

Share Stuff Project Review

The goal was create a multi user web site where people could post stuff that they were willing to share with other members. The site was designed to focus primarily on books, movies and cd's but can also accept other items as miscellaneous.

It was implemented using apache web server, php server side scripting as an interface to oracle database. As outlined in the initial project targets, the site includes the ability to

- post items for sharing
- display pages for all items for borrowing with the ability to checkout and return items.
- list of all items owned by user logged in with ability to remove items from circulation
- Ability to make comments on items that have been borrowed.
- Ability to queue up on an item that is not currently available and to cancel queue
- reports on user sharing statistics

Over and above the stated deliverables, extra functionality includes

- simple security login scheme that polls database for user id and passwords.
- Key word seach across multiple text fields (case sensitive)

The site implements all tables as submitted in original project specification and builds on top of it (eg adds views, use of sequence for auto generation of primary index keys etc). Functionality is largely complete with some minor incomplete corner cases.

- bookItem and item are complete but cditem and videoitem are only partially complete. The functionality should be identical but time is requiered to implement and align look and feel of results pages.
- Some planned standalone Reports were not implemented the information was met through other views. (eg total items in pool, total value of items in pool, user goodness all have combined view from mystuff -> show totals) .
- New user sign up page not implemented

The project implements all key deliverables and adds some small incremental items while leaving out a few originally planned dedicated reports in favour of redistributing information access for better integrated view.

While there are many more things that can be done to improve the site, the goal of establishing the foundational functionality has been reached. Other supplementary functionality would improve usefulness might include

- new user page
- email integration
- auto generated emails for regular report/reminders
- more reports on item circulation statistics
- better implementation of security - right now all pages are publicly available and directly accessible by anyone who knows the php page names. Currently Passwords are stored directly in database in clear text.

Specified Demonstrations

1. **(1 point) Selection and projection query:** mystuff.php displays all items owned by a specific user (the one currently logged in)
2. **(1 point) Join query:** showallBooks displays all books that are a result of joining a book with its parent item through a view.
3. **(1 point) Division query:** mystuff -> show Totals displays a number of reports. The presidentail report shows all users who have borrowed all items shared by the president.
4. **(1 point) Aggregation query:** mystuff-> show Totals, shows totals by person and availability as well as rolling up subtotals.
5. **(2 points) Nested aggregation with group-by:** From the myStuff page, the ShowTotals button displays a number of reports including at the bottom a report showing Max, count and totals for all users offering an average value greater than the overall average value.
6. **(1 point) Delete operation:** delete self; removes all items owned by user as well as deletes the current user using update cascade on both the items and the user.
7. **(1 point) Update operation:** mystuff shows the availability of items owned by the current user. Available is constrained to be either T or F. In addition to the constraint the user interface is designed to toggle value to prevent any possible illegal error input.
8. **(2 points) Graphical user interface:** html frames were used with buttons and http links to achieve user interface utilizing server side scripting for all updates. Note that for most queries, the related sql statements are also displayed at the bottom of the queries to assist the TA.

Some additional items that might be worth noting:

- user passwords are checked and validated based on table persons
- views were used with books, cd, videos to simplify the is-a relationship handling while preserving the fact that most relationships were through the parent item table.
- Since the site was designed around multi-user, the drop down login setting to change user perspective was used to verify the user specific views of the collection of items.
- sequences where used to autogenerate primary key indexes for some relations.
- key word search was implemented to search all text fields across items, books, cd, video including fields specific to the subclasses. The resulting display was produced by concatenating string fields for consolidated display. Test directly through sqlplus showed that alter session set NLS_COMP='LINGUISTIC' would allow search ignoring case for the session but this command had no effect through the OCI interface and so search remains case sensitive.
- Where possible, the use of web and php interface to constrain user input to valid ranges was implemented.

Files Submitted

- cs304.TeamShare.zip contains the complete html/php for the website
- It also contains a pair of sql files that are used to insert the tables and the test data.
- makeProj06.sql - creates the tables, views and sequences used for the relations.
- in3.sql populates with some test data for users, items and other tables.

- Also included is a summary list of sql statements for reference as well as a larger secondary set of test data
-