Inconsistent databases

- Often arise in data integration.
- Suppose have a functional dependency name $\longrightarrow$ salary and two tuples (John, 10K) in source 1, and (John, 20K) in source 2.
- One may want to clean data before doing integration.
- This is not always possible.
- Another solution: keep inconsistent records, and try to address the issue later.
- Issue = query answering.

Inconsistent databases cont'd

- Setting:
  - a database D;
  - a set of integrity constraints IC

    e.g. keys, foreign keys, functional dependencies etc
  - a query Q
- D violates IC
- What is a proper way of answering Q?
- Certain Answers:

$$\text{certain}_{IC}(Q, D) \;=\; \bigcap_{D_r \text{ is a repair of } D} Q(D_r)$$

Repairs

- How can we repair an instance to make it satisfy constraints?
- If constraints are functional dependencies: say $A \rightarrow B$ and we have

A	B	C
a1	b1	c1
a1	b2	c2

  we have to delete one of the tuples.

- If constraints are referential constraints, e.g. $R[A] \subseteq S[B]$ and we have

R:

A	C
a1	c1
a2	c2

S:

B	D
a1	d1
a3	d2

  then we have to add a tuple to S.

Repairs cont'd

- Thus to repair a database to make it satisfy IC we may need to add or delete tuples.
- Given D and D', how far are they from each other?
- A natural measure: the minimum number of deletions/insertions of tuples it takes to get to D' from D.
- In other words,

$$\delta(D, D') \;=\; (D - D') \cup (D' - D)$$

- A repair is a database D' so that
  - it satisfies constraints IC, and
  - there is no D'' satisfying constraints IC with $\delta(D, D'') \subset \delta(D, D')$

How many repairs are there?

Can easily be exponential even for keys: i.e. $\sqrt{2}^N$.

A	B
1	0
1	1
2	0
2	1
...	...
...	...
n	0
n	1

plus key $A \rightarrow B$ $\stackrel{\text{REPAIR}}{\Longrightarrow}$

A	B
1	$\cdot$
2	$\cdot$
...	...
n	$\cdot$

I.e. for $N = 2n$ tuples we have $2^n = \sqrt{2}^N$ repairs.
(A side remark: this construction gives us $\sqrt[c]{c}^n$ repairs for any number c. What is the maximum of $\sqrt[c]{c}$?)

Query answering

- Recall $\text{certain}_{IC}(Q, D) \;=\; \bigcap_{D_r \text{ is a repair of } D} Q(D_r)$.
- Computing all repairs is impractical.
- Hence one tries to obtain a rewriting Q':

$$Q'(D) \;=\; \text{certain}_{IC}(Q, D).$$

- Is this always possible?

Query rewriting: a good case

- One relation $R(A, B, C)$
- Functional dependency $A \to B$
- Query Q: just return R
- If an instance may violate $A \to B$, then we can rewrite Q to $R(x, y, z) \wedge \forall u \forall v \ \big(R(x, u, v) \to u = y\big)$ or

```
SELECT * FROM R
WHERE NOT EXISTS (SELECT * FROM R R1
                  WHERE R.A=R1.A AND R.B <> R1.B)
```

- This technique applies to a small class of queries: conjunctive queries without projections, i.e.

```
SELECT * FROM R1, R2 ...
WHERE <conjucntion of equalities>
```

Query rewriting: a mildly bad case

- One relation $R(A, B)$; attribute A is a key
- Query $Q \;=\; \exists x, y, z\ \big(R(x, z) \wedge R(y, z) \wedge (x \neq y)\big)$
- When are certain answers false?
- If there is a repair in which the negation of Q is true.
- What is the negation of Q?
  - $\neg Q \;=\; \forall x, y, z\ \big((R(x, z) \wedge R(y, z)) \rightarrow x = y\big)$
- This happens precisely when R contains a perfect matching
- But checking for a perfect matching cannot be expressed in SQL.
- Hence, no SQL rewriting for $\text{certain}_{IC}(Q)$.

Query rewriting: the worst

- One can find an example of a rather simple relational algebra query Q and a set of constraints IC so that the problem of finding

$$\mathsf{certain}_{IC}(Q, D)$$

  is coNP-complete.

- In general for most types of constraints one can limit the number of repairs but they give rather high complexity bounds
  - typically classes "above" PTIME and contained in PSPACE – hence almost certainly requiring exponential time.

Other approaches

- Repair attribute values.
  - A common example: census data. Don't get rid of tuples but change the values.
  - Distance: sum of absolute values of squares of differences new value – old value
  - Typically one considers aggregate queries and looks for approximations or ranges of their values
- A different notion of repair.
  - Most commonly: the cardinality of $(D - D') \cup (D' - D)$ must be minimum.
  - This is a reasonable measure but the complexity of query answering is high.