A02: Database Application

By: Nikki Burr, Nikki Buzianis, and Mujib Rahman

Design:

Create a SQL database with the following requirements:

Notice that the requirements are minimum requirements. Teams may add additional tables and/or functionality if they want.

- It includes a table called **Customer** with the following columns:
    - CustomerID (a unique whole number)
    - FirstName
    - LastName
    - Phone
    - ArtistID
    - BillingID
- It includes a table called **Billing** with the following columns:
    - BillingID
    - ArtistID
    - CustomerID
    - Cost
    - ApptDate
- **Artist**
    - ArtistID
    - ArtistName
    - The tale **Customers** needs to include at least 25 Customers (rows)
    - The tale **Billing** needs to include at least 5 teams (rows)

The Database Application needs to provide the following functionality:

1. Allow the user to do the following:
    - add a customer
    - remove a customer
    - list all customers
    - edit/update customer info
2. Allow the user to change a customer record (e.g. to allow a customer to change the artist and/or to change the artist)
   Note: the unique ID (primary key) should not be changeable.

3. Allow the users to query the data in at least three different ways.
   The queries should provide meaningful information in the Basketball domain. I list a few examples that came up in class. Feel free though to come up with your own queries and to make it a unique and interesting application.
   i. Monthly revenue for artist (filtering)
   ii. Sort clients by name/date (sorting)
   iii. Customers by artist (filtering & sorting)
   iv. Sort by price (sorting)

   Examples from Margaret:

   - *One query should include filtering (select only some of the rows)*
     e.g. list players from a given team / position / city
     e.g. list all point-guards of a given team (more complex but it includes filtering)
   - *One query should include sorting*
   - *One query should provide data from both tables*
     e.g. list names of the players together with the names of the teams they belong to
     e.g. list the names of the teams, the corresponding cities, and the number of players on their team (provided the number of players is not stored in a team column)

4. Additional functionality as needed

GUI:

Strive to write a user interface that is intuitive and easy to use.

Can we choose our own project? Yes. Here are two options:

- Change the theme only
  In this case, you would follow the same instructions but with different database tables. In this case, the database tables would need the same amount of columns or optionally more.
  E.g. Instead of players and teams, you could have a library theme with books and patrons, or a restaurant reservation theme with tables and guests, or an aviation theme with flights and airports, etc.
- Come up with your own database application
  In this case, you and your partner can come up with your o plan of similar complexity (or more)

However, you'd need to confirm it with me before submitting the design (e.g. on Monday after class or Wednesday before class)

In either case, it needs to be a new project that is started from scratch.

To help the teams stay on task there will be three submissions for this assignment

- **Milestone 1 - Design**
  At this point, the team needs to have completed the design phase and the database. Submit a video recording that is 2 - 4 minutes long and includes the following:
  1. **The design:**
     - Show a diagram of the database tables and point out the columns you added
     - Describe three queries you plan to include in your application - one for each of the requirements above.
     - Identify which query requires data from both tables and describe the columns that are accessed.
     - Mention whether you plan to include additional queries. There is no need to provide a complete list at this point.
  2. **The database:**
     - The database tables need to be implemented
     - Each database table needs to be filled with at least 3 records
     - Print the content of each database table
     - Show both the database code and the output of the print statements.
  3. Clearly describe the teamwork with your partners and the pebble distribution as described in the first team assignment.
     This part needs to be discussed with the team before it is recorded.
     - Include how often you met.
     - Identify which parts were created together as a team, and which parts were created by individuals. In the latter case, describe who worked on those parts.
     - Describe the quality of the teamwork. (E.g., Were the members responsive? Did the team members teach each other with respect? etc.)
     - Include the pebble distribution. If the distribution is uneven, describe the reason for the difference in pebbles.