**Database Application Development**

**Spring 2019**

**Online Movie Ticket Purchase System**

**Overview**

You will form groups of four to five people in this project. It is recommended that each group consists of at least four people given the amount of work. Please read the whole document carefully before starting your project.

Your assignment is to design an online movie ticket purchase system. You can get yourself familiar with the systems by visiting websites such as fandango.com, movietickets.com. You will design the database, insert some sample data, and implement a set of required features. Each feature will be implemented as one or more Oracle PL/SQL procedures/functions. You do **NOT** need to write a graphic user interface.

**Assumptions:**

You can make the following assumptions in this project.

1. The system needs to store data about users. Each user has user id, name of user, email, address, and password.
2. The system needs to store data about cinemas. Each cinema belongs to a company (e.g., AMC, Regal). Each cinema has a cinema ID, cinema name, address, phone, and has one or more auditoriums.
3. Each company (like AMC) has a company ID, company name.
4. Each company has discounts for child and senior. The discount rate may vary. E.g., company A may have a rate of 0.8 (i.e., 20% off) for child and 0.9 (i.e., 10% off) for senior.
5. Each cinema auditorium has an auditorium ID (only unique within the cinema), cinema ID, auditorium type (regular, 3D, or IMAX), and capacity (#of seats). 3D auditorium can show both regular and 3D movies. Regular auditorium can only show regular movies. IMAX auditorium can only show IMAX movies.
6. Each IMAX cinema auditorium has a seat map with a varchar type seat ID. E.g., '9A' means a seat in row 9. The combination of cinema ID, cinema auditorium ID, and seat ID will uniquely identify a seat in the whole system.
7. The system stores information about movies, including movie ID, title, release\_date, rating (e.g., 'PG 13'), average imdb review score, length of the movie (in hours and minutes).
8. The system stores show time of a movie at a cinema, including a show time ID, movie id, auditorium id, cinema id, start time of the show, format (regular, 3D, or IMAX), whether the show is full (no more tickets can be purchased), whether the show allows selection of seat (typically for IMAX format), base price (for adults).
9. For shows that allow selection of seats (typically IMAX movies), the system keeps track of whether a seat in the seat map table is still available for that show time.
10. The system stores information about purchase transactions, including a transaction ID, user ID, show ID, purchase time, quantity (#of tickets), status (1 means paid, 0 not paid, 2 canceled), and total amount of the transaction, which includes sum of price of each ticket in the transaction plus a $1.50 fee per ticket and a 6% sales tax.
11. The system stores information about tickets. Each ticket is associated with a purchase transaction. Each ticket has a unique ticket ID, a ticket type (adult, child, or senior), price for that ticket (equals base price \* applicable discount rate depending on the type of ticket), optional cinema ID, auditorium ID, and seat ID (identifying the assigned seat), and whether the ticket has been issued (after the transaction has been paid in full).
12. The system stores information about payment. Each payment has a unique payment ID, ID of the transaction the payment is applied to, a payment type (1 means payment, 2 means refund), payment method (1 credit card/gift card, 2 debit card, 3 paypal), last 4 digits of payment card, amount of payment, and payment time. It is possible to make multiple payment to a transaction (e.g., using a gift card plus a credit card) as long as the total payment reaches the total amount of the transaction.
13. The system stores reviews about cinema auditoriums, including review ID, user ID, cinema ID, a numerical score from 1 to 5, and a textual content and review time.

**Features: (those with \*\*\* are more difficult ones and may require at least twice or more time to finish than other features. Be aware of this when you assign features)**

**Please check special cases and normally if an error message is printed you should end the procedure.**

**Features for user management:**

1. Registration: a user needs to provide name, email, address, and password. The procedure should check whether the email already exists in the users table. If so, please print a message saying the user already exists. Otherwise insert the user into users table and print the new user ID.
2. Login: Allow a user to log on by providing email and password. Please check whether email exists and password matches. If not, please print a message to indicate the error. Otherwise print a message to indicate user has logged on. The procedure should return a value 1 for success login and 0 for unsuccessful log in.

**Features for review and statistics:**

1. Add a review for a cinema. Input includes a user ID, cinema ID, review score, and content of review.
   1. First check whether user ID, cinema ID are valid and review score is from 1 to 5. If not print an error message.
   2. Please also check if the same user has left a review for the same cinema within 30 days. If so print an error message that you cannot review the same place twice within 30 days. Otherwise insert a new review with the review time is current time.
2. \*\*\* Compute statistics for movies: given a time period (start and end dates), for each movie showing in this period, compute total ticket sales as well as occupancy rate.
   1. Total ticket sales is the sum of total amount of paid transactions for that movie
   2. Occupancy rate is total number of tickets sold for that movie divided by maximal possible number of tickets (maximal possible #of tickets = capacity of the cinema auditorium showing the movie\*#of shows in that auditorium and then sum over all such cinema auditoriums). E.g., suppose a movie is shown in a cinema auditorium with capacity 100 for 10 times and a cinema auditorium with capacity of 50 for 10 times, but only sold 300 tickets, the occupancy rate is 300/(100\*10+50\*10)=300/1500=0.2.
   3. Please order results by descending order.

Hint: 1) it is easier to compute total tickets sold for a movie and maximal possible number of tickets separately. 2) Even if a show has sold zero tickets it still needs to be counted in the maximal possible number of tickets. 3) You can use subquery in from clause.

1. Compute statistics for each cinema. Input is a date range. Compute statistics for each cinema in this period, including total tickets sold, total sales (in dollars), occupancy rate (computed as total tickets sold divided by maximal possible tickets sold) and average review score during the period.
2. \*\*\* Computer statistics for a given user. Input includes user id and a date range. First check whether the user id is valid. If not print an error message. Otherwise compute for that user the following statistics:
   1. #of paid transactions,
   2. #of tickets bought, total money spent
   3. #of canceled transactions,
   4. the most frequently visited cinema (with the maximal number of paid transactions).

Hint: to compute most frequently visited cinema, you can use subquery in having clause.

1. \*\*\* Identify suspicions reviews. Suspicious reviews are those 1) very short (no more than x characters long where x is input and with extreme scores (1 or 5); or 2) the reviewer has never purchased any tickets for shows at the reviewed cinema. For example, if a reviewer A has left a review for cinema X. But only users B, C, and D have purchased tickets at X (they have transactions with shows at X), then A's review for X is fake. Please print out review id and reason for fake review.

Hint: for the second condition, use subquery.

GOOD LUCK!