Andrew Zurn
CSCI 331 – HW 3
Relational Languages
10/3/13

Relational Algebra

1) All_Course_aIDs ← π aID (Training_Course)
   All_DB_aIDs ← π aID (σ aTitle = 'Databases' (Technology_Area))
   All_DB_Course_aIDs ← All_Course_aIDs * All_DB_aIDs
   Result ← π cID, cTitle, cHours (All_DB_Course_aIDs **J** aID = aID (Training_Course))

2) All_Course_aIDs ← π aID (Training_Course)
   All_DB_aIDs ← π aID (σ aTitle = 'Databases' (Technology_Area))
   All_DB_Course_aIDs ← All_Course_aIDs * All_DB_aIDs
   All_DB_Course_cIDs ← π cID (All_DB_Course_aIDs **J** aID = aID (Training_Course))
   EIDs_Taken_DB_Course ← π eID (All_DB_Course_cIDs **J** cID = cID (Takes))
   Result ← π eID, eFirst, eLast, eTitle (EIDs_Taken_DB_Couse **J** eID = eID (Employee))

3) All_Course_aIDs ← π aID (Training_Course)
   All_DBNW_aIDs ← π aID (σ aTitle = 'Databases' OR aTitle = 'Networks' (Technology_Area))
   All_DBNW_Course_aIDs ← All_Course_aIDs * All_DBNW_aIDs
   All_DBNW_Course_cIDs ← π cID (All_DBNW_Course_aIDs **J** aID = aID (Training_Course))
   EIDs_Taken_DBNW_Course ← π eID (All_DBNW_Course_cIDs **J** cID = cID (Takes))
   Result ← π eID, eFirst, eLast, eTitle (EIDs_Taken_DBNW_Course **J** eID = eID (Employee))

4) All_DB_aIDs ← π aID (σ aTitle = 'Databases' (Technology_Area))
   All_NW_aIDs ← π aID (σ aTitle = 'Networks' (Technology_Area))
   All_DB_Course_cIDs ← π cID (All_DB_Course_aIDs **J** aID = aID (Training_Course))
   All_NW_Course_cIDs ← π cID (All_NW_Course_aIDs **J** aID = aID (Training_Course))
   EIDs_Taken_DB_Course ← π eID (All_DB_Course_cIDs **J** cID = cID (Takes))
   EIDs_Taken_NW_Course ← π eID (All_NW_Course_cIDs **J** cID = cID (Takes))
   EIDs_DB_No_NW_Course ← EIDs_Taken_DB_Course – EIDs_Taken_NW_Course
   Result ← π eID, eFirst, eLast, eTitle (EIDs_DB_No_NW_Course **J** eID = eID (Employee))

5) All_DB_aIDs ← π aID (σ aTitle = 'Databases' (Technology_Area))
   All_NW_aIDs ← π aID (σ aTitle = 'Networks' (Technology_Area))
   All_DB_Course_cIDs ← π cID (All_DB_Course_aIDs **J** aID = aID (Training_Course))
   All_NW_Course_cIDs ← π cID (All_NW_Course_aIDs **J** aID = aID (Training_Course))
   EIDs_Taken_DB_Course ← π eID (All_DB_Course_cIDs **J** cID = cID (Takes))
   EIDs_Taken_NW_Course ← π eID (All_NW_Course_cIDs **J** cID = cID (Takes))
   EIDs_DB_And_NW_Course ← EIDs_Taken_DB_Course ∩ EIDs_Taken_NW_Course
   Result ← π eID, eFirst, eLast, eTitle (EIDs_DB_And_NW_Course **J** eID = eID (Employee))

6) All_eIDs ← π eID (Employee)
   All_eIDs_Taken_Course ← π eID (Takes)
   All_eIDs_Not_Taken_Course ← All_eIDs – All_eIDs_Taken_Course
   Result ← π eID, eFirst, eLast, eTitle (All_eIDs_Not_Taken_Course **J** eID = eID (Employee))

7) Takes_Copy ← π eID (eID2), cID (cID2) (Takes)
   Takes_Cross_Takes_Copy ← Takes x Takes_Copy
   EIDs_Diff_Courses ← π eID (σ eID = eID2 AND cID <> cID2 (Takes_Cross_Takes_Copy))
   Result ← π eID, eFirst, eLast, eTitle (EIDs_Diff_Courses **J** eID = eID (Employee)

8) CIDs_With_Titles ← π cID, aTitle (Training_Course **J** areaID = aID (Technology_Area))
EIDs_Takes_Course ← π eID, aTitle (CIDs_With_Titles * Takes)
EIDs_Takes_Course_Copy ← π eID (eID2), aTitle (aTitle2) (EIDs_Takes_Course)
EIDs_Takes_Cross ← EIDs_Takes_Course x EIDs_Takes_Course_Copy
EIDs_1+_Area ← π eID (σ eID = eID2, aTitle <> aTitle2 (EIDs_Takes_Cross))
Result ← π eID, eFirst, eLast, eTitle (EIDs_1+_Area)

9) Lowest_Salary ← ƒ Min(Salary) (Employee)
Result ← Lowest_Salary **J** MIN_Salary = Salary (Employee)

10) All_Internet_aIDs ← π aID (σ aTitle = 'Internet' (Technology_Area))
All_Internet_cIDs ← π cID (All_Internet_aIDs **J** aID = aID (Training_Course))
EIDs_Takes_CIDs ← π eID, cID (Takes)
EIDs_Taken_All_Internet ← EIDs_Takes_CIDs ÷ All_Internet_cIDs
Resut ← π eID, eFirst, eLast, eTitle (EIDs_Taken_All_Internet **J** eID = eID (Employee))

11) All_EIDs ← π eID (Employee)
All_Internet_aIDs ← π aID (σ aTitle = 'Internet' (Technology_Area))
All_Internet_cIDs ← π cID (All_Internet_aIDs **J** aID = aID (Training_Course))
All_Combos ← All_EIDs x All_Internet_cIDs
EIDs_Takes_CIDs ← π eID, cID (Takes)
Diff ← All_Combos – EIDs_Takes_CIDs
Matches ← π eID (EIDs_Takes_CIDs – Diff)
Result ← π eID, eFirst, eLast, eTitle (Matches **J** eID = eID (Employee))

12) All_EIDs_All_CIDs ← π eID, cID (Takes)
EIDs_Join_Course ← π eID, areaID (All_EIDs_All_CIDs **J** cID = cID (Training_Course))
Tech_Area_Join_EIDs ← π eID, aID, aTitle (EIDs_Join_Course **J** areaID = aID (Technology_Area))
Count_EIDs_Per_Area ← (aTitle ƒ Count(eID) (Tech_Area_Join_EIDs)
COUNT_RESULT ← π aID (TechAreaID), aTitle (Title), COUNT_eID (Total_Enrollees) (Count_EIDs_Per_Area)

13) All_Leaders ← π eID, aID (Employee **J** eID = aLeadID (Technology_Area)
All_Course_Title ← π aID, aTitle, cID (Technology_Area **J** aID = areaID (Training_Course))
EIDs_Taken_Courses_Titles ← π eID, aTitle (Takes **LOJ** cID = cID (All_Course_Title))
Result ← π eFirst, eLast, aTitle (EIDs_Taken_Courses_Titles **J** eID = eID (Employee))

Relational Calculus

∃ ∀

1) { ***c***.cID, ***c***.cTitle, ***c.***cHours | **TRAINING_COURSE**(**c**) and (∃***a***)(**TRAINING_AREA**(**a**) and ***c***.areaID = ***a***.aID) }

2) { ***e***.eID, ***e***.eFirst, ***e***.eLast, ***e***.eTitle | **EMPLOYEE**(***e***) and (((∃***c***)(**TAKES**(***c***) and ***c***.eID = ***e***.eID*)* and ((∃***c***)**TRAINING_COURSE**(**c**) ***c***.areaID = ***a***.aID) and ((∃***a***)(**TRAINING_AREA**(**a**) and aTitle = 'Databases')) }

3) { ***e***.eID, ***e***.eFirst, ***e***.eLast, ***e***.eTitle | **EMPLOYEE**(***e***) and (((∃***c***)(**TAKES**(***c***) and ***c***.eID = ***e***.eID*)* and ((∃***c***)**TRAINING_COURSE**(**c**) ***c***.areaID = ***a***.aID) and ((∃***a***)(**TRAINING_AREA**(**a**) and ***a***.aTitle = 'Databases' or ***a***.aTitle = 'Networks')) }

4)

5)

6)
7)
8)
9)
10)
11)
12)