1) m = (course × classroom × section)

n = (σ section.course_id = course.course_id ∧ section.building = classroom.building ∧ course.title = 'Database System Concepts' m)

π Classroom_Capacity ← classroom.capacity (n)

2) π Student_Name ← student.name, Course_Title ← course.title (σ student.ID = takes.ID ∧ takes.grade = 'A' ∧ student.dept_name = course.dept_name ∧ takes.course_id = course.course_id (course × student × takes))

3) σ tot_cred = 0 (student)

4) π instructor.name (σ classroom.capacity < 100 ∧ instructor.ID = teaches.ID ∧ section.room_number = classroom.room_number (instructor × teaches × section × classroom))

5) γ building; max(instructor.salary) → maximum_salary (department ⋈ instructor)

6) a = γ course_id; count(ID) → number_of_students takes

b = π title,course_id,number_of_students (a ⋈ course)

π title,course_id, section.semester,section.year,number_of_students (b ⋈ section)

7) m=γ course_id;count(ID) → count_ID takes

n =γ ;MAX(count_ID) → max (m)

res=m ⋈ (count_ID = max) n ⋈ course

π course.title res

8) Sec_Hig_SAL = π instructor.salary (σ instructor.salary < d.salary

(instructor x ρ d (instructor)))

Sec_Hig_SAL_RES = γ;MAX(instructor.salary) → second_max (Sec_Hig_SAL)

newres = (Sec_Hig_SAL) - (Sec_Hig_SAL_RES)

Top_Two_High_Sal = π salary (instructor) - newres

π instructor.salary,instructor.name (Top_Two_High_Sal ⋈ instructor)

9) takes ⋈ σ semester != 'Summer' (takes)

10) γ dept_name; max(salary) → salary (instructor) ⋈ (σ instructor.salary = salary (instructor))