Comp 430/533
Assignment 3

Entity Relationship Diagramming

The goal of this assignment is to develop expertise in going from a textual description of a business model to a formal database model.

You will create an entity-relationship model / diagram for the system described below.

You must prepare your ERD using **some electronic drawing program** (there are many programs, like Microsoft Visio, that have ERD capabilities built in; but it is also possible to draw a nice ERD using something like PowerPoint or Google drawings).

This is a 2 person assignment. You may fully collaborate with your partner and all names should be included on the submission. Groups will be set up on Canvas and only one person in each group should submit the assignment.

You MUST use the Chen +(min, max) notation given in class. Credit will not be given if you use any other notation.

What to turn in

Upload a pdf of your final model and answers to Canvas. This should be a 2-page document. The first page should have ONLY the team member names and netIds. The second page should be the model.

Grading

The model is worth 99 points. Everyone starts with full credit, and deductions are made based on errors and / or missing items. Systemic issues are

typically penalized once. Common deductions are for missing attributes, incorrect use of notation or errors in, missing relationships, vague relationship names, etc.

The question at the end is worth 1 point.

Academic Honesty

The following level of collaboration is allowed on this assignment: You may discuss the assignment fully with your partner, but only at a high level with your other classmates. What is not allowed is direct examination of anyone else's model (on a computer, email, whiteboard, etc.) or allowing anyone else to see your group's model.

You may use the search engine of your choice to lookup additional details on the ERD notation, but may not use it to find solutions to the assignment.

1 ER Model – Ice Cream Food Truck Business

1.1 Background

A local entrepreneur has recently purchased a food truck that serves soft-serve ice cream and related items. Seeing an opportunity, the business owner has decided to focus on selling ice cream to the Rice University community. We are going to design a database that will help the owner and her staff run, manage, and grow the business.

1.2 What to Model

Model the database as described here. Make sure that your design is capable of storing products with choices (e.g. vanilla or chocolate ice cream) and recording exact orders. For example, if someone orders 2 kid cones, one with vanilla ice cream and M&M topping, and the other with chocolate ice cream with peanut topping, we should be able to see that on that ticket those two combinations were ordered. We should be able to tell that peanut topping was paired with chocolate ice cream and M&M was paired with vanilla.

The food truck sells products. Products are ice cream cups and cones (small, medium, large), ice cream sundaes (brownie, hot fudge, etc.) and drinks, to name a few. Products are identified by a unique one to three letter alphanumeric code and also have unique names. That is, either the code of the name is sufficient to uniquely identify each product. The code is a convenient shorthand for the product name.

Products are only sold during events. Each event is identified by a name and start date/timestamp. For example there may be multiple "Jones College Study Break" events, but each will have a unique start date/time. Events also have a planned and an actual end date and time and a total dollar sales (initially set to zero). This information may be used (later) to analyze product sales to determine the most popular products and / or profitable events. Events have status. This includes values like "Planned", "Confirmed", "Cancelled", and "Completed".

To accommodate different types of events, the truck has a number of different menus, each of which has a different combination of products and prices (e.g. Rice discount menu, Cups and Cones only, ...). Each menu has a unique name. Products may be assigned to one or more menus, with a different price on each menu, for example a Kid Cone may cost $3 on one menu, but $2 on a different, subsidized menu.

Each product is assembled from ingredients. Ingredients have unique names and may be used in multiple products. Some products are atomic, and consist of a single ingredient. Examples of these single ingredient products are sodas, ice cream novelties (e.g. a fudgsicle), and a bottle of water. For each ingredient used in a product, we know the quantity, unit of measure (e.g. "oz"), and the detail. Details indicate choices for an ingredient, or descriptors such as "#10", "short", or "tall". For example, the ingredients for a Kid Cone might be:

QTY	UNIT	INGREDIENT	DETAIL
6	oz	ice cream	choice
1	oz	ice cream topping	choice
1	each	cone	#10
1	each	napkin	short

When someone orders a Kid Cone, the server asks the customer to choose between the available choices for ice cream (e.g. Chocolate, Vanilla, Strawberry) and topping (e.g. Sprinkles, M & Ms, peanuts, etc.). The database should contain a list of the available choices for each ingredient. If no choice is needed, a detail may be specified in the database instead. For example, there are long and short napkins, and the database specifies which to give out based on the product selected.

The ingredients for a can of soda might be:

QTY	UNIT	INGREDIENT	DETAIL
1	can	soda	choice

Some ingredients, like ice cream, are divisible. People should be able to order more than one flavor of ice cream in a product, but not exceed the total amount of ice cream specified. For example, the 6 oz of ice cream in our Kid Cone could be 3 oz of vanilla and 3 oz of chocolate.

During each event, we keep track of tickets. Tickets are groups of products purchased at the same time. For example, a family may buy a Kid Cone, a Brownie Sundae, and a Milkshake. Each ticket records the date-time of the ticket, the total number of products sold, and the total dollar amount. Each ticket is assigned an arbitrary unique ticket id. We keep track of the products purchased on each ticket, as well as the choices/details associated with the sold product(s). Multiple payment methods (e.g. cash, credit card, venmo) can be used to pay each ticket. Our database should record each payment type and amount used to pay for each ticket.

To simplify usage and inventory calculations later, we record all of the

information about each product sold on a ticket, including the quantity and unit for each ingredient actually used. For the ticket mentioned earlier, the Kid Cone might be Vanilla (6 oz) with M &M topping (1 oz), the Brownie Sundae might be Strawberry Ice Cream (4 oz) and Chocolate Ice Cream (4 oz) with peanut topping (1 oz) and chocolate chip topping (1 oz) and the Milkshake might be Chocolate Ice Cream (6 oz), sugar (0.75 oz), etc. We also record the ingredients with details (instead of choices), including napkins, straws, etc. Basically, we should be able to reconstruct the exact contents of a ticket, its products, and the ingredients used by examining this sales record.

The truck owner has decided to hire Rice students to help staff the truck. Each student has a unique netId as well as a name and a social security number. The truck is staffed in shifts. Each shift is 2-4 hours long. Each shift has a start and end date-time. There must be 1 student working each shift, but there may be up to 2. Each student has a role for the shift (e.g. Order taker, Order filler, Driver)

2 Our Food Truck

(1 point) Our ice cream truck business should be named: