Database Versioning



Kai Herrmann, Database Technology Group, TU Dresden

DATABASE VERSIONING – OUR MOTIVATION





CODEL – A RELATIONALLY COMPLETE DATABASE **EVOLUTION LANGUAGE**

- CREATE TABLE $R(c_1, \ldots, c_n)$;
- DROP TABLE *R*;
- ADD COLUMN c AS $f(c_1, ..., c_n)$ INTO R;
- DROP COLUMN *c* FROM *R*;
- DECOMPOSE TABLE R INTO S (s_1, \dots, s_n) [, $T(t_1, \dots, t_m)$];
- [OUTER] JOIN TABLE *R*, *S* INTO *T* WHERE *cond*;
- AGGREGATE $R(g_1, ..., g_n)$ WITH $a_1 = f_1(G, V), ..., a_m = f_m(G, V)$ [INTO S];
- PARTITION TABLE *R* INTO *S* WITH cond_*S* [, *T* WITH cond_*T*];
- MERGE TABLE R, S INTO T;
- RENAME TABLE R INTO R';
- RENAME COLUMN *c* IN *R* TO *c*';
- Codel is inspired by PRISM/PRISM++ (C. Curino, H. J. Moon, A. Deutsch, and C. Zaniolo, "Automating the database schema evolution process," VLDB J., Dec. 2012.)
- We showed relational completeness of CoDEL.

INVERDA – OUR SOLUTION

EVOLUITION CONTINUE AppA-0; PARTITION TABLE Person INTO Kids WITH age<18, Person WITH age<=18; **EVOLUTION COMMIT AS AppA-1;**

EVOLUTION START; CREATE TABLE Person (name STRING, age INTEGER, address TEXT); ALTER TABLE Person ADD PK pk_person(name); CREATE TABLE Company (name STRING, manager STRING); ALTER TABLE Company ADD PK pk_comp(name); ALTER TABLE Company ADD FK fk_comp (manager) REFERENCES Person(name); **EVOLUTION COMMIT AS AppA;**

EVOLUTION CONTINUE AppA-0; DECOMPOSE TABLE Person INTO Person(name, age), PersonAddress(name, address); ADD COLUMN address INTO Company AS "; MERGE TABLE PersonAddress, Company INTO Customer; **EVOLUTION COMMIT AS AppB**; 3



DATALOG – THE KEY TO INVERTIBILITY



Goal:

 $Source = get_{Source} \left(put_{Target} (Source) \right)$ $Target = get_{Target} \left(put_{Source} (Target) \right)$

PARTITION T INTO R WITH cond_R, S WITH cond_S DEFAUL R;

- $get_{Target} \begin{cases} R(p,a) \leftarrow T(p,a), cond_{R}(p,a), \neg R^{-}(p) \\ S(p,a) \leftarrow T(p,a), cond_{S}(p,a), \neg S^{+}(p,_), \neg S^{-}(p) \\ S(p,a) \leftarrow S^{+}(p,a) \\ T'(p,a) \leftarrow T(p,a), \neg cond_{R}(p,a), \neg cond_{S}(p,a) \end{cases}$
- $T(p,a) \leftarrow R(p,a)$ $T(p, a) \leftarrow S(p, a), \neg R(p, _)$ $get_{Source} \begin{cases} T(p,a) \leftarrow T'(p,a) \\ S^+(p,a) \leftarrow S(p,a), R(p,a'), a \neq a' \end{cases}$ $S^{-}(p) \leftarrow R(p, a), \neg S(p, _), cond_{S}(p, a)$ $R^{-}(p) \leftarrow S(p, a), \neg R(p, _), cond_{R}(p, a)$

FURTHER RESEARCH QUESTIONS

• General topics

- Evolution and versioning of role based models
- Explicit query optimization
- External level
 - Versioning model
 - Invariant variants
- Internal level
 - Incremental materialization, Zero downtime
 - Evaluation and advisor for materialization
 - Alternative materialization