CS2102 Tutorial 4: SQL 2

Questions to be discussed: 2(a)-(b), 3(a)-(b), 4, 5, 6

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.

(a) Query A

```
1  SELECT  *
2  FROM     bar b
3  WHERE   EXISTS (
4     SELECT  1
5     FROM     foo f
6     WHERE   f.f > 100
7        AND  f.a = b.a
8  );
```

(b) Query B

```
1  SELECT  *
2  FROM     bar b
3  WHERE   EXISTS (
4     SELECT  1
5     FROM     foo f
6     WHERE   f.f > 100
7  )
8     AND  f.a = b.a;
```

(c) Query C

```
1  SELECT  *
2  FROM     bar b
3  WHERE   EXISTS (
4     SELECT  1
5     FROM     foo f
6     WHERE   f.f > 100
7        AND  a = b.a
8  );
```

(d) Query D

```
1  SELECT  *
2  FROM     bar b
3  WHERE   EXISTS (
4     SELECT  1
5     FROM     foo f
6     WHERE   f.f > 100
7        AND  a = a
8  );
```
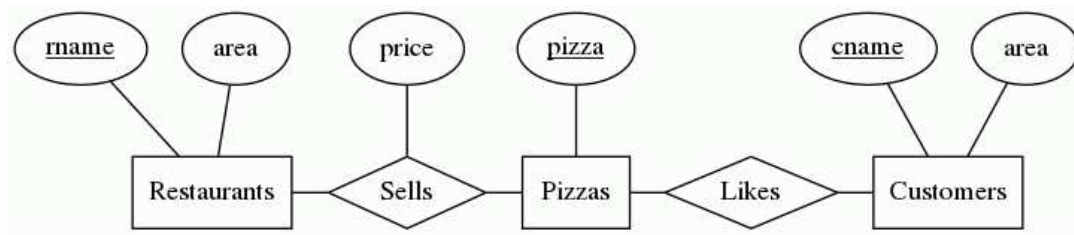
(e) Query E

```
1  SELECT  *
2  FROM     bar b
3  WHERE   EXISTS (
4     SELECT  1
5     FROM     foo f
6     WHERE   f.f > 100
7        AND  f.a = b.a
8        AND  b > 20
9  );
```

2. Questions 2 to 5 are based on the pizza database schema used in the tutorial 1
   (T01.sql); we show its ER diagram below.

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

   (a) Query A

```
1  SELECT DISTINCT cname
2  FROM    Likes L
3  WHERE   EXISTS (
4    SELECT 1
5    FROM    Sells S
6    WHERE   S.rname = 'Corleone Corner'
7      AND   S.pizza = L.pizza
8  );
```

   (b) Query B

```
1  SELECT cname
2  FROM    Customers C
3  WHERE   NOT EXISTS (
4    SELECT 1
5    FROM    Likes L, Sells S
6    WHERE   S.rname = 'Corleone Corner'
7      AND   S.pizza = L.pizza
8      AND   C.cname = L.cname
9  );
```

   (c) Query C

```
1  SELECT DISTINCT rname
2  FROM    Sells
3  WHERE   rname <> 'Corleone Corner'
4    AND   price > ANY (
5      SELECT price
6      FROM    Sells
7      WHERE   rname = 'Corleone Corner'
8  );
```

   (d) Query D

```
1  SELECT rname, pizza, price
2  FROM    Sells S
3  WHERE   price >= ALL (
4    SELECT S2.price
5    FROM    Sells S2
6    WHERE   S2.rname = S.rname
7      AND   S2.price IS NOT NULL
8  );
```

3. Write an SQL query to answer each of the following questions on the pizza database *without using aggregate functions*. Remove duplicate records from all query results.
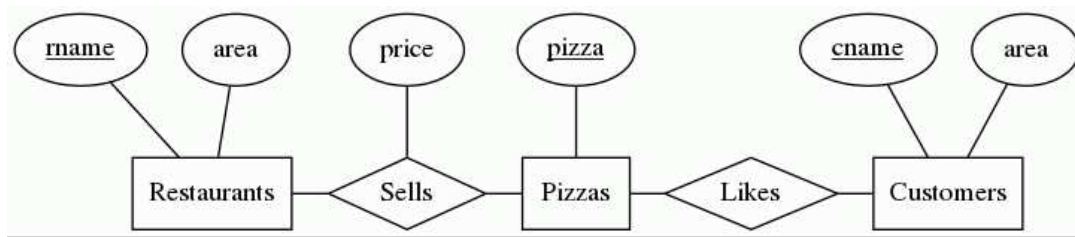
(a) Find pizzas that Alice likes but Bob does not like.

(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, P_{min}) where P is a pizza that is available in area A (*i.e.*, there is some restaurant in area A selling pizza P) and P_{min} is the *lowest* price of P in area A.

(d) Find all tuples (A, P, P_{min}, P_{max}) where P is a pizza that is available in area A (*i.e.*, there is some restaurant in area A selling pizza P), P_{min} is the *lowest* price of P in area A and P_{max} 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 Q_1 and Q_2*) for this query.
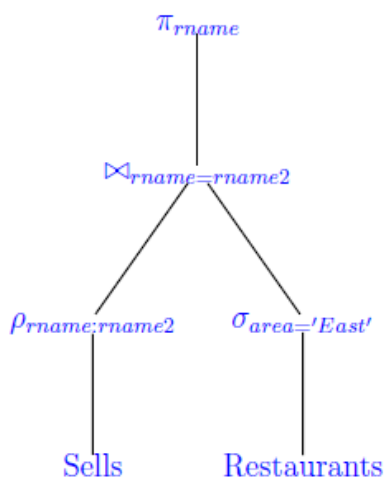
   Q_1 Query 1

   ```
   1  SELECT  DISTINCT  S.rname
   2  FROM    Sells S JOIN Restaurants R
   3    ON    S.rname = R.rname AND R.area = 'East';
   ```
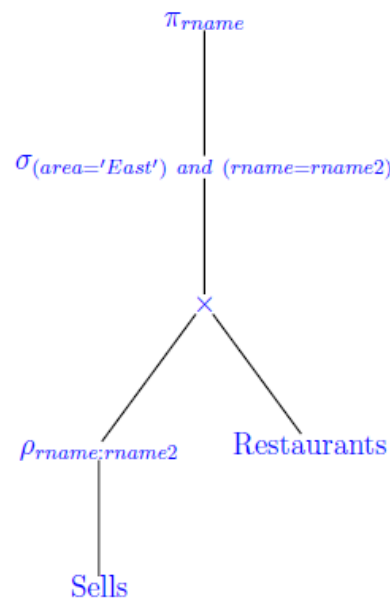
   Q_2 Query 2

   ```
   1  SELECT  DISTINCT  S.rname
   2  FROM    Sells S, Restaurants R
   3  WHERE   S.rname = R.rname
   4    AND R.area = 'East';
   ```

   The semantics of these two SQL queries are defined by the relational algebra expressions shown below. Discuss whether Q_1 and Q_2 are equivalent queries.

(a) Q_1

(b) Q_2

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 the restaurants that do not sell any pizza are to be excluded. The following are two possible SQL answers (*denoted by Q_1 and Q_2*) for this query.

Q_1 Query 1

```
1  SELECT  DISTINCT  S.rname
2  FROM    Sells S JOIN Restaurants R
3    ON    S.rname = R.rname AND R.area = 'East'
4  UNION
5  SELECT  DISTINCT  S.rname
6  FROM    Sells S JOIN Likes L
7    ON    S.pizza = L.pizza AND L.cname = 'Lisa';
```
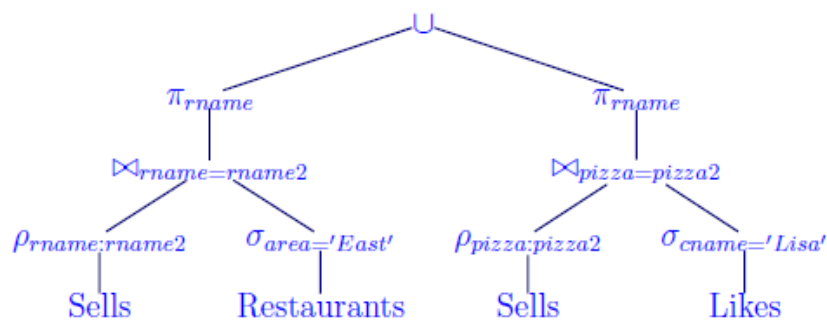
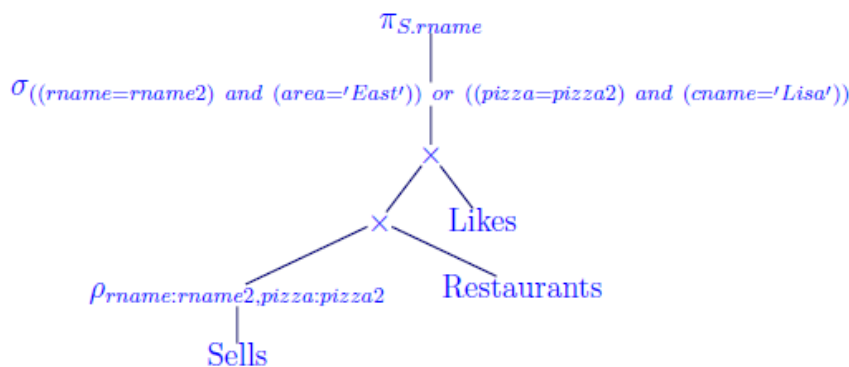Q_2 Query 2

```
1  SELECT  DISTINCT  S.rname
2  FROM    Sells S, Restaurants R, Likes L
3  WHERE   (S.rname = R.rname AND R.area = 'East')
4    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 Q_1 and Q_2 are equivalent queries.

(a) Q_1

(b) Q_2

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

```
1  CREATE TABLE Offices (
2    office_id     INTEGER ,
3    building      TEXT NOT NULL ,
4    level         INTEGER NOT NULL ,
5    room_number   INTEGER NOT NULL ,
6    area          INTEGER ,
7    PRIMARY KEY (office_id),
8    UNIQUE (building , level , room_number)
9  );
10
11 CREATE TABLE Employees (
12   emp_id        INTEGER ,
13   name          TEXT NOT NULL ,
14   office_id     INTEGER NOT NULL ,
15   manager_id    INTEGER ,
16   PRIMARY KEY (emp_id),
17   FOREIGN KEY (office_id) REFERENCES Offices (office_id)
18     ON UPDATE CASCADE ,
19   FOREIGN KEY (manager_id) REFERENCES Employees (emp_id)
20     ON UPDATE CASCADE
21 );
```

Suppose that the office with office_id = 123 needs to be renovated. Write an 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
0	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