Revision History

Version	Date	Comment
1.0.0	10/31/2019	Initial Release
1.0.1	11/14/2019	“ALL” option for i_status in Screen 13: Admin Filter User; Modify Screen 18: Manager Filter Theater 2d

CS 4400 Phase 3 Instruction

Fall 2019

I. Overview

In this phase, each team will implement a list of stored procedures for the Atlanta Movie database. The listed store procedures are aimed to guide you through all the required functionalities in this project. Feel free to also use them in your phase 4 if applicable. Each team is required to turn in a zip archive by the due date. Make sure to review the check list before submission.

II. Check List

1. Start with an empty database with name **"Team*[Team#]*"** (e.g. "Team1")
2. Load **ALL** initial data into your database (see InitialData.xlsx on Canvas) and **DO NOT** include additional data in your submission
3. Load provided stored procedures into your database ("CS4400_Fall2019_Phase3_Examples.sql")
4. Implement **ALL** required stored procedures
    a. Procedure's name **MUST** be exact
    b. Input **MUST** be in the exact order
        i. However, some input values can be optional
        ii. "**ALL**" is a reserved input keyword for any applicable dropdown fields (see UI in project description). For example, for admin_filter_user, i_status can have value "ALL" and the store procedure returns users with any appropriate status
        iii. Hint: Some domains for stored procedure have default value "" (e.g. VARCHAR). Other domains have default value NULL. These default values are important for handling the optional inputs. In this project, treat the default value for VARCHAR and CHAR as "" and all the other domains as NULL
    c. Output table's name **MUST** be exact (if applicable)
    d. Output table's attribute naming **MUST** be exact (if applicable)

        i. Use your judgement from reading the project description to decide the domain for output attributes
    e. Output table's attribute ordering **MUST** be exact (if applicable)
5. One person in each team submit a zip archive containing the following files:
    a. **Team[*TeamNumber*]_readme.txt** (e.g. Team1_readme.txt)
        i. Include team members' names and GT account usernames
        ii. Include any notes you would like us to know for grading
    b. **Team[*TeamNumber*]_DB.sql** (e.g. Team1_DB.sql)
        i. This is your database dump file containing initial data and all required modifications (stored procedures, views, etc.)
        ii. **DO NOT** modify initial data in your submission
    c. **Team[*TeamNumber*]_SP.sql** (e.g. Team1_SP.sql)
        i. This is the file with all your required stored procedures **AND** the provided stored procedures

III. Required Stored Procedures

Screen 1: User login

1. Procedure: user_login
2. Input: i_username, i_password
    a. i_username, i_password: VARCHAR
3. Output: UserLogin (username, status, isCustomer, isAdmin, isManager)
    a. isCustomer: INT (0: is not customer, 1: is customer)
    b. isAdmin: INT (0: is not admin, 1: is admin)
    c. isManager: INT (0: is not manager, 1: is manager)

[PROVIDED] Screen 3: User register

1. Procedure: user_register
2. Input: i_username, i_password, i_firstname, i_lastname
    a. i_username, i_password, i_firstname, i_lastname: VARCHAR

Screen 4: Customer-Only register

1. Procedure: customer_only_register
2. Input: i_username, i_password, i_firstname, i_lastname
    a. i_username, i_password, i_firstname, i_lastname: VARCHAR

Screen 4: Customer add credit card

1. Procedure name: customer_add_creditcard

2. Input: i_username, i_creditCardNum
    a. i_username: VARCHAR
    b. i_creditCardNum: CHAR

Screen 5: Manager-Only register

1. Procedure name: manager_only_register
2. Input: i_username, i_password, i_firstname, i_lastname, i_comName, i_empStreet, i_empCity, i_empState, i_empZipcode
    a. i_username, i_password, i_firstname, i_lastname, i_comName, i_empStreet, i_empCity: VARCHAR
    b. i_empState, i_empZipcode: CHAR

Screen 6: Manager-Customer register

1. Procedure: manager_customer_register
2. Input: i_username, i_password, i_firstname, i_lastname, i_comName, i_empStreet, i_empCity, i_empState, i_empZipcode
    a. i_username, i_password, i_firstname, i_lastname, i_comName, i_empStreet, i_empCity: VARCHAR
    b. i_empState, i_empZipcode: CHAR

Screen 6: Manager-Customer add credit card

1. Procedure: manager_customer_add_creditcard
2. Input: i_username, i_creditCardNum
    a. i_username: VARCHAR
    b. i_creditCardNum: CHAR

Screen 13: Admin approve user

1. Procedure: admin_approve_user
2. Input: i_username
    a. i_username: VARCHAR

Screen 13: Admin decline user

1. Procedure: admin_decline_user
2. Input: i_username
    a. i_username: VARCHAR

Screen 13: Admin filter user

1. Procedure: admin_filter_user
2. Input: i_username, i_status, i_sortBy, i_sortDirection
    a. i_sortBy has to be one of the following (**EXACT** naming): username, creditCardCount, userType, status. Default sort by username
    b. i_sortDirection is either ASC or DESC. Default is DESC.
    c. i_username: VARCHAR
    d. i_status can be one of the following (**EXACT** naming): Approved, Pending, Declined, ALL
3. Output: AdFilterUser (username, creditCardCount, userType, status)
    a. userType has the following values: User, Customer, Admin, Manager, CustomerManager, CustomerAdmin

Screen 14: Admin filter company

1. Procedure: admin_filter_company
2. Input: i_comName, i_minCity, i_maxCity, i_minTheater, i_maxTheater, i_minEmployee, i_maxEmployee, i_sortBy, i_sortDirection
    a. i_sortBy has to be one of the following (**EXACT** naming): comName, numCityCover, numTheater, numEmployee. Default sort by comName
    b. i_sortDirection is either ASC or DESC. Default is DESC.
    c. i_comName: VARCHAR
    d. i_minCity, i_maxCity, i_minTheater, i_maxTheater, i_minEmployee, i_maxEmployee: INT
3. Output: AdFilterCom (comName, numCityCover, numTheater, numEmployee)

Screen 15: Admin create theater

1. Procedure: admin_create_theater
2. Input: i_thName, i_comName, i_thStreet, i_thCity, i_thState, i_thZipcode, i_capacity, i_managerUsername
    a. i_thName:, i_comName, i_thStreet, i_thCity, i_managerUsername: VARCHAR
    b. i_thState, i_thZipcode: CHAR
    c. i_capacity: INT

Screen 16: Admin view company detail (Employee)

1. Procedure: admin_view_comDetail_emp
2. Input: i_comName
    a. i_comName: VARCHAR

3. Output: AdComDetailEmp (empFirstname, empLastname)

Screen 16: Admin view company detail (Theater)

1. Procedure: admin_view_comDetail_th
2. Input: i_comName
    a. i_comName: VARCHAR
3. Output: AdComDetailTh (thName, thManagerUsername, thCity, thState, thCapacity)

Screen 17: Admin create movie

1. Procedure: admin_create_mov
2. Input: i_movName, i_movDuration, i_movReleaseDate
    a. i_movName: VARCHAR
    b. i_movDuration: INT
    c. i_movReleaseDate: DATE

Screen 18: Manager filter theater

1. Procedure: manager_filter_th
2. Input: i_manUsername, i_movName, i_minMovDuration, i_maxMovDuration, i_minMovReleaseDate, i_maxMovReleaseDate, i_minMovPlayDate, i_maxMovPlayDate, i_includeNotPlayed
    a. i_manUsername, i_movName: VARCHAR
    b. i_minMovDuration, i_maxMovDuration: INT
    c. i_minMovReleaseDate, i_maxMovReleaseDate, i_minMovPlayDate, i_maxMovPlayDate: DATE
    d. i_includedNotPlay: BOOLEAN (NULL or FALSE: all movies, TRUE: only include not played movies)
3. Output: ManFilterTh (movName, movDuration, movReleaseDate, movPlayDate)

Screen 19: Manager schedule movie

1. Procedure: manager_schedule_mov
2. Input: i_manUsername, i_movName, i_movReleaseDate, i_movPlayDate
    a. i_manUsername, i_movName: VARCHAR
    b. i_movReleaseDate, i_movPlayDate: Date

Screen 20: Customer filter movie

1. Procedure: customer_filter_mov
2. Input: i_movName, i_comName, i_city, i_state, i_minMovPlayDate, i_maxMovPlayDate

    a. i_movName, i_comName, i_city, i_state: VARCHAR
    b. i_minMovPlayDate, i_maxMovPlayDate: Date
3. Output: CosFilterMovie (movName, thName, thStreet, thCity, thState, thZipcode, comName, movPlayDate, movReleaseDate)

Screen 20: Customer view movie

1. Procedure: customer_view_mov
2. Input: i_creditCardNum, i_movName, i_movReleaseDate, i_thName, i_comName, i_movPlayDate
    a. i_creditCardNum: CHAR
    b. i_movName, i_thName, i_comName: VARCHAR
    c. i_movReleaseDate, i_movPlayDate: Date

Screen 21: Customer view history

1. Procedure: customer_view_history
2. Input: i_cusUsername
    a. i_cusUsername: VARCHAR
3. Output: CosViewHistory (movName, thName, comName, creditCardNum, movPlayDate)

[PROVIDED] Screen 22: User filter theater

1. Procedure: user_filter_th
2. Input: i_thName, i_comName, i_city, i_state
    a. i_thName, i_comName, i_city, i_state: VARCHAR
3. Output: UserFilterTh (thName, thStreet, thCity, thState, thZipcode, comName)

[PROVIDED] Screen 22: User visit theater

1. Procedure: user_visit_th
2. Input: i_thName, i_comName, i_visitDate, i_username
    a. i_thName, i_comName, i_city, i_username: VARCHAR
    b. i_visitDate: DATE

[PROVIDED] Screen 23: User filter visit history

1. Procedure: user_filter_visitHistory
2. Input: i_username, i_minVisitDate, i_maxVisitDate
    a. i_username: VARCHAR
    b. i_minVisitDate, i_maxVisitDate: DATE

3. Output: UserVisitHistory (thName, thStreet, thCity, thState, thZipcode, comName, visitDate)