University of Puerto Rico - Mayaguez Campus

Department of Electrical and Computer Engineering

ICOM 5016 - Introduction to Database Systems

Professor: Manuel Rodriguez

**Term Project – Backend System for Disaster Site Resources Locator**

**Phase I – Conceptual Design**

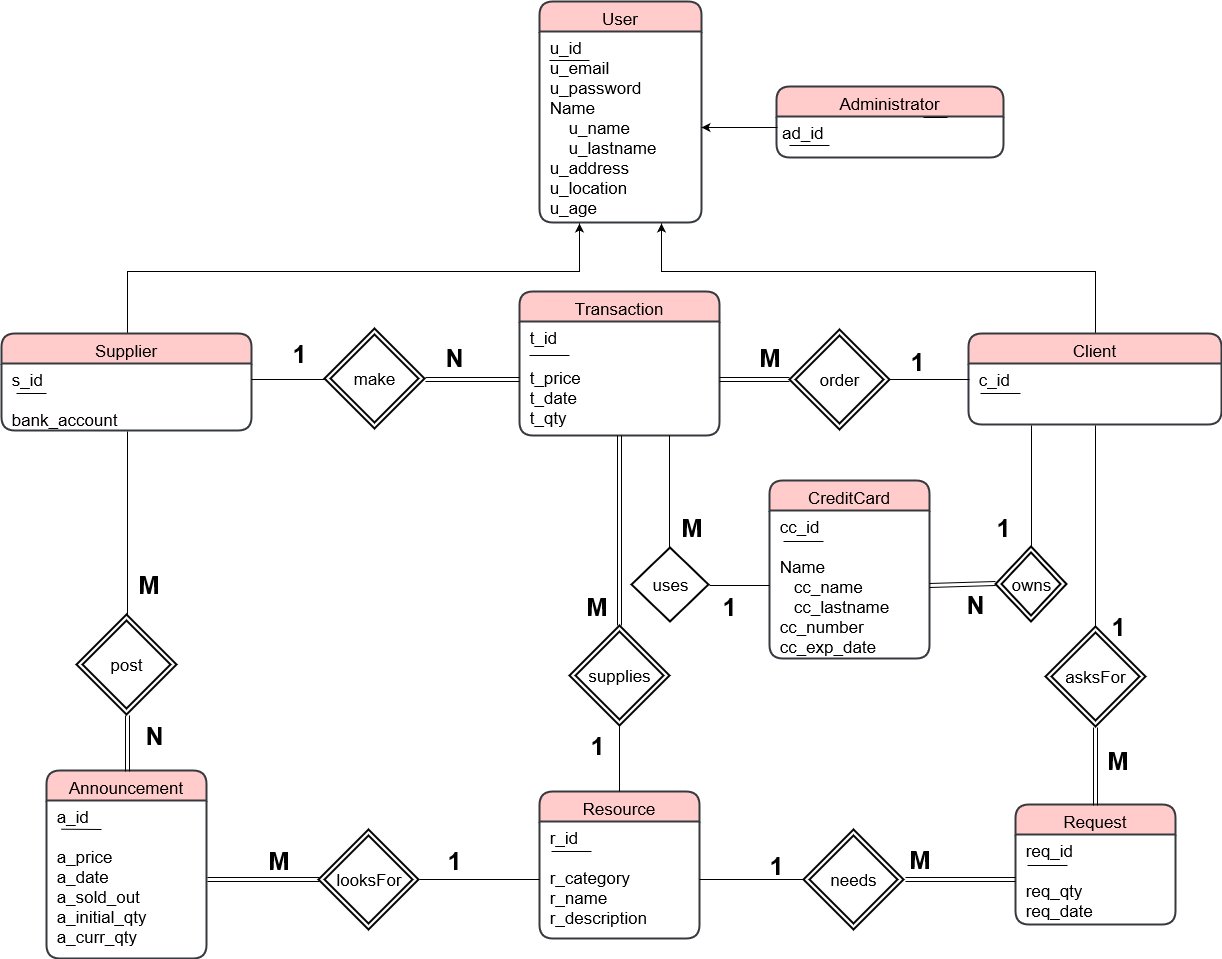
By:

Nicole Matos Feliciano

Gilissa Matos Hernández

December 5, 2017

Entity-Relationship Model



Schema Design with E-R Model

User( -- user of system (inherit administrator, supplier and client)

u\_id serial primary key, -- user id

u\_email varchar(50), -- user email

u\_password varchar(20), -- user password

u\_name varchar(20), -- user name

u\_lastname varchar(20), -- user last name

u\_address varchar(50), -- user address

u\_location varchar(25), -- user location

u\_age int -- user age to see if user is older than 18 years old

)

Administrator( -- administers the system (inherited from users)

ad\_id serial primary key, -- administrator id

u\_id references User(u\_id) -- user id

)

Supplier( -- supplies by announcements (inherited from users)

s\_id serial primary key, -- supplier id

u\_id references User(u\_id), -- user id

bank\_account integer -- supplier bank account

)

Client( -- client get donations or purchases (inherited from users)

c\_id serial primary key, -- client id

u\_id references User(u\_id)) -- user id

Transaction( -- transaction made between supplier and client

t\_id serial primary key, -- transaction id

s\_id references Supplier(s\_id), -- supplier id

c\_id references Client(c\_id), -- client id

t\_price float, -- transaction price

t\_date Date, -- transaction date

t\_qty int -- transaction quantity (of resources)

)

CreditCard( -- credit card registered by a supplier (can own more)

cc\_id serial primary key, -- credit card id

cc\_name varchar(20), -- credit card name

cc\_lastname varchar(20), -- credit card last name

cc\_number integer, -- credit card number

cc\_exp\_date Date -- credit card expiration date

)

Announcement( -- announcement made by a supplier

a\_id serial primary key, -- announcement id

s\_id references Supplier(s\_id), -- supplier id

a\_price float, -- announcement price per unit

a\_date Date, -- announcement date

a\_sold\_out boolean, -- announcement status (if sold out or not)

a\_initial\_qty int, -- announcement initial quantity

a\_curr\_qty int -- announcement current quantity

)

Request( -- request made by a supplier

req\_id serial primary key, -- request id

c\_id references Client(c\_id), -- client id

req\_qty int, -- request quantity

req\_date Date -- request date

)

Resource( -- resource of system

r\_id serial primary key, -- resource id

r\_category varchar(20), -- resource category (medicine, water, clothing, food, etc)

r\_name varchar(20), -- resource name (aspirin, nikini, t-shirt, etc)

r\_description varchar(50) -- resource description (50 tablets, 6 bottles, medium, etc)

)

Relationships

1. **make** - This is a one to many relationship, since an element from entity Supplier is related with more than one element from entity Transaction. A supplier can have many transactions, but a transaction can only be part of one supplier. In this relationship Transaction must have total participation and is declared a weak entity.
2. **needs** - This is a one to many relationship, since an element from entity Resource is related with more than one element from entity Request. A resource can be requested many times, but a request can only be about one resource. In this relationship Request must have total participation and is declared a weak entity.
3. **asksFor** - This is a one to many relationship, since an element from entity Client is related with more than one element from entity Request. A client can have many requests, but a request can only be made by one client. In this relationship Request must have total participation and is declared a weak entity.
4. **order -** This is a many to one relationship, since an element from entity Client is related with more than one element from entity Transaction. A client can have many transactions, but a transaction can only be order by one client. In this relationship transaction must have total participation and is declared a weak entity.
5. **uses -** This is a many to one relationship, since an element from entity Credit Card is related with more than one element from entity Transaction. A credit card can be used many transactions, but a transaction can only be placed with one credit card.
6. **owns -** This is a many to one relationship, since an element from entity Client is related with more than one element from entity Credit Card. A client can own many credit cards, but a credit card can only be owned by one client. In this relationship Credit Card must have total participation and is declared a weak entity.
7. **supplies -** This is a many to one relationship, since an element from entity Resource is related with more than one element from entity Transaction. In this relationship Transaction must have total participation and Transaction is declared a weak entity.
8. **looksFor -** This is a many to one relationship, since an element from entity Resource is related with more than one element from entity Announcement. In this relationship Resource must have total participation and Announcement is declared a weak entity.
9. **post -** This is a many to many relationship, since an element from entity Supplier is related with more than one element from entity Announcement. Likewise, an element from entity Announcement is related with more than one element from entity Supplier. In this relationship Announcement must have total participation and Announcement is declared a weak entity.