Completeness of Queries over SQL Databases

Werner Nutt and Simon Razniewski
Introduction

- Data Quality research investigates how good data is

- Dimensions of Data Quality are:
  - Correctness
  - Timeliness
  - Completeness
Completeness

- **Query answering** over incomplete data: extensively studied
  - Codd: NULL values [1975]
  - Imielinski/Lipski: Representation systems [1984]

- **Query completeness**: Little attention
  - Razniewski/Nutt: Only on missing records [VLDB 2011]
Bolzano is in the Province of South Tyrol

- Trilingual province in the north of Italy
- Has its own school administration
Incompleteness in the school data

Facts in real world

- result(Paul, Music, A)
- result(Giulia, Music, A)

Facts in school database

- result(Paul, Music, NULL)

Missing information in the school database:
- no grade for Paul (missing value)
- no entry for Giulia (missing record)
Consequence: Query answers are incorrect

Query Q: "How many pupils have grade A in Music?"

According to ideal database:
\[ \text{Q(} \text{Paul, Music, A} \text{)} = 2 \]
\[ \text{result(Paul, Music, A)} \]
\[ \text{result(Giulia, Music, A)} \]

According to available database:
\[ \text{Q(} \text{Paul, Music, NULL} \text{)} = 0 \]
\[ \text{result(Paul, Music, NULL)} \]

→ If data is incomplete, query answers become incorrect.
Use Metadata to guarantee completeness!

... vocational schools use the information system of the province to manage grades.

However, we may know whether parts of a database are complete, e.g.,

- “The grades from vocational schools are complete”
- “The Music grades from primary schools are complete”

⇒ Idea: Assess completeness of a query using completeness assertions for (parts of) tables.
Reasoning about query completeness

Grades from vocational schools are complete

All Music grades from primary schools are complete

Biology grades from high schools are complete

I want to know “How many pupils have grade A in Music?“ Can I trust the query answer?

You cannot, because information about pupils from high schools could be missing

Space of possible information

Assertions about partial completeness
Reasoning about query completeness (2)

1. Formalization:
   - incomplete dbs
   - assertions about db completeness

   Grades from vocational schools are complete

   All Music grades from primary schools are complete

   Biology grades from high schools

   Space of possible information

   I want to know
   “How many pupils at vocational schools have grade A in Music?”
   Can I trust the query answer?

   You can, because all needed information is complete in the database

2. Reasoning methods

3. Implementation techniques
   [Demo today]
Running example: Schema

result(name, subject, grade)

pupil(name, schoolName, schoolType)
Formalization: Incomplete database

When talking about incompleteness, we need a complete reference

An *incomplete database* $D$ is a pair of

an *ideal database* $D^i$ and

an *available database* $D^a$

$$D = (D^i, D^a)$$

such that

each record in $D^a$ is less informative than some record in $D^i$
Example: Incomplete database

\[ D^i \]

\[
\text{result}(\text{Paul, Music, A})
\]
\[
\text{result}(\text{Giulia, Music, A})
\]
\[
\text{pupil}(\text{Paul, Verdi, Voc})
\]

\[ D^a \]

\[
\text{result}(\text{Paul, Music, NULL})
\]

less informative than
Formalization: Query completeness

Query Q

“The answer to Q is complete“

Notation: \( \text{Compl}(Q) \)

Semantics:

\[(D^i, D^a) \models \text{Compl}(Q) \iff Q(D^i) = Q(D^a)\]
Formalization: DB completeness

Table completeness statement assert partial completeness of a db table. E.g.,

“The available database contains all subjects taken by pupils at vocational school.”

Formally:

\[ \text{result}^i(n,s,g), \text{pupil}^i(n,sn,\text{Voc}) \rightarrow \exists g' \, \text{result}^a(n,s,g') \]

Every result of a pupil from a vocational school according to the ideal db is also in the available db (but the grade may be missing).
Example: DB completeness

\[
\text{result}^i(n,s,g), \text{pupil}^i(n,sn,Voc) \rightarrow \exists g' \ \text{result}^a(n,s,g')
\]

holds over the incomplete database \((D^i,D^a)\)

\[\begin{align*}
\text{D}^i & : \\
\text{result}(Paul, \text{Music}, A) \\
\text{result}(Giulia, \text{Music}, A) \\
\text{pupil}(Paul, \text{Verdi}, \text{Voc})
\end{align*}\]

\[\begin{align*}
\text{D}^a & : \\
\text{result}(Paul, \text{Music}, \text{NULL})
\end{align*}\]

because \(\text{result}(Paul, \text{Music}, \text{NULL})\) is in \(\text{D}^a\)
The reasoning problem

Table completeness statements $C$

Grades from vocational schools are complete

I want to know “How many pupils at vocational schools have taken Music?“ Can I trust the query answer?

Query $Q$

Grades from vocational schools are complete

All Music grades from primary schools are complete

Space of possible information

Does $C$ imply $\text{Compl}(Q)$?
Reasoning: The principle

Query: “Pupils at vocational schools that took Music“

\[ Q_{\text{pupils}}(x) : \text{result}(x, \text{Music}, g), \text{pupil}(x, \text{sn}, \text{Voc}) \]

1. Assume \( Q_{\text{pupils}} \) returns \( x' \) over \( D^i \)

2. See which facts must be in \( D^i \)

\[
\begin{align*}
... \\
\text{result}(x', \text{Music}, g') \\
\text{pupil}(x', \text{sn}', \text{Voc}) \\
...
\end{align*}
\]
Reasoning: The principle (2)

3. Use table completeness to derive facts in $D^a$

“All subjects taken by pupils at vocational schools there“
\[ \text{result}^i(n, s, g), \text{pupil}^i(n, sn, Voc) \rightarrow \exists g' \text{ result}^a(n, s, g') \]

“All pupils there“
\[ \text{pupil}^i(n, sn, st) \rightarrow \text{pupil}^a(n, sn, st) \]
Reasoning: The principle (3)

4. Query the available database

"Pupils at vocational schools that took Music"

\[ Q(D^a) = \{x'\} \rightarrow x' \text{ is also in } Q(D^a) \]

Conclusion: Query Q is complete given the table completeness statements
Reasoning summary

1. Assume, Q returns a generic answer $x'$ over $D_i$
2. See which facts must be in $D_i$
3. Use table completeness to derive facts in $D^a$
4. Evaluate $Q(D^a)$
5. If $x'$ is returned, the query is complete

Reasoning is NP-complete for DBs without NULLs

Is that unique?

How to evaluate over databases with NULLs?

[Razniewski/Nutt VLDB 2011]
What is the Meaning of NULL?

- No grades were given in the Pottery course?
  Non-existing value
- Paul received a grade, but the grade was not recorded?
  Unknown value
- It is unknown, which of the two is the case?
  Ambiguous NULL

→ NULLs may indicate incomplete information, but need not
→ Usage of NULLs is ambiguous
Reasoning over databases w/ NULLs

Q_{pupils}(x):-\text{result}(x, \text{Music}, A), \text{pupil}(x, \text{sn}, \text{Voc})

"Pupils at vocational schools with A in Music"

1. Assume $Q_{pupils}$ returns $x'$ over $D^i$

2. See which facts must be in $D^i$
Challenge 1: How can we adapt the reasoning to NULLs?

- In general, the reasoning has to be done for both cases
  - Reasoning is in $\Pi^P_2$

- If NULLs stand only for unknown values, then no NULLs can appear in $D_i$ and therefore the second case cannot apply
  - Reasoning is NP-complete
## Reasoning with NULLs: Complexity

<table>
<thead>
<tr>
<th>NULLs mean unknown values</th>
<th>NULLs mean inapplicable values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries w/o selfjoins</td>
<td>PTIME</td>
</tr>
<tr>
<td>Queries w/ selfjoins</td>
<td>NP-complete</td>
</tr>
</tbody>
</table>
Challenge 2: How to compute answers of complete queries?

\[ Q(g) : - \text{result}(Paul, s, g) \]

"All grades of Paul"

\[ Q(D^a) = \begin{cases} 
\{\text{NULL, A, B}\} & \text{if NULL stands for a non-existing value} \\
\{A, B\} & \text{if NULL stands for an unknown value} \end{cases} \]
Reasoning with both kinds of NULLs

Result: Reasoning has the same complexity as reasoning with NULLs standing for inapplicable values (in $\Pi_2^P$)

Result: If a query is complete, tuples that contain unknown NULLs can be forgotten in the query answer
Make different NULLs explicit

Ambiguity can be resolved by boolean guards

<table>
<thead>
<tr>
<th>name</th>
<th>subject</th>
<th>grade</th>
<th>grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paul</td>
<td>...</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>Giulia</td>
<td>...</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>Maria</td>
<td>...</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>Andrea</td>
<td>...</td>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

Allows to count how many pupils received a grade (2-3)

In practice, boolean guards possibly already used where needed
Outcome

- Extended framework
  - Partial databases with NULLs
  - TC statements with projections
  - Reasoning for different meanings of NULL

- Complete queries can be evaluated by standard SQL database engines

- Complexities between $\text{PTIME}$ and $\Pi^P_2$
Conclusion

- **Query completeness** assessment is **practically relevant**

- **Reasoning over SQL databases** is possible

- **Demo** at http://magik-demo.inf.unibz.it/
Questions?