CS2102: Database System

P01: ER Diagram AY2023/24 Sem 2: Week 05

Objective

The objective of this team project is for you to apply what you have learned in class to design and develop a database application using PostgreSQL. The project is to be done in teams of four students. The project consists of the following four tasks:

(P01) Design an ER data model for the application. Your ER model should capture as many of the application's requirements as possible.

(P02) Translate your ER data model into a relational database schema. Your relational schema should capture as many of the application's constraints as possible.

(P03) Implement an SQL or PL/pgSQL functions/procedures for each of the functionalities listed in Application Functionalities.

(P03) Implement triggers as required by the specification.

Deliverables

Each team is to upload a **pdf** file named "`teamNNN.pdf`" where "`NNN`" is the three digit team number according to your project group number. You should add leading zeroes to your team number (*e.g.*, `team005.pdf`). Submit your **pdf** file on Canvas assignment named "Project 01" (`https://canvas.nus.edu.sg/courses/52825/assignments/93146`). Only the **latest** submission that is submitted **before the deadline** will be marked.

As a general guideline, your ER diagram should:

- not capture constraints not specified in the application's constraints.
  - Any constraints that can be captured but did not may be penalized.
- capture as many of the application's constraints as possible as per the specification.
  - Any constraint specified in the specification but is not captured or incorrectly added may be penalized.

The submitted **pdf** file must be at most **6 pages** with font size of **at least 12 points** and consists of the following:

- The project team number and names of team members on the first page.
- The constraints required from the specification.
  - The constraints should be categorized by the elements in the ER diagram (*e.g.*, entity or relationship sets).
  - Indicates the constraints that are not captured by the ER diagram clearly.
- ER data model of your application.
  - You may *optionally* use the ER diagram tools from https://thisisadi.yoga/cs2102/ERD/.
  - If your ER diagram is too large (*e.g.*, spanning more than one page), you may want to include a single-page simplified ER diagram (*i.e.*, with non-key attributes omitted) before presenting the detailed ER diagram.
  - Your ER diagram cannot be too small and it must be readable when printed.
    * The font of the ER diagram must be readable.
    * Unreadable texts on ER diagram may be omitted from grading even if they contain correct elements.
    * You may rearrange your ER diagram manually in the ER diagram tools to produce a more vertically oriented ER diagram.
- Justification for any non-trivial design decisions made.
  - This includes *but not limited to* the use of ISA hierarchies, aggregates, and any ternary relationship set.

Deadlines

The deadline for the submission is **12:00** of **Saturday** of Week 05 (*i.e.*, **17 February 2024**). Only submissions through Canvas will be accepted. Submissions through other means (*e.g.*, emails) will not be accepted.

Late Submissions

2 marks (*out of 6*) will be deducted for submissions up to **1 day late**. Submissions late for more than 1 day will receive **0 mark** and will not be graded.

Templates

ER Diagram

You are given the file ERD.pptx containing shapes that can be used to create your ER diagram. It also has some guides on good practices related to ER diagram. You may choose to use this to create your ER diagram.

Report

You are given the file P01.docx containing a template for your report with an initial constraint and ER diagram provided. You may choose to continue from the given template or rewrite the constraint and/or ER diagram.

You are encouraged to write detailed constraints with the following suggested phrasing:

- **Lower Bound**
  - "A student may work on 0 projects"
  - "A student must work on at least 1 projects"
- **Upper Bound**
  - "A student may work on more than 1 projects"
  - "A student may work on at most 1 projects"
- **Condition on Bound**
  - "A student *working on a project* may use 0 GPUs"
  - "A student *working on a project* may use more than 1 GPUs"
- **Non-Trivial Design**
  - "A student must be exactly one the following: undergraduate or graduate"
  - "Works is implemented as an aggregate because [... *justification* ...]"
  - "Match is implemented as a ternary relationship involving two entities from clubs and one entity from stadium because [... *justification* ...]"
    - You must provide justification

You should also clearly indicate which constraints are enforced and which are not enforced by the ER diagram. The template file P01.docx contains the suggested way to arrange the constraints and how to clearly specify which are enforced and which are not. If you choose not to follow the template, you must ensure that your format clearly shows this information.

Grading

The following grading scheme will be used. The details of the grading scheme is hidden.

Validity of ER Diagram (0 Mark)

- The ER diagram should not contain invalid constructs.
- The ER diagram should be readable when printed on an A4 paper.
- While no mark is allocated for this, note that the use of invalid constructs and/or unreadable ER diagram may cause deductions in other areas as the meaning is unknown.
  - We will look only at the ER diagram when determining data and constraint requirements.

Data Requirements (1 Mark)

- The ER diagram should capture all the data (*e.g.*, attributes) needed by the application's requirement.

Constraint Requirements (3 Marks)

- The ER diagram should capture as many of the application's requirements as possible.
- The ER diagram should not add constraints not specified in the application's requirements whenever possible.

Analysis (2 Marks)

- The report should contain a thorough assessment of the constraints requirements.
- The report should correctly mention the constraints not captured.