Final Project: Milestone 2

Derek Caramella, Lisa Pink, & Tapan Pradyot

Department of Computer Science, University of Rochester

CSC 461: Database Systems

Dr. Zhupa

October 14, 2021

Task A: Draw an ER Diagram

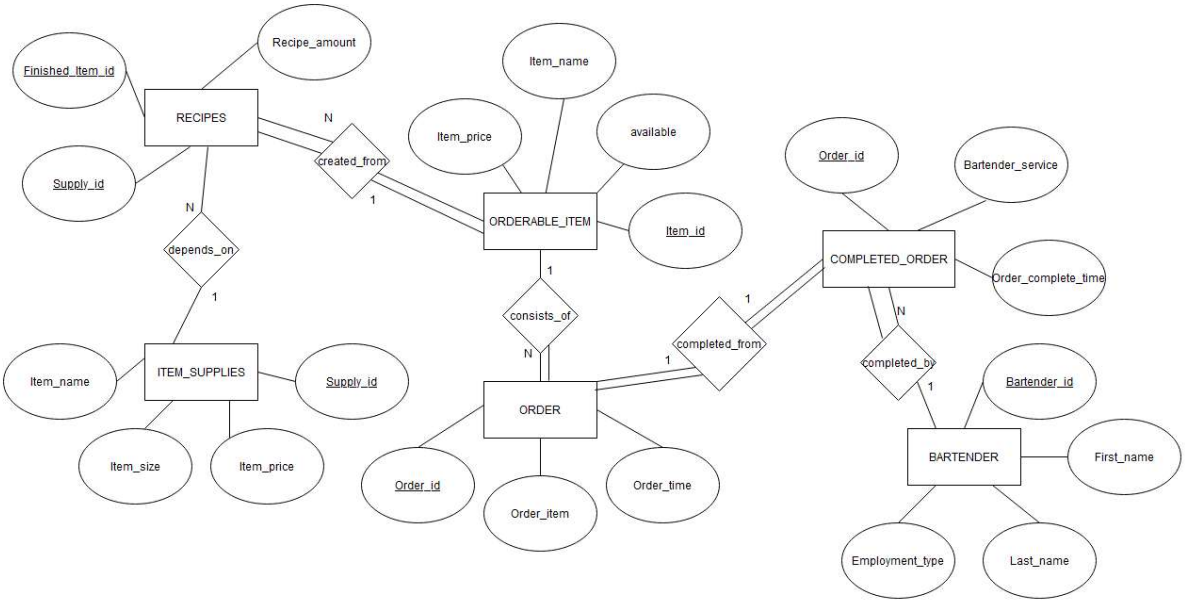

Figure 1: ER Diagram

The ER diagram consists of six entities: ORDER, COMPLETED_ORDER, BARTENDER, ITEM_SUPPLIES, RECIPES and ORDERABLE_ITEM. Each entity has a primary key that uniquely identifies the tuple. The attributes of **ORDER** are Order_id (primary key), Order_item, and Order_time. The **ORDERABLE_ITEM** entity consists of the attributes Item_id (primary key), Item_name, Available and Item_price. **ITEM_SUPPLIES** contains attributes Supply_id (primary key), Item_name, Item_size, and Item_price. The **BARTENDER** entity has attributes Bartender_id (primary key), First_name, Last_name, Employment_type. The **COMPLETED_ORDER** entity has attributes Order_id (primary key), Bartender_service, and Order_complete_time. The **RECIPES** entity has attributes Finished_item_id, Supply_id and Recipe_amount

Each ORDERABLE_ITEM is *created_from* one RECIPES. Each ORDER *consists* of one ORDERABLE_ITEM. Each COMPLETED_ORDER is *completed_by* a BARTENDER. Each COMPLETED_ORDER is *completed_from* one ORDER, creating a 1:1 relationship. Each RECIPES *depends_on* ITEM_SUPPLIES for the creation of an order.

There are no weak entities, weak relationships, or class hierarchies in our schema.

Task B: Relational Database Design Using ER-to-Relational Mapping

I. ER to Relation Mapping Algorithm

1. *Mapping of Regular Entity Types*. For each regular entity type *E* in the ER, a relation *R* is made that includes the simple attributes of *E*. A key attribute is chosen for each entity type *E* as the primary key. The six relations are displayed below in Figure 2:

ORDERS

Order_id	Order_item	Order_time

ORDERABLE ITEMS

Item_id	Item_name	Item_price	Available

RECIPES

Finished_Item_id	Supply_id	Recipe_amount

ITEM_SUPPLIES

Supply_id	Item_name	Item_size	Item_price

BARTENDERS

Bartender_id	First_name	Last_name	Employment_type

COMPLETED_ORDERS

Order_id	Bartender_service	Order_complete_time

Figure 2: Mapping of Regular Entity Types

2. *Mapping of Weak Entity Types*. We skip this step because the ER model does not contain any weak entity types.
3. *Mapping of Binary 1:1 Relation Types*. For each 1:1 relationship *R* in the schema, relations *S* and *T* identify the corresponding entity types in *R*. The foreign key approach is utilized by map between entities.
    - For the *completed_from* relationship, the foreign key Order_id from the COMPLETED_ORDER entity references the primary key Order_id from the ORDER entity.
4. *Mapping of 1:N Relationship Types*. There are four 1:N relationship types in the ER model.
    - For the *created_from* relationship, the foreign key attribute Finished_item_id of RECIPES references the attribute Item_id of ORDERABLE_ITEMS.
    - In the *consists_of* relationship, the attribute Item_id of ORDERABLE_ITEM is the foreign key that references the attribute Order_item in the ORDER entity. Partial participation exists for ORDERABLE_ITEMS because we can have a record from ORDERABLE_ITEMs that has not been purchased in ORDER.
    - The *completed_by* relationship between BARTENDER and COMPLETED_ORDER. Each COMPLETED_ORDER is *completed_by* one BARTENDER, but each BARTENDER can complete multiple COMPLETED_ORDERs. The foreign key Bartender_service from the COMPLETED_ORDER entity references the primary key Bartender_id from the BARTENDER entity. Partial participation exists for BARTENDER because we can have bartender who did not complete any orders (i.e., if a bartender is a new hire).
    - The *depends_on* relationship between RECIPES and ITEM_SUPPLIES. The attribute Supply_id of RECIPES references the primary key Supply_id of ITEM_SUPPLIES entity. Each recipe is created using multiple Supply_id of the entity ITEM_SUPPLIES hence creating a 1:N relationship. Partial participation exists by both RECIPES and ITEM_SUPPLIES because we can have recipes that do

not have the required, and we can have items in ITEM_SUPPLIES that are not being used by the RECIPIES.

5. *Mapping of Binary M:N Relationship Types*. No M:N relationships exist in the ER model; subsequently, this step is bypassed.
6. *Mapping of Multivalued Attributes*. No multivalued attributes exist in the ER model; subsequently, this step is bypassed.
7. *Mapping of N-ary Relationship Types*. No N-ary relationships exist in the ER model; subsequently, this step is bypassed.
8. Applicable to only to EER models, thus, the step is skipped, and the mapping process has been completed.
9. Applicable to only to EER models, thus, the step is skipped, and the mapping process has been completed.

The ER to Relation mapping is summarized below in Figure 3:

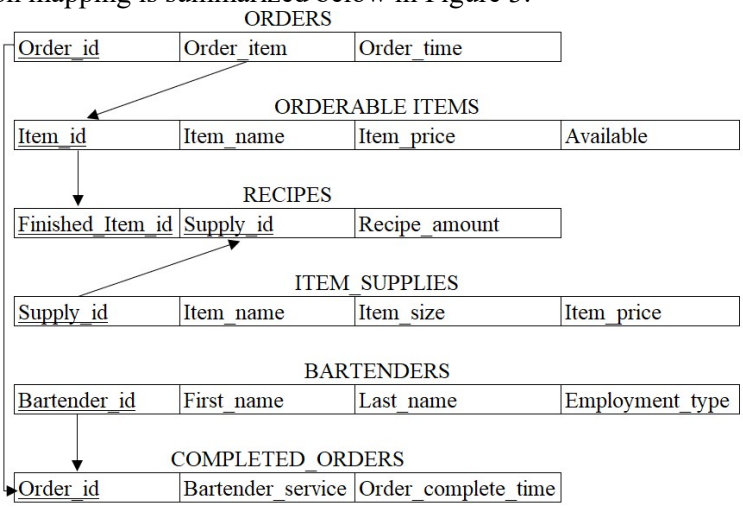

Figure 3: ER to Relation Mapping

The entities and relations can be summarized further in the table below:

Relation Name	ER Diagram Components
ORDER	E(Order) + R(consists_of)
COMPLETED_ORDER	E(Completed_order) + R(completed_from)
ORDERABLE_ITEM	E(Orderable_item) + R(created_from)
BARTENDER	E(Bartender) + R(completed_by)
ITEM_SUPPLIES	E(Item_supplies)
RECIPES	E(Recipes) +R(depends_on)

Table 1: Summary of Mapping

II. Database Schema

- **ORDER:** The contains information about the drink orders at the bar. The primary key is Order_id, as each order has a different identification number. Moreover, the Order_id acts as a foreign key to the COMPLETED_ORDER table. Additionally, the Order_item is a foreign key to Item_id in the ORDERABLE_ITEM entity. In the case of a DELETE operation, the delete is cascaded onto the COMPLETED_ORDER table. The attributes of this relation are listed below:
    - **Order_id:** Uniquely identifies the order. It is of datatype INT (5).

    - **Order_item:** Tells which type of drink was ordered. This attribute is of datatype VARCHAR (50).
    - **Order_time:** States the time at which the order occurred. The datatype is DATETIME.
- **COMPLETED_ORDER:** This relation tracks orders that have been completed by the bartenders. The primary key is Order_id. The Order_id acts as a foreign key to the ORDER table. Additionally, the Bartender_service is a foreign key to Bartender_id in the BARTENDER entity. In the case of a DELETE operation, the delete is cascaded onto the ORDER table. The attributes of this relation are listed below:
    - **Order_id:** Uniquely identifies the order that has been completed. This attribute is of datatype INT (5).
    - **Bartender_service:** Provides the Bartender identifier of the bartender that has completed the order with datatype INT (5).
    - **Order_complete_time:** Provides the time at which the order was completed. The datatype is DATETIME.
- **ORDERABLE_ITEM:** The table contains information about items that are on the drink menu. The primary key is Order_id. The Order_id is a foreign key to the RECIPES table and the ORDER entity. For DELETE operations, the DELETE is cascaded onto RECIPES. The attributes of this relation are listed below:
    - **Item_id:** Uniquely identifies the item that can be ordered from the menu, the datatype is INT (5)
    - **Item_name:** Holds the name of the drink on the menu. The data type is VARCHAR(50).
    - **Item_price:** States the price of the drink. This attribute is of datatype DECIMAL(4,2).
    - **Available:** This a binary attribute that takes 1 or 0 value. It is allowed to have a NULL value. The attribute datatype is TINYINT(1).
- **BARTENDER:** The entity holds relevant information about the bartenders that are employed by the bar. The primary key is Bartender_id. Moreover, the Bartender_id serves as a foreign key to the COMPLETED_ORDER entity. DELETE operations set NULL values in the COMPLETED_ORDER entity. The attributes of this relation are listed below:
    - **Bartender_id:** Uniquely identifies the bartender with datatype INT(5).
    - **First_name:** The first time of the bartender. The datatype is VARCHAR (50).
    - **Last_name:** The last name of the bartender. The datatype is VARCHAR(50).
    - **Employment_type:** Specifies whether the bartender is a full-time, part-time, or inactive employee with datatype VARCHAR(9).
- **ITEM_SUPPLIES:** This relation contains information about the supplies need to produce the drinks at the bar. The primary key is Supply_id; moreover, the Supply_id acts a foreign key to the RECIPES entity. The DELETE operation cascades onto the RECIPES entity. The attributes of this relation are listed below:
    - **Supply_id:** Uniquely identifies the supply with datatype is INT(5).
    - **Item_name:** The name of the bottle. The data type is VARCHAR (13).
    - **Item_size:** The size of the bottle purchased within management supply chain, INT (5). It is allowed to have **NULL** values.
    - **Item_price:** The cost of the bottle purchased within management supply chain, DECIMAL(5,2). It is allowed to have **NULL** values.

- **RECIPES:** This relation contains recipes to create an Orderable item in the entity ORDERABLE_ITEM. The primary key is a combination of attributes Supply_id and Finished_item_id. In case of a DELETE operation, no action is taken on other entities; however, corner case exists: if a single record of finished_item_id remains, the DELETE operation is prevented. The Finished_item_id acts a foreign key to the ORDERABLE_ITEM entity and the Supply_id acts as a foreign key to the ITEM_SUPPLIES entity.
    - **Supply_id:** Identifies the supply to be used in the Recipe, INT(5).
    - **Finished_item_id:** References the PK of entity ORDERABLE_ITEM, INT(5)
    - **Recipe_amount:** References the quantity of an Item to be used in the creation of drink, INT (5).

Task C: Milestone 1 Feedback Response

Feedback: I am just a little puzzled about your target group and who the users would be.

Response: The application's target audience are bar owners looking to view revenue streams, identify marketing channels, and appraise employee performance.