Relational Algebra – Class Exercise – 2

Consider the following schema (primary key fields are underlined):

The attributes `Student.major`, `Track.major`, and `TrackRequirements.major` contain just the acronym for the major, e.g., 'CptS','EE', etc.

`Course(courseno, credits, enroll_limit, classroom)`

`-- "courseno"` includes a string, like 'CptS451'.

`Tracks(major, trackcode, title)`

`Student(sID, sName, major, trackcode)`

`-- Student(major,trackcode)` is a foreign key referencing `Tracks(major,trackcode)`

`Enroll(courseno, sID, grade)`

`-- Enroll(courseNo)` is a foreign key referencing `Course(courseno)`
`-- Enroll(sID)` is a foreign key referencing `Student(sID)`

`Prereq(courseno, preCourseNo)`

`-- Prereq(courseNo)` is a foreign key referencing `Course(courseNo)`
`-- Prereq(preCourseNo)` is a foreign key referencing `Course(courseNo)`

`TrackRequirements(major, trackcode, courseno)`

`--TrackRequirements(major,trackcode) is a foreign key referencing Tracks(major,trackcode)`
`-- TrackRequirements(courseNo)` is a foreign key referencing `Course(courseNo)`

Based on the above database schema, write the following queries in relational algebra. Assume set semantics for all operations.

https://dbis-uibk.github.io/relax/calc/gist/6f3e57cb72d66f6c74438b210d6a7ba7

Write the following queries in Relational Algebra:

1. Find the distinct courses that 'SYS' track students in 'CptS' major are enrolled in. Return the courseno and credits for those courses. Return results sorted based on `courseNo`.

2. Find the sorted names, ids, majors, track codes, and total credits of the students who took (enrolled in) more than 18 credits (19 or above).

3. Find the distinct courses that only 'SE' track students in 'CptS major have been enrolled in. Return the courseNo's of those courses.

4. Find the students who have enrolled in the courses that Diane enrolled and earned the same grade Diane earned in those courses. Return the student name, sID, major as well as the courseno and grade for those courses.

5. Find 'CptS' major students who enrolled in a course for which there exists a prerequisite that the student got a grade lower than "2". (For example, Alice (sid: 12583589) was enrolled in CptS355 but had a grade 1.75 in prerequisite CptS223.) Return the names and sIDs of those students and the courseno of the course (i.e., the course whose prereq had a low grade).

Solutions:

1. τ courseNo (π courseNo, credits (σ major ='CptS' and trackcode='SYS' (Student ⋈ Enroll ⋈ Course)))

2. τ sName (π sName,sID,major,trackcode,totalCredits (Student ⋈ σ totalCredits>18 (γ sID;sum(credits)->totalCredits (Enroll ⋈ Course))))

3. π courseNo (σ major='CptS' and trackcode='SE' (Student ⋈ Enroll)) -
   π courseNo (σ major<>'CptS' or trackcode ≠ 'SE' (Student ⋈ Enroll))

4. DianesCourses = σ sName='Diane' (Student ⋈ Enroll)
   enroll_2 = ρ Enroll2 (Enroll)
   student_2 = ρ Student2 (Student)
   enrolled_in_samecourse =
      DianesCourses ⋈ (Enroll.courseNo=Enroll2.courseNo and Enroll.grade=Enroll2.grade) enroll_2

  π Student2.sName, Student2.sID, Student2.major, Enroll2.courseNo, Enroll2.grade
  (enrolled_in_samecourse ⋈ (Enroll2.sID = Student2.sID and Student.sID ≠ Student2.sID) student_2)

5. prereq_courses = (ρ Enroll1 (Enroll) ⋈ Prereq)

   the_courses = (ρ Enroll2 (Enroll))

   R = prereq_courses ⋈(Prereq.preCourseNo = Enroll2.courseNo and Enroll1.sID=Enroll2.sID and Enroll2.grade<2) the_courses

   π Student.sName, Student.sID, Enroll1.courseNo (Student ⋈(Student.sID=Enroll1.sID) R)