Runder

Mike, Kyle, & Scott

CMPT 307:  Assignment 8

Your Database Design: the Conceptual and Logical Models

The goals of this assignment are:

* To do the data "discovery" phase of your project.
* To apply this process to create an ER/EER Diagram - your "conceptual model".
* To write a set of Relation Schemas from your ER Diagram
* To carefully review the structure of your data model in preparation of implementation

**When done, submit the following:**

* **List of team members**
* **Name desired for you database**
* **Statement of purpose of the project and its tasks (what it can accomplish)**
* **a list of the business rules you think apply to your data**
* **an ER diagram as a jpg file here (exported from MySQLWorkbench)**
* **a final list of your normalized schemas**

**You can submit one zip file for your group.**

GENERAL INSTRUCTIONS:

        This lab will be done with your database group.  (I know some are just individuals...)

**The lab is four parts:**

In part 1 you are asked to conduct the data discovery and analysis phase and identify the Entities and Attributes you will need

In part 2 you will construct an ER diagram to complete a Conceptual Model for your project.

In part 3 you will map your conceptual data model expressed in your ER/EER diagram into a logical data model.  This will be refined, and will be your database definition.

Finally, in part 4, you will normalize the schemas you created.

(Software packages can generate the database from the EER diagram now, but traditionally, the logical model was the next step to write the SQL to build a database; and it still serves as a useful step for applying Normalization.  These schemas will be used for Normalization in a later assignment.)

**Part 1: The Conceptual Data Model**

Data Discovery – three things to create:

1. Statement of purpose and list of tasks  (you may have this from your proposal, but now is a good time to review this)

**Create an easy way to get info about ski runs**

2. List of Business Rules

All chairlifts must have a run

Each run must have only one difficulty

All runs must have an elevation

All runs must have a chairlift

All Chairlifts must have a name

Multiple runs can have the same chairlift

3. List Entities and their Attributes.

Run - Length, name, Elevation, Chairlift, Food available?, Medical?

Difficult - Green Circle, Blue Square, Black Diamond, Double black Diamond

Chairlifts?

Details:

1. If your project will be used by other (known) people, you should interview them and gather data related documents in use.   Discover what they need.

**N/A**

You can flesh out some of this material, but don’t neglect to meet with or talk with them.

The results of  this process should be a **list of business rules** and a  **list of subjects and their characteristics** (these are obviously the entities and attributes you'll need)

All of these lists will be useful as you model data and also when you write your project report and present the material to the class.

2.  From your Initial Subject List and Characteristics.  This list should be as complete as you can make it.  You will edit it to remove duplicates and create names that are descriptive. Review if any characteristics should be entities and visa versa.

This is your initial list of **Entities and their Attributes.**

Based on this Discovery process you can now create a

**statement of purpose and a list of required tasks for the database.  Submit this too.**

**Part 2: Conceptual Data Model**

You will use mysqlworkbench to generate an ER/EER diagram for your group database.

**Establish Relationships between all the Entities on your list**

You need this to create your ER diagram.

**An organized approach is described here:**

A. Identify Relationships between Tables

  Work with one table at a time.  For that table, consider if any direct relationship exists with another table.  Systematically proceed to pair it up with every other table in the model.  To find direct relationships between two tables ask these types of questions:

Ask "Can a single record in the first table may be <associated with> **one or more** records in the second table?"  These are associative relationships.

Also ask if a subject in the first table <owns, has, is part of, contains, etc> **one or more** subjects in the second table.  These are ownership oriented relationships.

  Ask if a subject in the first table <makes, visits, teaches, pays, etc> **one or more** subjects in the second table.  These are Action-oriented relationships.

  When a potential relationship is not clear try making up sample data for the tables as this can often bring more clarity.

  Before continuing, reverse the two tables and ask the questions again.

  Proceed through all the possible combinations of tables to find all the direct relationships in your model.

  The answers to the questions will tell you IF a relationship exists, and if so, what TYPE it is:

one-to-one, one-to-many, or many-to-many.  Record each relationship for the next step.

B. Define the Table-Connection for each Relationship

  Based on the answers from step A, proceed through each relationship to establish the attributes and style that will connect the tables involved.

one-to-one:  Decide which table has the dominant role; this is the main table. (For instance Managers manage Departments, so Managers dominates.) The other is the subordinate table. If neither seems subordinate in the relationship, then arbitrarily assign one this role.

Place a copy of the primary key from the main table in the subordinate table.  That attribute is now a foreign key in the subordinate table.

one-to-many: Place a copy of the primary key of the "one" table into the "many" table.  Again this is now a foreign key in the many table.

many-to-many: The easiest solution is to create a new "linking" table between the two tables.  We will discuss in class the problems that lead to this solution.  The linking table consists of the primary keys from both tables.  These will form the composite key of the linking table.  Next consider all the attributes from each table, and decide if any fit better as members of the linking table.

C. Set the Characteristics for each Relationship

  The "characteristics" of a relationship are the degree and type of participation and what happens when a record of one table is deleted.

Type of Participation: Does a record HAVE to exist in one table before a record may be entered in the other table?  If so, this is full participation.  If not it is optional participation. For instance every record in ORDER must be associated with a record in CUSTOMER (full participation), but not every CUSTOMER record must have an active ORDER (optional).  A symbol showing this should be placed on each side of the relationship in your ER diagram.

Degree of Participation: You have determined this in step A.  How many records in this table may be associated with a single record on the other table?  It will be from 0 through either an upper limit or without a limit.  It is often shown on ER diagrams as (min, max).  (1, 1) is a one-to-one, (0, N) is unrestricted.

Deletion:  We must be concerned when a record in a table that is on the "one" side of a relationship is deleted, because it may leave "orphaned" data in the other table.  You may choose to Restrict deletion so that records may not be deleted if related records exist in the subordinate table. (Related records must be deleted first.)  Or you may choose to Cascade, which means the related records are deleted automatically.  Restrict is the normal choice.

  Ex: Ask "If a record in the CUSTOMER table is deleted, should related records in the "ORDER" table be deleted too?" If not, choose Restrict.  If participation is optional for both tables, you may choose neither.

**2. Create an ER Diagram for the group database.**

  The work in this step should be divided up in your group as fairly as possible.  Since we are small groups, you may just want to work together.  The completed diagram should be an organized and clear document that you can present to the organization for discussion and suggestions.

  This ER diagram should show all the tables, each will list all its attributes, it will show all the relationships, and should indicate the participation constraints of each table in it.  Try to take care with foreign keys and relationships.  This document will be used to generate your actual database.  You may have to go back and change some of what mysqlworkbench does automatically.

**Turn in your ER Diagram on the course web site (use jpg format for the submission).**

**Part 3: The Logical Model**

This part is fairly easy once you have your model.

Translate your ER Model into a **Relational Data Model** consisting of a collection of relational schemas.  The essential steps are described in in chapter 9.6 and are worth reading through.  The result will be a set of relational schemas that describe the database.  These may also be converted directly into your database.  An example of a relational schema follows:

**EMPLOYEE{SSN, FName, MInt, LName, BirthDate, EmpAddress, EmpCity, EmpState, EmpZip, JobType}**

These schemas give you a chance to **review** your database!

A careful review of your model ensures that your database structure is sound.

Check all Table and Field names:

* is each descriptive and meaningful
* is each unambiguous (e.g., PhoneNumber is unclear - home?, cell?, work?)
* is each a unique name (only where it is a foreign key should it be repeated in another table; even then it may be different)
* avoid acronyms and abbreviations

Look for the following potential problems:

Problematic Fields:  for each attribute or field be sure that:

* it represents a characteristic of the subject of the table
* it contains just a single value (multivalued fields imply a many-to-many relationship and should be broken out into separate linking tables)
* it cannot be broken down into parts (such as an address)
* it cannot be calculated (calculated values are often wrong because the fields they depend on may have been changed)
* it is unique within the database

Problematic Entities (relations/tables):  check that every entity has these qualities:

* it represents a single subject
* it has a primary key
* it has no unnecessary duplicate fields, i.e., minimize redundant data (from other tables)
* foreign keys make sense and you don't have more than you need

Review each primary key:

* does it uniquely identify each record
* will it always contain unique values
* can it be optional for some records - if so, it could be unknown and thus not a key
* can the value ever change - if so, it’s a bad key

Consider the relationships between your entities carefully.  This is often an error prone part of the database.

Check your data model for problems with extra data, missing entities, update issues, missing relationships or logic problems.

**Part 4.  Normalization of Your Schemas (to BCNF)**

Work with your logical model from Part 3 to normalize your database schemas.

Please explain any issues you encountered.  If you choose to denormalize any parts of the database explain why.

Submit a list of your final schemas with keys identified.  These should be in BCNF (unless otherwise explained).

a.  Determine the functional dependencies for every relational schema in your logical model by creating a functional dependency diagram for each schema.  Explain any assumptions that are needed to justify any dependencies that are not obvious.

b.  Every schema should be normalized to Boyce-Codd Normal Form, unless you have specific reasons to normalize to a lower level.

c.  Check if you have any multivalued (4NF) dependencies.  Decompose your relations to fix these as needed.

Submit your final schemas.  It is quite possible you did not need to make any changes to your original schemas.