University of Edinburgh
School of Informatics
INFR11199 - Advanced Database Systems (Spring 2024)

Tutorial Sheet 1

1. (Single-Table SQL) Consider a database of published scientific papers with the following schema, where each primary key is underlined:

   Conference(<u>conference_id</u>, conference_name, organiser)

   Paper(<u>paper_id</u>, title, field, citations, year_published, conference_id)

   Ownership(<u>paper_id</u>, <u>researcher_id</u>)

   Researcher(<u>researcher_id</u>, researcher_name, affiliation, email)

   Write a SQL query for each task below.

   (a) Find the 10 papers with the highest numbers of citations, ordered from most to least. Break ties by paper title in alphabetical order.

   Solution:

   ```
   SELECT *
   FROM Paper
   ORDER BY citations DESC, title ASC
   LIMIT 10;
   ```

   (b) Find the name and email for every researcher whose affiliation starts with the string 'University of'.

   Solution:

   ```
   SELECT researcher_name, email
   FROM Researcher
   WHERE affiliation LIKE 'University of%';
   ```

(c) Find the total number of published papers per research field.

Solution:

```
SELECT field, COUNT(*)
FROM Paper
GROUP BY field;
```

(d) Find the total number of published papers per research field. Do not report fields with less than 10 papers.

Solution:

```
SELECT field, COUNT(*)
FROM Paper
GROUP BY field
HAVING COUNT(*) >= 10;
```

(e) Find the research field with the highest number of papers published after the year 2020. Assume there are no ties.

Solution:

```
SELECT field
FROM Paper
WHERE year_published > 2000
GROUP BY field
ORDER BY COUNT(*) DESC
LIMIT 1;
```

2. (Multi-Table SQL) Consider the same database schema from the previous question. Write a SQL query for each task below.

(a) Find the name of all conferences that featured database papers in 2024.

Solution:

```
SELECT conference_name
FROM Conference INNER JOIN Paper
  ON Conference.conference_id = Paper.conference_id
WHERE field = 'databases' AND year_published = 2024
GROUP BY conference_id, conference_name;
```

The same query can be expressed using an alternative join syntax:

```
SELECT conference_name
FROM Conference, Paper
WHERE Conference.conference_id = Paper.conference_id AND
      field = 'databases' AND year_published = 2024
GROUP BY conference_id, conference_name;
```

If a conference publishes multiple database papers in 2024, we need to make sure the conference_name appears only once in the result. However, we cannot use 'DISTINCT conference_name' to resolve this, because if there are 2 conferences that have different conference_ids but share the same conference_name, we need to make sure the conference_name shows up in the results twice (once for each unique conference_id), which is why we include the GROUP BY clause.

(b) Find the name of the conference that featured the paper with the highest number of citations. Assume there is only one such paper.

Solution:

```
SELECT conference_name
FROM Conference INNER JOIN Paper
  ON Conference.conference_id = Paper.conference_id
ORDER BY citations DESC
LIMIT 1;
```

(c) For each researcher, find the researcher name and the highest citation count of one of their paper. Include researchers that have not published a paper.

Solution:

```
SELECT researcher_name, MAX(citations)
FROM Researcher LEFT OUTER JOIN
    (Paper INNER JOIN Ownership
      ON Paper.paper_id = Ownership.paper_id)
  ON Researcher.researcher_id = Ownership.researcher_id
GROUP BY Researcher.researcher_id, researcher_name;
```

3. (CQ Minimization) Consider the CQ

$$Q(x, y) \text{ :- } R(y, x, z), R(w, v, z), T(x, z), T(x, a), R(y, x, a)$$

where x, y, z, v, w are variables and a is a constant value. Compute the minimal CQ of Q.

Solution: We are going to apply the Minimization algorithm discussed in the lecture. Recall that this is a non-deterministic algorithm since the order in which the atoms of Q are considered is not predetermined. On the other hand, we can consider the atoms of Q in any order since the algorithm always computes the same minimal CQ (up to variable renaming). We are going to consider the atoms in Q from left to right.

- The first step is to check whether there is a query homomorphism from

  $$Q(x, y) \text{ :- } R(y, x, z), R(w, v, z), T(x, z), T(x, a), R(y, x, a)$$

  to

  $$Q(x, y) \text{ :- } R(w, v, z), T(x, z), T(x, a), R(y, x, a).$$

  This is indeed the case witnessed by the following query homomorphism

  $$h \;=\; \{x \mapsto x, y \mapsto y, z \mapsto a, w \mapsto y, v \mapsto x, a \mapsto a\}.$$

  It is easy to verify that

  $$\begin{aligned} &h(\{R(y, x, z), R(w, v, z), T(x, z), T(x, a), R(y, x, a)\}) \subseteq \\ &\quad \{R(w, v, z), T(x, z), T(x, a), R(y, x, a)\}. \end{aligned}$$

  Thus, we can remove the atom $R(y, x, z)$.

- The second step is to check whether there is a query homomorphism from

  $$Q(x, y) \text{ :- } R(w, v, z), T(x, z), T(x, a), R(y, x, a)$$

  to

  $$Q(x, y) \text{ :- } T(x, z), T(x, a), R(y, x, a).$$

  This is indeed the case witnessed by the following query homomorphism

  $$h \;=\; \{x \mapsto x, y \mapsto y, z \mapsto a, w \mapsto y, v \mapsto x, a \mapsto a\}.$$

  It is easy to verify that

  $$h(\{R(w, v, z), T(x, z), T(x, a), R(y, x, a)\}) \subseteq \{T(x, z), T(x, a), R(y, x, a)\}.$$

  Thus, we can remove the atom $R(w, v, z)$.

- The third step is to check whether there is a query homomorphism from

  $$Q(x, y) \text{ :- } T(x, z), T(x, a), R(y, x, a)$$

to

$$Q(x, y) \text{ :- } T(x, a), R(y, x, a).$$

This is indeed the case witnessed by the following query homomorphism

$$h \;=\; \{x \mapsto x, y \mapsto y, z \mapsto a, a \mapsto a\}.$$

It is easy to verify that

$$h(\{T(x, z), T(x, a), R(y, x, a)\}) \subseteq \{T(x, a), R(y, x, a)\}.$$

Thus, we can remove the atom $T(x, z)$.

- The fourth step is to check whether there is a query homomorphism from

  $$Q(x, y) \text{ :- } T(x, a), R(y, x, a)$$

  to

  $$Q(x, y) \text{ :- } R(y, x, a).$$

  It is clear that this is not the case since there is no way to map the atom $T(x, a)$ to $R(y, x, a)$. Therefore, $T(x, a)$ cannot be eliminated.

- The fifth and final step is to check whether there is a query homomorphism from

  $$Q(x, y) \text{ :- } T(x, a), R(y, x, a)$$

  to

  $$Q(x, y) \text{ :- } T(x, a).$$

  As in the previous step, it is clear that there is no way to map the atom $R(y, x, a)$ to $T(x, a)$, and thus, there is no query homomorphism. Hence, $R(y, x, a)$ cannot be eliminated.

Consequently, the minimal CQ of Q is

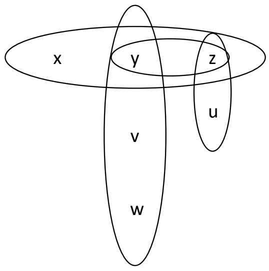
$$Q(x, y) \text{ :- } T(x, a), R(y, x, a).$$

4. (Acyclicity) Consider the CQ

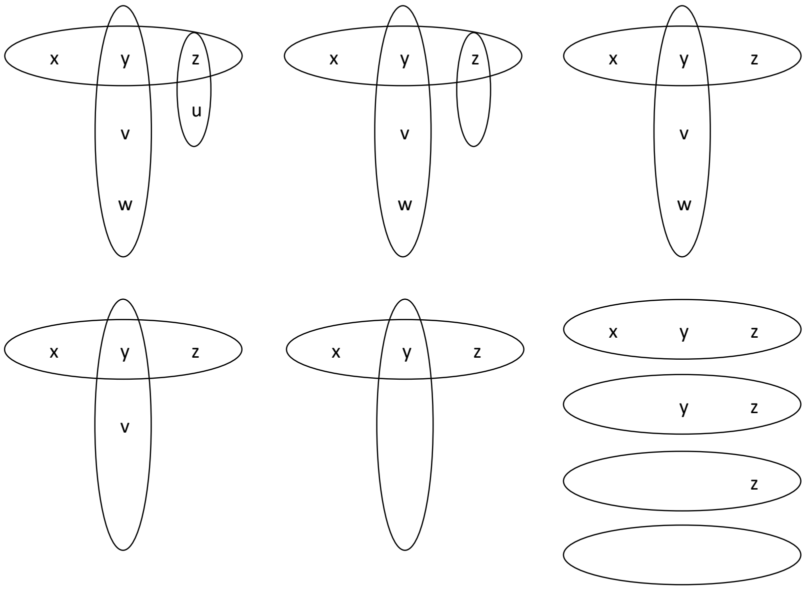
$$Q(x, y) \text{ :- } T(x, y, z), R(y, z), P(y, v, w), R(z, u).$$

Prove that Q is acyclic. In particular, apply the GYO-reduction to the hypergraph $H(Q)$ of Q and show that this leads to the empty graph. Give also the obtained join tree of $H(Q)$.

Solution: We first need to construct the hypergraph $H(Q)$ of Q, which follows:



We then apply the GYO-reduction on $H(Q)$. Here are the intermediate hypergraphs until we get the empty hypergraph:



The root of the obtained join tree is the set $\{x, y, z\}$, which has as children the sets $\{y, v, w\}$, $\{y, z\}$ and $\{z, u\}$.