CS586/486 Introduction to Databases

Fall 2016 Quarter

Assignment 2 – DDL & DML; SQL & Relational Algebra – Revised

Due: Thursday, 13 October 2015, at the beginning of class

You may do this assignment individually or you may work with one partner. That is, this assignment is to be completed by individuals or by teams of two students. You should only talk to the instructor, the TA and your partner about this assignment. You may also post questions to the Piazza discussion list.

Please turn in your completed assignments on paper. Put your last name, first name, the assignment number in that order in the first line of your assignment. List last name and first name for your partner, if you have one, on the second line of your assignment. (If you are working with a partner, turn in one assignment paper.)

Part I

(36 points) For the following exercises, you will be creating, modifying and querying SQL tables. For each item, show the SQL you used and the resulting state (*all rows*) of your table or tables (or the error message that SQL returns). Do all these tasks using SQL statements (rather than the interactive interface). You will be using data on Pokemon Go characters as posted on Piazza (which is adapted from http://pokemondb.net/spinoff/go/pokedex).

(a) Create a table with columns for Name, Stamina, Attack, Defense and Candy, with Name as the primary key.

(b) Insert rows for all characters with capture rate less than 10%.

(c) What happens if you try to insert Venusaur a second time?

(d) Modify your table to add columns for Capture Rate and Flee Rate.

(e) Update the existing rows in the table to add Capture Rate and Flee Rate information.

(f) Insert rows for characters with capture rate equal to 10%.

(g) Write a query to find all characters with Stamina less than 100.

(h) Create a second table with Name and Type information, with Name as a foreign key to the first table.

(i) Insert rows in the second table corresponding to all characters in the first table. For characters with multiple types, each type should be listed separately.

(j) What happens if you try to insert Bug as a type for Caterpie?

(k) What happens if you try to delete the row in first table for Venusaur?

(l) Write a query to find the average stamina of all characters with type Bug.

Parts II and III of this assignment are based on the Spy relational database.

Part II

Relational Algebra
Note: you won't be able to execute these queries, since we don't have an implemented system that supports user queries written in relational algebra.

(15 points) Write the following queries in Relation Algebra, using ONLY the *select*, *project* and *cross product* operators:

(a) Find the city and country for all agents paid more than 200000.
(b) List first and last names of all agents who speak Chinese.
(c) List the agent_id, cith and country of all agents with an affiliation titled FBI.

Part III

Write a single SQL statement for each of the following queries. Show the first five rows of the result for each query (or fewer, if the result is smaller) and the number of rows returned. You should be able to write these SQL queries using only the features covered in the first two sets of lecture notes.

Note: If you are asked to write a query two ways, you only need to include the results and row count once.

(a) (5 points) Find the agent id and salary in Pounds for all agents whose city is London. Name the result columns London_ids and Pounds_pay. (Assume the salary column is US dollars.)

(b) (5 points) Find the number of different skills.

(c) (5 points) Find the high, low and average salary for all agents who speak French.

(d) (10 points) Find the team name for all teams with at least one agent who speaks Vietnamese.
Do this query two ways: Once using NATURAL JOIN and once without any JOIN operator in the FROM clause.

(e) (5 points) List the team names for each agent, *including agents on no teams*. The result should have last name, city, agent_id and team name.

(f) (10 points) List the team name for each team that has an agent with CIA affiliation and an agent with FBI affiliation. Do this query twice: Once using INTERSECT and once without using that operator.

(g) (10 points) Find all agents with the KGB affiliation who cannot speak Russian.