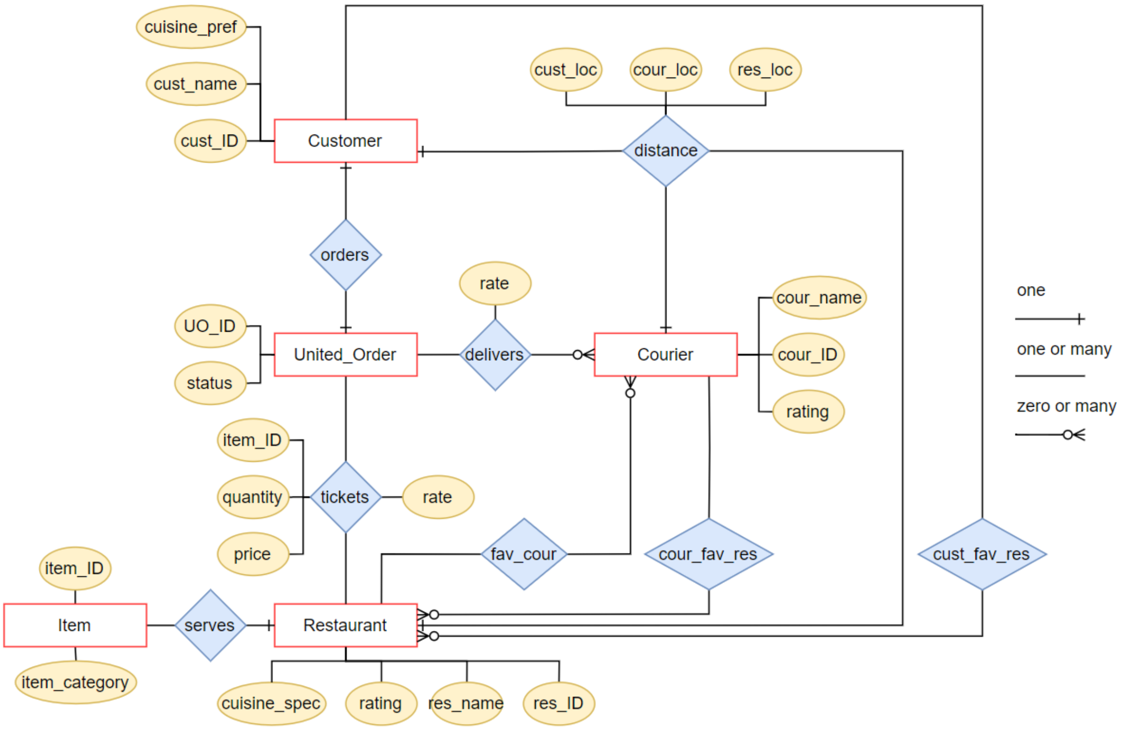
DIS assignment 1

Adit Fadhillah, Naomi Knudsen & Nikolaj Schaltz

May 2023

Exercise 1

a) Design a high-level E/R model for the described scenario.



b) Discuss your modeling choices.

We decided upon the entities from the assignment text. We were told that there were told that the platform was accessed by ”three external stakeholder groups”, and it was these groups that we made entities. We also created another entity called United_Order because we felt that the situation could be described better if some of the other entities had relationships with the order.
Sometimes we contemplated whether or not to make something an attribute or a relationship, and then we chose based on the design principle of simplicity. We also had a hard time with the address and distance part of the situation. In the end, because address and distance are so important for the situation we decided to create a distance relationship, that had some location attributes.

We used several binary relationships and one n-nary relationship of 4 to describe the situation. We used a ternary relationship between Customers, Couriers and Restaurants since we decided this was a better way to describe the situation; both Couriers and Customers should be able to pick a favorite restaurant. For the rest of the model, we found that n-ary relationships weren’t adequate to describe the situation of a United Order, and thus we used binary relationships.

On the right side of the picture, the arrows and their meaning are described.

In terms of arrows that describe how many relations each entity could have, we chose that for the relation ’order’, the entity customer can have one United_Order, and that a ’United_Order’ can only have one customer.

We chose that customers and couriers can have zero or many favorite restaurants and that restaurants can have zero or many favorite couriers. We also decided that a customer can only have exactly one order, and thus we used referential integrity.

The relations ’fav_cour’, ’cour_fav_res’ and ’cour_fav_res’ are weak relations, as they are binary relation that is not crucial for the whole system to work.

c) Translate your E/R model to the relational model.

We have tried to maintain referential integrity by using ON UPDATE NO ACTION for some of the Foreign keys in relationships.

Entities

```
CREATE TABLE Customer
(cust_ID CHAR(8),
cust_name CHAR(20),
cuisine_pref CHAR(20),
PRIMARY KEY (cust_ID));

CREATE TABLE Courier
(cour_ID CHAR(8),
cour_name CHAR(20),
rating INTEGER,
PRIMARY KEY (cour_ID));

CREATE TABLE United_Order
(UO_ID CHAR(10),
status INTEGER,
PRIMARY KEY (UO_ID));

CREATE TABLE Restaurant
(rest_ID CHAR(8),
rest_name CHAR(20),
cuisine_spec CHAR(20),
rating INTEGER,
PRIMARY KEY (rest_ID));

CREATE TABLE Item
(item_ID CHAR(8),
item_category CHAR(20),
PRIMARY KEY (item_ID));
```

Relationships

```
CREATE TABLE orders
(cust_ID CHAR(8),
UO_ID CHAR(10),
PRIMARY KEY (cust_ID, UO_ID),
FOREIGN KEY (cust_ID) REFERENCES Customer ON UPDATE NO ACTION,
FOREIGN KEY (UO_ID) REFERENCES United_Order ON UPDATE NO ACTION);

CREATE TABLE tickets
(rest_ID CHAR(8),
UO_ID CHAR(10),
item_ID CHAR(12),
quantity INTEGER,
price INTEGER,
rate INTEGER,
PRIMARY KEY (rest_ID, UO_ID),
FOREIGN KEY (rest_ID) REFERENCES Restaurant,
FOREIGN KEY (UO_ID) REFERENCES United_Order);

CREATE TABLE serves
(item_ID CHAR(8),
rest_ID CHAR(8),
PRIMARY KEY (item_ID, rest_ID),
FOREIGN KEY (item_ID) REFERENCES Item,
FOREIGN KEY (rest_ID) REFERENCES Restaurant ON UPDATE NO ACTION);

CREATE TABLE cour_fav_res
(cour_ID CHAR(8),
rest_ID CHAR(8),
PRIMARY KEY (cour_ID, rest_ID),
FOREIGN KEY (cour_ID) REFERENCES Courier,
FOREIGN KEY (rest_ID) REFERENCES Restaurant);

CREATE TABLE cust_fav_res
(cust_ID CHAR(8),
rest_ID CHAR(8),
PRIMARY KEY (cust_ID, rest_ID),
FOREIGN KEY (cust_ID) REFERENCES Customer,
FOREIGN KEY (rest_ID) REFERENCES Restaurant);

CREATE TABLE fav_cour
(rest_ID CHAR(8),
cour_ID CHAR(8),
PRIMARY KEY (rest_ID, cour_ID),
FOREIGN KEY (rest_ID) REFERENCES Restaurant,
FOREIGN KEY (cour_ID) REFERENCES Courier);

CREATE TABLE delivers
(UO_ID CHAR(10),
cour_ID CHAR(8),
rate INTEGER,
PRIMARY KEY (UO_ID, cour_ID),
FOREIGN KEY (UO_ID) REFERENCES United_Order,
FOREIGN KEY (cour_ID) REFERENCES Courier);

CREATE TABLE distance
(cust_ID CHAR(8),
cour_ID CHAR(8),
res_ID CHAR (8),
cust_loc CHAR(50),
cour_loc CHAR(50),
res_loc CHAR(50),
PRIMARY KEY (cust_ID, cour_ID, res_ID),
FOREIGN KEY (cust_ID) REFERENCES Customer ON DELETE NO ACTION,
FOREIGN KEY (cour_ID) REFERENCES Courier,
FOREIGN KEY (res_ID) REFERENCES Restaurant);
```

d) Create a schema and one example instance in the format accepted by RC-EVAL.

Instance

```
Customer
("Italian", "Rasmus", "CU111111")
("Thai", "Peter", "CU222222")
("Italian", "Naomi", "CU333333")
("Chinese", "Nikolaj", "CU444444")

Restaurant
("Italian", "Resorante_Italiano", "R1111111", 5)
("Thai", "Thai_Take_Away", "R2222222", 4)
("Chinese", "China-box", "R3333333", 1)

Courier
("Sophus", "CO111111", 4)
("Hanne", "CO222222", 5)
("Matias", "CO333333", 2)

United_Order
("u111111111", 0)
("u222222222", 1)
("u333333333", 0)
("u444444444", 1)

Item
("I1111111", "Pizza")
("I2222222", "Springrolls")
("I3333333", "Dumplings")
("I4444444", "Rice")

orders
("CU111111", "u111111111")

distance
("CU111111", "CO111111", "R3333333", "2300", "2300", "2300")
("CU111111", "CO111111", "R1111111", "2300", "2300", "2100")
("CU111111", "CO111111", "R2222222", "2300", "2300", "2770")
("CU111111", "CO222222", "R1111111", "2300", "2100", "2100")
("CU111111", "CO222222", "R2222222", "2300", "2100", "2770")
("CU111111", "CO222222", "R3333333", "2300", "2100", "2300")
("CU333333", "CO222222", "R1111111", "2100", "2100", "2100")

cust_fav_res
("CU111111", "R1111111")
("CU333333", "R1111111")
("CU222222", "R2222222")
("CU444444", "R3333333")

fav_cour
("R1111111", "CO111111")
("R2222222", "CO333333")
("R3333333", "CO333333")
```

Schema

```
Customer(cuisine_pref:string, cust_name:string, cust_ID:string)
Restaurant(cuisine_spec:string, res_name:string, res_ID:string, res_rating:int)
Courier(cour_name:string, cour_ID:string, co_rating:int)
United_Order(UO_ID:string, status:int)
Item(item_ID:string, item_category:string)
orders(cust_ID:string, UO_ID:string)
distance(cust_ID:string, cour_ID:string, res_ID:string, cust_loc:string,
                    cour_loc:string, res_loc:string)
cust_fav_res(cust_ID:string, res_ID:string)
fav_cour(rest_ID:string, cour_ID:string)
```

e) Write formulas of relational calculus to answer the following queries.

We dont have access to the rating that a customer gives after the delivery, therefore we are looking at the couriers and restaurants avg rating.

(i) Which united orders are available for a given customer based on her/his location?

```
EXISTS cour_loc, res_loc, cour_ID, cust_ID, res_ID, type.
Customer(_, cust_name, cust_ID) AND
Restaurant(_, res_name, res_ID, _) AND
Courier(cour_name, cour_ID, _) AND
distance(cust_ID, cour_ID, res_ID, cust_loc, cour_loc, res_loc)
AND cust_loc = cour_loc
AND cust_loc = res_loc
```

(ii) Which restaurants have been tagged as favorites and have received a poor rating from the same customer?

```
EXISTS cust_ID, res_ID, type.
Customer(_, cust_name, cust_ID) AND
Restaurant(_, res_name, res_ID, rating) AND
rating < 3 AND
cust_fav_res(cust_ID, res_ID)
```

(iii) Which couriers have been tagged as favorites by some restaurant and have received a poor rating from some customer?

```
EXISTS cour_ID, res_ID, type.
Restaurant(_, res_name, res_ID, _) AND
Courier(cour_name, cour_ID, co_rating) AND
fav_cour(res_ID, cour_ID) AND
co_rating < 3
```

(iv) Which customers gave a poor rating to all involved parties in a united order?

```
EXISTS type.
Courier(cour_name, _, co_rating) AND
Restaurant(_, res_name, _, res_rating) AND
co_rating < 3 AND
res_rating < 3
```

Exercise 2

a)

What PC models have a ram of 512 MB?

```
EXISTS PC.
PC(model, _, 512, _, _)
```

b)

Which manufacturers make laptops with a screen size of 15”?

```
EXISTS model.
Laptop(model, _, _, _, 15, _) AND
Product(maker, model, "laptop")
```

c)

Find the model number and price of all products (of any type) made by manufacturer E.

```
# c)
EXISTS x.
Product("E", model, _) AND
(Laptop(model, _, _, _, _, price) OR
PC(model, _, _, _, price) OR
Printer(model, _, _, price))
```

d)

Find the makers, model numbers, and prices of all offered devices.

```
# d)
EXISTS x.
(Laptop(model, _, _, _, _, price) OR
PC(model, _, _, _, price) OR
Printer(model, _, _, price)) AND
Product(maker, model, _)
```

e)

Find the model numbers of all color laser printers that are not made by manufacturer E.

```
# e)
EXISTS x.
Printer(model, _, _, price) AND
NOT Product("E", model, "printer")
```

f)

Find those manufacturers that sell PCs, but not Printers. Do not rely on the correctness of the “type” column in Product for this query.

```
# f)
EXISTS model.
(PC(model, _, _, _, _)) AND
Product(maker, model, _) AND

NOT EXISTS model.
(Printer(model, _, _, _) AND
Product(maker, model, _))
```

g)

Find those screen sizes that occur in two or more laptops.

```
# g)
EXISTS model1, model2.
Laptop(model1, _, _, _, screen, _) AND
Laptop(model2, _, _, _, screen, _) AND
NOT model1 = model2
```

h)

Find the manufacturers of Laptops with at least three different hard disk sizes.

```
# h)
EXISTS model1, model2, model3, ram1, ram2, ram3.
Laptop(model1, _, _, ram1, _, _) AND
Laptop(model2, _, _, ram2, _, _) AND
Laptop(model3, _, _, ram3, _, _) AND
Product(maker, model1, "laptop") AND
Product(maker, model2, "laptop") AND
Product(maker, model3, "laptop") AND
NOT ram1 = ram2 AND
NOT ram1 = ram3 AND
NOT ram2 = ram3
```

i)

Find the manufacturers who sell exactly three different Printer models.

```
# i)
FORALL type.
(EXISTS x. Printer(_, _, type, _))
IMPLIES (EXISTS model1, model2, model3.
Product(maker, model1, "printer") AND
Product(maker, model2, "printer") AND
Product(maker, model3, "printer") AND
NOT model1 = model2 AND
NOT model1 = model3 AND
NOT model2 = model3)
```

j)

Find those makers that manufacture every type of printer. Do not rely on the fact that there are only two types of printers in the given instance.

```
# j)
FORALL type.
(EXISTS x. Printer(_, _, type, _))
IMPLIES (EXISTS model. Product(maker, model, _)
AND Printer(model, _, type, _))
```

k)

Validate that the “type” column in Product is correct, i.e., only includes correct labels for PCs, Laptops, and Printers with the respective model number. Your query should be a closed formula that outputs the empty set if and only if the correctness is violated.

```
# k)
NOT EXISTS model, type.
(Product(_, model, type) AND
NOT ((PC(model, _, _, _, _) AND
type = "pc") OR

(Laptop(model, _, _, _, _, _) AND
type = "laptop") OR

(Printer(model, _, _, _) AND
type = "printer")))
```