1. Using either https://app.diagrams.net or actual pen and paper, take a stab at diagramming a database based on the ideas you previously brainstormed for your friend’s business:
   1. What entities do you include?
      1. Transactions
      2. Financial Data
      3. Marketing
      4. Customer Feedback
      5. Employee Information
   2. What attributes should be included in each entity?
      1. Transactions
         1. Timestamp
         2. Payment Method
         3. Amount of Sale
         4. Item Purchased
      2. Financial Data
         1. Date
         2. Store ID
         3. Vendor/Customer
         4. Total Revenue
         5. Expenses
         6. Profit Margin
         7. Payment
         8. Balance
      3. Marketing
         1. Consumer ID
         2. Website
         3. Visit Day/Time
         4. Social Media Engagement
         5. Rewards
         6. Order Placed
      4. Customer Feedback
         1. Feedback ID
         2. Customer Name
         3. Date
         4. Rating
         5. Comments/Open Feedback
      5. Employee Information
         1. Employee ID
         2. First Name
         3. Last Name
         4. Date of Birth
         5. Hire Date
         6. Department
         7. Email
   3. What are the relationships between entities? Based on what attributes?
      1. Looking closely at the entities and attributes that we created as a team, I’m finding that a lot of our data (especially **Financial**) can have more attributes created (or maybe a renamed entity?)
         1. Ex: Financial data doesn’t just include vendor/customer, but also different departments of the company in question
            1. With this, I can possibly draw a relation from **Financial Data** (department) to **Employee Information** (Department) if it was created (one to many relationship)
      2. Additionally, the data feels as though it is lacking attributes that can relate to other entities. Even though there are similar date/time attributes in entity tables, they don’t necessarily relate to other tables, as the date/times represent different pieces of information
         1. Ex: *Date for* ***Customer Feedback*** *entity doesn’t equal the visit day/time for* ***Marketing*** entity or the timestamp for ***Transactions*** *entity*
      3. **Marketing Entity** (Order Placed) can correspond with **Transactions Entity** (Item Purchased) possibly (one to one relationship)
         1. Same with **Marketing Entity** (Timestamp) to **Transaction Entity** (Timestamp) (one to one relationship)
      4. **Financial Data Entity** (Payment) to **Transaction Entity** (Payment Method) possibly?
         1. However, I feel as though Payment Method refers to merchant (ex: Visa, MasterCard, AMEX), Payment could be cash, card, check, etc.
         2. I also think that many attributes in **Transaction Entity** can relate to many in **Financial Data** Entity (many to many relationship)
      5. **Transaction Entity** (Item Purchased) to **Customer Feedback Entity** (Item Purchased) if that attribute was created for Customer Feedback
         1. If we created more attributes (*payment information for example*), it may have a closer relationship (many to many) with each entity
      6. **Customer Feedback Entity** (Customer Name) to **Financial Data Entity** (Vendor/Customer) (many to many relationship)
      7. **Customer Feedback Entity** to **Marketing Entity** (many to many relationship)
      8. **Employee Information Entity** (Department) to **Transactions Entity** (one to many relationship)
   4. What is the cardinality of each relationship? One-to-one, one-to-many, many-to many?
      1. Provided in red text above and in diagram that will be placed in repository

With your group:

1. Reconvene with your small group and take turns reviewing each of your diagrams.
   1. What are the common elements you have identified?
      1. Common Stuff:
         1. The use of entity tables
         2. Redefining entity fields and reducing table attributes to make information more comprehensive
         3. We didn’t have many cardinality lines tying attributes and entities together
   2. Where did each of your diagrams differ? What are the pros and cons of each approach?
      1. Differences
         1. Most of my team did not use the entity table titles/attributes that we created in group. They created entity titles that made sense for the example
         2. I used actual plain tables for my attributes/entities, while my team used entity relation tables
      2. Pros
         1. Some tables had organization of data that made sense because they redefined tables that we initially created
         2. Each of us in our team agreed that the way we created our tables was not beneficial to the entity relation charts we needed to build resolved tables
         3. Being able to see how each teammate chose to design their entity relation charts
      3. Cons
         1. Not having the appropriate entity and attribute fields created/addressed to make relational connections with ease