MET CS 669 Database Design and Implementation for Business
Term Project: Online DVD Rental Business

Objective
Create an initial design for the database schema for an online DVD rental business that is similar to the DVD rental portion of the business pioneered by *NetFlix*®. Map your conceptual design into your approved DBMS, and implement the database schema. We will call this *NetFlix*® subset the *classic default term project*.

Default Project Description
Design a database for a DVD rental business such as Netflix.com. The store rents DVDs online. The customer business model is illustrated below.

Image Source: http://www.netflix.com/HowItWorks?lnkctr=nmhhiw

In order to rent a movie, a person must be enrolled at the online store. There are two different membership programs. As quoted from *NetFlix*®:

> "With *NetFlix* you can rent as many DVDs from the comfort of your home and have them delivered to your door in about 1 business day. There are no late fees and no due dates, and shipping is free both ways. Programs start at $11.99 plus any applicable tax. With our most popular program, 3-at-a-time, you can rent as many DVDs as you want for just $17.99 a month. You keep a revolving library of up to 3 DVDs at a time and can exchange them for new available DVDs as often as you like. … Our 2-at-a-time program (limit 4 DVDs per month) is $11.99 plus any applicable tax per month." *(From http://www.netflix.com)*

Below is an example use case to aid in understanding the Netflix operation:

1. A customer signs up for the 3-at-a-time program.
2. The same customer adds 10 movies to their queue.
3. Netflix mails to the customer the first three movies in their queue.

4. The customer watches and returns the first movie to Netflix.
5. Netflix mails out the next movie in the queue to the customer, which is the fourth movie added to the queue in Step #3.
6. The customer closes their account, but only returns two of the three movies the customer has at home.
7. Netflix charges the customer $25 for the missing movie.

This use case is not exhaustive. There will be elements in your database mentioned in the Netflix specification, but not mentioned in this use case.

The database will store membership information for each person, the movies she/he rented, movies in the queue to be rented, when were these movies returned, and so on.
The rental history is used for two purposes:

- To give employees a basis to work from when they are asked what movie the customer has rented out or if it was lost in mail
- To determine if the movie was never mailed back by the customer

The database that you design must support all the usual value chain operations occurring at a video rental e-store. You will design the Entity-Relationship model (ERD) for this database, and its corresponding relational model. It is not possible to attempt a full production-capable database for a large enterprise such as NetFlix in the time period afforded by this course; such a database would have a few hundred tables. A general guideline is that to effectively create this design for this course, the logical ERD should have between 10 and 20 entities. This number is not exact and will vary according to your specific implementation. Example entities include Customer, Movie, Rental, and Queue.

Note that this implementation need not provide for streaming or downloadable movie content.

Non-Default Term Project
You are welcome to implement a project of your own definition that involves the design, implementation, and usage of a different database schema. Perhaps the default project seems too easy, or you have a different project in mind. You can submit a proposal for your project, and your facilitator will examine your proposal for scope and level of effort, to help assure that you learn as much as possible and to keep you from undertaking a project that is unreasonably large. Unless the scope of your project fits neatly within the course timeframe, you will need to classify the work into that which will be completed by semester end, and that which you will need to leave until after the course is completed. You will be expected to submit term project iterations comparable to those on the default project on the same schedule. It is fine if you or your employer have a use for the database that you design and implement.

Use Cases
Below are 10 use-cases. A complete schema design will allow all ten of these use cases to be addressed in full. For use cases where information is requested, *provide a single query to retrieve the information*. For use cases that require data modification, *develop a stored procedure that performs the modifications, and invoke the store procedure.* The store procedures you create should be designed to be reusable, by making use of one or more parameters. For example, the stored procedure for use case #3 should use a parameter to specify the particular customer's queue that will be modified.

1) A customer requests the titles of all movies that are directed by "George Lucas" or by “Rich Christiano”. Write a single query that retrieves this information.
2) Management requests the names of all currently active customers, as well as the name of the current plan in which each of these customers is enrolled. Write a single query that retrieves this information.
3) A customer wants to add a movie to their queue so that the newly added movie will be the next movie they receive. Develop a parameterized stored procedure that accomplishes this, then invoke the stored procedure for a customer of your choosing.
4) A customer requests the titles of all the DVD’s that he or she has *not* rented. Write a single query that retrieves this information for a customer of your choosing.
5) A customer cancels their membership and does not return a rented DVD, necessitating that a $25 DVD replacement fee be charged to their account. Develop a parameterized stored procedure that accomplishes this, then invoke the stored procedure for a customer of your choosing.
6) A customer enrolled in the two-at-a-time plan cancels their membership. When a customer cancels their membership, they become inactive, but their DVD queue and rental history remains in the database, in the event they return as a customer. Develop a parameterized stored procedure that accomplishes this, then invoke the stored procedure for a customer of your choosing.
7) Management requests the names of all movies that are currently sold out. A movie is sold out if all copies of the movie are currently rented and not yet returned. Write a single query that retrieves this information for management.
8) Management requests identification of the plan with the most customer enrollees, and for that plan, the name, number of DVDs allowed at one time, and the number of customer enrollees. Write a single query that retrieves this information for management.
9) Management requests the names of all customers, and for each customer, the titles of the movies that they rented multiple times. For each title, management would like to see the number of times it was rented by the customer, only including titles that the customer rented more than once. If a customer has no rentals, or did not rent any movies multiple times, management does not want to see them in the list. Write a single query that retrieves this information for management.
10) Management requests the titles of all movies, and for each movie, the number of different customers that rented the movie. They would like the list to be ordered from the highest number of different rentals to the lowest number. Multiple rentals of the same movie by the same customer only count as one unique rental. Management is interested in the number of different customers that rented the movie, but not whether the same customer rented the same movie more than once. Write a single query that retrieves this information for management.

Your logical ERD will be mapped to a relational database schema through the use of SQL. The schema should contain tables, primary and foreign keys, indexes, and optionally triggers and stored procedures. The primary and foreign keys will help enforce the relationships indicated in the logical ERD, and help enforce referential integrity. The tables need to be filled with some fictional data. Make sure you integrate sample data from the situations above, for example, ‘Spielberg’ should exist in your database. Some tables may just need a few rows. Barring extraordinary circumstances, each table should need no more than 15 rows to effectively demonstrate the correctness of the queries, triggers, and stored procedures. You may need to be creative when inserting the data so that the queries return reasonable results.

Make sure you assign a genre to the DVD movies. A non-exhaustive, example list of genres is given below:

- Thriller
- Sci-Fi
- Comedy
- Drama
- Miscellaneous

Focus on the key value chain tables required for the situations above and related situations. Do not include subschemas for credit card processing, accounting, human relations, marketing, or the many other functional areas of a large enterprise. Concentrate on the value chain operations, which are those involved in the basic processing of customer activities – the activities that directly provide value to the customer.

Indexes

Create and justify two indexes that are beneficial to queries against your schema. Include screenshots illustrating the creation of the two indexes, along with explanations as to why each index is beneficial (be specific).

Deliverables

1. A Word document containing a complete repository of the term project:
    a. The business rules
    b. Conceptual ERD or EERD
    c. Logical ERD or EERD
    d. Screenshots illustrating execution and results of the SQL addressing the 10 use cases
    e. Screenshots illustrating the creation of two indexes, along explanations as to why each index is beneficial (be specific)

2. An electronic zipped file of all key deliverables that will enable your instructor and facilitator re-create your schema and execute your queries against the schema. At a minimum, the following three scripts should be included. You may include additional scripts if it makes sense for your implementation.
    a. Your “create” script with DDL, which contains the SQL code which creates the tables, primary and foreign keys, and stored procedures
    b. Your “insert” script with data for populating the tables
    c. Your “use case” script which contains the queries and stored procedure invocations for each of the 10 use cases

Grading Criteria

We will grade you according to the following percentage breakdown.

Percent of project grade	Item graded
20%	Exposition – How clearly and persuasively the ideas and designs are presented, and how well organized your Term Project submission is.
30%	The completeness and correctness of your design. The business rules, conceptual and logical ERDs, and database schema will be examined, including the entities, relationships, cardinalities, and primary and foreign keys. Your tables should be normalized to BCNF, or accompanied with a valid reason why the table was not normalized to BCNF.
5%	Submitting all five iterative term project deliverables on time. Each deliverable should represent a good faith attempt for an initial iteration of each section.
5%	Correct, working SQL scripts that are consistent with the other deliverables in the project. Using a database user with an empty schema, your facilitator should be able to execute your create script, your insert script, and your query script(s) without errors.
5%	Correct, working primary, foreign, and not null constraints.
5%	The indexes and the justifications. You need to explain the benefit of each index in terms of physical access.
30%	SQL for the 10 use cases.