**Task List No. 5 (Normalization)**
*Bazy danych 2024*

Help design a database for the Organized Crime Department (OCD). The secret services have already compiled a list of the most compromised cities and the most malicious criminal activities. It has also been firmly established that if a gang engages in a criminal activity, it engages in it in all the cities where it is present, and if it is present in a city, it engages in all the criminal activities practiced anywhere.

Currently, the database consists of the relation: `Mafia(City, Gang, Activity, Boss, Income, ROI)`. It is also known that the functional dependencies are: `Gang` → `Boss`, `Boss` → `Income`, `Activity` → `ROI`.

**1. (0.5 points)** Give the definition of BCNF form. Is the above database in this form? Why? Propose a decomposition of the database into BCNF form that is reversible and preserves all dependencies. Do not decompose any table that is already in BCNF.

**2. (0.5 points)** Even though the database (after your modifications) is in BCNF, do you still see any obvious redundancy in it? What is its cause?

**3. (0.5 points)** Is there a reversible decomposition that allows us to eliminate this redundancy? Provide it. Why is it reversible?

**4. (0.5 points)** Write a relational algebra query that returns a non-empty result if the functional dependency `Activity` → `ROI` is violated in the `Mafia` relation.

**5. (1 point)** Consider the relation $S(F, M, R)$. A multivalued dependency $F \twoheadrightarrow M$ holds if for any two tuples $t_1, t_2 \in S$ such that $\pi_F(t_1) = \pi_F(t_2)$ there exists a tuple $t \in S$ such that:

1. $\pi_{FM}(t) = \pi_{FM}(t_1)$,
2. $\pi_{FR}(t) = \pi_{FR}(t_2)$.

Prove or provide a counterexample:

1. If $F \twoheadrightarrow M$, then $F \to M$.
2. If $F \to M$, then $F \twoheadrightarrow M$.

If in the database all multivalued dependencies depend on (super)keys, we say that the database is in the fourth normal form (4NF).

**6. (1 point)** One of the most important data models for graphs nowadays are the so-called *property graphs* (example in Fig. 1). Graphs in this model consist of vertices and edges. Vertices and edges can have multiple *properties*, which are key-value pairs. Propose a sensible way to store such graphs in a relational database.

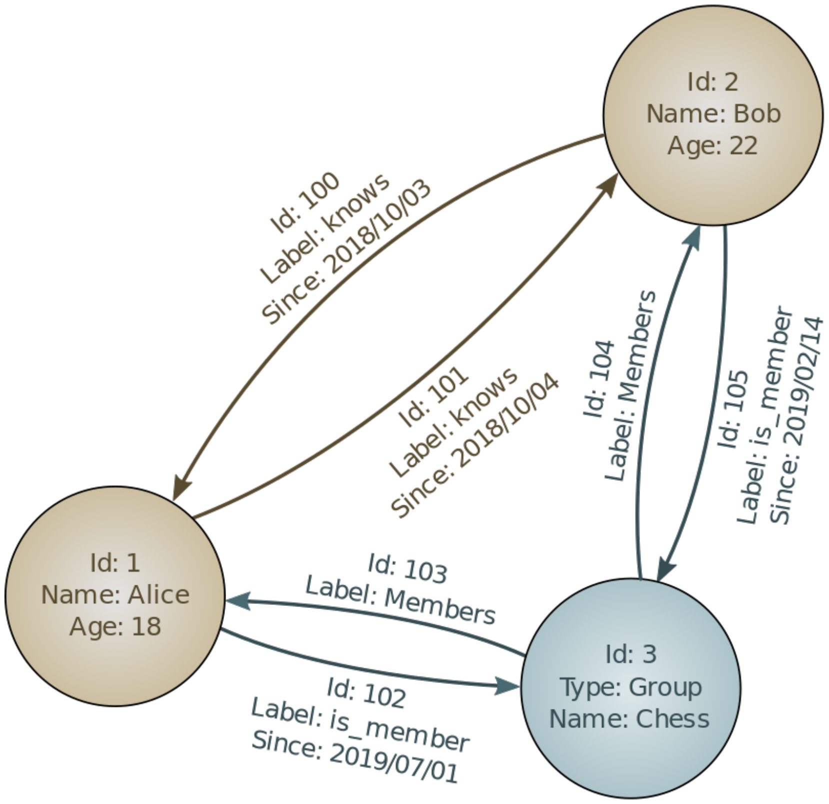


Figure 1: Property graph

**7. (1 point)** Traditionally, normal forms beyond BCNF (e.g., 4NF) were very rarely used. The theory of normalization beyond BCNF was considered distant from practice.

This was largely due to the fact that the ways of calculating joins in databases assumed that the join operator is binary (we join two tables, the result of the join with another table, etc.). In modern solutions, sometimes it is worth normalizing to the extreme – each table has at most one column that is not part of the key (this is so-called 6th normal form, 6NF). One of the advantages of such an approach is when joining multiple tables simultaneously (*worst-case optimal multi-way joins*).

Modify the previous solutions to store property graphs in 6NF.

**8. (1 point)** Consider the relation $R$, in which the functional dependency $\xi = \alpha \to \beta$ holds, where $\alpha \cap \beta = \emptyset$ and $\alpha$ and $\beta$ are subsets of attributes of $R$. Prove that the decomposition of relation $R$ according to $\xi$, into relations $R_1$ with attributes $\alpha \cup \beta$ and $R_2$ with attributes $\mathrm{attr}(R) \setminus \beta$, is reversible.