Databases

Week 1 - Introduction to Data modelling

Activity

- Students should create groups of 3-5 participants (no more, no less) numbered groups 1, 2, 3, etc.
- There should be an even number of groups.
- Each group should take it own business requirements. The requirements can not be shown or revised to any other group.
- For 20 minutes, group 1 will interview the group 2 (group 3 will interview group 4 and so on) to create the respective ERD.
- For 20 minutes, the groups swap their roles (interviewer and interviewed). Another ERD should be generated.
- In the final part (30 minutes), the answers will be discussed in the board.

The Music Database
for even groups

- The music database stores details of a personal music library, and could be used to manage your MP3, CD, or vinyl collection. Because this database is for a personal collection, it's relatively simple and stores only the relationships between artists, albums, and tracks. It ignores the requirements of many music genres, making it most useful for storing popular music and less useful for storing jazz or classical music. (We discuss some shortcomings of these requirements at the end of the section in "What it doesn't do".)
- We first draw up a clear list of requirements for our database:
  - The collection consists of albums.
  - An album is made by exactly one artist.
  - An artist makes one or more albums.
  - An album contains one or more tracks
  - Artists, albums, and tracks each have a name.
  - Each track is on exactly one album.
  - Each track has a time length, measured in seconds.
  - When a track is played, the date and time the playback began (to the nearest second) should be recorded; this is used for reporting when a track was last played, as well as the number of times music by an artist, from an album, or a track has been played.
- There's no requirement to capture composers, group members or sidemen, recording date or location, the source media, or any other details of artists, albums, or tracks.

The University Database
for odd groups

- The university database stores details about university students, courses, the semester a student took a particular course (and his mark and grade if he completed it), and what degree program each student is enrolled in. The database is a long way from one that'd be suitable for a large tertiary institution, but it does illustrate relationships that are interesting to query, and it's easy to relate to when you're learning SQL. We explain the requirements next and discuss their shortcomings at the end of this section.
- Consider the following requirements list:
  - The university offers one or more programs.
  - A program is made up of one or more courses.
  - A student must enroll in a program.
  - A student takes the courses that are part of her program.
  - A program has a name, a program identifier, the total credit points required to graduate, and the year it commenced.
  - A course has a name, a course identifier, a credit point value, and the year it commenced.
  - Students have one or more given names, a surname, a student identifier, a date of birth, and the year they first enrolled. We can treat all given names as a single object—for example, "John Paul." o When a student takes a course, the year and semester he attempted it are recorded. When he finishes the course, a grade (such as A or B) and a mark (such as 60 percent) are recorded.
  - Each course in a program is sequenced into a year (for example, year 1) and a semester (for example, semester 1).

Time to check the results

Groups presentations

See you next week