Computer Science Department

CSC 3300-001 Database Management Systems

Project 1 – KEY (09/02/2021; Fall 2021)

Beata Kubiak

1. Course(s) for which only one section was created in the Spring 2009 semester.

(πcourse\_id (σyear = 2009 ∧ semester = 'Spring' section)) - πs1.course\_id (σs1.course\_id = s2.course\_id ∧ s1.sec\_id <> s2.sec\_id ∧ s1.year = 2009 ∧ s1.semester = 'Spring' ∧ s2.year = 2009 ∧ s2.semester = 'Spring' (ρs1 section ⨯ ρs2 section))

1. Department(s) in the Painter building. We are interested in dept name(s) of this\these department(s) that has\have an office in the Painter building.

πdept\_name σbuilding = 'Painter' department

1. Grades given to students.

πgrade σgrade <> null takes

1. ID(s) of istructor(s) who is\are advisor(s).

πi\_id (σi\_id <> null advisor)

1. Course section(s) instructor(s) Srinivasan taught in the Fall 2009 semester. We are only interested in course\_id and sec\_id here, cause course section is uniquely identified by the values for {course\_id, sec\_id, semester, year} and values for attributes {semester, year} we already know (these are {'Fall', 2009} respectively).

(σsemester = 'Fall' ∧ year = 2009 teaches) ⨝ (σ name = 'Srinivasan' instructor)

1. Instructor(s) who taught course(s) offered by different department they are associated with. Retrieve IDs and names of these instructors only.

πID, name (σcourse.course\_id = teaches.course\_id ∧ instructor.dept\_name <> course.dept\_name ((instructor ⨝ teaches) ⨯ course))

1. IDs and names of students who don't have advisors. Hint: What does the tuple e.g. (s\_id:1, i\_id:null) of the advisor relation mean? Assumption: Value of any foreign key attribute can be null, but value of any primary key attribute can not be null.

(πID student) - (πs\_id σi\_id <> null advisor)

1. Course(s) (course\_id(s) and title(s)) that was\were offered in the Spring 2010 semester.

πcourse\_id, title (course ⨝ σsemester = 'Spring' ∧ year = 2010 section)

1. Course(s) that are a prerequisite to some course(s). We are interested in course\_id(s) only.

πprereq\_id prereq

1. Course(s) with prerereqisite(s) from other departments. We are interested in course\_id(s), title(s) and dept\_name(s) of a course with its prerequisite's course\_id(s), title(s) and deprt\_name(s).

π course.course\_id, course.title, course.dept\_name, p.course\_id, p.title, p.dept\_name (σprereq.prereq\_id = p.course\_id ∧ course.dept\_name <> p.dept\_name ((course ⨝ prereq) ⨯ ρp course))

1. Students who received grade A for course CS-190 in the Spring 2009 semester. Retrieve ID(s) of student(s) along with their name(s).

σcourse\_id = 'CS-190' ∧ semester = 'Spring' ∧ year = 2009 ∧ grade = 'A' takes

1. Course section(s) what student(s) enrolled in but noone taught. We are interested in course\_id(s), sec\_id(s), semester(s), year(s) here.

πcourse\_id, sec\_id, semester, year takes - πcourse\_id, sec\_id, semester, year teaches

1. Classroom(s) with the smallest capacity. We are interested in building and room\_number here.

πbuilding, room\_number (classroom ⨝ (πcapacity classroom - πc1.capacity (σc1.capacity > c2.capacity ((ρc1 classroom) ⨯ (ρc2 classroom)))))

1. Student(s) whose name(s) are the same as name(s) of some instructor(s). We are interested in ID(s) and name(s) of this\those student(s).

πstudent.ID, student.name σstudent.name = instructor.name (student ⨯ instructor)

1. Student(s) taught by their advisor. We are interested in ID(s) and name(s) of this\those student(s).

πstudent.ID, student.name σtakes.course\_id = teaches.course\_id ∧ takes.sec\_id = teaches.sec\_id ∧ takes.semester = teaches.semester ∧ takes.year = teaches.year ∧ teaches.ID = advisor.i\_id ((σstudent.ID = advisor.s\_id (student ⨯ advisor) ⨝ takes) ⨯ teaches)

1. Student(s) instructor with ID 15151 taught. We are interested in ID(s) and name(s) of this\those student(s).

σteaches.course\_id = takes.course\_id ∧ teaches.sec\_id = takes.sec\_id ∧ teaches.semester = takes.semester ∧ teaches.year = takes.year ((σteaches.ID = 15151 teaches) ⨯ takes)

1. Student(s) that got a grade A for both courses: CS-315 and CS-347.

σt1.ID = t2.ID (ρt1 (σcourse\_id = 'CS-315' ∧ grade = 'A' (takes)) ⨯ ρt2 (σcourse\_id = 'CS-347' ∧ grade = 'A' (takes)))

Another solution:  
  
πstudent.ID, student.name (σstudent.ID = takes.ID ∧ (takes.course\_id = 'CS-347' ∨ takes.course\_id = 'CS-315') ∧ takes.grade = 'A' (student ⨯ takes))

1. Course ID(s) and name(s) that were never offered.  
     
   (πcourse\_id course - π course\_id section) ⨝ course
2. Building(s) student(s) Brown had classes in the Spring 2010 semester.  
     
   πbuilding ((σname = 'Brown' student) ⨝ takes ⨝ (σsemester = 'Spring' ∧ year = 2010 section))
3. Name(s) and ID()s of student(s) that instructor(s) Gold taught.  
     
   πstudent.ID, student.name σtakes.course\_id = teaches.course\_id ∧ instructor.name = 'Einstein' ((student ⨝ takes ⨝ section) ⨯ (teaches ⨝ instructor))
4. (BONUS) Course section(s) taught by 2 instructors during the same semester and year.  
     
   π t1.course\_id, t2.course\_id sigma t1.course\_id = t2.course\_id and t1.sec\_id = t2.sec\_id and t1.semester = t2.semester and t1.year = t2.year ∧ t1.ID < t2.ID ((rho t1 teaches) x (rho t2 teaches))

-

π t1.course\_id, t2.course\_id sigma t1.course\_id = t2.course\_id and t1.sec\_id = t2.sec\_id and t1.semester = t2.semester and t1.year = t2.year ∧ t1.ID < t2.ID

and

t1.course\_id = t3.course\_id and t1.sec\_id = t3.sec\_id and t1.semester = t3.semester and t1.year = t3.year ∧ t1.ID < t3.ID

and

t3.course\_id = t2.course\_id and t3.sec\_id = t2.sec\_id and t3.semester = t2.semester and t3.year = t2.year ∧ t2.ID < t3.ID

((rho t1 teaches) x (rho t2 teaches) x (rho t3 teaches))