

University of Maryland at College Park

CMSC 424 - Database Design - Spring 2006

Application Development Project: iPod Video Database

1 Purpose of the project

Apple Computers recently introduced iPod with Video and also started making some TV Shows available for download (at \$2 a pop), in partnership with various TV Networks (ABC, NBC etc); there are also many videos available for free. You are to analyze the requirements of, design, implement, and demonstrate a database system that could be used as the backend database of such a service. (Note: You will essentially be duplicating the functionality of iTunes Store for most part; feel free to borrow ideas).

The main goal of this project is to design a relational database, called iPodVideoDB (a replacement for iTunes Video store), that can be used to automate the data management functions of this service, and create a web interface to interact with this database. You will also be required to populate this database using XML and HTML data available on the Web.

2 Application Requirements

You are building the backend database for Apple, which would essentially serve as the middle-man between the content providers (e.g. ABC) and the consumers (you). The consumer interaction is quite straightforward: they can search the available videos, and buy those that they like (for a specified cost, or for free if there is no cost). When you sign in, a credit card needs to be provided, which is billed on a periodic basis (I don't know Apple's exact policy or schedule).

The content provider interaction is not visible to us, and is probably more ad hoc in practice; we will use a simple model. We will assume that Apple provides the content providers with an interface to add Videos, and also allows them to unilaterally specify the price for the videos as well. Assume that Apple takes a 50% cut of all revenues (the bandwidth and hosting costs can be quite high). The exact percentage cut in practice probably depends on several other things. The specified percentage of the money charged to the consumers is also passed on to the content providers on a periodic basis. You should also allow new content providers to decide to join in, and start offering their own videos etc.

You are free to change the service model, subscription plans or anything else you might like if you feel your idea is better. In case of drastic changes, you might want to talk to the TA or me first.

2.1 Data

A brief description of the data that needs to be stored in the database follows.

- **Data about Videos:** Information about the videos available for purchase. This should include all the pertinent information about each video. Video's can be of different types: (1) TV Show Episodes, (2) Song videos, (3) Short Movies etc. Additional information about the videos should also be maintained; e.g. Emmy awards information. Director/Actor information should be maintained. You have freedom in designing the search interfaces the consumers are provided with. Finally, the *owner* of each video needs to be maintained (see below).
- **Customer Data:** Information about the customers. Along with the basic information, this needs to include information about all the videos purchased by the consumer etc. Users should also be allowed to rate videos; this information can be used to recommend shows to other users. Finally, the customer id, password, and credit card information should also be maintained.
- **Data about Content Providers:** The videos are provided by content providers, e.g. ABC, Disney etc. Other than maintaining information about which video is owned by whom, we also need to maintain the information about the content providers themselves, e.g. the contact person, the mailing address etc. In addition, we need to maintain the information about the *deal* that has been made with the content provider, in particular what percentage of the revenue generated by selling their videos should be sent to them, and also when it should be sent (1st day of every month etc). Assume the default percentage cut for Apple is 50%, but this may be different for some customers.

2.2 Tasks & Queries

This section gives only a few of the tasks and queries. You may choose to add and/or substitute some other more interesting ones depending on the emphasis of your project.

- **Extract-Transform-Load (ETL):** Populating the database (data entry) is one of the tasks that you have to *design*. Much of the auxiliary information stored by this database is available on the web. For example, information about TV Shows is available at TV Guide or Yahoo! TV (<http://tv.yahoo.com>). It is also available at IMDB.

Such Web data will most likely be in the form of HTML, or if you are lucky, XML documents. Automatically parsing such documents is an important task.

To make your task more concrete, you will be provided with an XML file that contains information about a few TV Shows or Movies, along with their prices (imagine this to be an

XML document sent by a Content Provider). You have to write a program/script to process this XML file, and automatically insert the extracted data into your database. The script should be able to take any XML file with the same format as the sample XML file, and parse it and insert the information into the database.

You should populate the rest of the tables with some data manually (e.g. by scraping some webpage manually). The amount of data in the database should be sufficient to answer some interesting queries (say 50-100 tuples per table should be more than enough).

- **Customer Interaction Queries:**

- **Register for the service:** Provide credit card information, choose a unique id etc.
- **Browse/Search for Video's:** A customer should be able to search the video collection by various search parameters such as the genre, actor name, director name, title, year of release, Emmy winners, Emmy nominated TV Shows etc. It should also be possible to search by *top sellers*, *highest rated videos* etc.
- **Display information about a Video:** This would include all known information about the video, including, if you have it in the database, the critic and customer reviews. This should also include information about the price of the video etc.
- **Buy Video:** If the video is free, the download by the user should still be recorded. The amount of the transaction should be noted down. It would be preferable to not charge the credit card immediately for each video; charging it say every month is a better idea (see Administrative Task queries below).
- **Rate a Video:** The customer should be able to rate Videos that she has seen previously, whether bought through or not through iPodVideoDB.

- **Content Provider (CP) Interaction Queries:** The main task that a CP would like to do is to add more videos for sale. You should provide such an interface to the users. *Anybody* should be able to walk in, register as a CP, and add videos and their prices.

- **Administrative Task Queries:** The main administrative tasks have to do with charging the customer credit cards and passing a percent of the revenue to the content providers. You should build a web interface that allows the administrator to click a button that does the above two tasks, and provides a summary after doing them.

There are many directions that you can explore to enhance your project for extra credit (upto 20% of the project grade/4% of the total grade). Some suggestions:

- Recommend videos to the users based on their preferences/ratings, and other auxiliary information that you have stored. There are many ways this can be done. Ask me if you want ideas (e.g. *collaborative flitering*).

- Add *social circle* information for consumers so they can see what their friends, their friends of friends, have bought etc.
- Store the customer credit card information *securely*.

Be creative.

3 Rules of the game

- **Groups:** The project is to be done in groups of 2 students. A roster for each group must be submitted to the TA by the date specified in the “Due Dates” section of class schedule. The groups are “self-policing” (e.g., each group is responsible for its own division of labor, scheduling, etc.). *Note: If an unreconcilable problem arises in your group, it is your responsibility to contact both the professor and the TA as soon as possible. After the project is due, it will be too late.*
- **Assumptions:** In cases where you have questions on the above description, it is acceptable to make assumptions about the application providing that: 1) they are explicitly stated in the report, 2) they don’t terribly conflict with any of the requirements specified above, and 3) they are ”reasonable”. If you have a question about the acceptability of any of your assumptions, check with the TA or the professor.
- **Reports:** A report should be handed in at the end of each phase (Due dates below).
- **Implementation:** The final phase of the project requires a working implementation of the system to be built, tested, and demonstrated. A large part of the project grade depends on the quality of this implementation. The implementation will be done as a client-server system in which a web server runs on your cluster unix account, accepts web queries, and connects to the Oracle DBMS to retrieve from the database.

4 Project Phases

Phase	Phase Name	Due Date
<i>0</i>	Group names to the TA	Feb 16, 2006
<i>I</i>	System Analysis and Specification, Conceptual Modeling.	Mar 16, 2006
<i>II</i>	Implementation, Testing, and Demo	May 2, 2006

5 Reports

The Phase I report must contain:

1. a *short* description of the purpose of the project and the purpose of this phase of the project.
2. a description of the scope of the project
3. the assumptions that you have made about the enterprise.
4. the graphical schema using the E-R model,
5. list of the attributes for each entity and relationship,
6. the relational schema obtained by mapping the E-R to relations, and their Boyce-Codd or 3rd Normal Form with keys.
7. a description of how you plan to populate the tables (including details of how you plan to parse the XML document provided).

The Phase II report must contain:

1. Phase I report with corrections addressing TA's feedback.
2. a description of the purpose of this phase of the project,
3. a description of the problems encountered in this phase and justification for the solutions.
4. any revisions made to the relational schema definition from Phase II,
5. a brief summary of your implementation efforts, the tools used etc.
6. a description of the system's limitations and the possibilities for improvements.
7. In addition, a demo of the system is required. All members of the group should attend this demo, to explain the aspects of the project for which they were responsible.