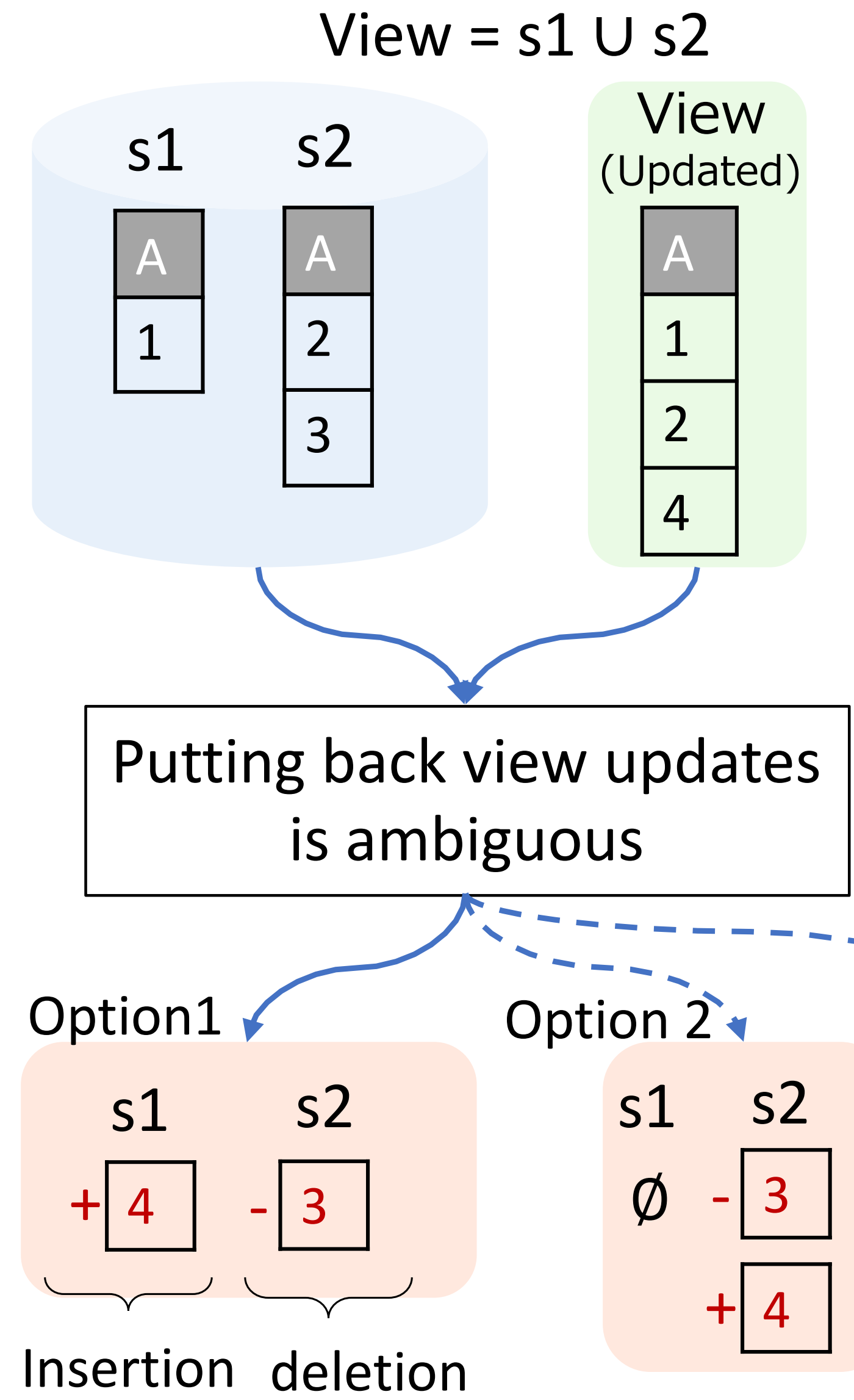
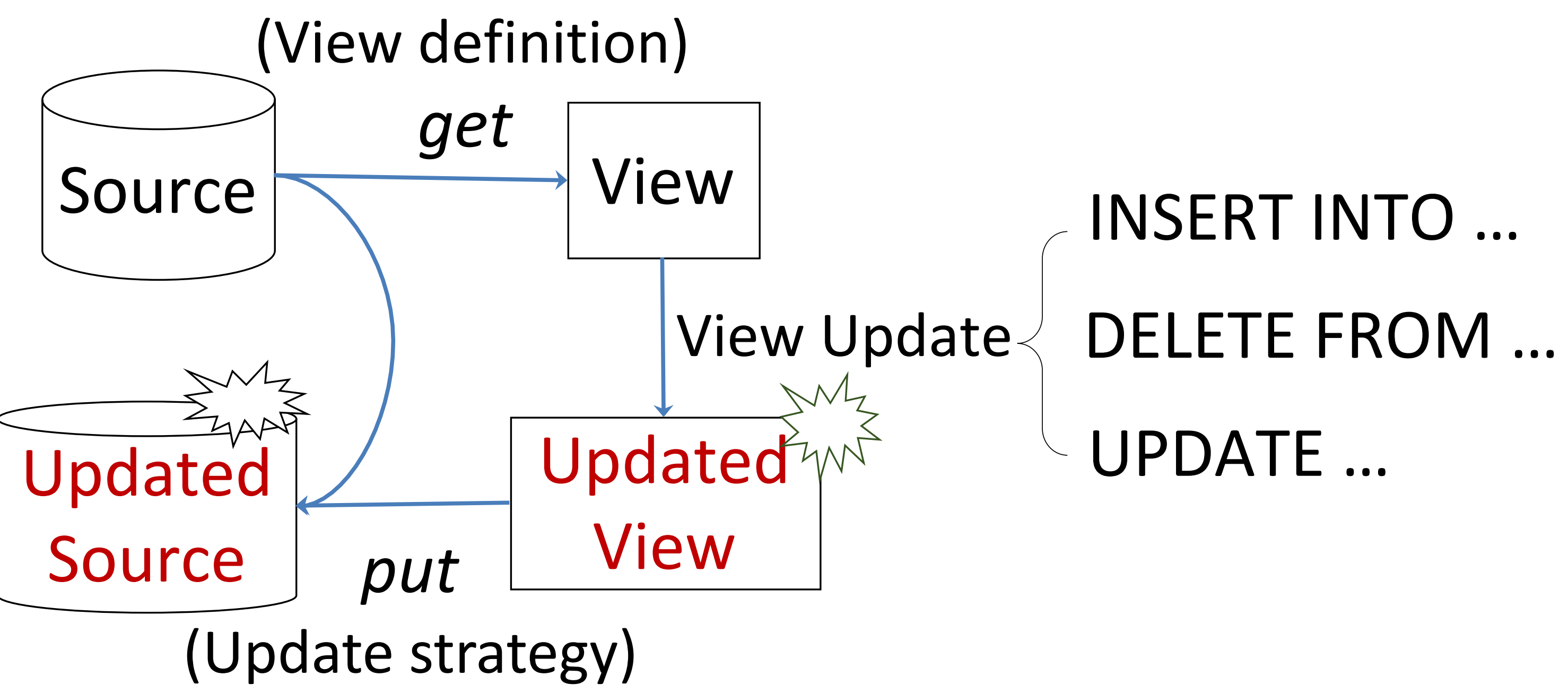


# The View Update Problem

# Programmable View Update Strategy



A putback strategy (Option1) specified in Datalog:

```
-s1(X) :- s1(X), not view(X).
-s2(X) :- s2(X), not view(X).
+s1(X) :- view(X), not s1(X), not s2(X).
```

Correctness?

Round-tripping properties:

```
put(Source, get(Source)) = Source
get(put(Source, UpdatedView)) = UpdatedView
```

A Bidirectional Transformation =  $\langle \text{get}, \text{put} \rangle$

# Validation Algorithm

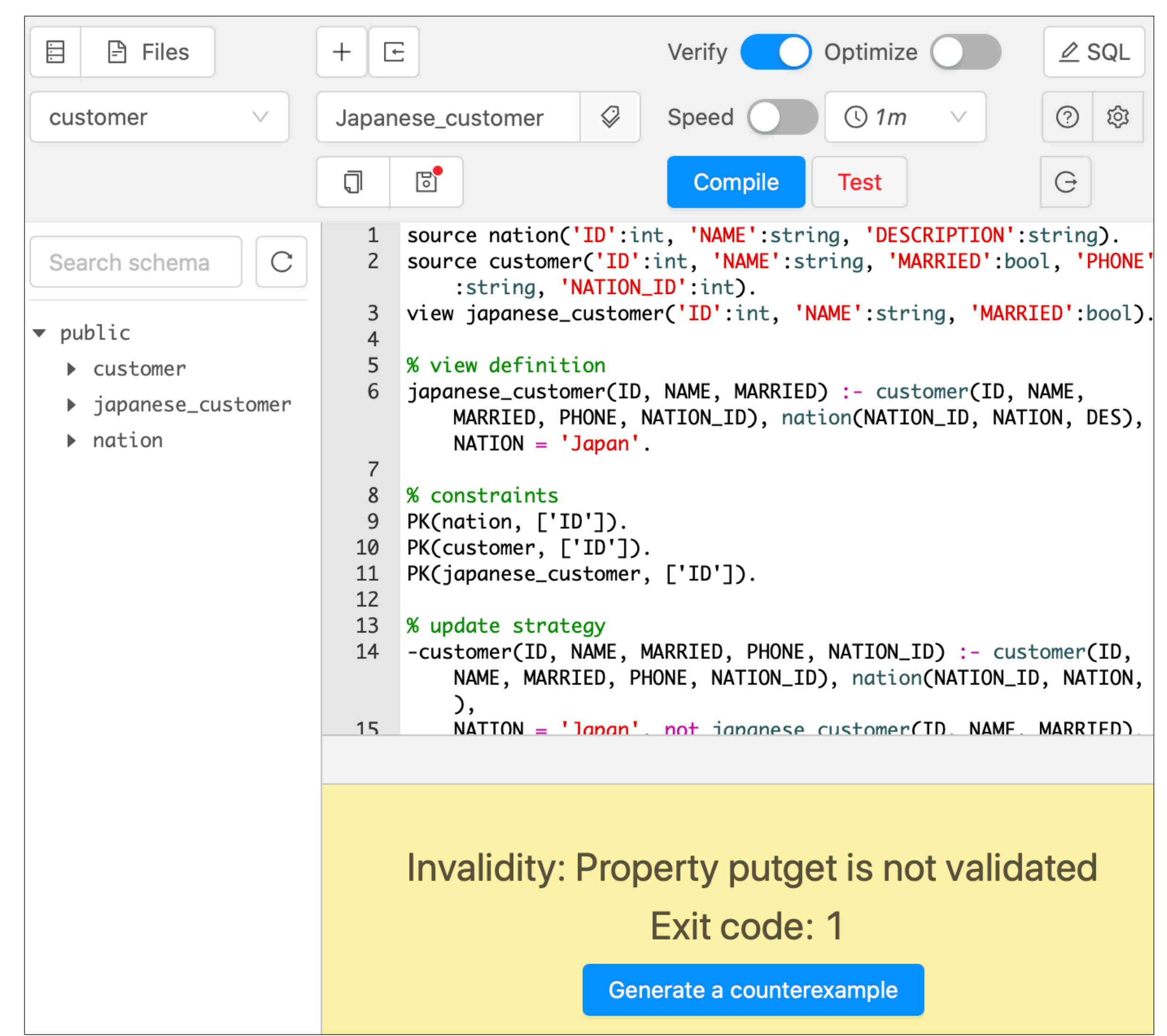
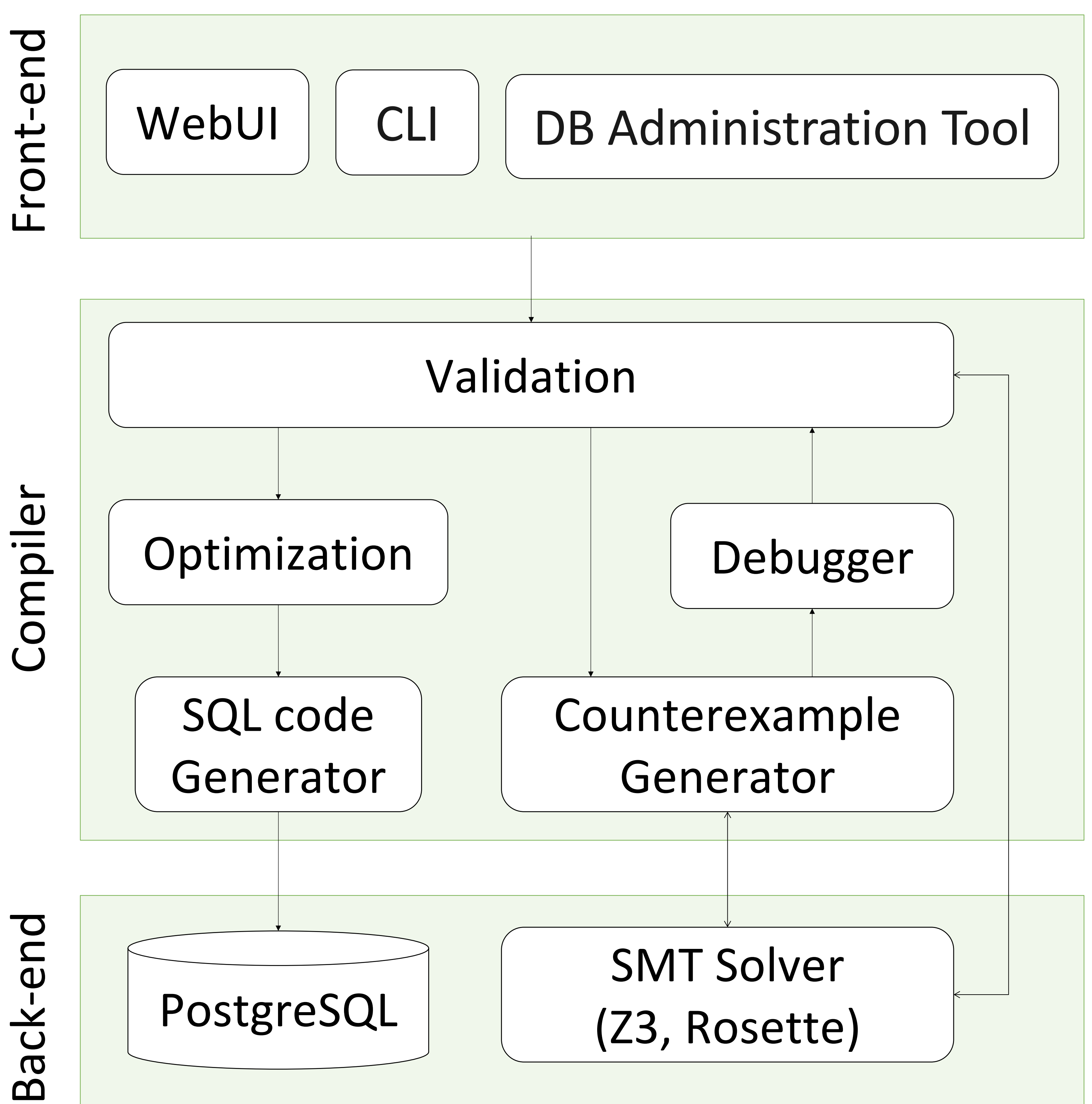
Input: A Datalog-written view update strategy.  
Output: The validity of the input Datalog program.

```

graph TD
    A[Overlap between insertions & deletions?] -- No --> B[GetPut? Existence of get?]
    A -- Yes --> C(Invalid)
    B -- No --> C
    B -- Yes --> D[PutGet?]
    D -- No --> C
    D -- Yes --> E(Valid)
  
```

The algorithm is sound and complete for LVGN-Datalog.

# The BIRDS System Architecture



## REFERENCES

[1] Project Page: <https://dangtv.github.io/BIRDS/>.  
 [2] V.-D. Tran, H. Kato, Z. Hu. Programmable View Update Strategies on Relations. *PVLDB*, 13(5):726–739, 2020.  
 [3] V.-D. Tran, H. Kato, Z. Hu. BIRDS: Programming view update strategies in Datalog. *PVLDB*, 13(12): 2897-2900, 2020.