CS2102 Tutorial 4: SQL 2

Wk 6, Sem 2, 2020/21

1. Consider a database consisting of the following two tables shown below.

bar

a	b
1	10
2	20
3	30
4	40

foo

f	a
100	2
200	7
300	3
400	2

For each of the following queries on the database, either state that the query is an invalid SQL query or show the query's output if the query is a valid SQL query.

```
select *
from bar b
where exists (
    select 1
    from foo f
    where f.f > 100
    and f.a = b.a
);
```

(a)

```
select *
from bar b
where exists (
    select 1
    from foo f
    where f.f > 100
)
and f.a = b.a;
```

(b)

```
select *
from bar b
where exists (
    select 1
    from foo f
    where f.f > 100
    and a = b.a
);
```

(c)

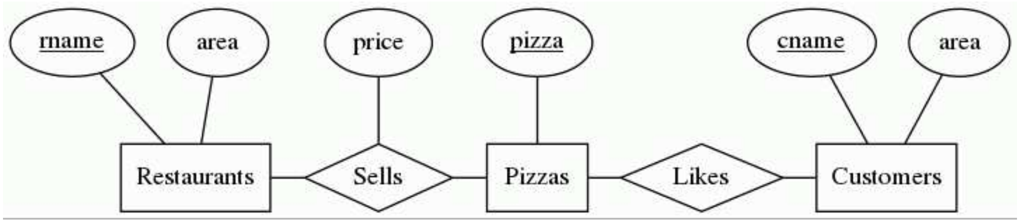
```
select *
from bar b
where exists (
    select 1
    from foo f
    where f > 100
    and a = a
);
```

(d)

```
select *
from bar b
where exists (
    select 1
    from foo f
    where f.f > 100
    and f.a = b.a
    and b > 20
);
```

(e)

2. Questions 2 to 5 are based on the pizza database schema used in the lectures; we show its ER diagram below.



For each of the following queries, write an equivalent SQL query that does not use any subquery.

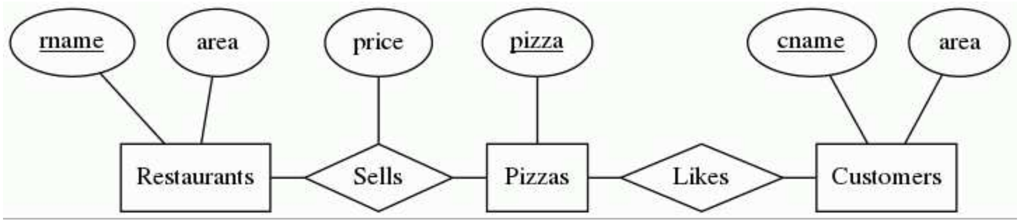
(a)
```
select distinct cname
from Likes L
where exists (
    select 1
    from Sells S
    where S.rname = 'Corleone␣Corner'
    and S.pizza = L.pizza
);
```

(b)
```
select cname
from Customers C
where not exists (
    select 1
    from Likes L, Sells S
    where S.rname = 'Corleone␣Corner'
    and S.pizza = L.pizza
    and L.cname = C.cname
);
```

(c)
```
select distinct rname
from Sells
where rname <> 'Corleone␣Corner'
and price > any (
    select price
    from Sells
    where rname = 'Corleone␣Corner'
);
```

(d)
```
select rname, pizza, price
from Sells  S
where price >= all (
    select S2.price
    from Sells  S2
    where S2.rname = S.rname
    and S2.price is not null
);
```

3. Write a SQL query to answer each of the following questions on the pizza database. Remove duplicate records from all query results.



   (a) Find pizzas that Alice likes but Bob does not like.

   (b) Find pizzas that are sold by at most one restaurant in each area; exclude pizzas that are not sold by any restaurant.

   (c) Find all tuples $(A, P, Pmin)$ where P is a pizza that is available in area A (i.e., there is some restaurant in area A selling pizza P), and $Pmin$ is the lowest price of P in area A.

   (d) Find all tuples $(A, P, Pmin, Pmax)$ where P is a pizza that is available in area A, $Pmin$ is the lowest price of P in area A, and $Pmax$ is the highest price of P in area A.

4. Consider the query to find distinct restaurants that are located in the East area. The following are two possible SQL answers (denoted by Q1 and Q2) for this query.

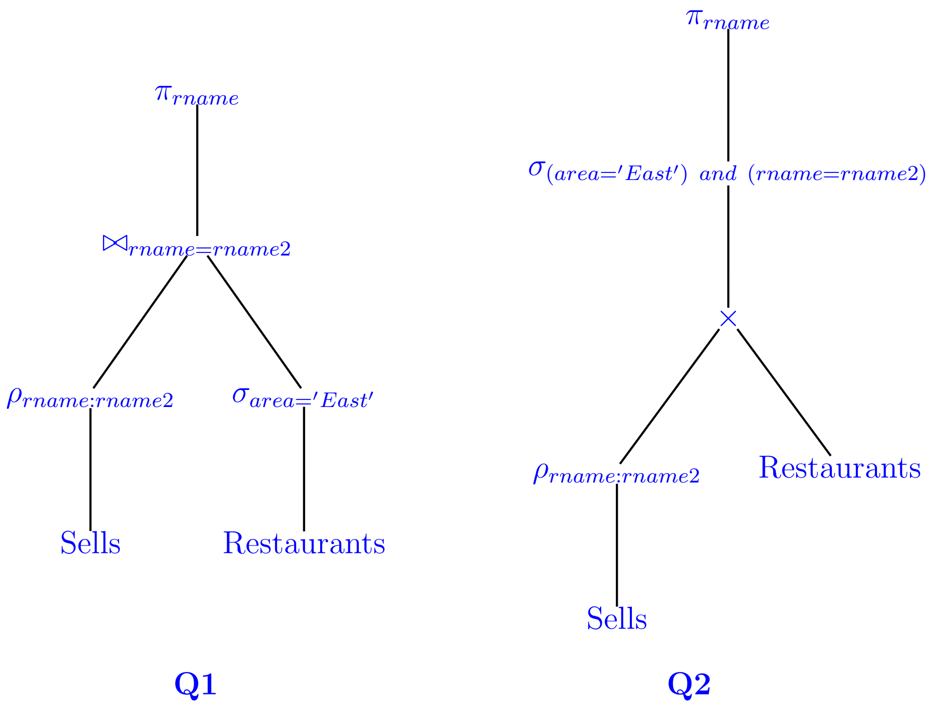
   Q1:

```
select distinct S.rname
from Sells S join Restaurants R on S.rname = R.rname
    and R.area = 'East';
```

   Q2:

```
select distinct S.rname
from Sells S, Restaurants R
where S.rname = R.rname
and R.area = 'East';
```

   The semantics of these two SQL queries are defined by the relational algebra expressions shown below. Discuss whether Q1 and Q2 are equivalent queries.



5. Consider the query to find distinct restaurants that are located in the East area or restaurants that sell some pizza that Lisa likes, where restaurants that do not sell any pizza are to be excluded. The following are two possible SQL answers (denoted by Q1 and Q2) for this query.

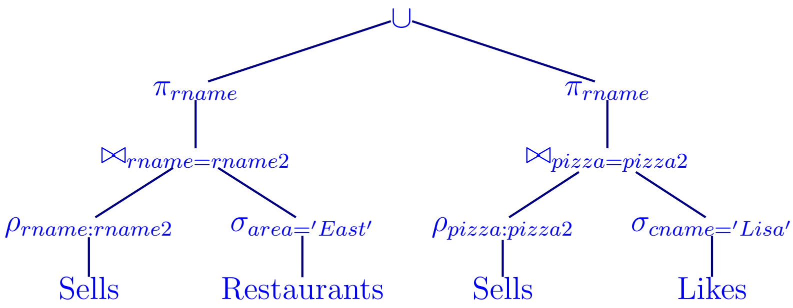
Q1:

```
select distinct S.rname
from Sells S join Restaurants R on S.rname = R.rname
    and R.area = 'East'
union
select distinct S.rname
from Sells S join Likes L on S.pizza = L.pizza
    and L.cname = 'Lisa';
```

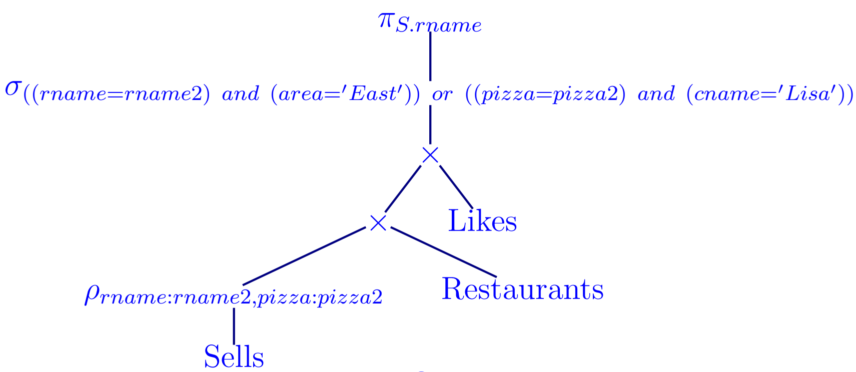
Q2:

```
select distinct S.rname
from Sells S, Restaurants R, Likes L
where (S.rname = R.rname and R.area = 'East')
or (S.pizza = L.pizza and L.cname = 'Lisa');
```

The semantics of these two SQL queries are defined by the relational algebra expressions shown below. Discuss whether Q1 and Q2 are equivalent queries.



Q1



Q2

6. Consider again the following relational schema discussed in Tutorial 2.

```
create table Offices (
    office_id   integer,
    building    text not null,
    level       integer not null,
    room_number integer not null,
    area        integer,
    primary key (office_id),
    unique (building, level, room_number)
);

create table Employees (
    emp_id      integer,
    name        text not null,
    office_id   integer not null,
    manager_id  integer,
    primary key (emp_id),
    foreign key (office_id) references Offices (office_id)
        on update cascade,
    foreign key (manager_id) references Employees (emp_id)
        on update cascade
);
```

Suppose that the office with `officeId` = 123 needs to be renovated. Write a SQL statement to reassign the employees located in this office to another temporary office located at room number 11 on level 5 at the building named *Tower1*.

7. Given the tables R and S shown below, compute the output of each of the following queries.

   (a) **select** $*$ **from** R **natural join** S;

   (b) **select** $*$ **from** R **inner join** S **on** R.A = S.A;

   (c) **select** $*$ **from** R **left outer join** S **on** R.A = S.A;

   (d) **select** $*$ **from** R **right outer join** S **on** R.A = S.A;

   (e) **select** $*$ **from** R **full outer join** S **on** R.A = S.A;

R

X	A	Y	B	Z
0	10	0	9	2
30	8	0	5	1
60	4	1	3	3
90	0	0	4	5

S

A	B	C	D
17	1	20	100
4	2	40	200
4	3	30	100
8	5	60	500