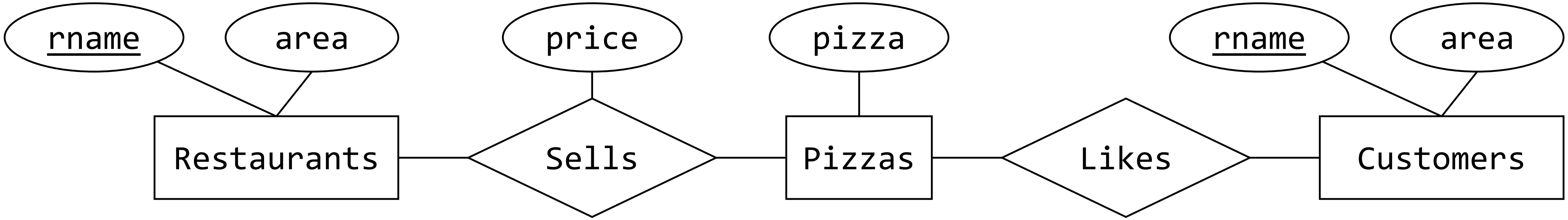
CS2102

Tutorial 04

Question 8

- Simplified ER diagram



Question 8(a)

- Question: Find all pizzas that ‘Alice’ likes but ‘Bob’ does not like.
- Separate the question:
  - Find all pizzas that ‘Alice’ likes
  - Find all pizzas that ‘Bob’ like

Question 8(a)

- Question: Find all pizzas that ‘Alice’ likes but ‘Bob’ does not like.
- Separate the question:
  - Find all pizzas that ‘Alice’ likes

```
SELECT pizza FROM Likes WHERE cname = ‘Alice’
```

  - Find all pizzas that ‘Bob’ like

```
SELECT pizza FROM Likes WHERE cname = ‘Bob’
```

Question 8(a)

- Question: Find all pizzas that ‘Alice’ likes but ‘Bob’ does not like.
- Separate the question:
  - Find all pizzas that ‘Alice’ likes

    ```
    SELECT pizza FROM Likes WHERE cname = ‘Alice’
    ```

  - Find all pizzas that ‘Bob’ like

    ```
    SELECT pizza FROM Likes WHERE cname = ‘Bob’
    ```

  - Get the difference!

Question 8(a)

- Question: Find all pizzas that ‘Alice’ likes but ‘Bob’ does not like.
- Separate the question:
  - Get the difference!

```
SELECT pizza FROM Likes WHERE cname = ‘Alice’
EXCEPT
SELECT pizza FROM Likes WHERE cname = ‘Bob’
```

Question 8(b)

- Question: For each customer, find the pizzas sold by restaurants that are located in the same area as the customer’s area. Exclude customers whose associated set of pizzas is empty.
    - Find what?
    - From which table?

Question 8(b)

- Question: For each customer, find the pizzas sold by restaurants that are located in the same area as the customer's area. Exclude customers whose associated set of pizzas is empty.
  - Find what?
    - Customer name and pizza
  - From which table?
    - Customer name is found in **`Customers`** table
    - Area of customer needs to be matched with restaurant are from **`Restaurants`** table
    - We need to know which pizza sold by which restaurant from **`Sells`** table

Question 8(b)

- Question: For each customer, find the pizzas sold by restaurants that are located in the same area as the customer’s area. Exclude customers whose associated set of pizzas is empty.
  - Simple Query:

```
SELECT DISTINCT C.cname, S.pizza
FROM Customers C, Restaurants R, Sells S
WHERE  C.area = R.area AND R.rname = S.rname;
```

Question 8(b)

- Question: For each customer, find the pizzas sold by restaurants that are located in the same area as the customer’s area. Exclude customers whose associated set of pizzas is empty.
  - Alternative Query: *inner join*

```
SELECT DISTINCT C.cname, S.pizza
FROM ( Customers C INNER JOIN Restaurants R ON C.area = R.area )
     INNER JOIN Sells S ON R.rname = S.rname;
```

Question 8(b)

- Question: For each customer, find the pizzas sold by restaurants that are located in the same area as the customer’s area. Exclude customers whose associated set of pizzas is empty.
  - Alternative Query: *inner join* (does order matter?)

```
SELECT DISTINCT C.cname, S.pizza
FROM ( Sells S INNER JOIN Restaurants R ON R.rname = S.rname )
     INNER JOIN Customers C ON C.area = R.area;
```

Question 8(b)

- Question: For each customer, find the pizzas sold by restaurants that are located in the same area as the customer’s area. Exclude customers whose associated set of pizzas is empty.
  - Alternative Query: *natural join*

```
SELECT DISTINCT C.cname, S.pizza
FROM Customers C NATURAL JOIN Restaurants R NATURAL JOIN Sells S;
```

Question 8(c)

- Question: For each customer, find the pizzas sold by restaurants that are located in the same area as the customer’s area. Include customers whose associated set of pizzas is empty.
    - Find what?
    - From which table?

Question 8(c)

- Question: For each customer, find the pizzas sold by restaurants that are located in the same area as the customer's area. Include customers whose associated set of pizzas is empty.
  - Find what?
    - Customer name and pizza
    - *Customers whose associated pizza is empty!*
  - From which table?
    - Customer name is found in **`Customers`** table
    - Area of customer needs to be matched with restaurant are from **`Restaurants`** table
    - We need to know which pizza sold by which restaurant from **`Sells`** table

Question 8(c)

- Question: For each customer, find the pizzas sold by restaurants that are located in the same area as the customer’s area. Include customers whose associated set of pizzas is empty.
  - Find what?
    - Customer name and pizza
    - *Customers whose associated pizza is empty!*
  - From which table?
    - Customer name is found in **`Customers`** table
    - Area of customer needs to be matched with restaurant are from **`Restaurants`** table
    - We need to know which pizza sold by which restaurant from **`Sells`** table
  - Try the simplest solution: *natural join*

Question 8(c)

- Question: For each customer, find the pizzas sold by restaurants that are located in the same area as the customer's area. Include customers whose associated set of pizzas is empty.
  - Try the simplest solution: *natural join*

```
SELECT DISTINCT C.cname, S.pizza
FROM Customers C NATURAL JOIN
     Restaurants R NATURAL JOIN Sells S;
```

Question 8(c)

- Question: For each customer, find the pizzas sold by restaurants that are located in the same area as the customer’s area. Include customers whose associated set of pizzas is empty.
  - Try the simplest solution: *natural join*

```
SELECT DISTINCT C.cname, S.pizza
FROM Customers C NATURAL JOIN
     Restaurants R NATURAL JOIN Sells S;
```

  - *Dangling tuple is not preserved*

Question 8(c)

- Question: For each customer, find the pizzas sold by restaurants that are located in the same area as the customer's area. Include customers whose associated set of pizzas is empty.
  - Try the simplest solution: *natural join*

    ```
    SELECT DISTINCT C.cname, S.pizza
    FROM Customers C NATURAL JOIN
         Restaurants R NATURAL JOIN Sells S;
    ```
  - *Dangling tuple is not preserved*
    - How to preserve: **left outer join**
    - There's actually `NATURAL LEFT JOIN`!

Question 8(c)

- Question: For each customer, find the pizzas sold by restaurants that are located in the same area as the customer's area. Include customers whose associated set of pizzas is empty.
  - Try the simplest solution: *natural join*

```
SELECT DISTINCT C.cname, S.pizza
FROM Customers C NATURAL LEFT JOIN
     Restaurants R NATURAL JOIN Sells S;
```

  - *This does not work!*
    - The order of operation is: $(C \rightarrow R) \bowtie S$
      - Where $\rightarrow$ is the NATURAL LEFT JOIN
    - Then the dangling tuple is lost from the last $\bowtie$ operation!
      - Invert the order to $C \rightarrow (R \bowtie S)$

Question 8(c)

- Question: For each customer, find the pizzas sold by restaurants that are located in the same area as the customer’s area. Include customers whose associated set of pizzas is empty.
  - Try the simplest solution: *natural join*

```
SELECT DISTINCT C.cname, S.pizza
FROM Customers C NATURAL LEFT JOIN
   ( Restaurants R NATURAL JOIN Sells S );
```

  - *Order of operation matters!*

Question 8(c)

- Question: For each customer, find the pizzas sold by restaurants that are located in the same area as the customer's area. Include customers whose associated set of pizzas is empty.
  - Try the simplest solution: *natural join*

```
SELECT DISTINCT C.cname, S.pizza
FROM Customers C NATURAL LEFT JOIN
   ( Restaurants R NATURAL JOIN Sells S );
```

  - Other answers are simply variations of this

Question 8(c)

- Question: For each customer, find the pizzas sold by restaurants that are located in the same area as the customer's area. Include customers whose associated set of pizzas is empty.
  - Alternative solution: *from natural left join to left join*

```
SELECT DISTINCT C.cname, S.pizza
FROM Customers C LEFT JOIN
   ( Restaurants R NATURAL JOIN Sells S )
     ON C.area = R.area;
```

Question 8(c)

- Question: For each customer, find the pizzas sold by restaurants that are located in the same area as the customer’s area. Include customers whose associated set of pizzas is empty.
  - Alternative solution: *from natural join to inner join*

```
SELECT DISTINCT C.cname, S.pizza
FROM Customers C LEFT JOIN
   ( Restaurants R INNER JOIN Sells S ON R.rname = S.rname )
     ON C.area = R.area;
```

Question 8(d)

- Question: Create the Dislikes relation as a VIEW.
  - Find what?
  - From which table?

Question 8(d)

- Question: Create the Dislikes relation as a VIEW.
  - Find what?
    - Customer name and pizza
  - From which table?
    - What customer likes can be obtained from Likes table

Question 8(d)

- Question: Create the Dislikes relation as a VIEW.
    - Find what?
        - Customer name and pizza
    - From which table?
        - What customer likes can be obtained from Likes table
        - But also need all customer and pizza pair
            - From Customers table and Pizza table

Question 8(d)

- Question: Create the Dislikes relation as a VIEW.
  - Find what?
    - Customer name and pizza
  - From which table?
    - What customer likes can be obtained from Likes table
    - But also need all customer and pizza pair
      - From Customers table and Pizza table
  - Simple Query:

```
SELECT C.cname, P.pizza FROM Customers C, Pizzas P
EXCEPT
SELECT * FROM Likes;
```

Question 9

Q1

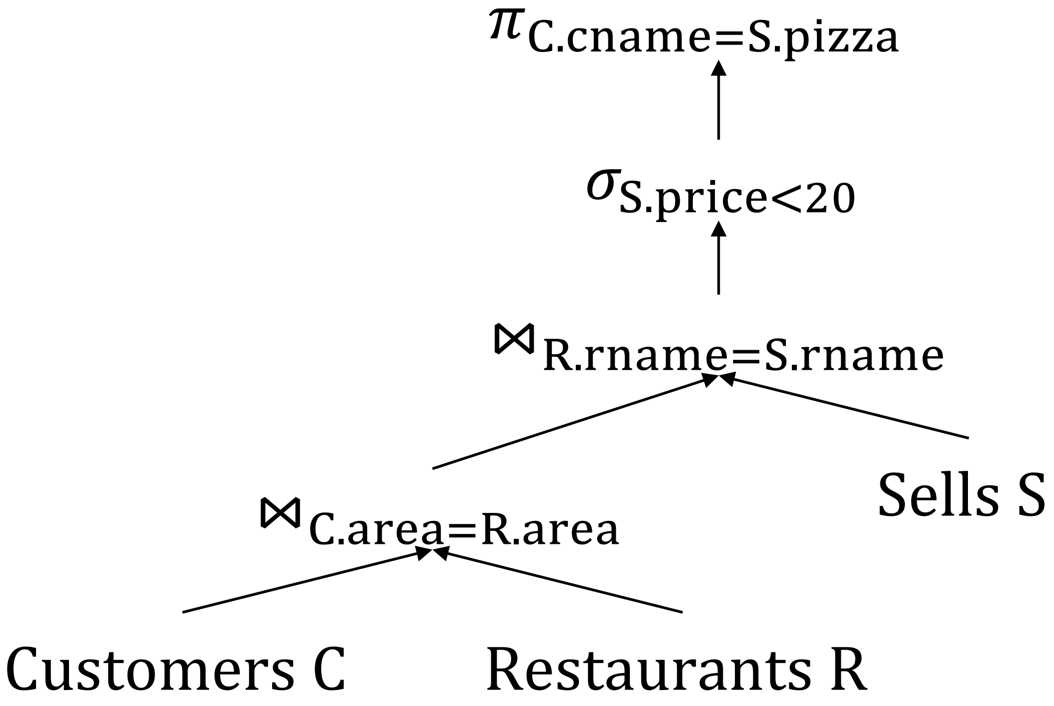
```
SELECT DISTINCT C.cname,
S.pizza
FROM   ( Customers C JOIN
Restaurants R ON C.area =
R.area )
       JOIN Sells S ON
R.rname = S.rname;
WHERE  price < 20;
```

Q2

```
SELECT DISTINCT C.cname,
S.pizza
FROM   ( Customers C JOIN
Restaurants R on C.area =
R.area )
       JOIN Sells S ON
R.rname = S.rname
       AND price < 20;
```

Question 9

Q1



$\pi_{\text{C.cname=S.pizza}}$

$\sigma_{\text{S.price<20}}$

$\bowtie_{\text{R.rname=S.rname}}$

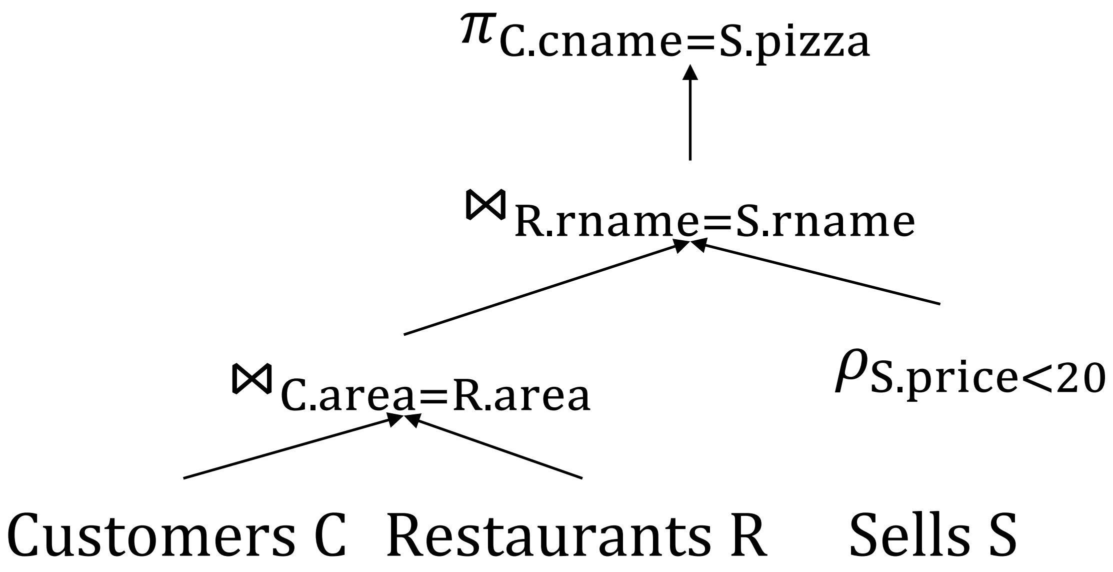
$\bowtie_{\text{C.area=R.area}}$

Sells S

Customers C

Restaurants R

Q2



$\pi_{\text{C.cname=S.pizza}}$

$\bowtie_{\text{R.rname=S.rname}}$

$\bowtie_{\text{C.area=R.area}}$

$\rho_{\text{S.price<20}}$

Customers C

Restaurants R

Sells S

Question 9

- How to compare?
  - Write it down first!

- Q1:
  - $\pi_{\mathrm{C.cname,S.pizza}}\left(\sigma_{\mathrm{S.price<20}}\left((\mathrm{C} \bowtie_{\mathrm{C.area=R.area}} \mathrm{R}) \bowtie_{\mathrm{R.rname,S.rname}} \mathrm{S}\right)\right)$
- Q2
  - $\pi_{\mathrm{C,cname,S.pizza}}\left((\mathrm{C} \bowtie_{\mathrm{C.area=R.area}} \mathrm{R}) \bowtie_{\mathrm{R.rname=S.rname}} \left(\sigma_{\mathrm{S.price<20}}(\mathrm{S})\right)\right)$

Question 9

- How to compare?
  - Remove outermost operators if they are the same

- Q1:
  - $\pi_{\text{C.cname,S.pizza}}\left(\sigma_{\text{S.price<20}}\left((\text{C} \bowtie_{\text{C.area=R.area}} \text{R}) \bowtie_{\text{R.rname,S.rname}} \text{S}\right)\right)$
- Q2
  - $\pi_{\text{C,cname,S.pizza}}\left((\text{C} \bowtie_{\text{C.area=R.area}} \text{R}) \bowtie_{\text{R.rname=S.rname}} \left(\sigma_{\text{S.price<20}}(\text{S})\right)\right)$

Question 9

- How to compare?
    - Remove outermost operators if they are the same

- Q1:
    - $\left(\sigma_{\text{S.price}<20}\left((\text{C} \bowtie_{\text{C.area}=\text{R.area}} \text{R}) \bowtie_{\text{R.rname,S.rname}} \text{S}\right)\right)$
- Q2
    - $\left((\text{C} \bowtie_{\text{C.area}=\text{R.area}} \text{R}) \bowtie_{\text{R.rname}=\text{S.rname}} \left(\sigma_{\text{S.price}<20}(\text{S})\right)\right)$

Question 9

- How to compare?
  - Simplify
  - Let $c_1\colon \mathrm{S.price} < 20$, $c_2\colon \mathrm{C.area} = \mathrm{R.area}$, and $c_3\colon \mathrm{R.rname} = \mathrm{S.rname}$

- Q1:
  - $\left(\sigma_{\mathrm{S.price<20}}\left((\mathrm{C} \bowtie_{\mathrm{C.area=R.area}} \mathrm{R}) \bowtie_{\mathrm{R.rname,S.rname}} \mathrm{S}\right)\right)$
- Q2
  - $\left((\mathrm{C} \bowtie_{\mathrm{C.area=R.area}} \mathrm{R}) \bowtie_{\mathrm{R.rname=S.rname}} \left(\sigma_{\mathrm{S.price<20}}(\mathrm{S})\right)\right)$

Question 9

- How to compare?
  - Simplify
  - Let $c_1\!: \mathrm{S.price} < 20$, $c_2\!: \mathrm{C.area} = \mathrm{R.area}$, and $c_3\!: \mathrm{R.rname} = \mathrm{S.rname}$

- Q1:
  - $\left(\sigma_{\mathrm{c}_1}\left((\mathrm{C} \bowtie_{\mathrm{c}_2} \mathrm{R}) \bowtie_{\mathrm{c}_3} \mathrm{S}\right)\right)$
- Q2
  - $\left((\mathrm{C} \bowtie_{\mathrm{c}_2} \mathrm{R}) \bowtie_{\mathrm{c}_3} \left(\sigma_{\mathrm{c}_1}(\mathrm{S})\right)\right)$

Question 9

- How to compare?
  - Simplify further
  - Let $A: C \bowtie_{c_2} R$

- Q1:
  - $\left(\sigma_{\mathrm{c}_1}\left(\left(\mathrm{C} \bowtie_{\mathrm{c}_2} \mathrm{R}\right) \bowtie_{\mathrm{c}_3} \mathrm{S}\right)\right)$
- Q2
  - $\left(\left(\mathrm{C} \bowtie_{\mathrm{c}_2} \mathrm{R}\right) \bowtie_{\mathrm{c}_3} \left(\sigma_{\mathrm{c}_1}(\mathrm{S})\right)\right)$

Question 9

- How to compare?
  - Simplify further
  - Let $A: C \bowtie_{c_2} R$

- Q1:
  - $\left(\sigma_{\mathrm{c}_1}(A \bowtie_{\mathrm{c}_3} \mathrm{S})\right)$
- Q2
  - $\left(A \bowtie_{\mathrm{c}_3} \left(\sigma_{\mathrm{c}_1}(\mathrm{S})\right)\right)$

Question 9

- How to compare?
  - We then have to show that these two are equivalent!

- Q1:
  - $\left(\sigma_{\mathrm{c}_1}\left(A \bowtie_{\mathrm{c}_3} \mathrm{S}\right)\right)$
- Q2
  - $\left(A \bowtie_{\mathrm{c}_3} \left(\sigma_{\mathrm{c}_1}(\mathrm{S})\right)\right)$

Question 9

- How to compare?
  - Partition S into $\sigma_{c_1}(\mathrm{S}) \cup \left(\mathrm{S} - \sigma_{c_1}(\mathrm{S})\right)$
  - Then $A \bowtie_{\mathrm{c}_3} \mathrm{S} \equiv \left(A \bowtie_{\mathrm{c}_3} \sigma_{c_1}(\mathrm{S})\right) \cup \left(A \bowtie_{\mathrm{c}_3} \left(\mathrm{S} - \sigma_{c_1}(\mathrm{S})\right)\right)$
  - And $\sigma_{\mathrm{c}_1}\left(A \bowtie_{\mathrm{c}_3} \mathrm{S}\right) \equiv \sigma_{\mathrm{c}_1}\left(A \bowtie_{\mathrm{c}_3} \sigma_{c_1}(\mathrm{S})\right) \cup \sigma_{c_1}\left(A \bowtie_{\mathrm{c}_3} \left(\mathrm{S} - \sigma_{c_1}(\mathrm{S})\right)\right)$
  - But $\sigma_{c_1}\left(A \bowtie_{\mathrm{c}_3} \left(\mathrm{S} - \sigma_{c_1}(\mathrm{S})\right)\right) = \emptyset$
  - And $\sigma_{c_1}\left(A \bowtie_{\mathrm{c}_3} \sigma_{c_1}(\mathrm{S})\right) = A \bowtie_{\mathrm{c}_3} \sigma_{c_1}(\mathrm{S})$
  - Therefore $\sigma_{\mathrm{c}_1}\left(A \bowtie_{\mathrm{c}_3} \mathrm{S}\right) \equiv A \bowtie_{\mathrm{c}_3} \sigma_{c_1}(\mathrm{S})$

Question 9

- Diagrammatically

$$\sigma_{c_1}\left(\begin{array}{c} A \\ \begin{array}{|c|}\hline A \\ A \\ \hline\end{array}\end{array} \bowtie_{c_3} \begin{array}{c} S \\ \begin{array}{|c|}\hline \sigma_{c_1}(S) \\ \hline \neg\sigma_{c_1}(S) \\ \hline\end{array}\end{array} = \begin{array}{c} A \bowtie_{c_3} S \\ \begin{array}{|c|}\hline A \bowtie_{c_3} \sigma_{c_1}(S) \\ \hline A \bowtie_{c_3} \sigma_{\neg c_1}(S) \\ \hline\end{array}\end{array}\right) = \begin{array}{c} \sigma_{c_1}(A \bowtie_{c_3} S) \\ \begin{array}{|c|}\hline \sigma_{c_1}\left(A \bowtie_{c_3} \sigma_{c_1}(S)\right) \\ \hline {\color{red}\sigma_{c_1}\left(A \bowtie_{c_3} \sigma_{\neg c_1}(S)\right)} \\ \hline\end{array}\end{array} = \begin{array}{c} \sigma_{c_1}(A \bowtie_{c_3} S) \\ \begin{array}{|c|}\hline A \bowtie_{c_3} \sigma_{c_1}(S) \\ \hline {\color{red}\emptyset} \\ \hline\end{array}\end{array}$$

Question 10(a)

- Question: What are the constraints that are not captured by this design?
  - Patients are identified by an SSN, and their names, addresses, and ages must be recorded.
  - Doctors are identified by an SSN. For each doctor, the name, specialty, and years of experience must be recorded.
  - Each pharmaceutical company is identified by name and has a phone number.

Question 10(a)

- Question: What are the constraints that are not captured by this design?
    - Patients are identified by an SSN, and their names, addresses, and ages must be recorded.
        - This is `Patients` entity-set
    - Doctors are identified by an SSN. For each doctor, the name, specialty, and years of experience must be recorded.
        - This is `Doctors` entity-set
    - Each pharmaceutical company is identified by name and has a phone number.
        - This is `PharmaceuticalCompanies` entity-set

Question 10(a)

- Question: What are the constraints that are not captured by this design?
  - For each drug, the trade name, and formula must be recorded. Each drug is sold by a given pharmaceutical company, and the trade name identifies a drug uniquely from among the products of that company. If a pharmaceutical company is deleted, you need not keep track of its products any longer.
  - Each pharmacy has a name, address, and phone number.

Question 10(a)

- Question: What are the constraints that are not captured by this design?
  - For each drug, the trade name, and formula must be recorded. Each drug is sold by a given pharmaceutical company, and the trade name identifies a drug uniquely from among the products of that company. If a pharmaceutical company is deleted, you need not keep track of its products any longer.
    - This is `Drugs` entity-set
  - Each pharmacy has a name, address, and phone number.
    - This is `Pharmacy` entity-set

Question 10(a)

- Question: What are the constraints that are not captured by this design?
  - For each drug, the trade name, and formula must be recorded. Each drug is sold by a given pharmaceutical company, and the trade name identifies a drug uniquely from among the products of that company. If a pharmaceutical company is deleted, you need not keep track of its products any longer.
    - ~~This is Drugs entity-set~~
    - Actually Drugs should be weak-entity-set
  - Each pharmacy has a name, address, and phone number.
    - This is Pharmacy entity-set

Question 10(a)

- Question: What are the constraints that are not captured by this design?
  - Every patient has a primary physician. Every doctor is the primary physician of at least one patient.
  - Each pharmacy sells several drugs and has a price for each. A drug could be sold at several pharmacies, and the price could vary from one pharmacy to another.

Question 10(a)

- Question: What are the constraints that are not captured by this design?
  - Every patient has a primary physician. Every doctor is the primary physician of at least one patient.
    - This is `PrimaryPhysician` relationship-set between `Patients` and `Doctors`
  - Each pharmacy sells several drugs and has a price for each. A drug could be sold at several pharmacies, and the price could vary from one pharmacy to another.
    - This is `Sells` relationship-set between `Pharmacies` and `Drugs`

Question 10(a)

- Question: What are the constraints that are not captured by this design?
  - Every patient has a primary physician. Every doctor is the primary physician of at least one patient.
    - This is `PrimaryPhysician` relationship-set between `Patients` and `Doctors`
    - Total participation constraints of `Doctors` is not captured
    - Key & total participation constraints of `Patients` is not captured
  - Each pharmacy sells several drugs and has a price for each. A drug could be sold at several pharmacies, and the price could vary from one pharmacy to another.
    - This is `Sells` relationship-set between `Pharmacies` and `Drugs`
    - But “Each pharmacy sells several drugs” is not captured

Question 10(a)

- Question: What are the constraints that are not captured by this design?
  - Doctors prescribe drugs for patients. A doctor could prescribe one or more drugs for several patients, and a patient could obtain prescriptions from several doctors. Each prescription has a date and a quantity associated with it. You can assume that, if a doctor prescribes the same drug for the same patient more than once, only the last such prescription needs to be stored.

Question 10(a)

- Question: What are the constraints that are not captured by this design?
  - Doctors prescribe drugs for patients. A doctor could prescribe one or more drugs for several patients, and a patient could obtain prescriptions from several doctors. Each prescription has a date and a quantity associated with it. You can assume that, if a doctor prescribes the same drug for the same patient more than once, only the last such prescription needs to be stored.
    - This is `Prescribes` relationship-set between `Patients`, `Doctors`, and `Drugs`

Question 10(a)

- Question: What are the constraints that are not captured by this design?
  - Doctors prescribe drugs for patients. A doctor could prescribe one or more drugs for several patients, and a patient could obtain prescriptions from several doctors. Each prescription has a date and a quantity associated with it. You can assume that, if a doctor prescribes the same drug for the same patient more than once, only the last such prescription needs to be stored.
    - This is `Prescribes` relationship-set between `Patients`, `Doctors`, and `Drugs`
    - Luckily, “only the last such prescription needs to be stored”
    - Since the triples (`patients`, `doctors`, `drugs`) should be unique

Question 10(a)

- Question: What are the constraints that are not captured by this design?
  - There is exactly one contract between a pharmacy and a pharmaceutical company if and only if that pharmacy sells some drug that is made by that pharmaceutical company. For each contract, you have to store a start date, an end date, and the text of the contract.

Question 10(a)

- Question: What are the constraints that are not captured by this design?
  - There is exactly one contract between a pharmacy and a pharmaceutical company if and only if that pharmacy sells some drug that is made by that pharmaceutical company. For each contract, you have to store a start date, an end date, and the text of the contract.
    - This is `Contracts` relationship-set between `Pharmacies` and `PharmaComp`

Question 10(a)

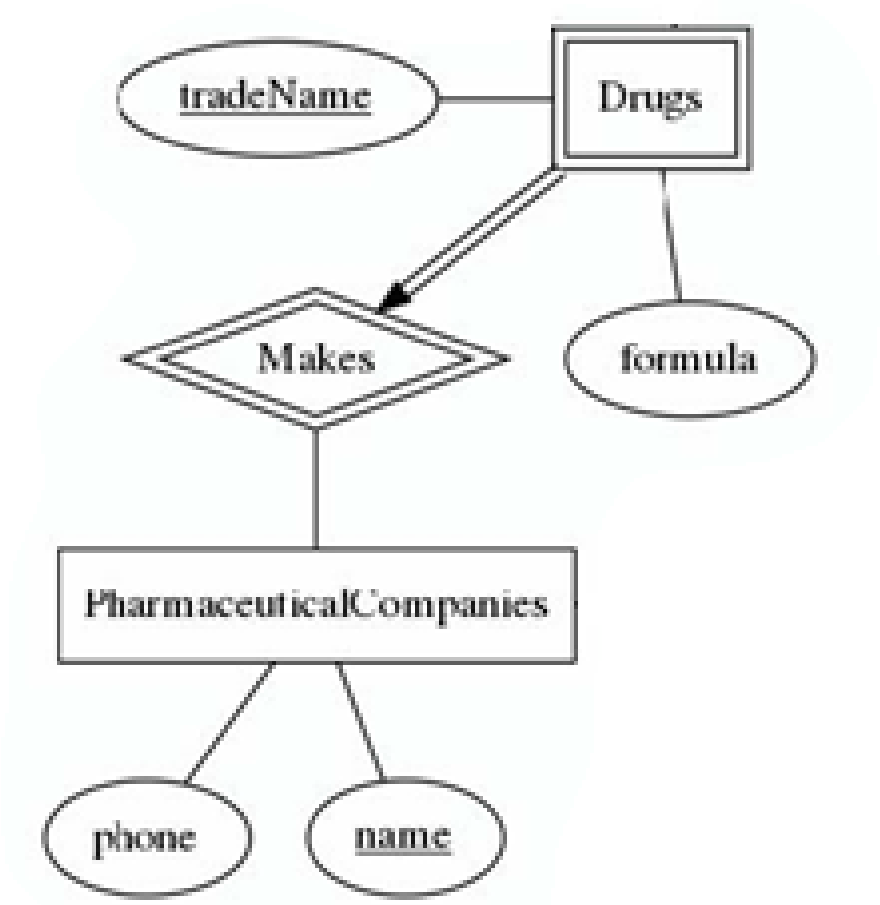
- Question: What are the constraints that are not captured by this design?
  - There is exactly one contract between a pharmacy and a pharmaceutical company if and only if that pharmacy sells some drug that is made by that pharmaceutical company. For each contract, you have to store a start date, an end date, and the text of the contract.
    - This is `Contracts` relationship-set between `Pharmacies` and `PharmaComp`
    - But does not capture "pharmacy sells some drug that is made by that pharmaceutical company"
    - That is from another table!

Question 10(a)

- Question: What are the constraints that are not captured by this design?
  - Not captured:
    1. `Drugs` as weak-entity-set
    2. Total participation constraints of `Doctors` w.r.t. `PrimaryPhysician`
    3. Key & total participation constraints `Patients` w.r.t. `PrimaryPhysician`
    4. Pharmacy sells more than one drug
    5. Exactly one contract if and only if pharmacy sells some drugs by the company

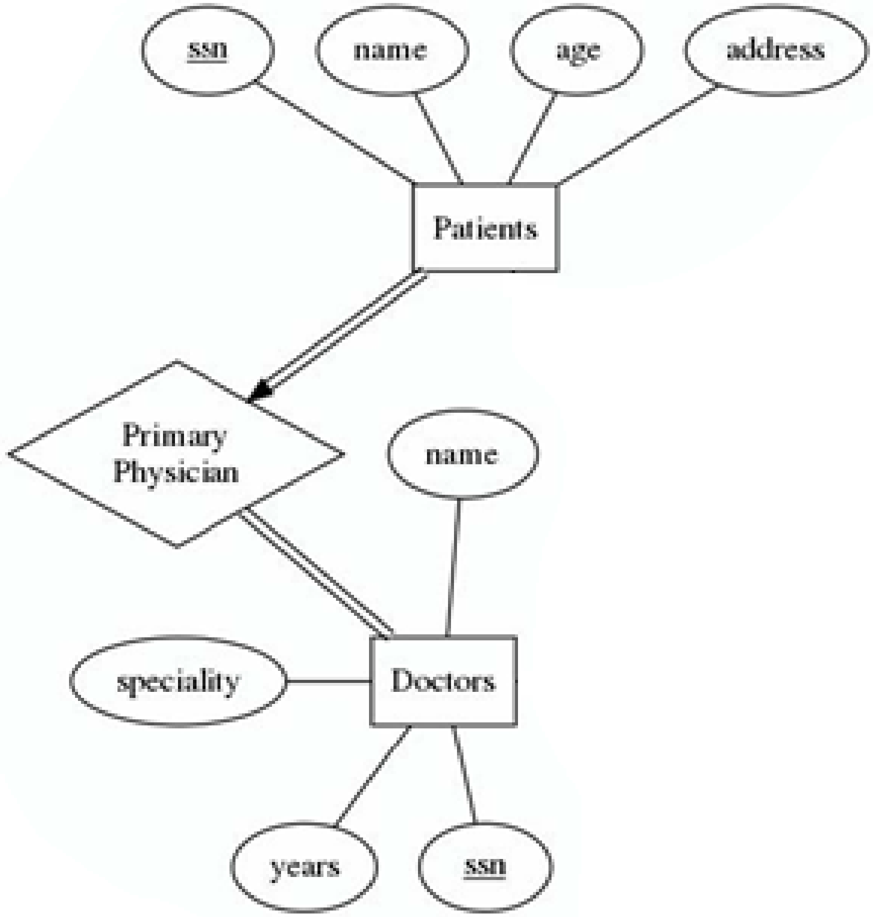
Question 10(a)

- Question: What are the constraints that are not captured by this design?
  - Drugs as weak-entity-set



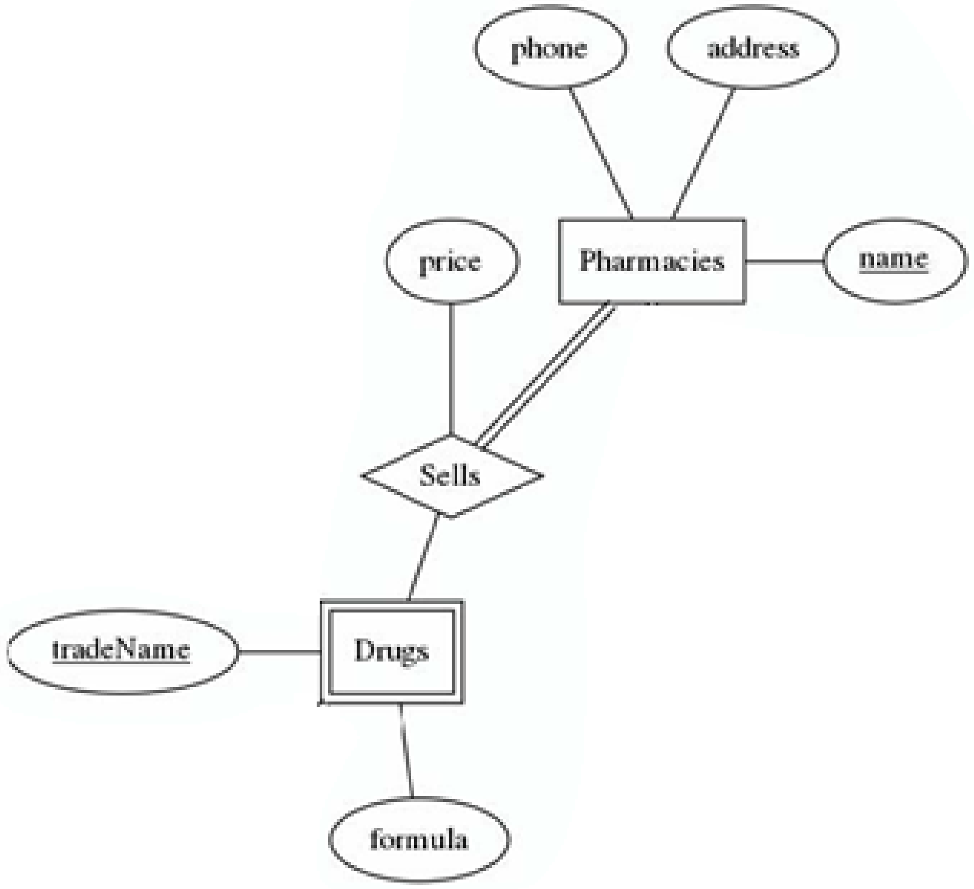
Question 10(a)

- Question: What are the constraints that are not captured by this design?
  - Total participation constraints of `Doctors` w.r.t. `PrimaryPhysician`
  - Key & total participation constraints of `Patients` w.r.t. `PrimaryPhysician`



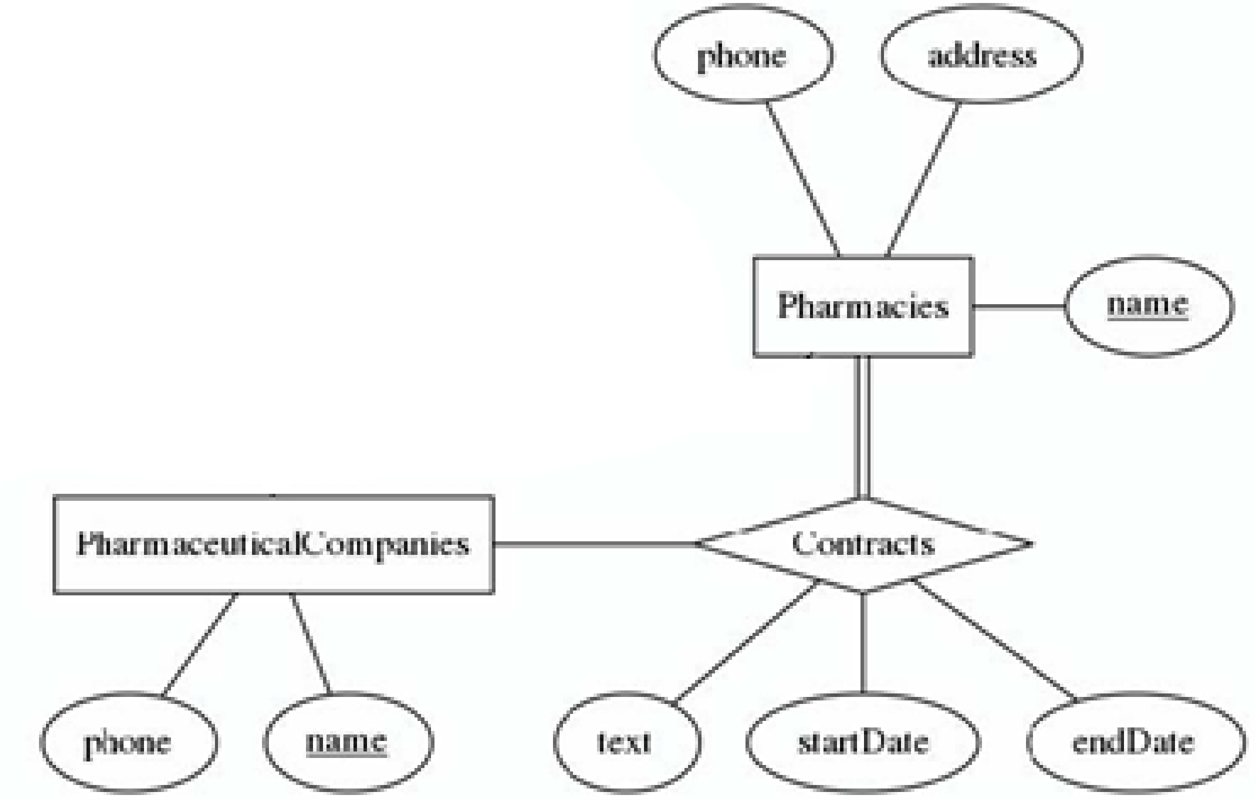
Question 10(a)

- Question: What are the constraints that are not captured by this design?
  - Pharmacy sells more than one drug
  - NOT FULLY CAPTURED!



Question 10(a)

- Question: What are the constraints that are not captured by this design?
  - Exactly one contract
    if and only if
    pharmacy sells some drugs
    by the company
  - NOT FULLY CAPTURED!


phone
address
Pharmacies
name
PharmaceuticalCompanies
Contracts
phone
name
text
startDate
endDate