Team 58 Project Proposal and Outline

Team 58: Alexander Wilson, Ethan Ruark
Project Name: Databar Events Planner

Overview: Databar is a bar with a large event space that can be rented out by its customers. In 2021, Databar hosted over 200 events. As their venue is becoming increasingly popular they are realizing the need to establish a database to help them organize as they grow. This database will be used to plan upcoming events, including the staffing and inventory needs associated with each event. Databar has the capacity to hold only one event per night. Each event needs to be appropriately staffed, including one worker with each job title. These events will include a drink special, which will be selected from the Drinks list. The drink special is available at a discounted price for the duration of the event. Drinks can have a maximum of five ingredients, but must have at least one (single liquor specials are allowed). Using this database, Databar will have confidence that it will never overbook its event space, events are fully staffed, and the appropriate ingredients for the nightly drink special are available.

Outline:

- **Employees**: The purpose of this entity is to track the workers at Databar. It will be used by management to staff upcoming events.
    - employee_ID: Int, auto_increment, unique, not NULL, PK
    - first_name: VarChar, not NULL
    - last_name: VarChar, not NULL
    - telephone: VarChar, not NULL
    - job_title: Varchar, not NULL (the following job title options will be implemented via dropdown menu when inputting new employees)
        - Bartender
        - Barback
        - Security
        - Manager
    - hourly_rate: decimal(18, 2), Not NULL
    - start_date: date, not NULL
    - Relationship: Worker entity will have a Many to one relationship with events entity. Each event will require workers with different job titles to operate properly. There must be one worker with each job title per event.. The employee_ID will be the FK inside of the event entity.

- **Events**: The purpose of this entity is to track upcoming events at Databar. It will have attributes such as employee IDs for the staff that will be running the event and the drink special for the night.
    - event_iD: int, auto_increment, unique, not NULL, PK
    - event_name: String, not NULL
    - event_date: date, not NULL

    - bartender: Int, FK (employee_ID), not NULL, exactly one
    - barback: Int, FK (employee_ID), not NULL, exactly one
    - security: Int: FK (employee_ID), not NULL, exactly one
    - manager: Int, FK (employee_ID), not NULL, exactly one
    - guest_count: int, not NULL, exactly one
    - drink_name: varchar, not NULL, FK (drink_name)
    - Relationship: The events entity will have a one to many relationship with the employees entity. The events entity will also have a one to one relationship with the drink entity.

- **Drinks** The purpose of this entity is to track event drink special options and their recipes.
    - drink_name : varchar, unique, not NULL, PK
    - ingredient_1: int, not NULL, FK (product_ID)
    - ingredient_2: int, can be NULL, FK (product_ID)
    - ingredient_3: int, can be NULL, FK (product_ID)
    - ingredient_4: int, can be NULL, FK (product_ID)
    - price: decimal(18,2), not NULL
    - Relationship: a 1:M relationship between Drinks and Events is implemented with drink_name as a FK inside of Events. A M:M relationship also exists between Drinks and Inventory, as implemented by the Inventory product_key utilized as a FK within Drinks.

- **Inventory**: The purpose of this entity is to track the ingredients inventory needed for event drink specials.
    - product_ID: int, auto-increment, unique, not NULL, PK
    - name: char, not NULL
    - category: char, not NULL (the following category choices will be implemented via dropdown menu when inputting new items)
        - Liquor
        - Beer
        - Wine
        - Mixer
        - Misc.
    - distributor: char, not NULL
    - bottle_cost: decimal(18, 2), can be NULL
    - Case_cost: decimal(18, 2), can be NULL
    - Relationship: a M:M relationship between Inventory is implemented with product_id as a FK inside of Drinks.