Students1: Xie Zhou 912143385

Student2: Ho Lun Sin 912316565

ECS165 HW4 Writeup

->We did part 1, part 2, part3, part5a, and partically part5b

**Part 1**: Our Schema includes 5 relations:

1) **Studentcourse**(SID, CID, TERM, UNIT, GRADE, LEVEL, CLASS, MAJOR, STATUS)

It explains a student-course relation. Within a TERM with a unique CID, a student’s performance in that course is explained. Also, a student’s LEVEL, CLASS, MAJOR, STATUS when taking that class is also recorded.

Note: UNIT here the unit a student has earned from the course, not the unit of the course if self.

Keys: SID, CID and TERM combined.

FDs:

SID, CID, TERM -> UNIT, GRADE, LEVEL, CLASS, MAJOR, STATUS

SID, TERM -> LEVEL, CLASS, MAJOR, STATUS

2) **CourseInfo**(CID, TERM, SUBJ, CRSE, SEC, UNIT)

It includes information about a course that has been provided in a certain term.

Note: UNIT here if for course not student.

Key: CID and TERM combined.

FD:

CID, TERM-> SUBJ, CRSE, SEC, UNIT

3) **CourseMeeting**(INSTRUCTOR, CCID, CTERM, TYPE, MON, TUE, WED, THU, FRI, TIMESTART, TIMEEND, BUILD, ROOM, SEATNUM)

This relation explains information about course meetings. TYPE value indicates if the meeting is a lecture or a discussion or a lab etc. MON through FRI values are all Booleans indicating on which day of a week a class happens. TIMESTART and TIMEEND describe the time a meeting starts and ends. The later three values BUILD, ROOM explains the location. SEATNUM is the number of student in that particular meeting. The maximum SEATNUM of a particular classroom would be the room capacity.

Keys: all attributes combined except for SEATNUM.

FDs:

CID,TERM->SEATNUM

therefore, CID,TERM,BUILD,ROOM->SEATNUM (this can be used for 5b)

4) **StudentInfo**(SID, SURNAME, PREFNAME, TERM, MAJOR)

The relation records the majors and terms of students.

Key: SID as key

FD:

SID -> SURNAME, PREFNAME, TERM, MAJOR

5) **InstrCourse**(Instructor, CID, TERM)

This relation is used for joining with other relations to include instructor of a certain course if needed.

Key: all attributes

**Updates:**

One advantage of our schema is that it is relatively intuitive to update. If we are to add a new course, we simply need to add it into the CourseInfo relation, then add the meeting times in CourseMeeting, and also add the Instructor’s name with the course’s CID into the InstrCourse relation. Once students finished the course, their information simply need to be added in StudentCourse. If we have a new student then his/her info just need to be added into StudentInfo. This process is also used when a new term happens, all student’s info would be added again with possible changes into the StudentInfo relation.

If we are to change something, say we need to change a course’s instructor, then it is simply done by updating the InstrCourse Instructor value with the CID and TERM provided. Then update all CourseMeeting INSTRUCTOR value with the CID and TERM provided.

In addition:

(When updating the StudentCourse table, TERM and Major attributes need to be updated in StudentInfo table for the same SID.

When updating the CourseMeeting table, update Instructor, CID, TERM in InstructerCourse table also.

When updating the StudentCourse table, if adding a new CID-TERM combination (which means it’s a new course object since CID-TERM is unique), update the same information in CourseInfo table.)

**Part2 and Part 3:**

Note:

1. readme.txt
2. “make”
3. All queries for part 3 and part 5 are in query\_codes.txt

**Part 3:**

3a)

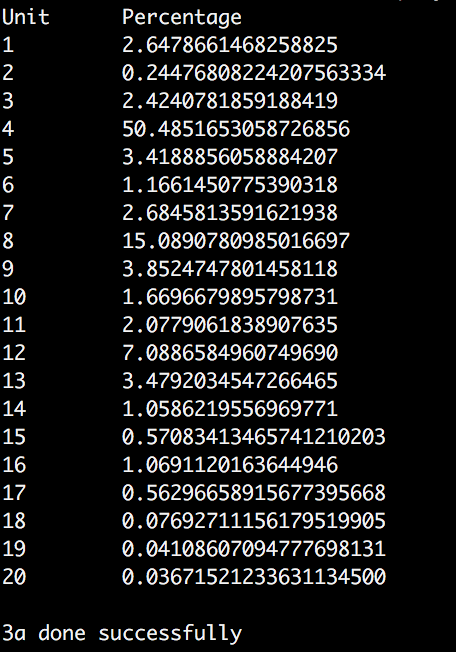


Figure 1 3a. result

3b)

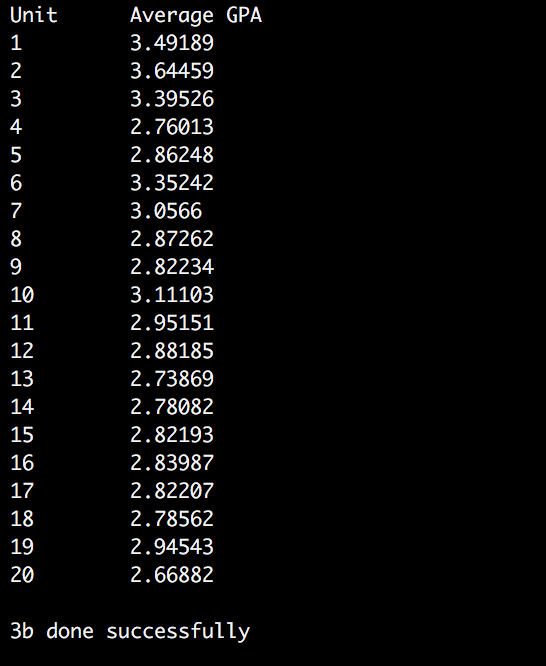


Figure 2 3b. result

3c)

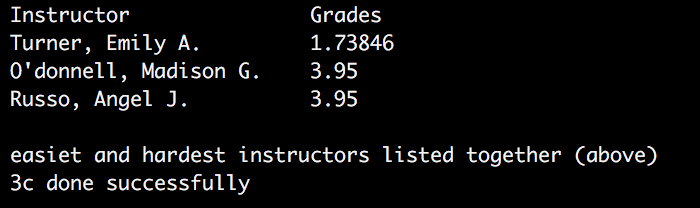


Figure 3 3c.result

3d)

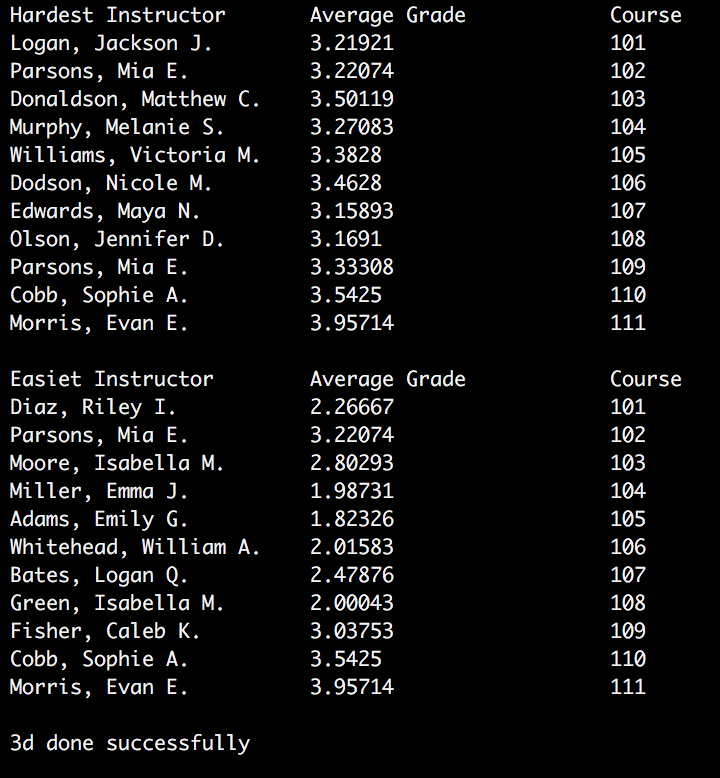


Figure 4 3d.result

3e)

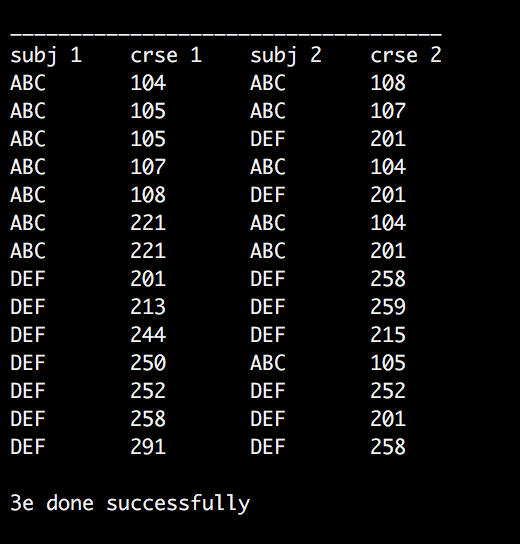


Figure 5 3e.result

3f)

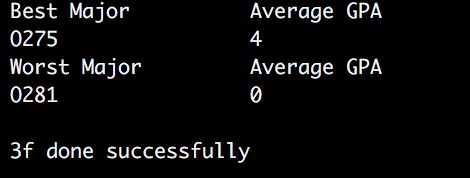


Figure 6 3f.result

3g)

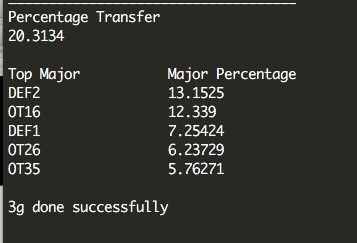
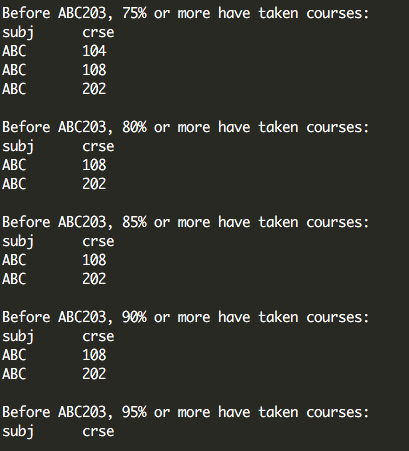


Figure 7 3g.result

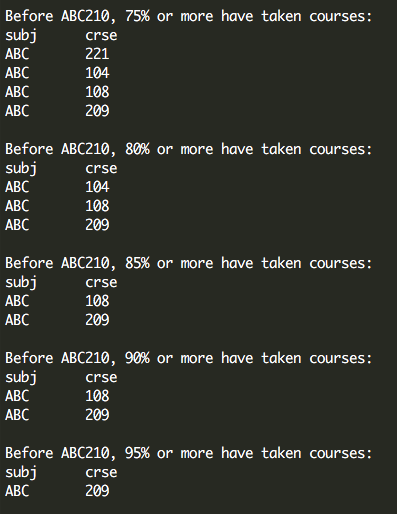
Extra credit:

5a)

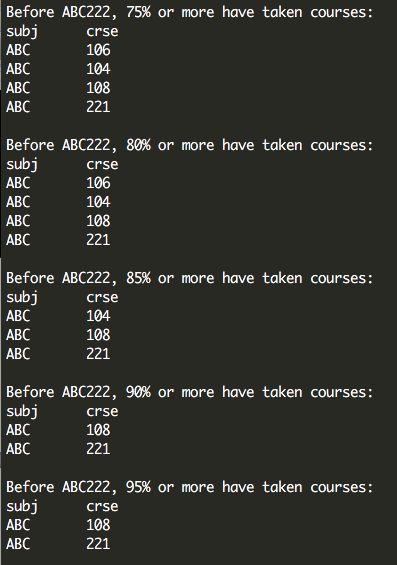
Prerequisites for ABC203:



Prerequisites for ABC 210:



Prerequisites for 222:



5b)

maximum capacity for each room:

1. For courses with different cid and sec, and everything else the same, add up their seat counts (same courses) ->seatnum
2. For courses with different cid, sec, subj, crse, and everything else the same, add up their seat counts (crosslisted courses) -> seatnum
3. For all courses, just collect their SEATNUM attribute (processed before inserting into db)

Find the maximum capacity of build-room by grouping build, rooms, and select max(seatnum) amount the three joined tables

Listed below are the capacities of all rooms (BUILD-ROOM):

