University of Toronto
CSC343, Fall 2021

Assignment 1:
Prerak Chaudhari — 1005114760
Jaakulan Subeethakumar — 1005225757

Part 1: Queries

1. *Extreme grades:* For each assignment find the highest and the lowest grade any group received. Report the assignment ID, highest grade, and lowest grade. If an assignment has no grades, it won't appear in the result.

   – What grade a group got on their assignment

   $GroupMarks(aID, gID, mark) := \Pi_{aID,gID,mark}(Group \bowtie Result)$

   – Compare two groups of the same assignment

   $GroupMarksCP(aID, gID1, mark1, gID2, mark2) :=$

   $\Pi_{T1.aID,T1.gID,T1.mark,T2.gID,T2.mark}(\sigma_{T1.aID=T2.aID}[(\rho_{T1}GroupMarks)\times(\rho_{T2}GroupMarks)])$

   – All groups that did not get the max grade

   $NotMax(gID) := \Pi_{gID1}(\sigma_{mark1<mark2}GroupMarksCP)$

   – Groups that got the maximum grade

   $gIDMax(gID) := \Pi_{gID}Group - NotMax$

   – All groups that did not get the min grade

   $NotMin(gID) := \Pi_{gID1}(\sigma_{mark1>mark2}GroupMarksCP)$

   – Groups that got the minimum grade

   $gIDMin(gID) := \Pi_{gID}Group - NotMin$

   – Highest grade for each assignment

   $AssignmentHighMarks(aID, highest) := \Pi_{aID,mark}(GroupMarks \bowtie gIDMax)$

   – Lowest grade for each assignment

   $AssignmentLowMarks(aID, lowest) := \Pi_{aID,mark}(GroupMarks \bowtie gIDMin)$

– Grade extremes for each assignment

$AssignmentGradeExtremes(aID, highest, lowest) :=$

$\quad AssignmentHighMarks \bowtie AssignmentLowMarks$

2. *Procrastinators:* For each assignment (that has at least one required file) and each group (that has more than one member), find the group member whose earliest submission of a required file came latest among the group members. Report the assignment ID, group ID, and userName. Groups that have submitted no required file on an assignment will not be reported for that assignment.

– List user memberships for multi-person groups

$$MultiPersonGroupMembership(userName, gID, status) :=$$
$$\Pi_{T1.userName,T1.gID,T1.status}(\sigma_{T1.userName \neq T2.userName \overset{\wedge}{} T1.gID=T2.gID}[(\rho_{T1}Membership)\times(\rho_{T2}Membership)])$$

– Required submissions by multi-person groups

$$RequiredSubmission(aID, sID, fileName, userName, gID, when) :=$$
$$Required \bowtie Submission \bowtie \Pi_{gID} MultiPersonGroupMembership$$

– User submissions that were not their earliest for that particular assignment

$$NotEarliestUserSubmission(aID, sID, fileName, userName, gID, when) :=$$
$$\Pi_{\substack{T1.aID,\\ T1.sID,\\ T1.fileName,\\ T1.userName,\\ T1.gID,\\ T1.when}} (\sigma_{\substack{T1.when>T2.when \\ \overset{\wedge}{} T1.userName=T2.userName \\ \overset{\wedge}{} T1.gID=T2.gID}} [(\rho_{T1}RequiredSubmission)\times(\rho_{T2}RequiredSubmission)])$$

– User submissions that were their earliest for that particular assignment

$$EarliestUserSubmission(aID, sID, fileName, userName, gID, when) :=$$
$$RequiredSubmission - NotEarliestUserSubmission$$

– Earliest user submissions that were not the latest of their group

$$NotLatestSubmission(aID, sID, fileName, userName, gID, when) :=$$
$$\Pi_{\substack{T1.aID,\\ T1.sID,\\ T1.fileName,\\ T1.userName,\\ T1.gID,\\ T1.when}} (\sigma_{T1.when<T2.when \overset{\wedge}{} T1.gID=T2.gID}[(\rho_{T1}EarliestUserSubmission)\times(\rho_{T2}EarliestUserSubmission)])$$

– Earliest user submissions that were the latest of their group

$$LatestSubmission(aID, gID, userName) :=$$
$$\Pi_{aID,gID,userName}(EarliestUserSubmission - NotLatestSubmission)$$

3. *Good students:* Find every student who meets the following conditions: (a) they have a grade on all the assignments, (b) their grades are all at least 70, and (c) if we consider assignments in order by their due date, their grades have never gone down. Report the userName of these students.

– Groups that got a mark of at least 70

$AtLeast70(gID, mark) := \Pi_{gID,mark}(\sigma_{mark \geq 70} Result)$

– Student membership tuples of said groups

$InterestedMemberships(userName, gID, status) := Membership \bowtie \Pi_{gID} AtLeast70$

– Student membership tuples that weren't graded or had a mark less than 70

$Failures(userName, gID, status) := Membership - InterestedMemberships$

– Students who meet criterion (a) and (b)

$PromisingUsers(userName) := \Pi_{userName} Membership - \Pi_{userName} Failures$

– userName and gID of said users

$PromisingUserGroups(userName, gID) := \Pi_{userName,gID} Membership \bowtie PromisingUsers$

– Tuples that contain the necessary attributes to determine criterion (c)

$InterestedAttributes(userName, gID, mark, aID, due) :=$

$[PromisingUserGroups \bowtie AtLeast70 \bowtie \Pi_{gID,aID} Group] \bowtie \Pi_{aID,due} Assignment$

– Students whose marks went down as per criterion (c)

$MarksWentDown(userName) :=$

$$\Pi_{userName}[\sigma_{\substack{T1.aID \neq T2.aID \\ \wedge \\ T1.gID \neq T2.gID \\ \wedge \\ T1.userName = T2.userName \\ \wedge \\ T1.due < T2.due \\ \wedge \\ T1.mark > T2.mark}} ([\rho_{T1} InterestedAttributes] \times [\rho_{T2} InterestedAttributes])]$$

– Usernames of students who meet criterion (a), (b) and (c)

$GoodStudents(userName) := PromisingUsers - MarksWentDown$

4. *Well-used tags:* Let's say that a tag is "well used" on an assignment if at least 3 different TAs used that tag for at least 3 different groups. Find the assignment whose number of well used tags was the highest. Report the assignment ID and assignment description. If there was a tie for highest, report them all.

   Cannot be expressed.

5. *Different group sizes:* Find every student who has worked in a group of three (three including themself) on some assignment, a group of two on some assignment (two including themself), and has also worked alone on some assignment. Report their username, highest mark on a group-of-three assignment, highest mark on a group-of-two assignment, and highest mark on a group-of-one assignment.

– Rename Membership relation to T1 for later use in Cartesian products

$T1(userName, gID, status) := \rho_{T1} Membership$

– Rename Membership relation to T2 for later use in Cartesian products

$T2(userName, gID, status) := \rho_{T2} Membership$

– Rename Membership relation to T3 for later use in Cartesian products

$T3(userName, gID, status) := \rho_{T3} Membership$

– Rename Membership relation to T4 for later use in Cartesian products

$T4(userName, gID, status) := \rho_{T4} Membership$

– Groups with at least 4 students

$$AtLeast4(user1, gID1, user2, gID2, user3, gID3, user4, gID4) :=$$
$$\Pi_{\substack{T1.userName, \\ T1.gID, \\ T2.userName, \\ T2.gID, \\ T3.userName, \\ T3.gID, \\ T4.userName, \\ T4.gID}} [\sigma_{\substack{T1.gID=T2.gID=T3.gID=T4.gID \\ \wedge \\ T1.userName \neq T2.userName \\ \wedge \\ T1.userName \neq T3.userName \\ \wedge \\ T1.userName \neq T4.userName \\ \wedge \\ T2.userName \neq T3.userName \\ \wedge \\ T2.userName \neq T4.userName \\ \wedge \\ T3.userName \neq T4.userName}} (T1 \times T2 \times T3 \times T4)]$$

– Groups with at least 3 students

$$AtLeast3(user1, gID1, user2, gID2, user3, gID3) :=$$
$$\Pi_{\substack{T1.userName, \\ T1.gID, \\ T2.userName, \\ T2.gID, \\ T3.userName, \\ T3.gID}} [\sigma_{\substack{T1.gID=T2.gID=T3.gID \\ \wedge \\ T1.userName \neq T2.userName \\ \wedge \\ T1.userName \neq T3.userName \\ \wedge \\ T2.userName \neq T3.userName}} (T1 \times T2 \times T3)]$$

– Groups with at least 2 students

$AtLeast2(user1, gID1, user2, gID2) :=$

$$\Pi_{\substack{T1.userName,\\ T1.gID,\\ T2.userName,\\ T2.gID}}[\sigma_{\substack{T1.gID=T2.gID \\ \wedge \\ T1.userName \neq T2.userName}}(T1 \times T2)]$$

– Groups with exactly 3 students

$Exactly3(userName, gID) := \Pi_{user1,gID1} AtLeast3 - \Pi_{user1,gID1} AtLeast4$

– Groups with exactly 2 students

$Exactly2(userName, gID) := \Pi_{user1,gID1} AtLeast2 - \Pi_{user1,gID1} AtLeast3$

– Groups with exactly 1 student

$Exactly1(userName, gID) :=$

$$\rho_{AtLeast1(user1,gID1)}(\Pi_{userName,gID} Membership) - \Pi_{user1,gID1} AtLeast2,$$

– Students who have worked in trio, duo and solo groups

$StudentsInAll3(userName) := [\Pi_{userName} Exactly3] \cap [\Pi_{userName} Exactly2] \cap [\Pi_{userName} Exactly1]$

– Marks of these students while in a group of 3

$TrioMarks(userName, trioMark) :=$

$$\Pi_{userName,mark}(StudentsInAll3 \bowtie Exactly3 \bowtie Result)$$

– Marks of these students while in a group of 2

$DuoMarks(userName, duoMark) :=$

$$\Pi_{userName,mark}(StudentsInAll3 \bowtie Exactly2 \bowtie Result)$$

– Marks of these students while in a group of 1

$SoloMarks(userName, soloMark) :=$

$$\Pi_{userName,mark}(StudentsInAll3 \bowtie Exactly1 \bowtie Result)$$

– Non-highest trio marks of these students

$NotHighestTrioMarks(userName, trioMark) :=$

$$\Pi_{T1.userName,T1.trioMark}(\sigma_{T1.userName \neq T2.userName \atop \wedge \; T1.trioMark < T2.trioMark}[(\rho_{T1} TrioMarks) \times (\rho_{T2} TrioMarks)])$$

– Highest trio marks of these students

$HighestTrioMarks(userName, trioMark) := TrioMarks - NotHighestTrioMarks$

– Non-highest duo marks of these students

$NotHighestDuoMarks(userName, duoMark) :=$

$$\Pi_{T1.userName,T1.duoMark}(\sigma_{T1.userName \neq T2.userName \atop \wedge \; T1.duoMark < T2.duoMark}[(\rho_{T1} DuoMarks) \times (\rho_{T2} DuoMarks)])$$

– Highest duo marks of these students

$HighestDuoMarks(userName, duoMark) := DuoMarks - NotHighestDuoMarks$

– Non-highest solo marks of these students

$NotHighestSoloMarks(userName, soloMark) :=$

$$\Pi_{T1.userName,T1.soloMark}(\sigma_{T1.userName \neq T2.userName \atop \wedge \; T1.soloMark < T2.soloMark}[(\rho_{T1} SoloMarks) \times (\rho_{T2} SoloMarks)])$$

– Highest solo marks of these students

$HighestSoloMarks(userName, soloMark) := SoloMarks - NotHighestSoloMarks$

– Report the student's username and their highest solo, duo and trio group grades

$DesiredRelation(userName, soloMark, duoMark, trioMark) :=$

$$HighestSoloMarks \bowtie HighestDuoMarks \bowtie HighestTrioMarks$$

6. *Consistent groups:* Find all pairs of groups that, for two different assignments, have the exact same students in them, and earned the exact same mark. Report the two group IDs, the two assignment IDs, and the mark. For example, you might report that group 7249 on Assignment 2 and group 8116 on Assignment 4 had the same group members and earned the same grade of 84. Do not include the same pair of groups twice as pseudo-duplicates (*e.g.,* Do not include both (7249, 8116) and (8116, 7249))

– The marks each student got on their assignments

$UserMarks(aID, gID, userName, mark) :=$

$$\Pi_{aID,gID,userName,mark}(Group \bowtie Membership \bowtie Result)$$

– Groups with at least 1 common student who got the same grade on both assignments

$Actual(gID1, userName, gID2) :=$

$$\Pi_{T1.gID,T1.userName,T2.gID}[\sigma_{\substack{T1.aID \neq T2.aID \\ \wedge \\ T1.gID \neq T2.gID \\ \wedge \\ T1.userName = T2.userName \\ \wedge \\ T1.mark = T2.mark}}([\rho_{T1} UserMarks] \times [\rho_{T2} UserMarks])]$$

– Ideally, the group pairings from the previous relation share the same students

$Expected(gID1, userName, gID2) :=$

$$\Pi_{gID1,userName,gID2}(\Pi_{gID1,gID2} Actual \bowtie_{gID1=gID} Membership)$$

– Group pairings that got the same grade but do not share the same members

$Failures(gID1, gID2) := \Pi_{gID1,gID2}(Expected - Actual)$

– Pair each tuple with itself, flipping the order of the gIDs

$TwoWayFailuresCP(gID1, gID2, gID3, gID4) :=$

$$\Pi_{\substack{T1.gID1, \\ T1.gID2, \\ T2.gID2, \\ T2.gID1}}(\sigma_{T1.gID1=T2.gID1}[(\rho_{T1} Failures) \times (\rho_{T2} Failures)])_{\substack{\\ \\ \\ T1.gID2 = T2.gID2}}$$

– Bidirectional failures of group pairings

$TwoWayFailures(gID1, gID2) :=$

$$[\Pi_{gID1,gID2}(TwoWayFailuresCP)] \cup [\rho_{T1(gID1,gID2)}(\Pi_{gID3,gID4} TwoWayFailuresCP)]$$

– Pairing groups with the same members and mark

$Success(gID1, gID2) := \Pi_{gID1,gID2} Actual - TwoWayFailures$

– Group pairings with no pseudo-duplicates

$NoDups(gID1, gID2) := \sigma_{gID1<gID2} Success$

– Report the assignment ids, group ids and mark

$$DesiredRelation(aID1, gID1, aID2, gID2, mark) :=$$

$$\Pi_{\substack{T1.aID,\\ gID1,\\ T2.aID,\\ gID2,\\ mark}}[\sigma_{\substack{gID1=T1.gID \\ \wedge \\ gID2=T2.gID \\ \wedge \\ gID1=Result.gID}} (NoDups \times [\rho_{T1}Group] \times [\rho_{T2}Group] \times Result)]$$

7. *Prolific TA:* For each assignment due in 2020, find the TA who was grader for the highest number of groups of all TAs. Report the TA, the assignment, and the highest grade that TA has given on the assignment. If there are multiple TAs tied for grading the most groups on an assignment, report them all. You may assume that for all assignments due in 2020, every group has received a grade.

   Cannot be expressed.

8. *Worked with top student:* Find all students who've been in a group with a student who has the highest mark in the entire database. (We said "a student" rather than "the student" because there could be several students tied for highest mark; we are looking for students who've worked with any of them.) For each of them, report their user name, their grade on the last assignment (last according to due dates), and the number of *different* students they have been in a group with.

   Cannot be expressed.

Part 2: Additional Integrity Constraints

1. A grader cannot be assigned to mark the same student for more than one assignment.

   – Student-Grader pairings for each group

   $Pairings(aID, gID, student, grader) :=$

   $$\Pi_{\substack{aID,\\ Group.gID,\\ Membership.userName,\\ Grader.userName}} [\sigma_{Group.gID=Grader.gID=Membership.gID}(Group \times Grader \times Membership)]$$

   $$\sigma_{\substack{T1.aID \neq T2.aID \\ \wedge \\ T1.gID \neq T2.gID \\ \wedge \\ T1.student = T2.student \\ \wedge \\ T1.grader = T2.grader}} [(\rho_{T1} Pairings) \times (\rho_{T2} Pairings)] = \emptyset$$

2. Students are not allowed to "overinvite": You can't invite more people to join your group for an assignment than the maximum number of students allowed to work together on that assignment.

   Cannot be expressed.

3. A group who submits one or more files after the deadline cannot receive a mark over 80 unless the grader is an instructor.

– When a group submitted a file

$SubmissionTimes(gID, when) := \Pi_{gID,when} Submission$

– All group submissions that weren't their latest

$NotLatestSubmission(gID, when) :=$

$$\Pi_{T1.gID,T1.when}[\sigma_{\substack{T1.gID=T2.gID \\ \wedge \\ T1.when<T2.when}} ([\rho_{T1} SubmissionTimes] \times [\rho_{T2} SubmissionTimes])]$$

– Latest submission time for each group

$LatestSubmission(gID, when) := SubmissionTimes - NotLatestSubmission$

– Append assignment ID and due date onto latest submission tuples

$LatestAssignmentSubmission(aID, gID, dueDate, submissionDate) :=$

$$\Pi_{aID,gID,due,when}(\Pi_{aID,gID,when}[LatestSubmission \bowtie Group] \bowtie Assignment)$$

– Groups who submitted past the due date

$PastDueDate(gID) := \Pi_{gID}(\sigma_{submissionDate>dueDate} LatestAssignmentSubmission)$

– What grade these groups got

$GroupGrades(gID, mark) := \Pi_{gID,mark}(PastDueDate \bowtie Result)$

– Late submitting groups who got a grade over 80

$Over80(gID) := \Pi_{gID}(\sigma_{mark>80} GroupGrades)$

– The graders of said groups

$Markers(gID, userName, type) := \Pi_{gID,userName,type}(Over80 \bowtie Grader \bowtie User)$

$\Pi_{type} Markers \subseteq \{\text{“}instructor\text{”}\}$