LAB 4 Worksheet

In this lab, we continue trying out SQL queries and also explore new features that we have covered in the class.

******NOTE : We will have a small submission at the end of the lab (either a new question of something from the below only !!)**

Please try out these queries below.
Note : the sample data given by us may not produce any result tuples. You may do one or both of the following:
- add a few more tuples while maintaining database consistency
- change query parameters

1. Add yourself as a student to any department here, and add yourself for a few courses from existing courses. You may see how primary and foreign keys prevent you from entering wrong data. For example, add yourself as a student twice (with same Id) but in different departments. Or add yourself in a non-existing department.

2. Write SQL queries for the following (choose proper values for your queries so that you get non-empty query result !):
a) List courses from a given department (say, English) which have no pre-requisites from same department. Also, as a separate query, list courses which have two level pre-requisite (for ecample course C1 has C2 as pre-requisite and C2 has C3 pre-requisite, then list C1 and C3.
b) Get faculty who are teaching in slot 6 in a given semester.
c) Get count of students from English department who have A+ grades in courses of non-English department

3. Create a view called facload that gives no. of courses and total credits of the courses being taught by all instructors in all semesters of years 2010 and 2011.
Use facload to find a department whose faculty load is highest in various semesters.

4. Write the following query using WITH clause :
i) List courses running in Taylor building that are courses of Comp. Sci. department and which are taken by students of Physics department.
ii) Do the above using nesting in FROM.
iii) Do the above using views. Define views and use them in your query.

5. We want to make database updates as follows:

- create a new room in Watson building (give suitable data for all attributes)
- find largest (in number of students taking that course) offered in this year and semesterby any department housed in the Watson building
- move this course into this room and release its earlier room
- reduce the salary of the instructor of that course by 1000
- increase the budget of this department by 5000

Should all of the above be done as a transaction ? Why ? How will you do it in SQL for postgresql ? (Read the manual for reference if required.