

Homework 4
1.264, Fall 2006
Data model for chemical manufacturer
Due: Thursday, October 12, 2004

A. Overview

In this homework, you will build the data model that will support your Web application. The dataset is on the servers—see the MIT Server for details. Refer to problem set 2 and your requirements for data definitions and a description of the features that must be supported by your data.

B. Data model

The assignment is to build the data model for the chemical manufacturer. A recommended set of steps is given, although you are free to approach this as you wish. You must use Visual Paradigm; select the Entity Relationship Diagram.

You are allowed to make the following assumptions:

- The international or UN number can be used as a primary or foreign key in all entities in which it appears. All chemical data has UN numbers.
 - All UN numbers that are in the Isolation and Emergency Response data are in the Chemical Product data, so the Product entity can be the parent entity in these relationships.
1. Define the Chemical Product entity. Use international (UN) number as the primary key. This defines a unique chemical product at the manufacturer level.
 2. Create a separate ChemicalName entity to hold the multiple names associated with each UN number. Don't keep any chemical names in the Chemical Product entity.
 - a. Build the relationship from ChemicalName to ChemicalProduct. In this entire homework, build the relationships right after building each new entity. Visual Paradigm will help create the foreign/primary key.
 3. Create a domain (validation) entity to store the valid values of the chemical class data (1 through 9, with decimal subdivisions). Chemical class should be a foreign key in the Chemical Product entity you define, which Visual Paradigm will create automatically for you when you create the relationship.
 4. Place the quantity limits in a separate QuantityLimit entity, by passenger and cargo vehicle. These should not be in the Chemical Product entity. The separate entity should hold the UN number, vehicle type (cargo or passenger) and quantity limit.
 - a. You will need to create a vehicle type domain entity before building the QuantityLimit entity.
 5. Define an EmergencyResponse entity based on the Emergency Response dataset.

- a. You do not need to keep the multiple chemical names in the Emergency Response entity. Use the UN number as the primary key. (The UN number can be used to get the list of the multiple names derived from the Product entity.)
 - b. Decide whether the ISOLATION attribute is necessary or not.
 - c. A chemical should have zero or one emergency response (guide number).

6. Define the Emergency Detail entity.
 - a. Decide the primary key and relationship to the previous emergency response entity.

7. Define an entity based on Isolation.
 - a. You do not need the multiple chemical names; use the UN number as the primary key. The isolation data is the same for all rows with the same UN number.

8. Define the chemical classes, groups and their relationships:
 - a. You have already defined a Chemical Class validation entity in step 3 above. It will hold data on all the classes (1-9) found in the database, including decimal numbers such as 1.1.
 - b. Define a Chemical Group validation entity, which will hold just the values A, B, C, D and E, corresponding to the groups in the customer and carrier datasets.
 - c. Define the mapping between Chemical Classes and Chemical Groups, as defined in the carrier and customer descriptions in homework 2. It will need to hold the following data:
 - i. Group A contains class 7
 - ii. Group B contains class 1
 - iii. Group C contains classes 2, 3 and 8
 - iv. Group D contains classes 4, 5 and 6
 - v. Group E contains class 9
 - d. Remember that your data model doesn't hold any data; it only has entities and relationships.

9. Define the carrier entity. Use the certificate number as the unique identifier. Include all the carrier attributes except the modes it operates and the groups it handles, which are more complex relationships.
 - a. You may include city, state and zip code all within this table, even though it violates third normal form (or Boyce-Codd normal form). Technically, you should have only zip code in this table, and then a ZipCode table with the city and state based on the zip code.

10. Define the customer entity in the same way as the carrier.

11. Define a Modes domain entity. This will hold the 4 modes (highway, rail, water, air), one row per mode, in the actual database you build in the next homework.
 - a. Then relate the carriers to the modes that they can offer. This is a many-many relationship, so you'll need an associative table.
 - b. Do the same for the modes that the customers can use.

12. Define entities and relationships between the carriers and the chemical groups they transport. Do the same for the customers and the chemical groups they can receive. These are also many-many relationships that require associative tables.
13. Define an entity to hold the manufacturer orders. This contains data specific to the entire order. See homework 2 for what the order contains.
 - a. Don't call this entity 'Order', since ORDER is a database (SQL) keyword and will cause confusion.
 - b. Treat purchase order versus credit card as simply as you wish. It is ok in this homework, for simplicity, to just have a single attribute that hold the credit card or purchase order number, since we're not modeling payment in detail.
14. Define an entity to hold the order detail for each order. (There can be many items ordered in a single order.) See homework 2.
 - a. This contains line item details for the order: it stores the individual chemical products and the related data (quantity, price) in the order. You should store the price of each product, since prices may change and the price in the product catalog at a later time may not be the price in effect when the order was placed (even though we're not handling price changes in detail in this homework).
15. If you have additional requirements from your homework 2 solutions, include them.
16. **For all entities: define the primary key, foreign keys (if any), all attributes and all relationships. You do not have to choose a data type for each attribute (integer, etc.) You do not have to worry about which attributes can be null or must be unique (other than primary keys).**

You should have about 20 entities as your final result. Your model should be in fifth-normal form; if you follow the suggestions in the homework, it will be, with the exception of the state, city and zip code data. (Purists might debate large spill vs. small spill and a few other items!)

C. Assignment

1. Hand in the data model electronically .
 - a. You must use the Visual Paradigm drawing tool.
 - b. Please make your model fit on one page if possible, no more than two pages otherwise.
 - c. Please lay out your model so that it is clear: avoid relationship lines crossing each other, place related entities near each other, etc., as much as possible.
2. The data model must include all entities, keys (primary and foreign), attributes, and relationships (including their types, such as one-to-many). Indicate keys explicitly for each entity. Include domain relations (validation entities). You do not need to be careful about null, unique or data type for attributes.
3. Hand in the UML diagrams electronically .