

Assignment 2

Submission instructions:

- The assignment should be submitted by pairs.
- You should submit project jar file (Java).
- In this assignment, All DB functions should use **Hibernate** and **HQL**.
- Make sure you have the table from the previous assignment (MediaItems, Similarity). The solution to the previous assignment is in the moodle.
- Don't submit code of tables or triggers creation.
- Don't submit the external jar files.
- Clean the DB before submit the assignment.
- You should use java packages and not submit the assignment with the default package (i.e. name the package with some name for your choice). We won't check assignment with default package.
- You should test your jar file and verify it doesn't throw any exception before you submit it! Make sure it works!
- The jar file should contain your project: all Java/XML sources, without external jars.
- The name of your project and jar file should be of form id1_id2 where id is your ID number.
- Deadline: **12/12/2020**

Assignment instructions:

1. Oracle SQL Developer
 - a. Create a table **Users** with the following columns:
 - USERID (NUMBER(11,0)) – primary key
 - USERNAME (VARCHAR2(200))
 - PASSWORD (VARCHAR2 (200))
 - FIRST_NAME(VARCHAR2 (200))
 - LAST_NAME(VARCHAR2 (200))
 - DATE_OF_BIRTH (TIMESTAMP)
 - REGISTRATION_DATE(TIMESTAMP)
 - b. Create a table **Administrators** with the following columns:
 - ADMINID (NUMBER(11,0)) – primary key
 - USERNAME (VARCHAR2(200))
 - PASSWORD (VARCHAR2 (200))
 - c. Create a table **LoginLog** with the following columns:
 - USERID (NUMBER(11,0)) – primary key (foreign key to Users)
 - LOGINTIME (TIMESTAMP) - primary key
 - d. Create a table **History** with the following columns:
 - USERID (NUMBER(11,0)) – primary key (foreign key to Users)
 - MID (NUMBER(11,0)) - primary key (foreign key to MediaItems)
 - VIEWTIME (TIMESTAMP) - primary key

2. Java

- a. Add ojdbc jar.
- b. Add to your project Hibernate jars.
- c. Generate using Hibernate tool the ORM.
- d. Make new Java class Assignment.java
- e. Write the Java function `isExistUsername` :
 - Signature: `public static boolean isExistUsername (String username)`
 - The function returns true if the received username exist in the table USERS otherwise false.
- f. Write a Java function `insertUser`:
 - Signature: `public static String insertUser(String username, String password, String first_name, String last_name, String day_of_birth, String month_of_birth, String year_of_birth)`
 - The function checks if the username exist in the USERS table, in case of positive answer returns null else insert the user to the table USERS including registration_date field and returns the user id (USERID).
 - Use generator class="increment" for USERID.
- g. Write a Java function `getTopNItems` :
 - Signature: `public static List<Mediaitems> getTopNItems (int top_n)`
 - The function retrieves from the table Mediaitems first top_n items (mid descending order).
- h. Write a Java function `validateUser`:
 - Signature: `public static String validateUser (String username, String password)`
 - The function compares received values with existing in the data base.
 - The function return USERID if the values are equal to the values in the table otherwise return "Not Found".
- i. Write a Java function `validateAdministrator`:
 - Signature: `public static String validateAdministrator (String username, String password)`
 - The function compares received values with existing in the data base.
 - The function return ADMINID if the values are equal to the values in the table otherwise "Not Found".
- j. Write a Java function `insertToHistory`:
 - Signature: `public static void insertToHistory (String userid, String mid)`

- The function inserts the row to the History table with current server time.
 - Print “The insertion to history table was successful <server time>”.
- k. Write a Java function `getHistory`:
- Signature: `public static List<String, Date> getHistory (String userid)`
 - The function retrieves from the tables History and MediaItems users's items.
 - The function return List of pairs <title,viewtime> sorted by VIEWTIME in ascending order.
- l. Write a Java function `insertToLog`:
- Signature: `public static void insertToLog (String userid)`
 - The function insert the row to the LoginLog table with current server time
 - Print “The insertion to log table was successful <server time>”.
- m. Write a Java function `getNumberOfRegisteredUsers` :
- Signature: `public static int getNumberOfRegisteredUsers(int n)`
 - The function receives an integer number n
 - The function retrieves from the table Users number of registered users in the past n days
 - The function return integer number
- n. Write a Java function `getUsers`:
- Signature: `public static List<Users> getUsers ()`
 - The function retrieves from the table Users all users
- o. Write a Java function `getUser`:
- Signature: `public static Users getUser (String userid)`
 - The function retrieves from the table Users user's information
 - The function return objects Users

Good Luck, Yafit & Bar.