Assignment 3

Database Design Theory and Normalization

Due Date: December 1st at 11:59pm

Points: 100

Read the note on academic Integrity.

Academic Integrity: https://utsc.utoronto.ca/aacc/academic-integrity-matters

Instructions:

Note: Late submissions will be penalized 10% for each day up to December 4th 11:59 pm. Any submissions after December 4th will not be marked.

There are two parts in this assignment.

- Part 1 (3 questions – 40 points)
  - Requires you to design ER Diagram given specification using any online drawing tool.
  - Requires conversion from ER diagrams to relational schemas and improved version of ER diagrams.
  - Requires you to design an ER diagram given a relational schema.
  - **Note:** Handwritten solutions/diagrams will not be accepted.
- Part 2 (6 questions with subparts - 60 points)
  - Requires you to answer questions on functional dependencies and normal forms.

You can work in a group of up to 3. Please submit only one copy of your solutions per team.

Submission:

https://markus.utsc.utoronto.ca/cscc43f22/

For this assignment, you need to submit all typed solutions in a single PDF named **Assignment3.pdf**. You can use LaTeX or any other word processing software. ER Diagrams should be computer generated as well and ***should not be hand drawn***.

Part 1: Entity Relationship Model [40 marks]

Question 1 [25 marks]

For this question, you will design Entity Relationship Diagram and a database schema for an Ice Hockey Tournament.

System Functional Requirements:

The administrator of tournament website and the database is the primary user who manages the data associated with the tournament that includes information about players, teams, coaches, matches, referees, etc.

The administrator wants following features included in your design and describes these features as follows:

For each match, we store the series and the tournament day on which it takes place, which match it is, (e.g., first match, second match, etc.) the date with day, month, year, the teams involved in the match with the name of the city and the coach, and finally for each team whether that team played at home. Each team can have multiple coaches and for each coach, we record their name, salary, and city of birth.

We store the name and the surname of each player in each team with his date of birth and main position. We store, for each day, how many points each team has, and we also store for each match, the player of each team who played and in which position each player played (the position can change from one game to another). For each match, we store the referee, with first name, surname, city, and region of birth.

The match played as scheduled must be distinguished from those postponed. For a postponed match, we store the date in which it is actually played. We also identify the matches played in a city other than that of the home team; for each of these, we store the city in which it took place, as well as the reason for the variation venue. For each player, we are interested in the city of birth.

We also record the contracts between players and teams including the past contracts with the date of beginning and end of the contracts for each player in each team. It is possible that a player can have different contracts with the same team in different periods. For all contracts we wish to know the date of commencement. Similarly, we also record contracts between coach and teams.

Question 1.1 [15 marks]:

Design and draw an ER schema that captures the information given above. Your schema should model explicitly entities and relationships in the domain of Ice Hockey Tournament, also their attributes, generalization relationships, keys, and cardinality constraints.

Make necessary assumptions in order to complete your schema and state those assumptions along with the diagram.

You may use domain knowledge to complete information missing in the given specifications.

Question 1.2 [10 marks]:

Remove any generalizations (subclass), unnecessary data and structural redundancies, multi-valued attributes, optional relationships, and irregularities from your model.

If you did modify your ER schema, show the improved version of ER Diagram in this stage.

By removing weak entity sets (if any), translate your ER diagram into Relational schema following the `PostgreSQL` syntax.

Question 2 [15 marks]

We wish to carry out a reverse engineering operation. That is, given a relational database, we wish to construct its conceptual representation using the E-R model. The database is for an application concerning trains and railway stations and is made up of the following relations:

- STATION (<u>Code</u>, Name, City) with a referential constraint between the attribute City and the CITY relation;
- CITY (<u>Code</u>, Name, Region);
- ROUTE (<u>From</u>, <u>To</u>, Distance), with referential constraints between the attributes From and the relation STATION and between the attribute To and the relation STATION; this relation contains all and only the pairs of stations connected directly by a route (that is without intermediate stations);
- TRAINTIMETABLE (<u>Number</u>, From, To, DepartureTime, ArrivalTime) with referential constraints between the attributes From and the relation STATION and between the attribute To, and the relation STATION;
- TRAINROUTE (<u>TrainNumber</u>, <u>From</u>, <u>To</u>) with referential constraints between the attribute TrainNumber and the relation TRAINTIMETABLE and between the attributes From and To and the relation STATION;

- STOPTIME (TrainNumber, Station, Arrival, Departure) with referential constraints between the attribute TrainNumber and the relation TRAINTIMETABLE and between the attribute Station and the relation STATION;
- ACTUALTRAIN (TrainNumber, Date, DepartureTime, ArrivalTime) with a referential constraint between the attribute TrainNumber and the TRAINTIMETABLE relation;
- ACTUALSTOP (TrainNumber, Date, Station, Arrival, Departure) with a referential constraint between the two attributes TrainNumber and Station and the STOPTIME relation.

Question 2.1 [15 marks]

For the above schema, construct its conceptual representation using the E-R model.

Recall that derived redundancy can be present in terms of attribute, entity, or relationship.

Find out any such *derived redundancy* in this model.

Part 2: Functional Dependencies and Normalization [60 marks]

Question 1 [5 marks] Consider a relation S with six attributes A, B, C, D, E, and F. You are given the following dependencies: $AB \rightarrow C$, $BC \rightarrow D$, $D \rightarrow E$, $CF \rightarrow B$.

a) What are all the non-trivial functional dependencies that follow from the given functional dependencies? Make sure that the functional dependencies you list have exactly one attribute on the right hand side.

b) What are all the candidate keys of S?

c) What are all the superkeys of S that are not candidate keys?

Question 2 [5 marks] Consider a relation R with five attributes A, B, C, D, and E. You are given the following dependencies: $A \rightarrow B$, $BC \rightarrow E$, and $ED \rightarrow A$.

a) List all the candidate keys for R.

b) Is R in 3NF? Explain why or why not.

c) Is R in BCNF? Explain why or why not.

Question 3 [15 marks] For all of the parts below, assume you are given a relation R with four attributes A, B, C, and D. In each part you are also given a set of functional dependencies, assume those are the only dependencies that hold for R and do the following:

(i) Identify the candidate key(s) for R.

(ii) Identify the best normal form that R satisfies (1NF, 2NF, 3NF, or BCNF).

(iii) If R is not in BCNF, give a lossless-join decomposition of R into BCNF.

a) $C \rightarrow D$, $C \rightarrow A$, $B \rightarrow C$

b) $B \rightarrow C$, $D \rightarrow A$

c) $AB \rightarrow C$, $AB \rightarrow D$, $C \rightarrow A$, $D \rightarrow B$

Question 4 [15 marks] In each part below you are given a relation with attributes as well as a set of functional dependencies that hold for that relation. For each part, do the following:

(i) List all the candidate keys for the relation.

(ii) Give a lossless-join and dependency-preserving decomposition of the relation into 3NF.

a) R(A, B, C, D, E); $\{A \rightarrow BC, CD \rightarrow E, B \rightarrow D, E \rightarrow A\}$

b) S(A, B, C, D, E, F); $\{C \rightarrow A, CD \rightarrow E, A \rightarrow B, D \rightarrow F\}$

c) $T(A, B, C, D, E)$; $\{ A \rightarrow CD, B \rightarrow CE, E \rightarrow B\}$

Question 5 [10 marks] For each of the following relation schemas and dependencies do the following:

(i) Find all the 4NF violations.

(ii) Decompose the relations into a collection of relation schemas in 4NF.

a) $R(A, B, C, D)$ with MVD's $A \twoheadrightarrow B$ and $B \twoheadrightarrow CD$.

b) $R(A, B, C, D, E)$ with MVD's $A \twoheadrightarrow B$ and $AB \twoheadrightarrow C$ and FD's $A \rightarrow D$ and $AB \rightarrow E$.

Question 6 [10 marks] Give counterexample relations to show why the following rules for MVD's do not hold. Hint: apply the chase test and see what happens.

a) If $A \twoheadrightarrow BC$, then $A \twoheadrightarrow B$.

b) If $A \twoheadrightarrow B$, then $A \rightarrow B$.