Example Case to 3NF Tables

Purpose

The COIS20025 textbook and assignments have numerous exercises that require you to take a given case and generate a data design for the case. The data design will typically consist of an ERD, 3NF table designs including keys (obviously) and sample data.

The purpose of this document is to give an example of how to do this. It walks you through the process of completing this task.

Preparation

This document draws on some existing resources including:

- A PDF document titled "Developing Entity Relationship Diagrams (ERDs)"
- Powerpoint slides and accompanying MP3 audio of a lecture explaining normalisation.

Both these documents are available from the Assignment 2, Part A Resources page. For T2, 2006 the URL for this page is

> http://webfuse.cqu.edu.au/Courses/2006/T2/COIS20025/Assessment/Item_2/Part_A_Resources/

It is important that you become familiar with these resources before going any further.

The Case

This document will use the same case as used in the "Developing Entity Relationship Diagrams (ERDs)" document. It is included below.

> A company has several departments. Each department has a supervisor and at least one employee. Employees must be assigned to at least one, but possibly more departments. At least one employee is assigned to a project, but an employee may be on vacation and not assigned to any projects. The important data fields are the names of the departments, projects, supervisors and employees, as well as the supervisor and employee number and a unique project number.

You are required to:

1. Create an ERD with cardinality notation.
2. Create 3NF table designs.
3. For each of the entities identified, design tables and identify the possible candidate keys, the primary key, a probable foreign key, and potential secondary keys.
4. Use sample data to populate the fields for three records.

Create the ERD

This process is already shown in the "Developing Entity Relationship Diagrams (ERDs)" document. Please read through that document for an explanation of how it is done.

The fully attributed ERD developed in that document is shown in Figure 1

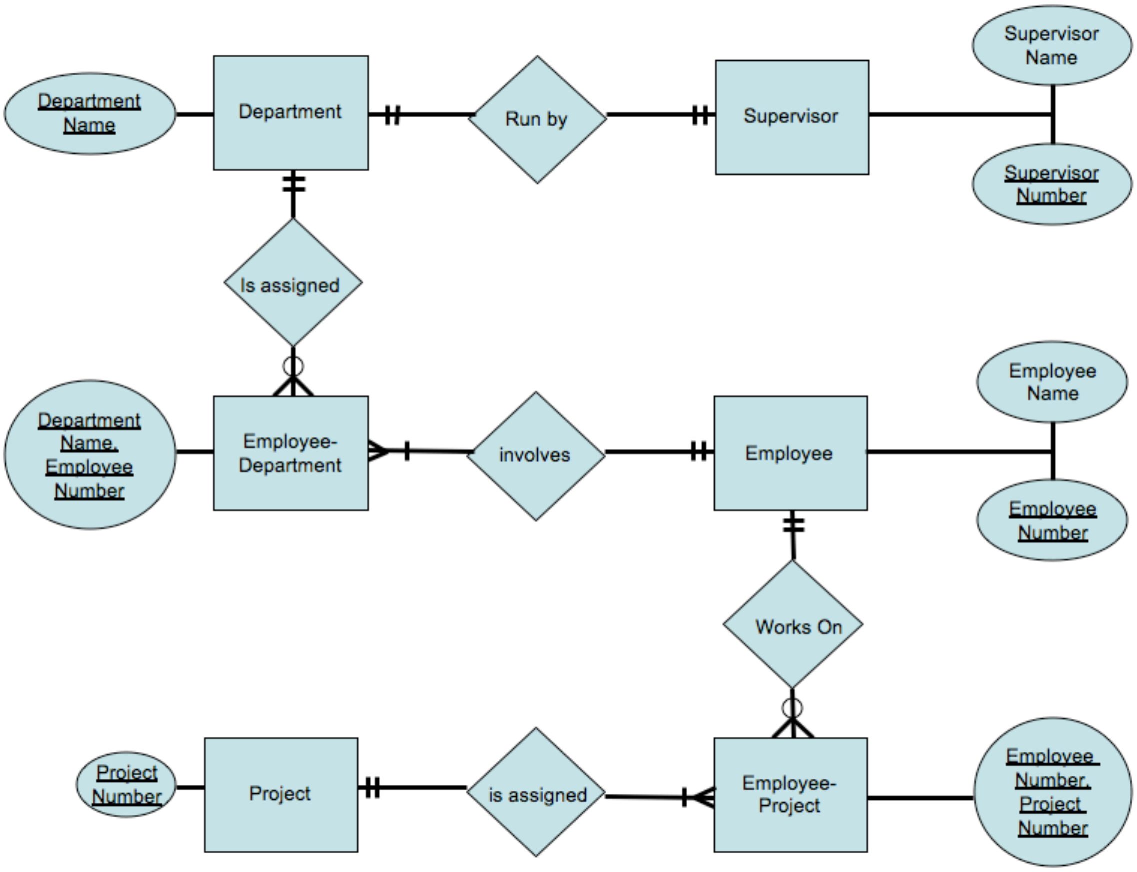

Figure 1
Completed ERD.

Create the 3NF Tables

This process has two steps:

1. Convert the ERD into tables expressed using "Standard Notation Format" – explained in the textbook.
2. Ensure that the tables are in 3NF.

The next two sections work through those steps.

Convert the ERD into SNF

The simple process is

- Create a table/relation for each entity from the ERD.

- The entity attributes become the attributes for the table/relation.
- Potentially, include some foreign keys to represent relationships shown in the ERD.

Create a table for each ERD entity

There are 6 entities in Figure 1 so there are six tables with the same names: Department, Supervisor, EmployeeDepartment, Employee, EmployeeProject, Project.

Entity attributes become table attributes

Department: (DepartmentName)

Supervisor: (SupervisorNumber, SupervisorName)

EmployeeDepartment(DepartmentName, EmployeeNumber)

Employee(EmployeeNumber, EmployeeName)

EmployeeProject(EmployeeNumber, ProjectNumber)

Project(ProjectNumber)

Obviously this is a very simple case, with simple data. But even with that it would be a touch silly to have tables that only contain a primary key such as the Department and Project tables above.

Some of this is solved in the next step.

Include some foreign keys to represent relationships

Let's list the relationships and take each in turn

- Run By
  Can be represented by adding SupervisorNumber to the Department table
  Department: (DepartmentName, SupervisorNumber)
- Is assigned (employee-department)
  Is already represented in the EmployeeDepartment table
- Involves
  Also represented in the EmployeeDepartment table
- Works on
  Represented in the EmployeeProject table
- Is assigned (employ-project)
  Also repsented by the EmployeeProject table.

This means no other foreign keys are really needed.

This leaves the Project table with only a primary key. The primary key is meant to uniquely identify instances of an entity. The only data being stored in this table is the primary key, it has nothing to identify. The choices are

- Remove the table. It's not needed if it only has the primary key.

- Add more attributes to the table.

It's the second point that makes more sense. In an organization you aren't going to know projects by their number. There would be a project name. So the Project table becomes

Project: (ProjectNumber, ProjectName)

You could also do similar things for the EmployeeDepartment and EmployeeProject tables. For example, storing the start date that an employee joined the department and project.

EmployeeDepartment(DepartmentName, EmployeeNumber, StartDate)

EmployeeProject(EmployeeNumber, ProjectNumber, StartDate)

Normally, this sort of process would not be required. Its required here because this case has been designed to be very simple so that it can act as a nice introduction.

Are the tables in 3NF?

This is where you check whether the tables are in 3NF and if not you should normalise them by breaking up the tables into smaller tables.

First, lets list the current tables.

Department: (DepartmentName, SupervisorNumber) – SupervisorNumber is a foreign key

Supervisor: (SupervisorNumber, SupervisorName)

EmployeeDepartment(DepartmentName, EmployeeNumber, StartDate)

Employee(EmployeeNumber, EmployeeName)

EmployeeProject(EmployeeNumber, ProjectNumber, StartDate)

Project: (ProjectNumber, ProjectName)

To check whether these tables are in 3NF you must answer the following questions for each table

1. Does the table contain any repeating groups?
   If not, and the table has a primary key then it is First normal form (1NF)
2. Does the table contain any partial dependencies?
   If not, and it is in 1NF then it is in 2NF
3. Does the table contain any transitive dependencies or derived attributes?
   If not, and it is in 2NF then it is in 3NF

If you don't know what these questions mean than you need to view the slides and listen to the audio to the normalisation lecture that is available from the course website. The lecture also shows how to go about normalising the tables if required.

As it turns out, these tables are in 3NF.

Identify the keys

The exercise also requires that we identify:

- Primary keys
  A field or collection of fields that can be used to uniquely and minimally identifies an instance of an entity. We've already done this as part of developing the ERD.
- Candidate keys.
  In some cases, it is possible that there might be different primary keys in the one table. In this case, the tables are very small and there are no other primary keys in any of the tables.
- Foreign Keys
  Represents a link between two tables. Already identified in the above.
- Secondary keys.
  Normally data is accessed via the primary key. Relational databases are implemented to make this access fast. With more complex tables you often want to be able to quickly access data via a field that is not the primary key. This is the purpose of secondary keys, sometimes called an index. These tables are too simple to have them.

Populate the tables with sample data

The important considerations to keep in mind while providing your sample data include:

- Ensure you stick by the constraints placed by primary keys
  Each primary key should identify a unique entity. For example, there is no duplicates values for DepartmentName in Table 1.
- Keep the connections implied by foreign keys.
  For example, each SupervisorNumber in Table 1 has a matching entry in Table 2.
- The data should attempt, where possible, to make sense within the domain of the case.
  For example, it would proably not make sense for employee 55 to be involved with project 65. From Table 6 you can see that project 65 is Mowing. Employee 55, as shown by Table 3 belongs to the Finance department. A member of the Finance department mowing the lawns doesn't make sense.

DepartmentName	SupervisorNumber
Finance	3
HumanResources	2
Buldings and Grounds	1

Table 1
Department table sample data

SupervisorNumber	SupervisorName
1	Fred
2	Barney
3	Wilma

Table 2
Supervisor table sample data

DepartmentName	EmployeeNumber	StartDate
Finance	55	Homer
HumanResources	105	Bart
Buildings and Grounds	1234	Lisa

Table 3
EmployeeDepartment table sample data

Employee Number	EmployeeName
55	Homer
105	Bart
1234	Lisa

Table 4
Employee table sample data

EmployeeNumber	ProjectNumber	StartDate
55	43	10/05/2005
1234	65	03/08/2005
105	103	04/02/2006

Table 5
EmployeeProject table sample data

ProjectNumber	ProjectName
43	Finance System Upgrade
65	Mowing
103	CEO Search

Table 6
Project table sample data