CSE3241 Project Checkpoint 1
Entities and Relationships

By Chalina Russell, Joshua Stark, Keming He, Shunian Xiang
Team SHRX, The Ohio State University
May 31st, 2023

In a NEATLY TYPED document each student, individually, needs to submit (Upload it to Carmen) the following document by the Checkpoint 1 due date, providing the following:
IMPORTANT NOTE: You need to have a clear description of the project requirements and the use cases before start working on checkpoint 1, otherwise you will have to re-do many parts of the project later.

1. **Based on the requirements given in the project proposal, list the entities to be modeled in this database. For each entity, provide a list of associated attributes (entities usually have multiple attributes).**

   Existing entities and their attributes for the project are listed below:

   - CUSTOMER
     - Username (primary key)
     - Password
     - Name
       - First_name
       - Last_name
     - Email
     - Creation_date
     - Billing_info
     - DOB

   - SUBSCRIPTION (superclass)
     - Start_date (partial key)
     - End_date (partial key)

- MONTHLY (subclass of SUBSCRIPTION)
    - Monthly_rate

- YEARLY (subclass of SUBSCRIPTION)
    - Yearly_rate

- RATING_AND_REVIEW
    - Date_and_time (partial key)
    - Rating_value
    - Review_description

- ANIME
    - Title (primary key)
    - Description
    - Genre
    - Price
    - Release_year

- EPISODE
    - Ep_no (partial key)
    - Title
    - Description
    - Duration
    - Release_date

- STUDIO
    - Name (primary key)
    - Description
    - Address
    - Website

2. **Propose at least two additional entities that it would be useful for this database to model beyond the scope of the project requirements. Provide a list of possible attributes for the additional entities and possible relationships they may have with each other and the rest of the entities in the database. Give a brief, one sentence rationale for why adding these entities would be interesting/useful to the stakeholders for this database project.**

   Additional entities and attributes and their justifications are listed below:

   - WATCHLIST
     - Attributes:
       - Name (partial key)
       - Description
       - Public_private_flag
     - Rationale:

       WATCHLIST is added as a weak entity to mimic its real-world counterpart - most streaming sites nowadays let customers bookmark shows they are watching for easy access. It's a weak entity due to being associated with a particular customer.
   - SALES_REP:
     - Attributes:
       - Username (primary key)
       - Password
       - Name
         - First_name
         - Last_name
       - DOB
       - Email
       - Creation_date
     - Rationale:

       SALES_REP is added as a strong entity for two reasons: it appeared multiple times in the use cases; and it has connections with both ANIME and STUDIO. Though SALES_REP has similar attributes compared to CUSTOMER, we chose not to generalize it in order to emphasize its unique role.

3. **Suppose we want to add a new instance to one of your entities in the database. How would we do that, given the entities and relationships you've outlined above? Given your above description, is it possible to add a new instance of an entity to your database without knowing the data of any other entity related to that entity? If not, revise your model to allow for an instance to be added as separate entities.**

    - **Case 1 of adding a new SALES_REP entity:**
      For strong entities, we can add a new instance without knowing the data of related entities. SLAES_REP is a strong entity, we can add a new instance of SALES_REP without knowing the data from other entities. We only need to create an instance of SALES_REP and provide attributes of SLAES_REP such as Username, Email, Password, etc.

    - **Case 2 of adding a new WATCHLIST entity:**
      For weak entities, it is not possible to add a new instance of an entity to the database without knowing the data of another entity related to that entity. WATCHLIST is a weak entity. To create a new instance of WATCHLIST, we would need to know the identifier of CUSTOMER that the new watchlist item is associated with.

      If we want to add a new instance for a weak entity like WATCHLIST without knowing the data of associated entities, one approach could be to create surrogate keys for WATHCHLIST, effectively turning it into a strong entity. A surrogate key is a unique identifier that is not derived from application data, and is often used in situations like this where we want to add flexibility to the data model.

      However, this approach does have some trade-offs. It can make data entry and management more flexible, but it can also make the data more complex, and it can introduce potential risk of inconsistent data if the relationships of entities are not mapped carefully. What's more, while it is technically possible to add a new instance without knowing the data of any related entities, we would still need to establish the relationships at some point in order for the data to be useful and meaningful.

4. **Determine at least three other informal update operations and describe what entities would need to have attributes altered and how they would need to be changed given your above descriptions. Include one example for each of the additional entities you proposed in question 3 above.**

   Additional informal update operations:

   A. Changing the password of a Sales rep: When a SALES_REP forgets his password and needs to reset the password, we need to update the Password attribute of the SALES_REP entity.

   B. Changing the description of a Customer’s Watchlist: When a CUSTOMER decides to change the description of one of their WATHCHLIST, we need to update the respective Description of the WATHCHILIST entity.

   C. Changing a Customer's Contact Information: If a customer updates their contact information (e.g., email, phone number), we would need to update the respective attribute(s) in the CUSTOMER entity.

5. **Provide an ER/EER diagram for your database. Make sure you include ALL entities and relationships in question 1, 2, above INCLUDING the entities for question 3 also, and remember that EVERY entity in your model needs to connect to some other(s) entity in the model via some kind of relationship (No entity should be isolated). Consider both sides of each relationship, do not forget about specifying the cardinality/participation for each one (1 to 1, 1 to N, N to 1, or N to N).**

   The Enhanced Entity Relationship Diagram for the project is posted below as Figure 1:

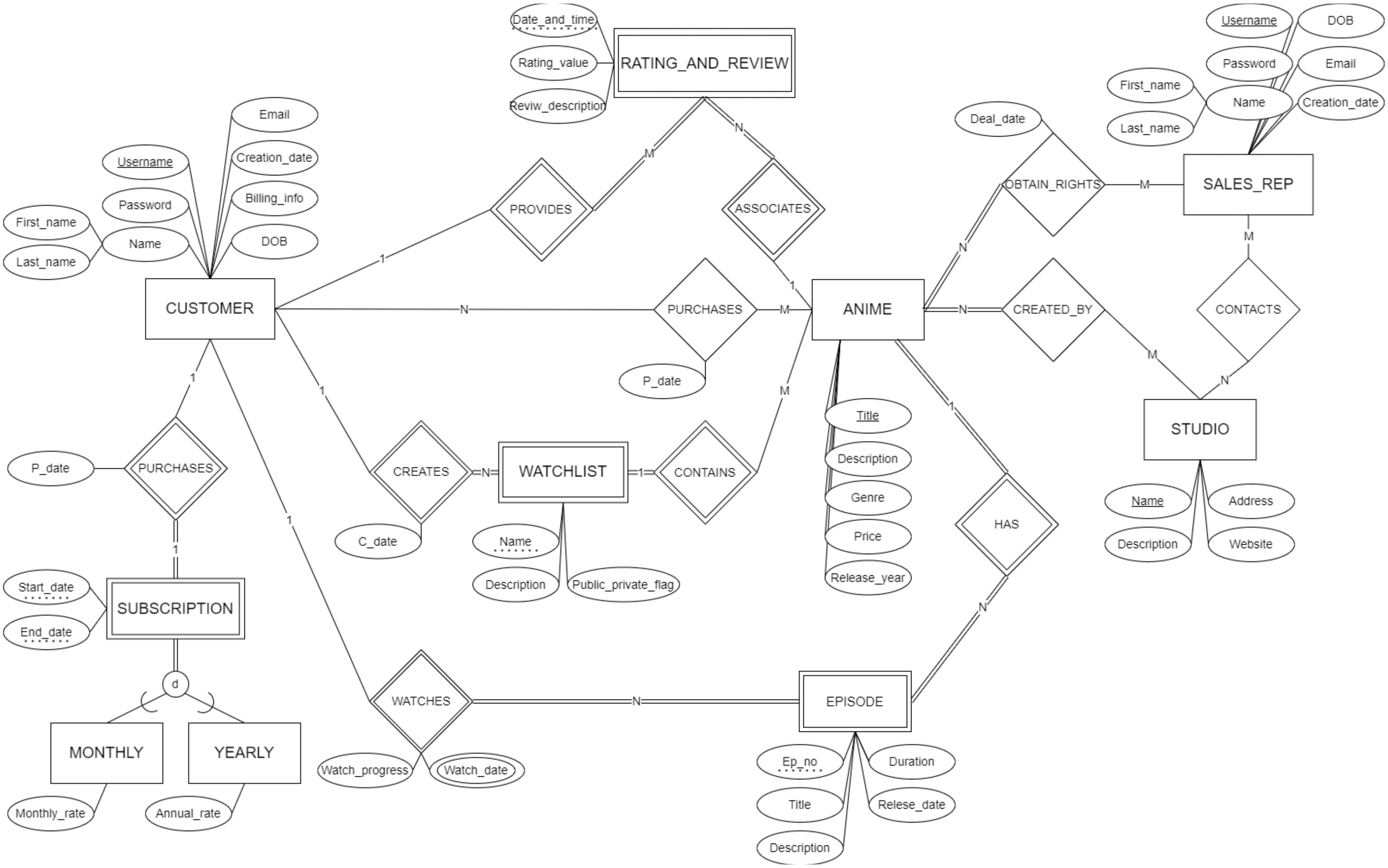

Figure 1 Team SHRX's EER Diagram

6. **Map your ER model to a relational schema. Indicate all primary and foreign keys with different indicators. Each foreign key should point to the attribute it references. Your schema should look similar to those in the power points.**

The Relational Schema for the project is posted below as Figure 2:

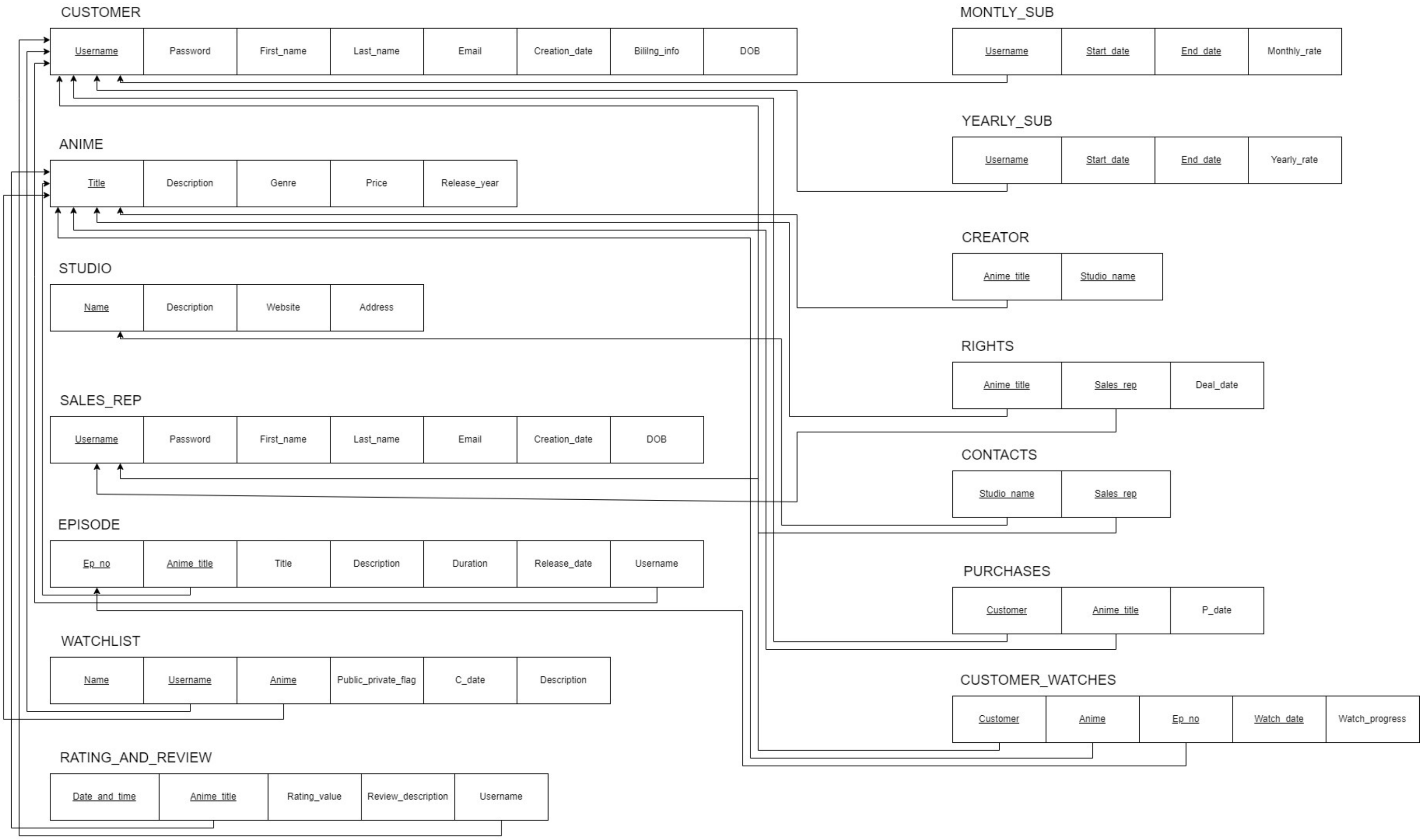

Figure 2 Team SHRX’s Relational Schema

7. Each team member, individually, needs to fill out the Peer-evaluation form provided and submit it to Carmen. Be sure to include yourself in the evaluation.

Please DO NOT zip the report file when you submit so that the grader can give you detailed feedback in Carmen.