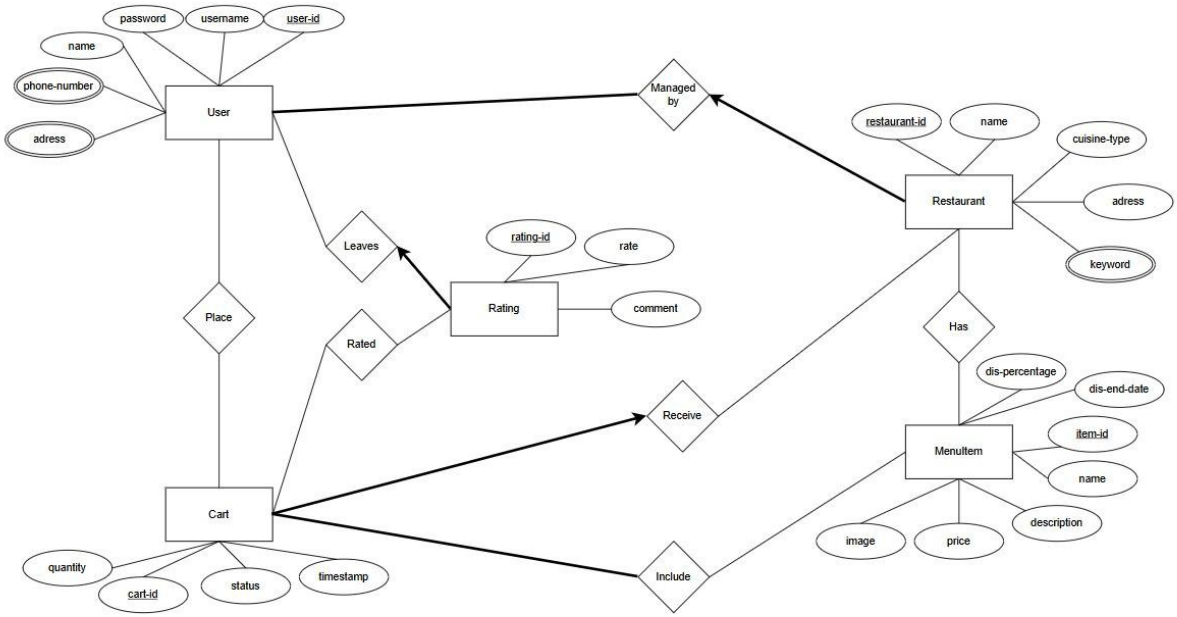
Aden Duru Çelik -Ahmet Fırat-Barış Andiç

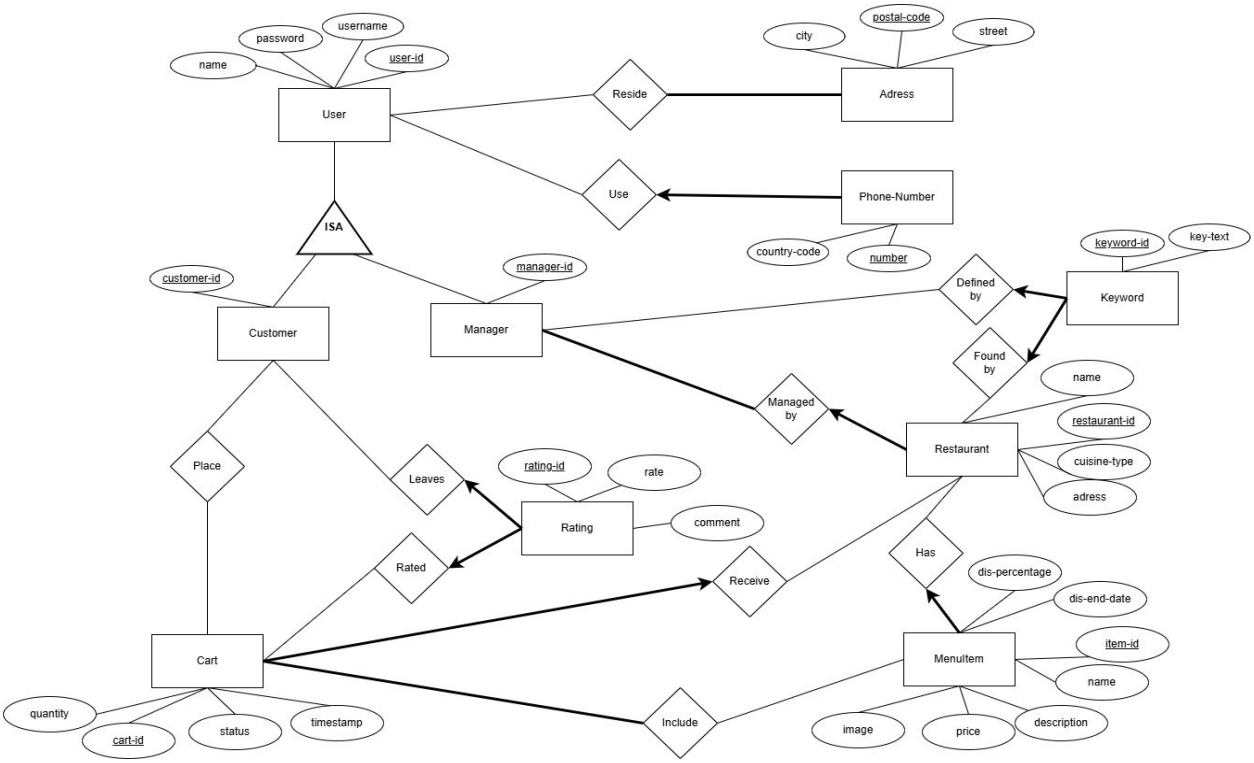
S028194-S029491-S028573

CS202-Spring 2025

-OLD ER DIAGRAM



-NEW ER DIAGRAM



We extracted phone_number and address from User. We converted them from attribute to entity. We also created customer and manager as ISA for user. And we added primary key to them. We converted keyword from attribute to entity in Restaurant.

-BASE RELATIONS AND CONSTRAINTS

User(user_id, username, password, name)

Restaurant(restaurant_id, name, cuisine_type, address,manager_id)
-foreign key(manager_id) references Manager
MenuItem(item_id, restaurant_id, name, description, price, image, dis_percentage, dis_end_date)

-foreign key(restaurant_id) references Restaurant

Cart(cart_id, customer_id, restaurant_id,item_id, quantity, status, timestamp)

-foreign key(customer_id) references Customer

-foreign key(restaurant_id) references Restaurant

-foreign key (item_id) references MenuItem

Rating(rating_id, customer_id, restaurant_id, cart_id, rate, comment)

-foreign key(customer_id) references Customer
-foreign key(restaurant_id) references Restaurant

-foreign key(cart_id) references Cart

Customer(customer_id,user_id)
-foreign key(user_id) references User

Manager(manager_id,user_id)
-foreign key(user_id) references User

Phone -Number(country_code,number,user_id)
-foreign key(user_id) references User

Adress(city,postal_code,street,user_id)
-foreign key(user_id) references User

Keyword(keyword_id,key_text,restaurant_id,manager_id)
-foreign key(restaurant_id) references Restaurant
-foreign key(manager_id) references Manager

-FUNCTIONAL DEPENDENCIES

user_id ->username, password, name

restaurant_id →name, cuisine_type, address
item_id →name, description, price, image, dis_percentage, dis_end_date
cart_id → quantity, status, timestamp
rating_id → rate, comment
number →country_code
postal_code →city,street
keyword_id →key_text
manager_id →user_id
customer_id → user_id

-NORMALIZATION

All relations in the database are normalized up to 3NF

-1NF: All attributes contain atomic values and each relation has a unique primary key. Multi-valued attributes like phone numbers, addresses, and restaurant keywords were extracted into separate tables (PhoneNumber, Address, Keyword) to preserve 1NF.

-2NF: There are no partial dependencies. Every non-key attribute is fully functionally dependent on the whole primary key.

-3NF: No transitive dependencies exist. All non-key attributes depend only on the primary key and not on other non-key attributes.

-DESIGN DECISIONS AND JUSTIFICATIONS

In designing this database, several key decisions were made to ensure data integrity, reduce redundancy, and optimize query performance.

Each major entity (User, Restaurant, MenuItem, Cart, Rating) was assigned a unique primary key to uniquely identify each record. Subtypes such as Customer and Manager were modeled using an ISA relationship with User, allowing role-based extension without duplication.

Foreign keys were used extensively to enforce referential integrity. For instance, MenuItem is linked to Restaurant, and Cart is linked to both User and Restaurant.

User-related multi-valued attributes like phone numbers and addresses were moved to separate PhoneNumber and Address tables to conform to 1NF. Similarly, restaurant keywords were stored in a separate Keyword table and linked through an associative RestaurantKeyword relation.

A many-to-many relationship between Cart and MenuItem was captured using the Include table, enabling a cart to contain multiple items and an item to appear in multiple carts.

Ratings were associated with Users, Restaurants, and Carts to track structured feedback. Storing ratings in a separate table improves scalability and flexibility, allowing multiple ratings per user and restaurant.

Functional dependencies were carefully considered to achieve normalization up to 3NF, minimizing update anomalies and ensuring a robust structure. Overall, the schema balances normalized design with practical performance, forming a solid foundation for a restaurant ordering and review platform.