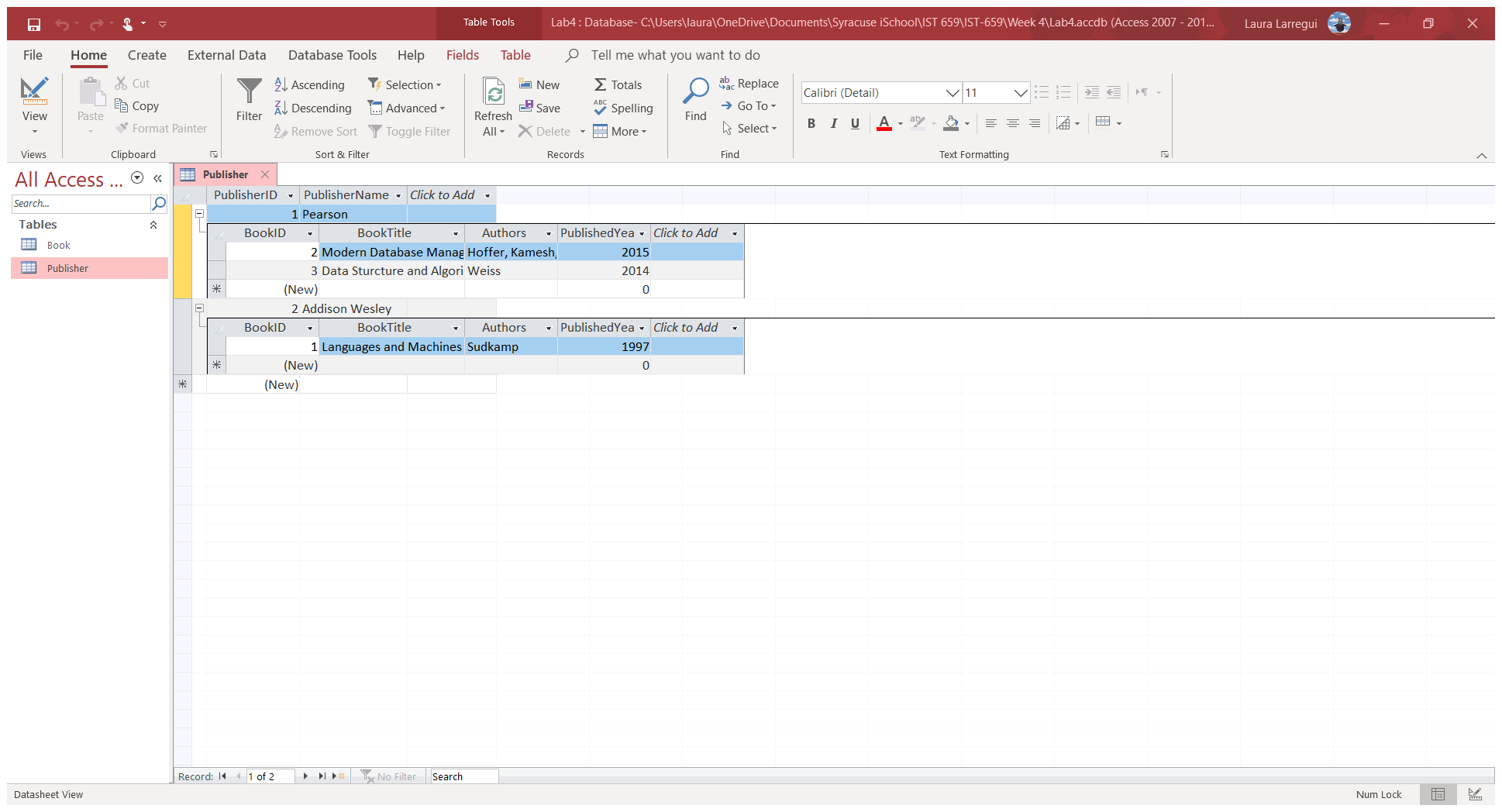
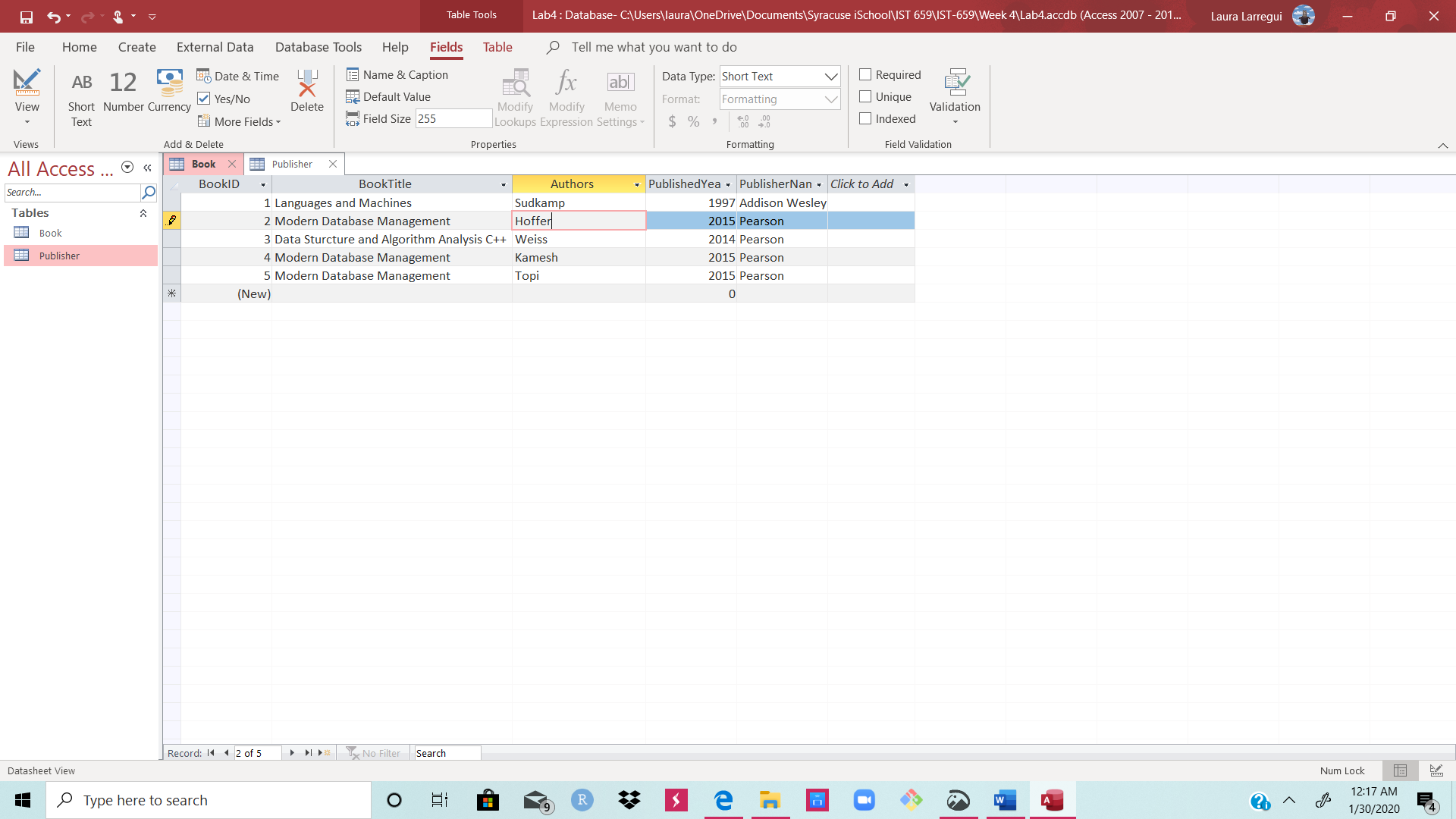
## Name: Laura C. Larregui Date: January 29, 2020

## Lab 04 - Normalization

Part 1 – Normalizing an Existing Data Set

1. Case Study 1- Rad Chad's Rad Bikes
   1. Relational Notation
      1. OrderData (*OrderNum*, Line, ItemNumber, Description, QtyOrdered, PriceEach)
      2. Order (OrderNum, *CustNum*, OrderedDate)
      3. Customer (CustNum, Name, PostalCode, State)
   2. Business Rules
      1. A Customer can place one or more orders. An Order belongs to one customer.
      2. An Order can have one or more OrderData. An OrderData belongs to one order.
2. Case Study 2- More Bikes
   1. Relational Notation
      1. OrderItem (OrderNum, Line, QtyOrdered, *ItemNumber*)
      2. Product (ItemNumber, PriceEach, Description)
         1. I added PriceEach under Product, but I do see some inconsistencies with the prices.
   2. Business Rules
      1. A Product can belong to one or many OrderItems. An OrderItem can have one Product.
3. Case Study 3- Books Again
   1. Screenshot
   2. The Book table is in 0NF because the Author field is multivalued. I would create a new record for each author.



* 1. With Book table I 1NF, I have identified BookTitle and Authors as Candidate Keys and PublishedYear, PublisherName, and BookID as Non-Keys. PublisherName, a non-key, depends on BookTitle, a candidate key. PublisherName would need to be remove from the Book relation into a new entity like the one we already have, Publisher.
  2. Now that the Book table is in 2NF, I will look if there are any non-keys dependent on other non-keys. I do not see any transitive functional dependency. Thus, the Book table is in 3NF.

Part 2 – Normalizing VideoData for VidCast

1. Process:
   1. We start with VideoData ( VideoID, VideoTitle, Username, UserTier, MinTierFollowers, StreamStart, Duration, ContentRating, RatingDescription)
   2. There are no multivalued attributes. Thus, the table is in 1NF.
   3. I have identified Username, VideoTitle, and VideoID as candidate keys. The rest of the attributes are non-keys.
   4. I can see that UserTier and MinTierFollowers are two non-keys that depend on each other, and both of these non-keys depend on Username. Thus, I will create an entity called User, and then I would have a 2NF relation.
   5. ContentRating and RatingDescription are two non-keys that depend on each other. Therefore, I will create another entity called Rating (RatingID as Primary Key).
2. Relational Notation
   1. User (Username, UserTier, MinTierFollowers)
   2. VideoData ( VideoID, VideoTitle, *Username*, StreamStart, Duration, *RatingID*)
   3. Rating ( RatingID, ContentRating, RatingDescription)
3. Business Rules
   1. A user can be the author of one or many VideoData. A VideoData can only have one User.
   2. A VideoData can only have one Rating. A Rating can be applied to one or many VideoData.