445 Course Project

Yanlei Diao
UMass Amherst
Fall 2009
Project Requirements

- Basic functionalities of a movie database
  - Searches
  - User login, rating, and reviews
  - DB admin
  - Constraints

- Performance requirements
  - A few seconds for most queries

- An extension beyond above
  - Social networking among movie fans
  - Customized advertisements
Data Sets

- Movies: (name, year)
- Directors: name + movies directed
- Actors: name + movies played
- Actresses: name + movies played
- MPAA ratings: movie + mpaa rating
- Genres: movie + genre
- User table: login (key), name, password, age, gender, location, school, movie interest (a list of genres, and a list of favorite movies)
Step 1: ER Diagram

- **Entity sets**
  - Movies, Directors, Actors(Actresses) (or People + role?), Genres?, MPAA rating table, Movie reviews...
  - Users (login, name, age...)
  - Online movie company?

- **Relationship sets**
  - A movie has a director, a cast, a genre (optional), a rating (optional), a list of reviews (optional)...
  - Users can rate and write reviews for movies
  - Users can be friends with each other...

- **Constraints**
  - A user can rate a movie once
  - Age constraint (left to php?)...
Step 2: Relational Model

- An entity set translated to a table
  - Set the primary key
  - Can introduce an ID to be the key; use ‘auto_increment’

- A relationship set translated to a table or embedded in an existing table
  - Attributes (Not Null?)
  - Foreign keys
  - Primary key

- Implement complex constraints using triggers or in PHP
  - PHP: program as you want
Proposal due on Oct 13

1. Extension beyond basic functionality and performance requirements
2. E/R diagram and relational schema for your application
3. Milestones and timeline for this semester
4. Responsibilities of group members
Step 3: Parse & Load Data Set

- Load movie data set into your defined tables
  - Data format: see the project description
  - Movie dataset (all movies ever made) on Edlab:
    /usr/net/ftp/pub/cs445/f2009/project/movie
  1. Parse data into the format defined in your schema
  2. Change permission of your directory and files to be **publicly accessible**
  3. Load parsed data files into your tables in MySQL

- Similarly, load user dataset into your tables
  - Data format: see the project description
  - User dataset (1M users) on Edlab:
    /usr/net/ftp/pub/cs445/f2009/project/user
More about Data Loading…

- If you decide to use movie_id:
  - Declare the field to be ‘auto_increment’;
  - Load movies into table M; movie ids are auto generated;
  - Load each other data file (e.g., Acting) first without the movie id field (say, into Table A1).
  - Join A1 with the movie table M; insert all result tuples into a new table (A2) to be the real Acting table;
  - Delete the old table (A1).

- Constraints:
  - User can rate a movie once: how do you implement in DB?
  - Other constraints, e.g., age constraint: implement in PHP?
Step 4: Test Queries & DB Tuning

- Test queries are released on the project web page.
- Test your database using the test queries.
  - If answer not correct, check schema & queries.
- If performance is not good,
  - Add indexes:
    ```sql
    CREATE INDEX index_name ON table_name ( column_name(s) )
    ```
  - Revisit the schema design
**Step 5: Web Application Development**

- **PHP/MySQL tutorial**
  

- **PHP on Edlab**
  
  
  - Group name and password to access Edlab
  - Create ‘www’ subdirectory
  - Upload your php files
  - Also need username/password to access CS PHP site

- **In-class Q&A session: Nov 17**
Step 6: Implementing Extension

- Can start as early as the development of the movie DB
- Or can wait until after the movie DB + PHP is developed...
Midterm Report due on Nov 12

1. Data loading completed
2. Performance tuning completed
3. Test queries passed
4. PHP development started
5. Implementation of your extension started
6. Update your timeline indicating which of the planned tasks are completed
7. Refine the tasks that remain to be completed