Adaptive Query Formulation to Handle Database Evolution

George Papastefanatos1, Panos Vassiliadis2, Yannis Vassiliou1

1National Technical University of Athens, Dept. of Electrical and Computer Eng., Athens, Hellas
{gpapas,yv}@dbnet.ece.ntua.gr
2University of Ioannina, Dept. of Computer Science, Ioannina, Hellas
pvassil@cs.uoi.gr

Current database systems are continuously evolving environments, where design constructs are
- added
- removed
- modified

Evolution is not handled by current DBMS with an automatic way, but rather they require great human effort

Existing Queries affected:
- Syntactically - i.e., become invalid
- Semantically - i.e., query must conform to the new database semantics

Adaptation of SQL queries and views
- time-consuming task
- treated in most of the cases manually by the administrators/developers

Graph-based representation of database constructs (i.e., relations, views, constraints, queries)

Annotation of graph with rules for adapting queries to database schema evolution

Mechanism for performing what-if analysis for potential changes of database configurations

Graph-based modeling
- Database Constructs mapped to directed graphs
  - Relations
  - Conditions (covering database constraints and query conditions)
  - Queries
  - Views
- Graph Semantics
  - Nodes represent Database Constructs, i.e., relation nodes, attribute nodes, query nodes, etc.
  - Edges represent Relationships Between Constructs, i.e., schema edges, map-select edges, operand edges, etc.

Transformed Query Graph
- Annotated Query Graph
- Event
- Transformed Query Graph

Extending SQL With Evolution Semantics
- ON <event> TO <element> THEN <policy>
  - E.g. SELECT Emp#, NAME, AGE FROM V
  - ON condition addition TO V THEN propagate,
  - ON attribute deletion TO V.AGE THEN block

Example
- Annotated Query Graph
- Event
- Transformed Query Graph

Visualization Tool