign keys to indicate important attributes and relationships about your data  
  
Every row in each table should to be uniquely identified by its primary key.(You may use simple or composite primary key).
- Every table should be involved in a link. No isolate tables included. (每个表要有外键，或者其他表的外键指向)
- Your design should contain no circular links (对于表之间的外键方向，不能有环)
- Each table should always have at least one mandatory ("Not Null") column(including the primary key but not the id column)
- Tables with no other unique columns than possibly a system-generated ID is not allowed. (除了主键自增的id之外，需要有其他unique约束的列)
- Use appropriate types for different fields of data
- Your design is as easy to expand as possible. (Especially in three-person's group)

## Task 2: Import data

- Design scripts to import data into your database from those two files ( **course\_info.json** and **select\_course.csv**).
- Finding ways to improve the efficiency of time consuming during your importing process, and make comparison between different importing ways.
- Make sure all data is successfully imported. (Especially in three-person's group)

## Task 3: Compare database and file

- Store the data into a database table. Then use DML (Data Manipulation Language) in SQL to do simple analysis of your db. Record the execution time of your algorithm.
- Store the data into a file, and then load it into RAM. The data in RAM can be any format you preferred. Design an algorithm to simple analysis of your file. In this case, you can reorganize data into some other format for faster retrieval. Record the execution time of your algorithm.

You can do some of the following advanced requirements (but not limit to the following ones) to challenging yourself. (Three-person's group should do better in this part)

1. High concurrency and transaction management
2. User privileges management
3. Database index and file IO
4. Compare performance of multiple databases with file system over different operation systems

## Report Structure:

(The words in report for three-person's group should be larger than 2000)

### Part 1. Group Info and Contribution

Need to write down the specific contribution content and the percentage of each team member.

### Part 2. Task 1

Provide a clearly formatted diagram of table structure that generated by datagrip.

Given clearly explanations for the designing of your database, tables and some of columns (if need).

### Part 3. Task 2

Introduce how to design scripts of importing data, and given the main code of your script.

Introduce how to improve the efficiency of importing process, and given the main code and experimental data of your efforts. To make sure the experiments should be reasonable and the comparisons should be fare.

simple attribute  $[A, B] \times \rightarrow [A] [B] \checkmark$   
attributes depend on the full key.  
non-key attributes not depend on each other.

老师, 学生, 数据库管理员, 领导, 校外人员

procedures



教师信息,但不能查看除了自己以外的同学的选课信息

数据库管理员: 查看, 修改所有记录. | 加课

校领导：查看所有信息

校外人员: 查看课程信息, 书院信息, 教师信息

Tech: 对某些 database 的权限 - 可实现

记录权限 > 未知  
column 权限

## Part 4. Task 3

### 1. Data and environment


Introduce the data (If you reorganized), operating system and the programming language you choose. How you organize the data.

### 2. Experiment design

List all comparisons item by item, and to introduce what you want to prove in experiments, then show your results in tables and figures, and analysis the differences of different methods.

In this part, you need to introduce what you want to prove in the experiments. **DO NOT copy any text from Internet.**

Several website can give you some ideas to research, but those are only theory description:

- 
1. [Advantage of database management system over file system](#)
  2. [Advantages of Database Management System](#)
  3. [Characteristics and benefits of a database](#)

### What to submit:

1. Submit a pdf file as your report into sakai website before **April 12th 13:00 pm**.
2. A SQL file about your create table query.
3. Submit necessary scripts (all scripts in task2 are needed).

class table : ID (int) 自增 key

totalCapacity int 30

courseID char GE232

prerequisite 外键 null / (...)

coursehour int 32

CourseCredit int 1

CourseName char "体育IV"

classname char "游泳2班"

courseDept char "体育中心"

teacher char 赵飞 (一对多)

classlist (weeklist) = 1 " 2 " 3 " ... (一对多)

外键  
foreign key (country)  
references countries  
(country-code).  
must be a key

location = "田径"

classtime = "1-2"

weekday = "3"

④ 关于 weeklist 多对多的问题

## Course table

② weekday 要不要一对多

prerequisite

course

(A & B) & (B & C)

belong to course

level

null
1

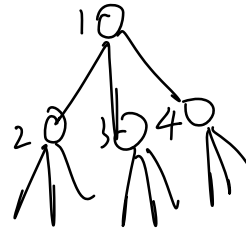
id	element	is-And	belong-to course	level
1		Yes.	A	1
2		Yes	A	2
3		✓	A	2
4	B	✓	A	3
5	C	✓	A	3
6	C	✓	A	3
③	从表上查		A	3

A: (B and C) or (C and D)

Course

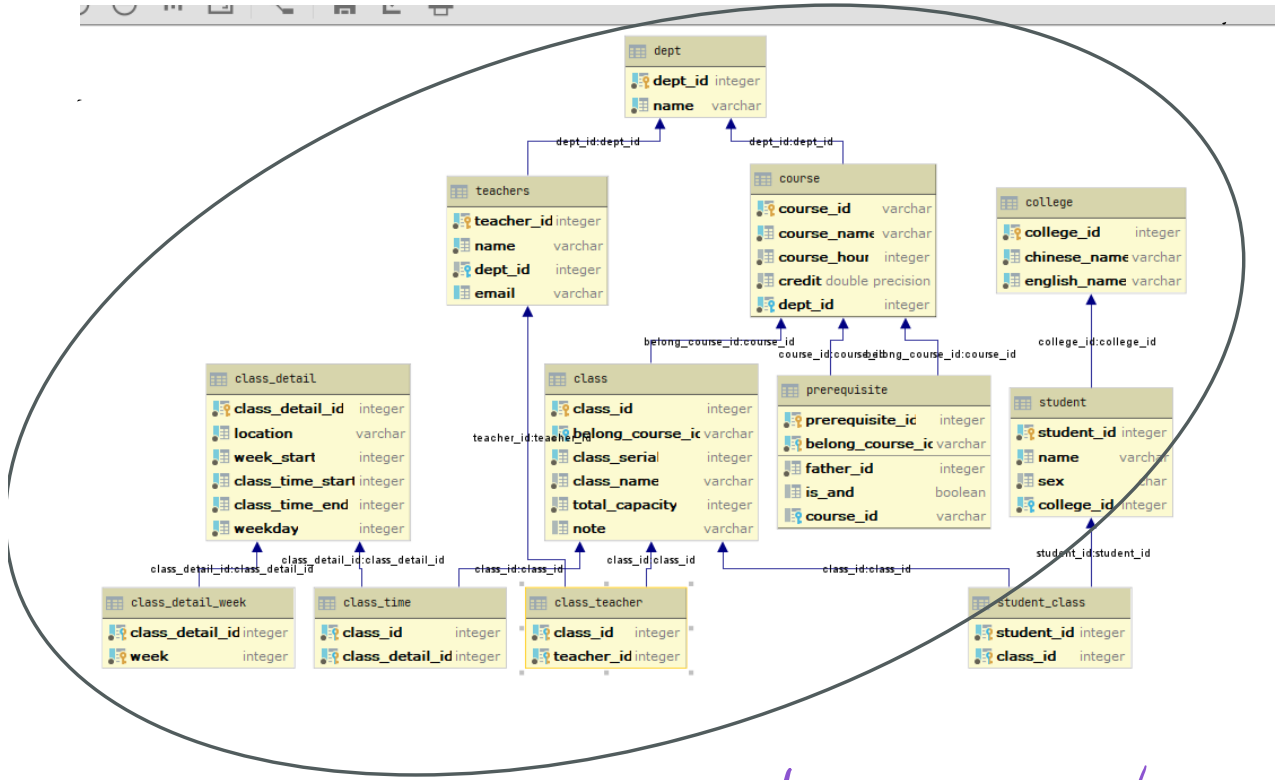
id	is Add	upper-id	course.
①	✓	0	E
②		1	E
③		1	E

E: (A|B) & (C|D)



	id	upper	belong to course name	is-And	course name
(A B) & (C D)	1	0	E	✓	null
(A B)	2	1	E	X	null
(C D)	3	1	E	X	null
A	4	2	E	✓	A
B	5	2	E	✓	B
C	6	3	E	✓	C

D | 7 | 3 | 4 | V | D |



belong to course lesson-name/lesson-number/  
subname/subnumber

写下如何通过表来判断先修课

系统权限.

encrypted.

数据库对象上的操作权限