Problem statement

Given a conjunctive query

\[ Q = \bigwedge_{i=1}^{m} A_i \land B_j \]

over a schema with access restrictions over the schema relations and integrity constraints \( \Sigma \) and a cost function \( C \) find a plan that minimises the cost to answer \( Q \) under \( C \).

Access restrictions over relations require values of certain attributes to be given as inputs to access relations.

The integrity constraints \( \Sigma \) are given by tuple-generating dependencies (TGDs):

\[ \forall_{x_1, \ldots, x_n} A \Rightarrow \exists_{y_1, \ldots, y_m} B \]

Access methods

- \( \text{Select} \) Details(cid, cname, pphone)
- \( \text{Product}(\text{pid}, \text{pname}, \text{pprice}) \)
- \( \text{Bought}(\text{cid}, \text{itemid}) \)

Example

<table>
<thead>
<tr>
<th>Relations</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer(cid)</td>
<td>CustomerDetails(cid, cname, pphone)</td>
</tr>
<tr>
<td>Product(pid, pname, pprice)</td>
<td>Bought(cid, itemid)</td>
</tr>
</tbody>
</table>

\[ Q: \text{pid, pname, itemid} | | CustomerDetails(cid, cname, pphone) | | Bought(cid, itemid) | | Product(pid, pname, pprice) | | Bought(cid, itemid) \]

Hidden database: \[ \text{CustomerDetails}(\text{cid}, \text{cname}, \text{pphone}), \text{Bought}(\text{cid}, \text{itemid}) \]

Prototype with LogicBlox

LogicBlox is a relational database geared toward analytics and predictions.

LogicBlox programs are implemented in the proprietary language LogicQL, derived from Datalog.

A PDQ server instance is started upon the creation or opening of an LB workspace.

PDQ is initialized with all relevant information from the workspace such as views and constraints.

PDQ optimizes LB rules, by offering equivalent rewritings with different cost.

During planning PDQ asks LB for its estimation about the cost of a specific subplan.

When a transaction ends successfully, PDQ is updated to account for the objects that have been created or destroyed.

REFERENCES


