ITR08 Lecture 8

Introduction to Databases

Thomas Krichel

2000–03–20

Reading

Roman, Steven (1999) "Access Database Design and Programming" (chapter 1)

ITR8 class (2001) "sofix file" http://wotan.liu.edu/home/krichel/itr8/xml/sofix.xml

Definitions

A "database" is a collection of related data.

A "database management system" (DBMS) is a computer system designed to

- to add, delete, and update the data in the database
- to provide various ways to view (on screen or in print) the data in the database

In fact, a database is most of the time understood as comprising *tables* of data!

Example table

#	*Title*	*C-year*	*R-year*	*Composer*
1	Serenade in E, K. 387	1788	1991	Mozart,W.A.
2	Serenade in c, K. 388	1788	1991	Mozart,W.A.
3	Symphony No.3 in E flat Op. 55 "Eroica"	1803	1998	van Beethoven, L
4	Symphony No. 4 in B flat Op. 60	1806	1998	van Beethoven, L

this is a table of works. Recall that in sofix, we have items, works and tracks...

flat and relational databases

A database is "flat" if it has one single table.

A database is "relational" if it has more than one single table.

How would a flat database for the sofix data that we have gathered look like?

Line 1:
Allegro maestoso | 7:42 | orchestra | Orpheus Chamber Orchestra | composer | Mozart, W. A. | Serenade in E, K. 387 | 1991 | 1788 | D 115273 | Deutsche Grammophon

Line 2:
Minuetto I – Trio | 4:10 | orchestra | Orpheus Chamber Orchestra | composer | Mozart, W. A. | Serenade in E, K. 387 | 1991 | 1788 | D 115273 | Deutsche Grammophon

Line 3:
Adagio | 6:01 | orchestra | Orpheus Chamber Orchestra | composer | Mozart, W. A. | Serenade in E, K. 387 | 1991 | 1788 | D 115273 | Deutsche Grammophon

this a just the first three lines...

Problems of flat databases

1. redundancy: many bytes are duplicated
2. update anomalies: imagine we want to change the number of an "item", but we forget to do it on one of the tacks
3. insertion anomalies: if we have a new work, but we do not know what tracks it has, we have to leave blank entries
4. deletion anomalies: if we delete all works on an item (in the sense of sofix), then we loose the data for the item.

A relational database for sofix data

I suggest to have three tables

1. a table of items
2. a table of works
3. a table of tracks

The table of items

It will take the form

item_number | *labelname* | *number*

Example:

1 | D 115273 | Deutsche Grammophon

to be done by mferran, and dstein, for 2001–03–27

The table of works

It will take the form

item_number | *work_number* | *compositionyear* | *recordingyear* | *role_1* | *name_1* | *role_2* | *name_2* | ...

Example:

1 | 2 | 1788 | 1991 | Serenade in C minor, K.388 | composer | Mozart, W. A. | orchestra | Orpheus Chamber Orchestra

(all in one line)

to be done by a group of three, for 2001–04–03

The table of tracks

It will take the form

item_number | *work_number* | *track_number* | *tracktitle* | *time*

Example:

1 | 2 | 2 | Andante | 4:24

to be done by a group of four, for 2001–04–03

Difficulties of relational databases

1. database design: how to split the flat database in several smaller databases
2. update anomalies: imagine we want to change the number of an "item", but we forget to do it on one of the tacks
3. maintaining integrity: if we remove a work in the table of works, we must make sure that we delete its tracks from the table of the tracks.
4. creating view: if we have data in different tables, it is more difficult to find out information that is dispersed around different tables. Example: which items have a track that is between 3 and 4 minutes long?

we will get through this, somehow