CSC343 - A1 Part 1

Q1)

-- all results for each assignment
AssignmentGrades(aID,mark) := Π aID,mark(Group ⋈ Result)

--all combinations of grades for the each assignment
InOrderGrades(aID1, mark1, aID2, mark2) :=
σ T1.aID = T2.aID (ρT1 AssignmentGrades X ρT2 AssignmentGrades)

--all the assignment grades that are not the maximum grades
NotMax(aID, mark) := Π aID1, mark1 σ aID1 = aID2 and mark1 < mark2 (InOrderGrades)

Max(aID, highest) = ρ(aID, highest) AssignmentGrades - NotMax

--all the assignment grades that are not the minimum grades
NotMin(aID, mark) := Π aID1, mark1 σ aID1 = aID2 and mark1 > mark2 (InOrderGrades)

Min(aID, lowest) := ρ(aID, lowest) AssignmentGrades - NotMin

Answer(aID, highest, lowest) := Max ⋈ Min

Q2)

-- all members for each group for each assignment
AssignmentGroups(aID, gID, userName):= Π aID, gID, userName (Required⋈Members⋈Group)

--a version of Submission that's better suited for our task
BetterSubmission(userName, gID, when) := Π userName, gID, when (Submissions)

--submission times for all tuples in AssignmentGroups
GroupSubTimes(aID, gID, userName, when) := BetterSubmission ⋈ AssignmentGroups

--all submissions that were not the latest in a group
NotLatest(aID, gID, userName, when) :=
Π T1.aID, T1.gID, T1.userName, T1.when σ T1.aID = T2.aID and T1.gID = T2.gID and
T1.when < T2.when (ρT1 GroupSubTimes X ρT2 GroupSubTimes)

Answer(aID, gID, userName) := Π aID, gID, userName (GroupSubTimes - NotLatest)

Q3)

--all assignments in database
AllAssignments(aID) := Π aID (Assignments)

--all students in database
AllStudents(userName) := Π userName σ type = “student” (User)

-- the checklist for the condition A that the student has all grades
ShouldBeA(userName, aID) := AllStudents X AllAssignments

--All students as well as their results for assignments
AllStudentResults(userName, aID) := Π userName, aID (Member ⋈ Result ⋈ Group)

--students who did not receive all their marks
NotinA(userName) := Π userName (ShoulbeA - AllStudentResults)

A(userName) := AllStudents - NotinA

--All students as well as their marks, similar to AllStudentsResults but better for next task
AllStudentMarks(userName, marks) := Π userName, mark (Member ⋈ Result ⋈ Group)

-- students who received less than 70 on some assignment
NotinB(userName) := Π userName σ mark < 70 (AllStudentMarks)

B(userName) := AllStudents - NotinB

--all students and their grades for each assignments with due dates
AllStudentGrades(userName, aID, mark, due) :=
Π userName, aID, mark, due (Membership ⋈ Result ⋈ Group ⋈ Assignment)

--mark combinations of all students where assignment for mark1’s is due before assignment for mark2
GradeComparer(userName, mark1, mark2) :=
ρ(userName, mark1, mark2) Π T1.userName, T1.mark, T2.mark σ T1.userName=T2.userName and T1.due<T2.due (ρT1 AllStudentGrades X ρT2 AllStudentGrades)

--students whose marks went down
NotinC(userName) := Π userName σ mark1 > mark2 GradeComparer

C(userName) := AllStudents - NotinC

Answer(userName) := A ∩ B ∩ C

Q4) Cannot be expressed

Q5)
--all students in a group with their group id
AllGroupMembers(userName, gID) := Π userName, gID (Group ⋈ Membership)

--all students who were in a group of 3 as well as their group ids
GroupOf3(userName, gID) :=
Π T1.userName, T1.gID σ T1.gID = T2.gID = T3.gID and T1.userName ≠ T2.userName and T2.userName ≠ T3.userName and T1.userName ≠ T3.userName (ρT1 AllGroupMembers X ρT2 AllGroupMembers X ρT3 AllGroupMembers)

--all students who were in a group of 2 or more
TwoOrMore(userName, gID) :=
Π T1.userName, T1.gID σ T1.gID = T2.gID and T1.userName ≠ T2.userName (ρT1 AllGroupMembers X ρT2 AllGroupMembers)

GroupOf2(userName, gID) := TwoOrMore - Groupof3

GroupOf1(userName, gID) := AllGroupMembers - GroupOf3 - GroupOf2

--all students who were in a group of 3 with their marks in that group
3Group(userName, mark1) := Π userName, mark1 ρ(userName, gID, mark1) (GroupOf3 ⋈ Result)

--all students who were in a group of 2 with their marks in that group
2Group(userName, mark2) := Π userName, mark2 ρ(userName, gID, mark2) (GroupOf2 ⋈ Result)

--all students who were in a group of 1 with their marks in that group
1Group(userName, mark3) := Π userName, mark3 ρ(userName, gID, mark3) (GroupOf1 ⋈ Result)

Answer(userName, mark1, mark2, mark3) := 1Group ⋈ 2Group ⋈ 3Group

Q6)
--all students and the groups they are in
AllStudentGroups(userName, gID, aID) := group ⋈ students

-- all combinations of students who are in each others groups
GroupCommonStudents(gid1, aid1, userName1, gid2, aid2, userName2) :=
σ T1.gid = T2.gid and T1.username ≠ T2.username (ρT1 AllStudentGroups X ρT2 AllStudentGroups)
-- not finished type setting

Q7) Cannot be expressed

Q8) Cannot be expressed

CSC343 - A1 Part 2

Q1)
--a version of Grader that's better for this task
BetterGrader(gID, marker) := ρ(gID, marker) (Grader)

--graders for every student and group for each assignment
AllGroupGraders(userName, gID, aID, marker) := Membership ⋈ Group ⋈ BetterGrader

--all combinations of the markers(graders) of each student's groups
GroupGraderCombinations(marker1, marker2) :=
ρ(marker1, marker2) Π T1.marker, T2. marker σ T1.userName=T2.userName and
T1.gID≠T2.gID (ρT1 AllGroupGraders X ρT2 AllGroupGraders)

Σ marker1 = marker2 GroupGraderCombinations = Ø

Q2) Cannot be expressed

Q3)
--a version of Grader that's better for this task
BetterGrader(gID, marker) := ρ(gID, marker) (Grader)

--a version of Submission that's better for this task
BetterSubmission(gID, subtime) := ρ(gID, subtime) Π gID, when (Submission)

--a version of User that's better for this task
BetterUser(marker, type) := ρ(marker, type) Π username, type (User)

-- submissions for all students and all groups for each assignment as well as the grader for said assignment and the due date
AllStudentSubmissions(gID, due, subtime, mark, marker) :=
Π gID, due, subtime, mark, marker (Group ⋈ Assignment ⋈ Membership ⋈ Result ⋈
BetterSubmission ⋈ BetterUser ⋈ BetterGrader)

σ subtime > due and mark > 80 and type≠ "instructor" = Ø