**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**

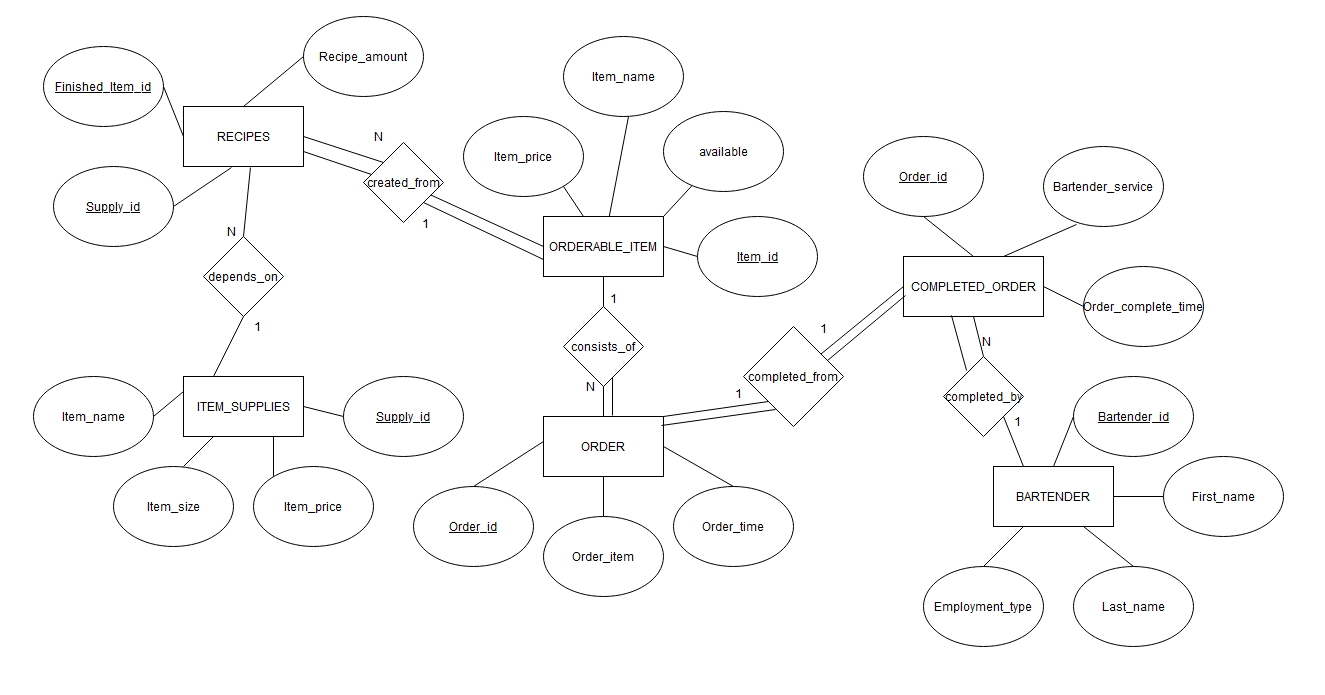


Figure 1: ER Diagram

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**

1. **ER to Relation Mapping Algorithm**
2. *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:

Table

Description automatically generated

Figure 2: Mapping of Regular Entity Types

1. *Mapping of Weak Entity Types*. We skip this step because the ER model does not contain any weak entity types.
2. *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.

1. *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.

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

Table

Description automatically generated with medium confidence

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

1. **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.