CS2102: Database Systems

Tutorial #5: *SQL (Part 3)*

Week 7

AY 2022/23 Sem 2

1 Discussions

The following questions are to be discussed during tutorial. All answers will be released with explanation.

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

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

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

Are these two queries *equivalent*?

No.

(a) Q_1

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

→ 先找所有 a=10 的 row，放到 table.
再 count table 里 column c 里 non-null 的个数.

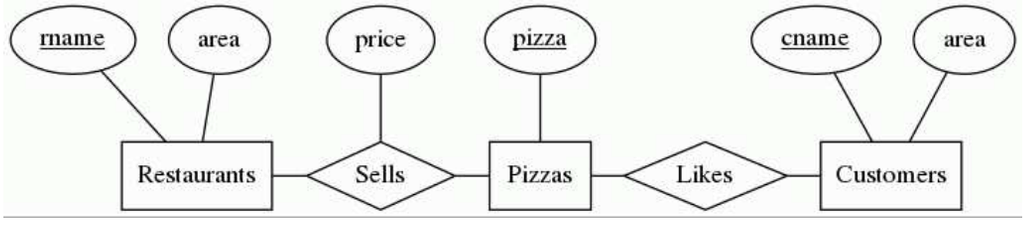
(b) Q_2

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

*: when row 'a=10' doesn't exist:
(a). still has a table. count(c) = 0. (1 row)
(b). GROUP BY a → No such Group → Count(c) won't evaluate. (0 Row).

2. **(Un-Aggregate)** 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.



(a) 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.

(b) 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.

(c) 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. Do **NOT** use the `HAVING` clause in your answer.

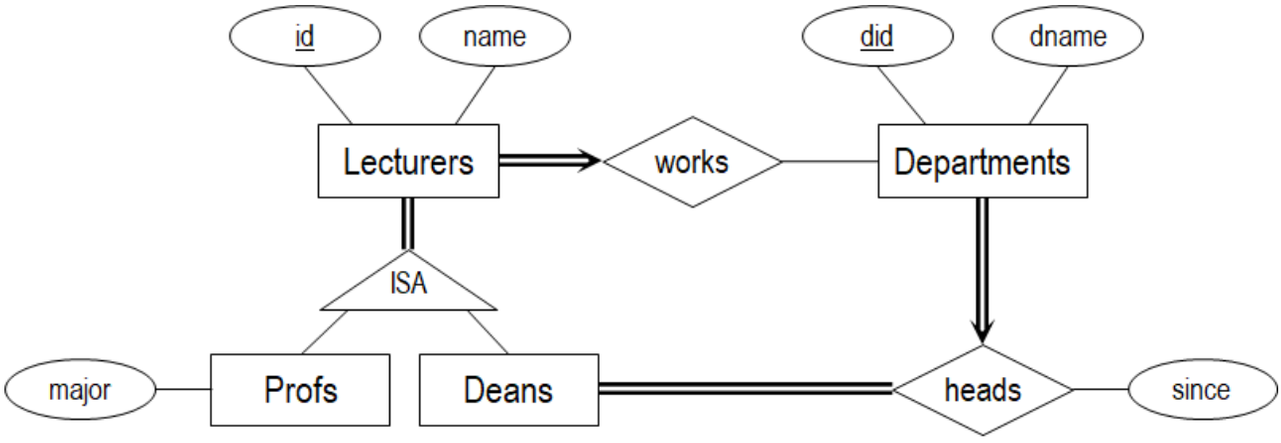
(d) For each restaurant R that sells some pizza, let $totalPrice(R)$ denote the total price of all the pizzas sold by R. Find all pairs $(R, totalPrice(R))$ where $totalPrice(R)$ is *higher* than the average of $totalPrice()$ over all the restaurants.

(e) 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.

2 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:

```
CREATE TABLE Students (
  matric   VARCHAR(9) PRIMARY KEY,
  sname    VARCHAR(50)
);
CREATE TABLE Projects (
  pid     VARCHAR(9) PRIMARY KEY,
  pname   VARCHAR(50)
);
CREATE TABLE Workings (
  pid      VARCHAR(9) REFERENCES Projects(pid),
  matric   VARCHAR(9) REFERENCES Students(matric),
  since    DATE,
  PRIMARY KEY(pid, matric)
);
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.