# Objectives

The objective of this assignment is to develop a working application that at a high-level is comprised of the following components:

* A MySQL Database
* Java POJO Application that will load the provided data files into appropriate tables
* A Front-End Web-Application that can be developed in the technology that you are most experienced with

The developer is completely free to design and implement this application in any way they see fit and will be evaluated on creativity, logical design, implementation and functionality.

The developer is also free to use any reference material available (i.e. books or from the web) to aid in creating this application to the best of his/her abilities.

**Presentation to the NextGate Team**

Once the application is complete you will be asked to present and demonstrate the application over a web-conference to the NextGate team. Please be prepared to present and answer the following:

* Present and demonstrate the working Front-End Application
* Present the database schema and tables that were created
* Present and walk through the source code and structure of the Java POJO application
* Present and walk through the source code and structure of the front-end web-application
* Answer any questions related to design and implementation choices

**Project Deliverables**

Please package or zip your completed project and source code and send it to us over email to: [gevik.nalbandian@nextgate.com](mailto:gevik.nalbandian@nextgate.com)

Any notes or special instructions needed to compile and build the project should be included.

**Expected Project Deadline**

We would consider a reasonable deadline to complete this assignment to be 3 working days (24 hours).

This is a soft deadline - if you feel you need extra time please let us know but be aware that we appreciate both quality and punctuality.

**Rules**

However, we do ask that you adhere to the following rules:

* Work on this project on your own and do not ask for help or questions outside of NextGate -- we are happy to answer any questions you may have.
* Do not simply copy sample code or applications from the web - we are interested in seeing your work

# MySQL Database

### Software Installation and Setup

Download and install the latest version of MySQL at:

<http://www.mysql.com/>

The reference manuals and tutorials can be also be found at:

<http://dev.mysql.com/doc/refman/5.6/en/index.html>

### Create a Database and a User

Once MySQL has been successfully installed on your local machine you will then need to:

1. Create a new database (it can be named anything you like).

Instructions can be found in the MySQL Tutorial:

<http://dev.mysql.com/doc/refman/5.6/en/database-use.html>

Example: create database ng\_music;

1. Create a user/password that you will use to access the schema.

Please do not use the "root" user when accessing the database through your application.

Example: create user 'ng\_user'@'%' identified by 'ng';

1. You will also need to "grant" privileges to the new user you have created in order for that user to have full access to the database

Example: grant all privileges on ng\_music.\* to ng\_user with grant option;

### Install NextGate Provided Database Scripts

We have provided 3 SQL Scripts for you:

* **create\_tables.sql**
* **drop\_tables.sql**
* **create\_db\_user.sql**

Run the create-tables.sql as the new user you have just created and this will create 3 tables that you will use for your application.

If you need to start over you can run the drop-tables.sql script and that will remove the tables allowing you to re-run the create-tables.sql

# Java: Build the Data Loader

### Prerequisites

* MySQL database has been installed and a user has been created
* NextGate create\_tables.sql script has been run and you have verified that the tables exist

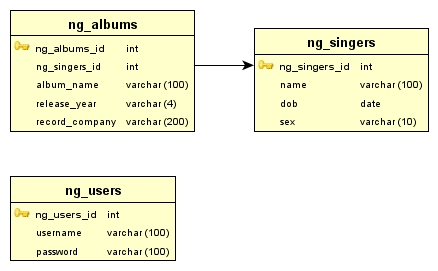
### NextGate Provided Data Files

We have provided 3 data files in simple text pipe-delimited format:

* ng\_albums.txt
* ng\_users.txt
* ng\_singers.txt

### Objectives

Build a java application that will read in a data file and load each row into the corresponding database table. The first line of each data file is the header so that should be ignored when you are loading. Below is the ER Diagram of the schema, use it check the field types and key constraints.



Note: it is not REQUIRED to use MySQL. It isn’t even required to use a relational database if you think some other data store technology would work better. If you choose to use a different technology, be prepared to discuss WHY.

# The Web-App

### Prerequisites

* A java loader has been created and run in order to populate the database tables with the dataset provided by NextGate.

### Objectives

Build a front-end web-application using any technology or toolset you feel most comfortable with.

It will be up to you to decide how simple or complex the design of this web-app will be so please do show off.

The front end UI application should at a minimum consist of the following elements:

1. Login Screen
   * User and Password Text Fields that validate against the User database table to control access into your web-application.
2. Search Screen
   * Upon successful authentication provide a means for the user to search for singers and albums.
3. Search Results Screen
   * Upon executing a search present the data onto a new search results screen showing the appropriate singer and album data.

# Extra Credit

All of the following are OPTIONAL. We put together this list as a suggestion for enhancements or design considerations that will be impressive if thoughtfully implemented. Please don’t over-reach. WE WILL NOT HOLD IT AGAINST YOU IF YOU DO NOT ATTEMPT ANY OF THE FOLLOWING. We will be more impressed if you complete the basic requirements in a thoughtful and comprehensive manner than if you sloppily attempt to implement everything in this list.

1. Login/general security:
   1. Cryptographically hashed passwords
   2. SSL (https) encryption
   3. Cleanse user input to SQL queries to prevent SQL injection.
   4. Include a way of adding new users through the GUI (perhaps only allowed for a specific user?)
2. GUI enhancements:
   1. An interface to add new entries to the album/singer tables.
   2. User-configurable skins (color pallet, widget style, etc.)
   3. Add an ‘about’ screen, with versioning info, library/license dependencies, etc.
   4. Results are sortable.
3. Database/data store enhancements:
   1. Extend the data model to include more fields which may or may not be searchable (stage name? band affiliations? Make record company its own table that relates to album and singer independently?).
   2. Use a different data storage/indexing technology (like Lucene). Supporting more than one data store technology would be even more impressive.
4. Data load enhancements:
   1. Add a hook for arbitrary formatting/normalization logic. This could take the form of one or more methods available to over-ride in a sub-class, or could be an optional groovy script, a class-loaded external jar or any number of things.
   2. Make it multi-threaded.
   3. Allow for very large input files to be processed