CS2102 Tutorial 4: SQL 2 — Wk 6, Sem 2, 2021/22

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

```sql
SELECT *
FROM   bar b
WHERE  EXISTS (
  SELECT 1
  FROM   foo f
  WHERE  f.f > 100
    AND  f.a = b.a
);
```

(b) Query B

```sql
SELECT *
FROM   bar b
WHERE  EXISTS (
  SELECT 1
  FROM   foo f
  WHERE  f.f > 100
)
  AND  f.a = b.a;
```

(c) Query C

```sql
SELECT *
FROM   bar b
WHERE  EXISTS (
  SELECT 1
  FROM   foo f
  WHERE  f.f > 100
    AND  a = b.a
);
```

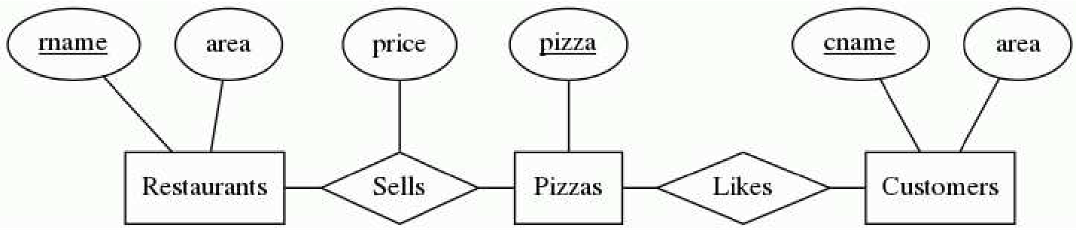
(d) Query D

```sql
SELECT *
FROM   bar b
WHERE  EXISTS (
  SELECT 1
  FROM   foo f
  WHERE  f.f > 100
    AND  a = a
);
```

(e) Query E

```sql
SELECT *
FROM   bar b
WHERE  EXISTS (
  SELECT 1
  FROM   foo f
  WHERE  f.f > 100
    AND  f.a = b.a
    AND  b > 20
);
```

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) Query 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) Query 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  C.cname = L.cname
);
```

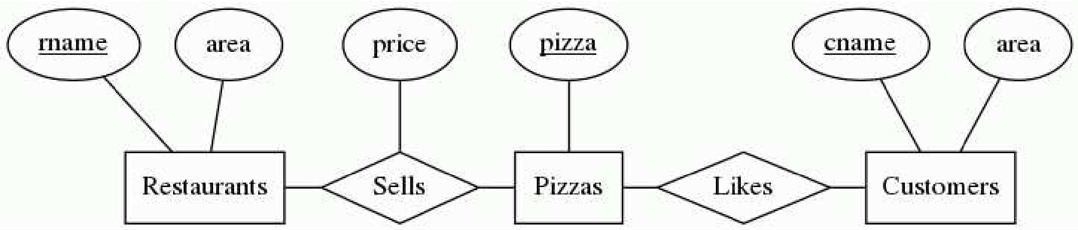
(c) Query C

```
SELECT DISTINCT rname
FROM   Sells
WHERE  rname <> 'Corleone Corner'
  AND  price > ANY (
    SELECT price
    FROM   Sells
    WHERE  rname = 'Corleone Corner'
);
```

(d) Query 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 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.

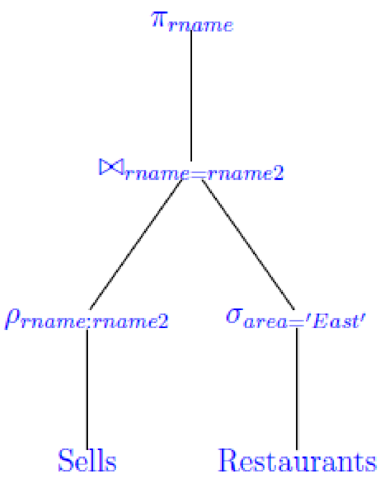
Q_1 Query 1

```sql
SELECT DISTINCT S.rname
FROM   Sells S JOIN Restaurants R
  ON   S.rname = R.rname AND R.area = 'East';
```

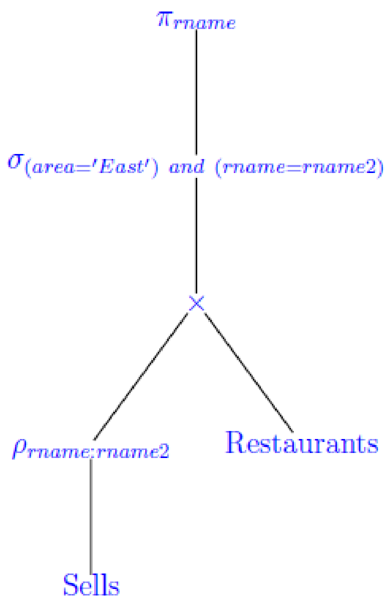
Q_2 Query 2

```sql
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 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.

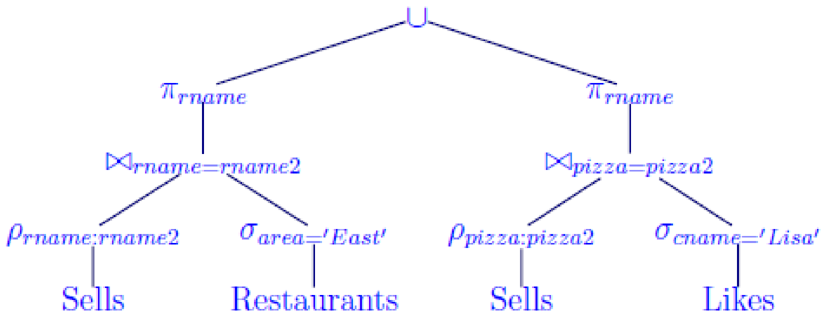
Q_1 Query 1

```
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';
```

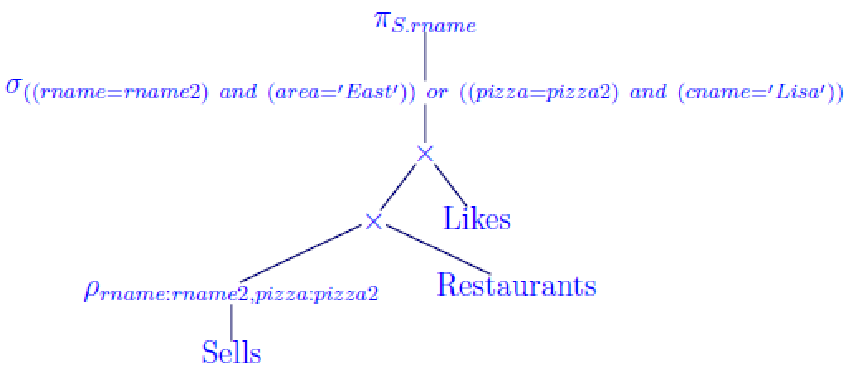
Q_2 Query 2

```
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 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.

```
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 `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