Discussions

1. **(Equivalence)** Consider the following relational schema:

```sql
CREATE TABLE R (
  a  INT PRIMARY KEY,
  b  INT,
  c  INT
);
```

```sql
CREATE TABLE S (
  x  INT PRIMARY KEY,
  y  INT REFERENCES R(a)
);
```

Are these two queries *equivalent*?

(a) Q_1

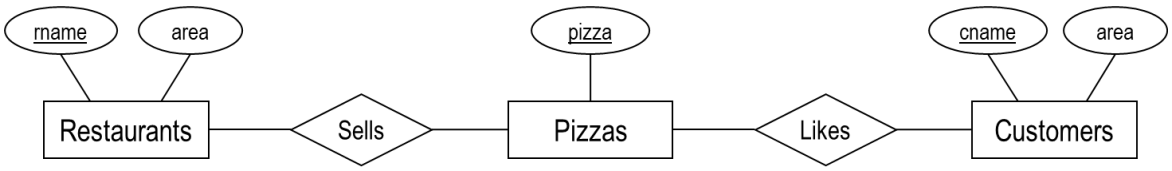
```sql
SELECT COUNT(c)
FROM   R
WHERE  a = 10;
```

(b) Q_2

```sql
SELECT COUNT(c)
FROM   R
WHERE  a = 10
GROUP BY a;
```

2. **(Aggregate)** This question is based on the pizza database schema shown below.

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



(a) Find all restaurant pairs (R_1, R_2) such that the price of the most expensive pizza sold by R_1 is *higher* than than that of R_2. Exclude restaurant pairs where R_2 do not sell any pizza.

(b) For each restaurant that sels some pizza, find the restaurant name and the average price of its pizzas if its average price is higher than $22.

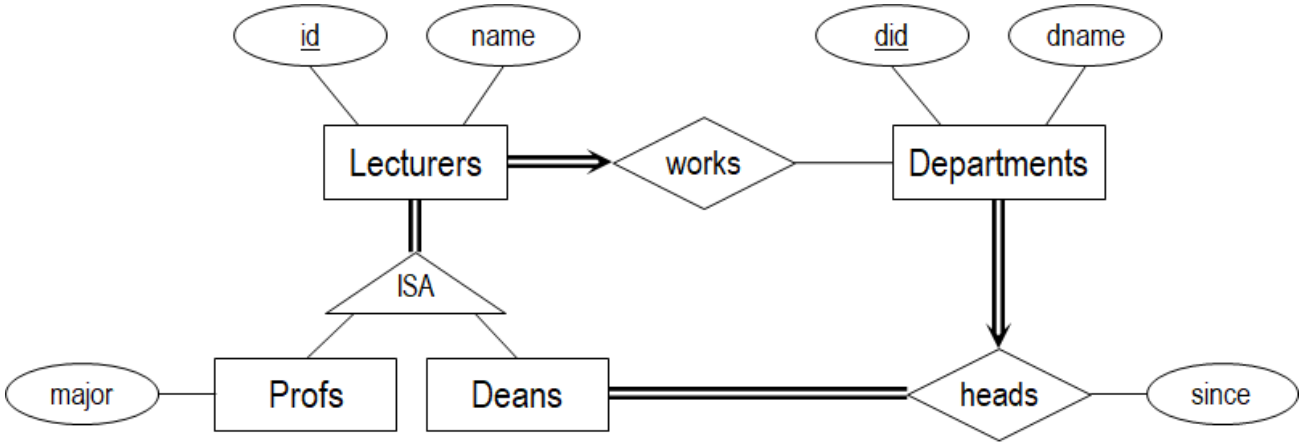
(c) For each restaurant R that sells some pizza, let $totalPrice(R)$ denote the total price of all the pizzas sold by R. Find all pairs $(R, totalPrice(R))$ where $totalPrice(R)$ is *higher* than the average of $totalPrice()$ over all the restaurants.

(d) Find the customer pairs (C_1, C_2) such that $C_1 < C_2$ 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.

Challenge

The answers to the following questions is given without explanation. Please discuss them on Canvas.

1. **(CTE)** Consider the following ER diagram.



   We assume that we have the following schema:

   - `Lecturers(id, name, did)`
   - `Profs(id, major)`
   - `Deans(id)`
   - `Departments(did, dname, id)`

   We say that a dean is ***important*** if the dean heads a department where either:

   - There are at least 20 professors working in the department excluding the dean, OR
   - There are at least 5 professors with different majors working in the department

   Find all the ***non-important*** dean. **Hint:** You may use CTE to simplify the problem.

2. **(Universal Quantification)** Consider the following schema:

```sql
CREATE TABLE Students (
  matric   VARCHAR(9) PRIMARY KEY,
  sname    VARCHAR(50)
);
```

```sql
CREATE TABLE Projects (
  pid     VARCHAR(9) PRIMARY KEY,
  pname   VARCHAR(50)
);
```

```sql
CREATE TABLE Workings (
  pid      VARCHAR(9) REFERENCES Projects(pid),
  matric   VARCHAR(9) REFERENCES Students(matric),
  since    DATE,
  PRIMARY KEY(pid, matric)
);
```

```sql
CREATE TABLE Category (
  pid     VARCHAR(9) REFERENCES Projects(pid),
  cname   VARCHAR(9),
 PRIMARY KEY(pid, cname)
);
```

Find all pair of distinct projects' `pid (p1, p2)` such that the two projects have exactly the same set of categories.