

1. Consider the following database schema consisting of two tables.

```
create table R (  
    a integer primary key,  
    b integer,  
    c integer  
);  
  
create table S (  
    x integer primary key,  
    y integer references R(a)  
);
```

For each of the following queries on the database, state whether the query is a valid SQL query.

```
select a, b, sum(c)  
from R join S on R.a = S.y  
group by a, b;  
(a)
```

```
select b, sum(c)  
from R join S on R.a = S.y  
group by a, b;  
(b)
```

```
select a, b, sum(c)  
from R join S on R.a = S.y  
group by a;  
(c)
```

```
select a, b, x, sum(c)  
from R join S on R.a = S.y  
group by a, b;  
(d)
```

```
select a, b, y, sum(c)  
from R join S on R.a = S.y  
group by a, b;  
(e)
```

```
select a, b  
from R join S on R.a = S.y  
group by a, b  
having sum(c) > 10;  
(f)
```

```
select a, b  
from R join S on R.a = S.y  
group by a, b  
having sum(x) > 10;  
(g)
```

```
select a, b  
from R join S on R.a = S.y  
where sum(c) > 10;  
(h)
```

```
select a, b  
from R join S on R.a = S.y  
group by a, b  
having sum(x) > sum(c);  
(i)
```

```
select b, sum(c)  
from R;  
(j)
```

2. Consider the following two SQL queries on the database in Question 1. Are these queries equivalent?

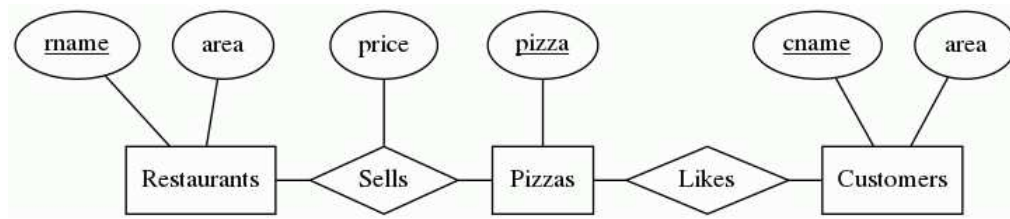
```
select count(c)
from R
where a = 10;
```

(a) Query 1

```
select count(c)
from R
where a = 10
group by a;
```

(b) Query 2

3. This question is based on the pizza database schema shown below.



Answer each of the following queries using SQL. For parts (a) to (e), remove duplicate records from all query results.

- Find the most expensive pizzas and the restaurants that sell them (at the most expensive price). Do not use any aggregate function in your answer.
- Find all restaurant pairs  $(R1, R2)$  such that the price of the most expensive pizza sold by  $R1$  is higher than that for  $R2$ . Exclude restaurant pairs where  $R2$  does not sell any pizza.
- For each restaurant that sells some pizza, find the restaurant name and the average price of its pizzas if its average price is higher than \$22. Do not use the **HAVING** clause in your answer.
- For each restaurant  $R$  that sells some pizza, let  $totalPrice(R)$  denote the total price of all the pizzas sold by  $R$ . Find all pairs  $(R, totalPrice(R))$  where  $totalPrice(R)$  is higher than the average of  $totalPrice()$  over all the restaurants.
- Find the customer pairs  $(C1, C2)$  such that  $C1 < C2$  and they like exactly the same pizzas. Exclude customer pairs that do not like any pizza. Do not use the **EXCEPT** operator in your answer.
- Write a SQL statement to increase the selling prices of pizzas as follows:
  - Increase by \$3 if the restaurant is located in 'Central',
  - Increase by \$2 if the restaurant is located in 'East',
  - Otherwise, increase by \$1.

4. In the lecture, we have discussed an algebraic approach to derive a SQL query involving universal quantification. In this question, we consider another approach to derive the same SQL query based on predicate logic.

The following predicate logic expression specifies the set of names of students who have enrolled in every course offered by the CS department.

$$\{S.name \mid S \in Students \wedge \forall C(C \in Courses \wedge C.dept = 'CS' \implies \\ \exists E(E \in Enrolls \wedge E.sid = S.studentId \wedge E.cid = C.courseId))\}$$

Rewrite the above expression into an equivalent expression without using any universal quantifier and implication operators, and translate it into an equivalent SQL query.