RDBMS and SQL, September-November 2022

Assignment 1: Relational Algebra

Anjan Mondal
MDS202208

October 14, 2022

The **Course Registration** database has five relations, *Students, Faculty, Courses, Instructors and Enrolment.* Sample Instances are as below...

Students

RollNo	*Name*	*Email*
MS2201	Sachin Tendulkar	sachin
MS2202	Brian Lara	lara
MS2203	Ricky Ponting	ponting
...	...	...

Faculty

Name	*Email*
Madhavan Mukund	madhavan
Rajeeva Karandikar	rlk
Sourish Das	sourish
...	...

Courses

Course Code	*Title*
DG1101	Mathematical Methods - Analysis
DG1102	Programming and Statistics with R
DG1103	Programming & Data Structures with Python
DG1106	RDBMS & SQL
DG1107	Visualization
...	...

Instructors

Course Code	Year	Semester	Teacher
DG1101	2022-2023	1	rlk
DG1102	2022-2023	1	sourish
DG1103	2022-2023	1	madhavan
DG1106	2022-2023	1	madhavan
DG1107	2022-2023	1	sourish
...	...	...	...

Enrolment

Rollno	*Course Code*	*Year*	*Semester*	*Grade*
MS2201	DG1102	2022-2023	1	NA
MS2203	DG1102	2022-2023	1	NA
MS2201	DG1103	2022-2023	1	NA
MS2201	DG1106	2022-2023	1	NA
MS2201	DG1107	2022-2023	1	NA
...	...	...	...	...

Write relational algebra queries that answer the following questions. Along with each query, provide a brief explanation of how it achieves the desired result. (For relational algebra, use the syntax given in the book by Silberschatz et al — see the slides posted on the course page.)

1. *List course codes and titles of all courses being offered in the current semester.*

Current Semester is 1 and the Year is 2022-2023.
Hence, for the relational query: First, we must have a Natural Join between Instructors and Courses with a common Course Code. Then, Select on the Current Semester and Year and lastly, project the Course Code and Titles.

$$\Pi_{Courses.Course\ Code,Title}(\sigma_{Semester=1 \wedge Year=2022-2023}(\\ Instructors \bowtie_{Instructors.Course\ Code=Courses.Course\ Code} Courses))$$

2. *List email addresses of all students enrolled for RDBMS & SQL in the current semester.*

Current Semester is 1 and the Year is 2022-2023.
Hence, for the relational query: First, do a natural join between Enrolment and Courses with a common Course Code. Then, select for current semester and courses with title RDBMS & SQL. Lastly, project the Name and Email columns.

$$\Pi_{Email}(\sigma_{Title="RDBMS\ \&\ SQL" \wedge Year=2022-2023 \wedge Semester=1}(\\ Enrolment \bowtie_{Enrolment.Course\ Code=Courses.Course\ Code} Courses))$$

3. *List names and email ids of faculty teaching two or more courses in the same semester.*

First, take the cartesian product of the Instructors relation with itself and select rows which don't have similar course codes with same teacher, semester and year. This gives us rows with faculty having more than one courses in a semester. Rename the relations before taking the cartesian product. Then, take the natural join of the previously obtained relation with the Faculty relation with Teacher and Email column being same from their respective relations. Finally project the Name and Email columns.

$$\Pi_{Name,Email}(\sigma_{I.CourseCode \neq In.CourseCode \wedge I.Teacher=In.Teacher I.Semester=In.Semester \wedge I.Year=In.Year} \\ (\rho_I(Instructors) \times \rho_{In}(Instructors)) \bowtie_{Email=In.teachers} Faculty)$$

4. *List names and email ids of faculty who have taught the same course more than once.*

First, take the cartesian product of the Instructor relation with itself and select on rows which have same faculty, same course but a different semester (either a different year or a different semester). This gives us rows with same faculty teaching the same course in a different year or semester (more than once). Now take a natural join of the previously obtained relation with the Faculty relation with Teacher and Email column being the same from their respective relations. Finally, project the Name and Email Columns.

$$\Pi_{Name,Email}(\\ \sigma_{(I.Year \neq In.Year\ \vee\ I.Semester \neq In.Semester) \wedge (I.CourseCode=In.CourseCode) \wedge (I.Teacher=In.Teacher)} \\ (\rho_I(Instructor) \times \rho_{In} Instructor)) \bowtie_{Email=I.Teacher} Faculty)$$

5. *List roll numbers and names of students enrolled for 3 or more courses in the same semester.*

First, take the cartesian product of the enrolment with itself and select on rows with different course id and same RollNo, Year and Semester. This would give us rows of students with more than one course in the same semester(Year, Sem). Rename relations before taking the cartesian product. Now take a cartesian product of the previously obtained relation with Instruction selecting for different course id

than the previous two relations but same RollNo, Year and Semester. This gives us rows with students taking 3 or more courses in the same Semester. Now take a natural join of the previously obtained relation with the relation Students with RollNo columns being same. Now take the projection of the RollNo and Name columns.

$$\Pi_{RollNo,Name}(\sigma_{Enr.Course\ Code \neq En.Course\ Code \wedge Enr.Course\ Code \neq E.Course\ Code \wedge E.Semester = Enr.Semester \wedge E.Year = Enr.Year \wedge E.RollNo = Enr.RollNo}($$
$$\sigma_{E.CourseCode \neq En.Course\ Code \wedge E.Semester = En.Semester \wedge E.Year = En.Year \wedge E.RollNo = Enr.RollNo}($$
$$\rho_E(Enrolment) \times$$
$$\rho_{En}(Enrolment)) \times \rho_{Enr}(Enrolment))$$
$$\bowtie_{E.RollNo = Students.RollNo} Students)$$

6. *List email addresses of all students enrolled for Programming & Data Structures with Python but not registered for RDBMS & SQL in the current semester. (Hint: Use set difference)*

Take a Natural Join of Enrolment and Courses where Course Code is equal. Select rows from the previous relation where Title is RDBMS & SQL and construct another relation by selecting rows where Title is Programming & Data Structures with Python. Subtract the first relation from the second. This gives us rows where students have taken only the Programming & Data Structures Course. Now take the relation from the previous substraction and do a natural join with the relation Students where RollNo matches. This adds the columns names and emails to the previous relation. Finally, take a projection of the Email column.

$$\Pi_{Email}((\sigma_{Title = "Programming\ \&\ Data\ Structures\ with\ Python"}($$
$$Enrolment \bowtie_{Enrolment.Course\ Code = Courses.CourseCode} Courses) -$$
$$\sigma_{Title = "RDBMS\ \&\ SQL"}(Enrolment \bowtie_{Enrolment.Course\ Code = Courses.CourseCode} Courses))$$
$$\bowtie_{Enrolment.Rollno = Students.RollNo} Students)$$

7. *List course codes and titles of all courses that have run exactly once. (Hint: Use set difference.)*

First, take the Cartesian Product of the relation Instructors with itself and select for rows that have different years and semesters but same Course Code. This gives us a relation with rows having courses which have run more than once. Rename the relations before taking the Cartesian Product. Now Project the Course Codes from the relation. Now take a projection of the Course Code column from the Instructors relation. Subtract from this relation the previously obtained relation using Cartesian Product. This gives us rows with Course Codes having run exactly once. Now take a natural join of the previous relation with the Courses relation with the Course Code columns being the same. Finally, project the Course Code and titles.

$$\Pi_{Courses.Course\ Code,\ Courses.Title}((\Pi_{I.CourseCode} Instructors - \Pi_{I.CourseCode}($$
$$\sigma_{(I.Year \neq In.Year\ \vee\ I.Semester \neq In.Semester) \wedge (I.Course\ Code = In.Course\ Code)}($$
$$\rho_I(Instructors) \times \rho_{In}(Instructors)))$$
$$\bowtie_{I.Course\ Code = Courses.Course\ Code} Courses)$$