

COSC 344

Assignment 2

Overview

In this assignment, you will turn your ER diagram into a set of Oracle tables, normalize your design, and populate your database.

Due date for assignment 2

This assignment is worth 15% of your final mark. It is due at **4 PM on August 21 (Friday)**

Description

1. Relational Schema

Revise your ER diagram based on the feedback from Assignment 1, and then use the 7-step algorithm introduced in Lecture 4 to convert the revised ER diagram to a relation schema. The result should be similar to the diagram discussed in Lecture 4. Underline the primary key, and show all attributes (no need to include the derived attributes). For referential integrity constraints, draw an arrow from the foreign key terminating on the primary key it refers to.

2. Normalization

Turn your schema diagram into a set of tables in BCNF. You should go to 1NF, 2NF, 3NF, then BCNF. It might happen that a relation may be already in a given normal form (e.g 3NF). In such a case, you just explain why it is in that normal form. Explain your steps in a manner similar to the example given in the attachment. If you cannot get to BCNF, explain why. Your write-up should be reasonable to follow and include no jumps in logic – it's possible that your BCNF schema may be a 'good representation' of your miniworld, but 'not be correct' based on your initial ERD.

3. Create Database

Create an SQL script that will create your tables in their BCNF form. The CREATE TABLE commands should be as complete as possible. This means where possible they should:

1. specify domains if some subset of normal data types is appropriate
2. specify the primary key
3. specify uniqueness, not null, and/or a default where appropriate
4. referential integrity constraints where appropriate

Depending upon your constraints, the order of table creation may be important.

Put a series of DROP TABLE commands ahead of the CREATE TABLE commands in your script so that the database can be reloaded without any error. Again the order may matter.

Populate your database with a series of INSERT commands. These should follow the CREATE TABLE commands. If at all possible, the INSERT commands for a given table should be grouped together and followed by a COMMIT.

The script described above must be named `load.sql` and is a deliverable. It should execute without errors. The allowed exception is a DROP TABLE error when the table does not exist.

Teamwork Model

The team leader coordinates the allocation of the tasks, and makes sure the task allocation is fair to every team member. For example, each member should map at least one entity type and one relationship type. Each member should normalize at least one relation and create at least one table.

At the end of the report there should be a summary of the teamwork, showing which part was done by which member, which parts have been discussed by all group members, and whether each member has contributed equally. You can add any comment on the teamwork in your group.

Assignment submission

The assignment **MUST** be submitted electronically. Include the following into your `cosc344_asgn2` folder:

1. A single document (prefer to be in Word format) that includes the following:
 - Revised ERD (if there is no change, just put the one in your 1st assignment), and explain what has been changed.
 - Relational schema diagram to show integrity constraints (refer to the example given in slide 23 of Lecture 3)
 - Normalization description
 - `load.sql`
2. the `load.sql` file

The team leader (or any team member) creates a folder named `asgn2_groupX`, where X is the group number. Make sure this folder contains all the files that need be submitted.

Change into the directory that **contains** this folder, and submit your assignment using the COSC344 submit script as follows:

```
$ submit344 asgn2_groupX
```

The `submit344` script is put at `/home/cshome/coursework/bin/`. If the above command does not work, please give the full pathname for the script, that is,
`$ /home/cshome/coursework/bin/submit344 asgn2_groupX`

The script displays its progress in submitting the assignment. You can check if your submission is successful from the output of the submission script. The lines after "File copied are : -" show what folder and file have been submitted. If you get an error saying "sh: /usr/bin/procmail: no such file or directory", just ignore it as it is related to the linking to your email account.

You can resubmit before the due date if you wish -- your last submission is the one that will be marked.