

## Individual Project #5

### INSTRUCTIONS

Please read the description and the requirements of Deliverable #5 carefully before responding. If you have any questions or clarifications, feel free to contact me via email ([aditimukherjee@ufl.edu](mailto:aditimukherjee@ufl.edu)), during my office hours, or by appointment

All submissions should be individual work only. Do not discuss your answers with your class mates or group members. Plagiarism will not be tolerated.

**Due Date:** Before class Apr 14, 2016 on Canvas.

No late submissions will be accepted. If you have trouble submitting assignments to the e-learning system, please contact me before the submission deadline.

**Page Limit:** There is no fixed page limit for this deliverable.

**Format:** You will submit 7 SQL statements and the results of running the queries in the attached database. Copy and paste all the statements and the diagram into a Word document that you can save and submit as a PDF. Please note the number of the problem statement clearly above each query. See page 3 of this document for an example of how to present the statements. The name of your file should be <LastName>\_<FirstName>\_IP5.pdf. Submissions that do not follow the formatting guidelines will not be graded.

### DELIVERABLE #5 – MULTIPLE TABLE QUERIES

For this deliverable, you will be writing queries to use the data in the TheHipp database.

**STEP 1:** Download the SQL script (TheHipp.sql) file attached to assignment IP#4. Run the scripts into MySQL as follows:

- Open the SQL script file using CTRL+SHIFT+O.
- Run the entire script using CTRL+SHIFT+Enter.

**STEP 2:** For each of the problem statements listed in below, please do the following:

- a) Write the SQL query that will achieve the desired result
- b) Run the query in the MySQL database.
- c) Copy and paste the results of the query below each SQL statement using the format on page 3 of this assignment. If your results have more than 10 rows only copy the **first 10 rows**.

Each query must satisfy the following criteria:

- a) They must not include any extra/unnecessary tables.
- b) The results must not include any extraneous columns.
- c) All column headings should be meaningful names. Please do not have columns with headings that contain expressions or aggregate functions.
- d) They must not use views unless specified in the instructions.

## Individual Project #5

### PROBLEM STATEMENTS

1. List the names of the customers in the database who have not bought a single ticket. You must submit two queries:
  - (A) Write this query using a sub-query
  - (B) Write this query using a join
 The output section of MySQL Workbench shows you the Duration for each query. Which query took longer to execute?
2. List the venue name, capacity, total revenue and the number of tickets sold for each venue. Write an efficient query that uses the minimum number of tables required.
3. List the name, revenue (total price paid for all the tickets), the promotion cost, the screen cost and the total profit (revenue - promotion cost - screening cost) for the 5 most profitable movies.
4. List the name, description and Venue ID for each event. In addition, if the event is a movie, list the genre. If the event is a play - list the name of the author. Your table should only have 4 columns. The fourth column should be called 'Genre/Author'. The results will have the same number of rows as the EventTable.
5. Create a View called TicketDetails that includes all the columns from the Ticket and the Event table for each ticket.
  - (A) Use the TicketDetails view in a query that returns the following information for each event: EventCode, event name, the number of shows, the screen cost, the total screening cost (number of shows \* screening cost), promotion cost, production cost, total cost (total screen cost + promotion cost + production cost), the expected revenue (use the BaseTicketPrice), the actual revenue, the total discounts given (expected revenue - actual revenue) and the profit (all costs - actual revenue)
  - (B) Use the Customer table and the TicketDetails to list the HippCode, name and email of each customer, along with number of tickets they have bought for theater (plays) and the number of tickets they have bought for movies. Sort your results in the descending order of the total number of tickets. Your results may look like the following but should have 199 rows.

HippCode	LastName	FirstName	Email	TheaterTickets	MovieTickets
29249	Yang	Shengda	shey@yahoo.com	41	4
91179	Burnell	Dana	danburnell@gmail.com	37	4
19670	Hill	Annie	anniehil@outlook.com	36	4

## Individual Project #5 Template

For each query, you must present your results using the following template.

### SELECT statement

```
SELECT * FROM Customer;
```

	CustomerID	LastName	FirstName
▶	123456	Pond	Amy
★	NULL	NULL	NULL