Team 6 Project Document

Database Application: Library Database

1. Types of data that can be added, modified, and edited

* Books
* Devices
* Media (movies, albums, songs, discs)

1. Types of user roles in your application

* Admin: create, view, update, and delete any students, faculties, and items (books, devices, media). Fine users.
* Student: borrow and return books, pay fines.
* Faculty: borrow and return books, pay fines.

1. Semantic constraints and the triggers

* Triggers #1: After the admin approves the processing item, if there’s any late or damaged fee, that amount is added to the balance table. After the insertion on the balance table, the trigger runs to update the outstanding fees of the user by adding the fee amount to the user’s initial fee amount.
* CREATE DEFINER=ina7exsoe5de1hlb@% TRIGGER balance\_AFTER\_INSERT AFTER INSERT ON balance FOR EACH ROW BEGIN
* UPDATE users SET fees = fees + NEW.lateamount + NEW.damagedamount
* WHERE userid = NEW.borrowerid;
* END
* Semantic constraints: Make sure that the users have an accurate visual representation of their current total fines reflecting the importance of real-time data
* Triggers #2: After the user successfully makes the payment for their late or damaged item, that amount is deleted from the balance table. After the deletion on the balance table, the trigger runs to update the outstanding fees of the users by subtracting the paid amount from the user’s current fee amount and at the same time transferring the paid amount into the earnings table.
* CREATE DEFINER=ina7exsoe5de1hlb@% TRIGGER balance\_AFTER\_DELETE AFTER DELETE ON balance FOR EACH ROW BEGIN
* DECLARE amount INT;
* UPDATE users SET fees = fees - OLD.lateamount - OLD.damagedamount
* WHERE userid = OLD.borrowerid;
* SET amount = OLD.lateamount + OLD.damagedamount;
* INSERT INTO earnings (amount, borrowerid, name, itemid, title) VALUES (amount, OLD.borrowerid, OLD.name, OLD.itemid, OLD.title);
* END
* Semantic constraints: Make sure that the users have an accurate visual representation of their current total fines reflecting the importance of real-time data. Plus, the earnings table is used to generate revenue reports.

1. Types of queries/reports in your application

* Reports:
* Total revenue from all users

SELECT users.userid, users.firstname, SUM(earnings.amount) AS totalamount

FROM users

LEFT JOIN earnings ON users.userid = earnings.borrowerid

WHERE users.userid IS NOT NULL AND users.firstname IS NOT NULL

GROUP BY users.userid, users.firstname

HAVING totalamount IS NOT NULL;

* Total revenue from all items

SELECT available.itemid,available.title,SUM(earnings.amount) AS totalamountFROMavailable LEFT JOINearnings ON available.itemid = earnings.itemid GROUP BYavailable.itemid, available.title HAVING totalamount IS NOT NULL;

* Total revenue from one user

SELECT earnings.itemid, earnings.title, earnings.name, earnings.amount FROM earnings WHERE earnings.borrowerid = ?

* Total revenue from one item

SELECT earnings.borrowerid, earnings.title, earnings.name, earnings.amount FROM earnings WHERE earnings.itemid = ?

* Queries:

1. These queries get all of the rows from their corresponding tables:

* Get all users from the **users table**
  + **SELECT \* FROM users**
* Get all items from the **available table**
  + **SELECT \* FROM available**
* Get all items from **rented table**
  + **SELECT \* FROM rented**
* Get all items from **processing table**
  + **SELECT \* FROM processing**
* Get all messages from **contact table**
  + **SELECT \* FROM contact**

2. These queries get the row equal to the entered values

* Get the row from the **users table** equal to the entered username
  + **SELECT \* FROM users WHERE username = ?**
* Get the row from the **users table** equal to the entered username and password
  + **SELECT \* FROM users WHERE username = ? AND password = ?**
* Get the row from the **admins table** equal to the entered username
  + **SELECT \* FROM admins WHERE username = ?**
* Get the row from the admins table equal to the entered username and password
  + **SELECT \* FROM admins WHERE username = ? AND password = ?**
* Get all the items from **rented table** equal to the current logged in users id
  + **SELECT \* FROM rented WHERE borrowerid = ?**
* Get all the fees from the balance table equal to the current logged in user id
  + **SELECT \* FROM balance WHERE borrowerid = ?**

3. These queries add a row to their corresponding tables

* Add a row to the **users table** equal to the entered values
  + **INSERT INTO users (userid, firstname, lastname, status, username, password, datesignedup, fees) VALUES (?, ?, ?, ?, ?, ?, ?, ?)**
* Add a row to the **admins table** equal to the entered values
  + **INSERT INTO admins (adminid, firstname, lastname, username, password) VALUES (?, ?, ?, ?, ?)**
* Add a row to the **available table** equal to the entered values
  + **INSERT INTO available (itemid, title, author, cover, type) VALUES (?, ?, ?, ?, ?)**
* Add a row to the **rented table** equal to the entered values
  + **INSERT INTO rented (rentedid, duedatems, borrowerid, name, itemid, title, author, cover, type) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)**
* Add a row to the **processing table** equal to the entered values
  + **INSERT INTO processing (processingid, duedatems, borrowerid, name, itemid, title, author, cover, type) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)**
* Add a row to the **balance table** equal to the entered values
  + **INSERT INTO balance (feeid, borrowerid, name, itemid, title, type, lateamount, damagedamount, productid) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)**
* Add a row to the **messages table** equal to the entered values
  + **INSERT INTO contact (contactid, name, email, message) VALUES (?, ?, ?, ?)**

4. These queries delete a row from their corresponding table

* Delete a row from the users table equal to the userid
  + **DELETE FROM users WHERE userid = ?**
* Delete a row from the available table equal to the itemid
  + **DELETE FROM rented WHERE rentedid = ?**
* Delete a row from the processing table equal to the processing id
  + **DELETE FROM processing WHERE processingid = ?**
* Delete a row from the balance table equal to the fee id
  + **DELETE FROM balance WHERE feeid = ?**
* Delete a row from messages table equal to the contact id
  + **DELETE FROM contact WHERE contactid = ?**

5. Update a row for the corresponding table equal to the entered values

* Update a row in the users table
  + **"UPDATE users SET `firstname` = ?, `lastname` = ?, `status` = ?, `username` = ?, `password` = ?, `datesignedup` = ?, `fees` = ? WHERE `userid` = ?**
* Update a row in the available table
  + **UPDATE available SET `title` = ?, `author` = ?, `cover` = ?, `type` = ? WHERE `itemid**` = ?