Guided Interaction: Rethinking the Query-Result Paradigm

Arnab Nandi
H.V. Jagadish
University of Michigan, Ann Arbor
• Databases have become really fast / efficient in going from query to result
• But does that solve the overall user need?
Outline

- Motivating Example
- Challenges
- Guided Interaction
Outline

• Motivating Example
• Challenges
• Guided Interaction
Motivating Example

- **Alex** and **Bob** meet a Senior Manager
  - Forget name, need to look up contact info.
  - All they remember: manager of small group of senior researchers
Motivating Example: Naïve Alex

- **Visits corporate social network website**

  1. Browses all the “advanced search” forms
  2. Uses Faceted Search interface to naively query for everyone in the company
  3. Realizes you can’t drill down by *seniority*
     - There isn’t a “*seniority*” field, but *age*…
  4. Goes back to “Birthday Search” form
     - Figures out senior employees are ~50
  5. Adds age range, drills further, finds person
Motivating Example: Expert Bob

- Opens up SQL Console to employee DB

  1. `SHOW TABLES;` // reads...

  2. `DESC TABLES;` // reads more...

  3. `SELECT emp.project, COUNT(*) AS c, AVG(emp.age) AS a
     FROM emp JOIN dept ON (emp.deptID = dept.ID)
     GROUP BY emp.project ORDER BY c ASC, a DESC LIMIT 3`

  4. `SELECT emp.name, emp.cubicleID
     FROM emp JOIN dept ON (emp.deptID = dept.ID)
     WHERE dept.name='Research' AND
     emp.project='DatabasePrj' AND
     emp.designation='Manager'`
Motivating Example

• Both users spent more time constructing and issuing sub queries

• Issued redundant / wrong queries

• On standard server, most queries take < 1 min
  • Session takes several minutes – hour!

• Most time was spent in constructing the right query
Outline

• Motivating Example
• **Challenges**
• Guided Interaction
Outline

• Motivating Example

• Challenges
  • User’s lack of Knowledge
  • Dependency of Information
  • Iterative and Incremental Querying
  • Imprecise User Query Intent

• Guided Interaction
Challenges

Lack of Knowledge

- Both users didn’t know about the
  - Schema
  - Data

- Naïve user Alex did not know about
  - Query Language
    either

- All 3 are needed to effectively issue queries

- Otherwise, most time is spent issuing trial-and-error queries to learn more about the DB
Challenges

Dependency of Information

3. Realizes you can’t drill down by *seniority*
   - There isn’t a “*seniority*” field, but *age*…
4. Goes back to “Birthday Search” form
   - Figures out senior employees are ~50

Naïve user
Alex

```
SELECT emp.project, COUNT(*) AS c, AVG(emp.age) AS a
FROM emp JOIN dept ON (emp.deptID = dept.ID)
GROUP BY emp.project ORDER BY c ASC, a DESC LIMIT 3
```

Average age & count per group

Database Expert Bob
Dependency of Information

- Finding out what age “Senior” meant required a secondary query

- Cannot really write as a subquery

- Dependency exists between final query and intermediate query results
Challenges

Iterative & Incremental Querying

• Observation: Users construct queries by first executing smaller parts
  • Cognitive capacity of users is limited

• Query may be declarative, but users prefer iterative / incremental construction

• Leads to a lot of requerying
Challenges

Imprecise Query Intent

DB Expert Bob was looking for some notion of “group” of small people

Hard to translate imprecise intents unless we’re aware of data

Only solution is to execute and see if answer worked

```
SELECT emp.project, COUNT(*) AS c, AVG(emp.age) AS a 
FROM emp JOIN dept ON (emp.deptID = dept.ID) 
GROUP BY emp.project ORDER BY c ASC, a DESC LIMIT 3
```
Challenges

• Our example was a simple one

• Challenges become much harder with complex needs
  • n-way JOINs, Nested queries, complex aggregates…

• Any database use-case with a human in the loop will face these problems
Solutions so far

- Application-level
  - Slick UIs, customized to use case

- No principled approach to solving overall user needs
  - *Where are my standardized operators for overall data interaction?*
  - *Set of rules I can follow when building such a system?*

- Related work:
  - QBE, VizQL(Tableau), AQUA, CONTROL, Telegraph and more
  - Solve thin slices of the overall problem
Outline

• Motivating Example
• Challenges
• Guided Interaction
Guided Interaction

- Principled Approach to solving these problems
- More holistic thinking
- To be included *inside* database

![Diagram showing the process of Guided Interaction](image)
Guided Interaction

• Set of 3 design principles
  • Enumeration
  • Insights
  • Responsiveness

• Database systems that keep these in mind can avoid the challenges discussed
Guided Interaction

Enumeration

- The database is responsible for effectively enumerating all possible valid interactions with the data.

- Removes burden of schema / data / language knowledge off the user
Guided Interaction

Enumeration: Example

• What does an enumeration-enabled query system look like?

• Important
  • One possible implementation
    • Focus on the concepts, not the idea!

• Portray simple use case
  • Can have many, far more complex systems built using these principles
Guided Interaction

Enumeration: Example

- Consider SQL query interface
  - With Partial Query Completion
- Typing in "em" has exposed projection, join, and selection options.
Guided Interaction

Insights

• The database must attempt to surface as many insights from the data as possible.

• Removes informational dependencies
• Aids expression of query intent
• Note: Should not overwhelm the user
Guided Interaction

Insights: Example

- Consider SQL interface with range / numeric value selection
- Visual / interactive feedback saves dependent query
- Does my DB let me build something like this?

WHERE emp.age > 60

Distribution of values in column
Guided Interaction

Responsiveness

• All interactions must be instantaneous even if inaccurate.

• Fluid data interaction is key to getting insights

• Tradeoff accuracy for near-instantaneous responses (i.e. <100ms*)

Guided Interaction

Responsiveness: Example

- SQL query interface, Partial Query Completion
- Need to deliver results in <100ms
Summary

• Shortcomings in the Query-Result Model
  • Challenges

• Proposed Solution: Guided Interaction
  • Enumeration
  • Insights
  • Responsiveness

• Designing DBs that abide by these principles overcomes these shortcomings
  • Many fundamental building blocks already exist
Thanks! Questions?

Join me at the OSU Database Group!

http://arnab.org