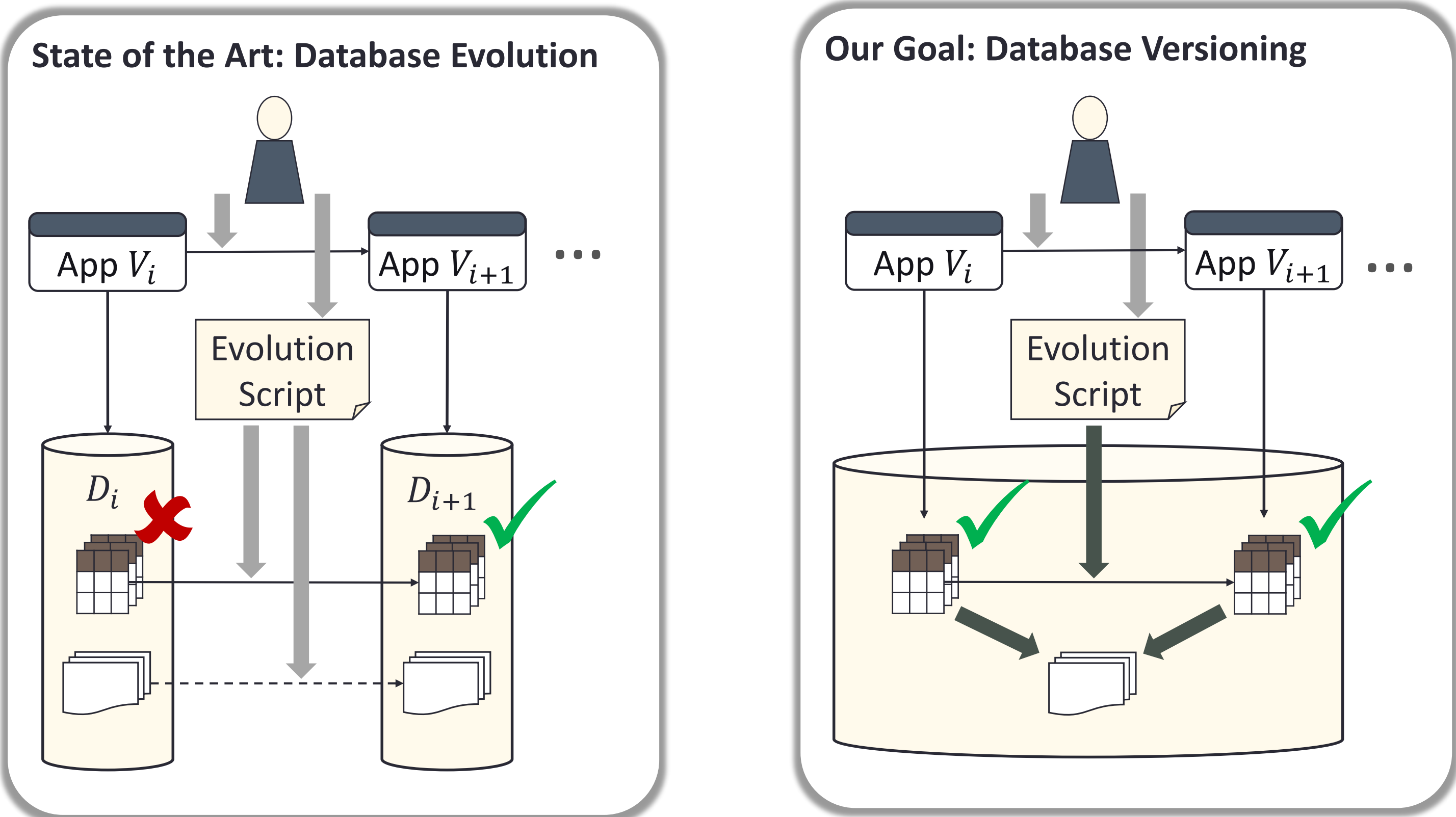


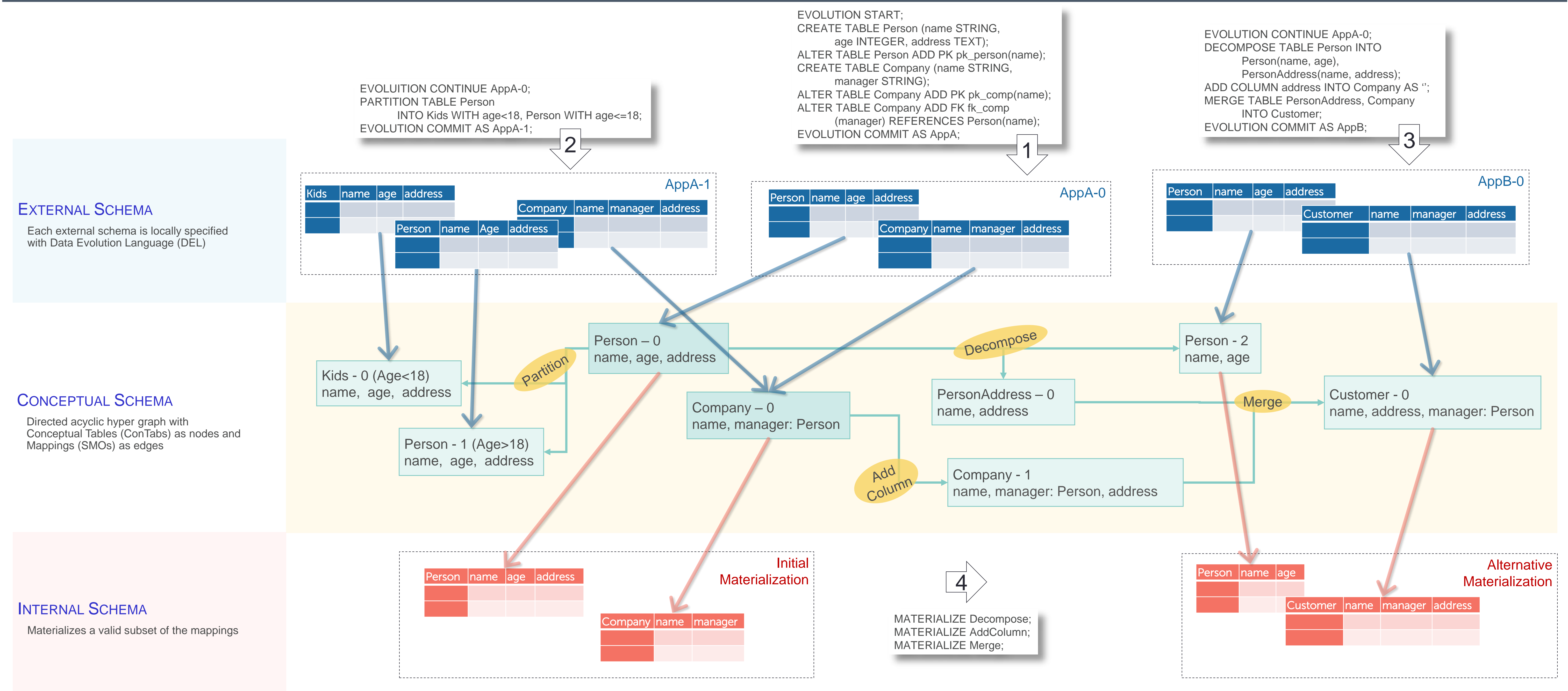
DATABASE VERSIONING – OUR MOTIVATION



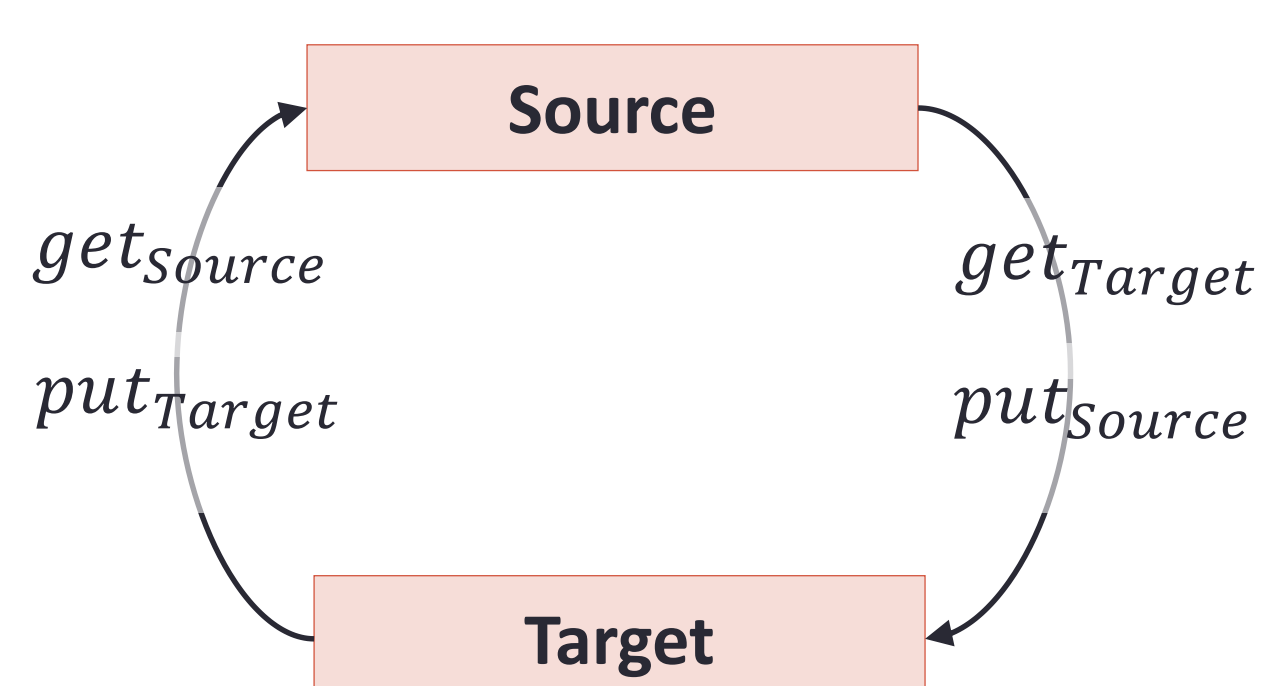
CODEL – A RELATIONALLY COMPLETE DATABASE EVOLUTION LANGUAGE

- CREATE TABLE $R(c_1, \dots, c_n)$;
 - DROP TABLE R ;
 - ADD COLUMN c AS $f(c_1, \dots, c_n)$ 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 $R(g_1, \dots, g_n)$ WITH $a_1 = f_1(G, V), \dots, a_m = f_m(G, V)$ [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



DATALOG – THE KEY TO INVERTIBILITY



Goal:

$$Source = get_{source}(put_{target}(Source))$$

$$Target = get_{target}(put_{source}(Target))$$

PARTITION T INTO R WITH $cond_R$, S WITH $cond_S$ DEFAULT R;

$$get_{target} \begin{cases} R(p, a) \leftarrow T(p, a), cond_R(p, a), \neg R^-(p) \\ S(p, a) \leftarrow T(p, a), cond_S(p, a), \neg S^+(p, a), \neg S^-(p) \\ S(p, a) \leftarrow S^+(p, a) \\ T'(p, a) \leftarrow T(p, a), \neg cond_R(p, a), \neg cond_S(p, a) \end{cases}$$

$$get_{source} \begin{cases} T(p, a) \leftarrow R(p, a) \\ T(p, a) \leftarrow S(p, a), \neg R(p, a) \\ T(p, a) \leftarrow T'(p, a) \\ S^+(p, a) \leftarrow S(p, a), R(p, a'), a \neq a' \\ S^-(p) \leftarrow R(p, a), \neg S(p, a), cond_S(p, a) \\ R^-(p) \leftarrow S(p, a), \neg R(p, a), cond_R(p, a) \end{cases}$$

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