CS 122A: Introduction to Data Management – Fall 2021

Homework 2: E/R and Relational Translation (100 points)

Due Date: **Wednesday, Oct 13 (06:00 PM)**

Submission

All HW assignments must be submitted online via the HW2 in **Gradescope**. See the table below for the HW 2 submission opportunities. Note that after 6 PM on Thursday, October 14th, no further HW 2 submissions will be accepted. (We will be releasing the solution at that time.) Please strive to get all your work in on time! If possible, try to save the one dropped assignment for the end of the term when you are most likely to want/need it.

Date / Time	**Grade Implications**
Wednesday, Oct 13 (06:00 PM)	Full credit will be available
Thursday, Oct 14 (06:00 PM)	10 points will be deducted

E-R based Relational Schema Design [100 pts]

You have successfully designed an E-R diagram for SWOOSH. Now, to set up a real database, you need to design a relational schema that can represent the objects described in the E-R diagram that you've created. **Specifically, you will need to design an appropriate collection of tables and create a SQL DDL statement for each table.** To make sure your design is a good one, you have given your E-R diagram to an external database consulting firm, DBInstructor, Inc., to have their experts check the correctness of your design and refine it if necessary. DBInstructor will provide you with the final E-R diagram soon. Since time is of the essence, you may want to start designing the required relational schema now based on your original E-R diagram. Your final design must be based on the E-R schema that DBInstructor provides, however. That schema will appear shortly after Thursday's 06:00 PM HW 1 "drop dead" deadline has passed. (You should get PostgreSQL installed and running in the meantime!)

As you work on your design, if you see opportunities to avoid creating excess relations that stem from relationships, do indeed avoid them so as to make the schema simpler. Clearly list all of your relations and their attributes (including their types), primary keys, foreign keys (including the referenced target relations), and any *not null* constraints by writing a SQL DDL statement for each table. The resulting design should capture the information and constraints of the E-R diagram as faithfully as possible, and you should verify that it runs on PostgreSQL. For types, choose one of the following types for each column:

Category	Type	Remark
NUMBER	integer	A number type for integer values.
	decimal(x,y)	A number type for real values where **x** is the maximum number of digits and **y** is the number of digits to the right of the decimal point.
STRING	text	A variable-length string type.
DATETIME	date	A type used for values with a date part but no time part. The format is '0000-00-00'.
	time	A type used for values with a time part. The format is '00:00:00'.
	timestamp	A type used for values with both a date part and a time part. The format is '0000-00-00 00:00:00'.
ENUMERATION STRING	enum	A string type with the value being restricted to a list of permitted values. (Use sparingly.)
JSON	json	A JSON type used to store a JSON document as a value.
BOOLEAN	boolean	A type for true or false values.

Make use of the entity, relationship, and attribute names from your final E-R (DBInstructor-provided) diagram when naming your tables and columns (to make it crystal-clear how your design corresponds to the E-R diagram). Note that some of your relationship or entity names might be **RESERVED** words (e.g. User). To get around this issue in PostgreSQL, you can use double quotes around the name of the table to create and query it, e.g. cs122a_hw."User", or you may rename the table to a similar non-reserved word such as "Users".

Again, all of the following information should be included in the DDL statements that you create for each table. Be sure to:

(a) [50pts] List the tables, columns, and column types in your design.

(b) [20pts] For each of your tables, identify its primary key column(s).

(c) [30pts] For each of your tables, identify its foreign key column(s) (and indicate which other table each one references) as well as any *not null* constraints and/or other integrity constraints (such as *unique* constraints).

Notes:

1) IsA relationships should be modeled using the Delta tables approach.
2) When an instance of a parent entity type in an IsA hierarchy is deleted, any/all associated child entity information associated with the instance should also be removed.
3) A Recurrence's day of repetition can be any of Mon, Tue, Wed, Thu, Fri, Sat, Sun.

Here is an example of some DDL statements for a few tables (which you can try running on your own PostgreSQL instance if you want to get your feet wet that way):

```
CREATE SCHEMA IF NOT EXISTS cs122a_hw;
DROP TABLE IF EXISTS cs122a_hw.Company;
DROP TABLE IF EXISTS cs122a_hw.Car;


CREATE TABLE cs122a_hw.Company (
      name        text NOT NULL,
      PRIMARY KEY (name)
);

CREATE TYPE car_type AS ENUM('sedan', 'hatchback', 'convertible', 'suv',
'pickup', 'van');

CREATE TABLE cs122a_hw.Car (
      VIN         integer,
      make        text,
      year        date NOT NULL,
      type        car_type NOT NULL,
      PRIMARY KEY (VIN),
      FOREIGN KEY (make) REFERENCES cs122a_hw.Company(name) ON DELETE
CASCADE
);
```

The best way to make sure that your solution is sound is to actually try it out on a real system – so go ahead and install **PostgreSQL** at this time on your favorite HW platform (presumably your laptop) and use it to verify that your PDF submission as a SQL file would actually run properly. (In other words, do **not** turn in your homework with DDL statements that you haven't actually tested first!)

IMPORTANT NOTE ON SUBMISSION LAYOUT:

Be sure to use the HW#2 Google Docs template as the basis for your submission, as its use is **mandatory**. The deliverable should first list the SQL DDL statements for creating tables for *entities*, including any supporting tables for *entity-related* information. It should then list **(on a new page please)** the additional SQL DDL statements to create any additional tables for *relationships*. (Please, please, please organize your DDL statements in this way!) The current template has one page for entities and their supporting tables and one page for the additional relationships' DDLs. Feel free to append more pages *at the end of each part* if you need more space than that for a part. Don't forget to download the template as PDF once you're done!