

## BUS 440 Database Management

### Project 1: Uptown Rentals Musical Instrument Rental System

#### Use Case:

As you learned from Assignment 1, Uptown Rentals is a new startup business that rents musical instruments to individuals. You have already designed a relational database management system for this company, which categorizes musical instruments by type (e.g., Strings, Woodwinds, Brass, etc.) and rental tier (e.g., Basic, Premium, etc.).

You also learned that the store has multiple employees with a primary role (e.g., cashier, consultant, admin clerk, manager) and email address. Store staff manage the rentals and track the returns. Customers must register with the store, providing their full name, age, address, and contact information. Each customer can rent more than one instrument at a time. Each rental includes the serial number of the rented instrument, daily rental fee, rental date, due date, and return date. Late returns incur a daily overdue fee based on the instrument's rental tier and the days overdue of the rented instrument. Fines are calculated per day late. Customers cannot rent new instruments until outstanding fines are paid. Uptown Rentals also tracks the condition of each instrument it rents. If the instrument is damaged upon return, or if the instrument needs periodic maintenance, the employee initiates a repair (maintenance), and the business tracks the repair cost and the maintenance date.

The business has been capturing data related to its inventory and rentals on spreadsheets but wants to migrate to a database solution. Figure 1 shows a sample of some of the current data:

Figure 1

Serial Num	Customer Name	Rental Date	Instrument Type	Rental Tier	Contact Email	Staff Name	Return Date	Due Date	Daily Rental Fee	Daily Overdue Fee
12878	Joseph Dow	12/4/2024	Flute	Basic	<a href="mailto:jdow@gmail.com">jdow@gmail.com</a>	Liz Conners	12/19/2024	12/20/2024	35.00	0
76887	Ric Martin	1/5/2015	Trumpet	Premium	<a href="mailto:Rm5@nc.rr.com">Rm5@nc.rr.com</a>	Liz Conners	1/12/2025	1/10/2015	40.00	5.00
76657	Ric Martin	1/5/2015	Bass Guitar	Premium	<a href="mailto:Rm5@nc.rr.com">Rm5@nc.rr.com</a>	Liz Conners	1/12/2025	1/10/2015	43.00	5.00
98223	Lauren Cox	1/5/2015	Flute	Basic	<a href="mailto:Lc49@gmail.com">Lc49@gmail.com</a>	Tom Lindel	1/12/2025	1/12/2015	35.00	0
12878	Luke Diago	12/20/2024	Flute	Basic	<a href="mailto:jdow@gmail.com">jdow@gmail.com</a>	Liz Conners	12/29/2024	12/30/2024	35.00	0
98223	Sue Mann	1/13/2015	Flute	Basic	<a href="mailto:sm3@gmail.com">sm3@gmail.com</a>	Tom Lindel	1/22/2025	1/22/2015	35.00	0
98223	Lauren Cox	1/25/2015	Flute	Basic	<a href="mailto:Lc49@gmail.com">Lc49@gmail.com</a>	Tom Lindel	1/30/2025	1/30/2015	35.00	0

#### Instructions:

From Assignment 1, take the feedback and modify your design. Then, build the schema and database from your modified design:

1. Build your EER model in MySQL Workbench and name it **uptown**. Include a screenshot of the model here.
2. Synchronize the EER to the build. Note that this process will create an empty schema.
3. Verify the build.
  - a. Verify all tables and their attributes and PK. For example, after synchronization, you could create a **rental** table that would include attributes such as the *serialNum*, *rentalDate*, *customerID*, etc.
  - b. Verify the FKs representing each relationship in your EER.

### Populate the tables with data:

4. Enter the data into the appropriate tables according to your data model design (at least 7 rows per table). (Note: Everyone should type the data from Figure 1 into the appropriate tables. However, the remaining data should be your own and different from others – each person should derive their own hypothetical data.)
  - a. Populate each table of the database by typing the data directly into each table, by importing it, or by using SQL code to insert it
  - b. Populate the parent tables first, then the child tables.

### Create SQL programs to answer questions of the data:

5. Open a new query tab and save it with the name, **uptownRentalsxy.sql** (where x is your first name and y is your last name).
6. Add a comment to the top of your script with your name, the purpose of the script, and the current date.
7. Type the “*use*” statement for uptown and run it: **use uptown;**
8. Create and run the following SQL programs, each with a preceding comment stating the query in English. Be sure to validate your results using an alternate method. **Screenshot your code and answer in this document for each of the following.**
  - a. What is the list of all instrument rentals in inventory? (Show the list displayed in Figure 1, along with any other rentals in your database.)
  - b. What are the youngest and oldest customers of Uptown Rentals? Write one SQL program to display both.
  - c. List the aggregated (summed) rental amounts per customer. Sequence the result to show the customer with the highest rental amount first.
  - d. Which customer has the most rentals (the highest count) across all time?
  - e. Which customer had the most rentals in January 2025, and what was their average rental total per rental?
  - f. Which staff member (name) is associated with the most rentals in January 2025?
  - g. For each customer that has an overdue rental, how many days have passed since the rental was due?
  - h. What is the total rental amount by Rental tier?
  - i. Who are the top three store staff members in terms of total rental amounts?
  - j. What is the total rental amount by instrument type, where the instrument type is *Flute* or *Bass Guitar*?
  - k. What is the name of any customer who has two or more overdue rentals?
  - l. List all of the instruments in inventory in 2025 that were damaged upon return or needed maintenance. Include the employee that handled the rental, the repair cost, and the maintenance date.
  - m. Create a query of your choice that includes a subquery.
  - n. Add another meaningful query of your choice. For example, you could create a query that answers the following question: What is the name of any customer who has rented 3 or more *Woodwind* instruments?

**Submit your project:** Two files are required:

1. This exact .docx file with all query answers and screenshots;
2. Screenshots of the data in each table; and
3. The .sql script.