Database Project: Video Preferences Website

You are in charge of setting up a video replay website. This site will use a database. The database will contain information about the different users registered on the site, the available videos, and the users' viewing history. Suggestions for videos to watch will also be generated.

Various shows will be available. Each show will be associated with a category (for example: culture, cinema, etc.). Some shows may have multiple episodes. After the release of the video, a video will be accessible on the site as replay for at least 7 days. After a certain amount of time, some videos will no longer be available for replay viewing. They will then be archived.

Different types of information will be available about the videos, such as their name, description, duration, release date, country of origin, availability of multi-language options, image format, and more.

A registered user will be characterized by a login, a username, and a password. They may also provide the following information: first name, last name, date of birth, and email address. A user will be able to indicate the categories of shows that interest them. They may also choose to subscribe to the site's newsletter, which will be generated weekly.

For a user who has created an account on the site, it will be possible to subscribe to shows. In the viewing suggestions, newly available episodes of the shows they have subscribed to will appear. The user can also mark certain videos as their favorites. The videos that are marked as favorites will be visible on the site, on a dedicated page for the user, allowing them to watch these videos later. It will also be possible to view videos that are close to their expiration date. Each viewing of a video will be recorded, and a user will also be able to review their viewing history.

Suggestions for videos to watch will also be generated. These suggestions will be based solely on the popularity of the videos by category. Popularity will be calculated based on the number of views during the last two weeks, in order to favor recent videos.

SQL queries, PL/SQL procedures, and integrity constraints will be indicated later in the project. They must be considered in the modeling of the database.

Project Deliverables

Modeling

Design the entity-relationship model of the database for task list management while respecting as much as possible the functional requirements specified in the statement.

Specify the integrity constraints on the model (constraints that the data must respect at all times).

The constraints must be indicated in text form (a paragraph containing the list of constraints to be implemented, which means expressed in SQL).

Establish the logical relational model of the database (all relations derived from the entity-relationship model).

A relation will be in the form **R(X1: T1, X2: T2, …, XN: TN)**, where **R** is the name of the relation, **Xi** are the names of the attributes, and **Ti** are abstract types (for example: **String** rather than **Varchar**). Do not forget to specify primary keys/foreign keys.

A report must be provided containing the E/R diagram, the list of integrity constraints, and the relational model.

Database Implementation

SQL scripts for table creation, table deletion, and data insertion for tests must be submitted.

SQL scripts can also contain commands for creating/deleting views.

Static integrity constraints (which can be implemented without using PL/SQL) must also be expressed in SQL in the table creation scripts.

A script should be provided to execute all other scripts in order to recreate the tables if necessary and to populate them with data.

SQL Queries

Write the queries to obtain the following results:

- Number of video views by video categories, for views from the last two weeks.
- Number of subscriptions, favorites, and watched videos per user.
- For each video, the number of views by French users, the number of views by German users, the difference between the two, sorted by the absolute value of the difference between the two.
- Episodes of shows that have at least twice as many views as the average views of other episodes of the same show.
- The 10 pairs of videos that appear most often simultaneously in a user's viewing history.

Define indexes to optimize your queries.

PL/SQL Procedures and Functions

- Define a function that converts the information of a video into JSON format.
- Define a procedure that will generate an initial text for the newsletter by adding the list of all releases of the week.
- Generate the list of popular videos, recommended for a user, meaning videos from the categories they are interested in.

Triggers

Implement the different integrity constraints specified in the project. You can define static or dynamic constraints. Implement the following integrity constraints:

- A user can have a maximum of 300 videos marked as favorites.
- Deleting a video will lead to its archiving in a table of videos that are no longer accessible through the replay site.
- To limit view spamming, a user cannot start more than 3 views per minute.

Project Submission

You must submit an archive containing:

- A short report explaining the choices made during the project implementation
- The E/R model
- The relational model
- The SQL scripts for creating/deleting tables/views with static constraints and data insertion scripts
- SQL queries
- Integrity constraint definitions (text version)
- PL/SQL scripts (functions, procedures, triggers)
- Indexes
- If necessary, define the start/end of transactions for your operations
- A script to test the procedures/functions/triggers