VESEL: VISUAL EXPLORATION OF SCHEMA EVOLUTION USING PROVENANCE QUERIES

Christos Athinaiou, Haridimos Kondylakis
Institute of Computer Science, FORTH-ICS, Heraklion, Crete, Greece
Computer Science Department, University of Crete, Crete, Greece

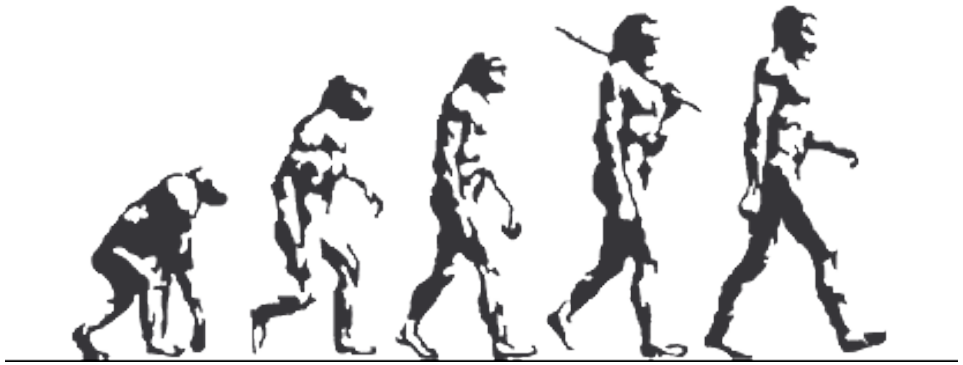
BIGVIS@EDBT 2019

SOME FACTS ABOUT DATABASES

Everything that exists it is only change"

-Heraclitus 535 BCE

- Database schemata are subject to continuous change
- Recent databases do not proactively support schema evolution
  - Developers have to migrate data from one database version to another using error-prone and complex ETL scripts

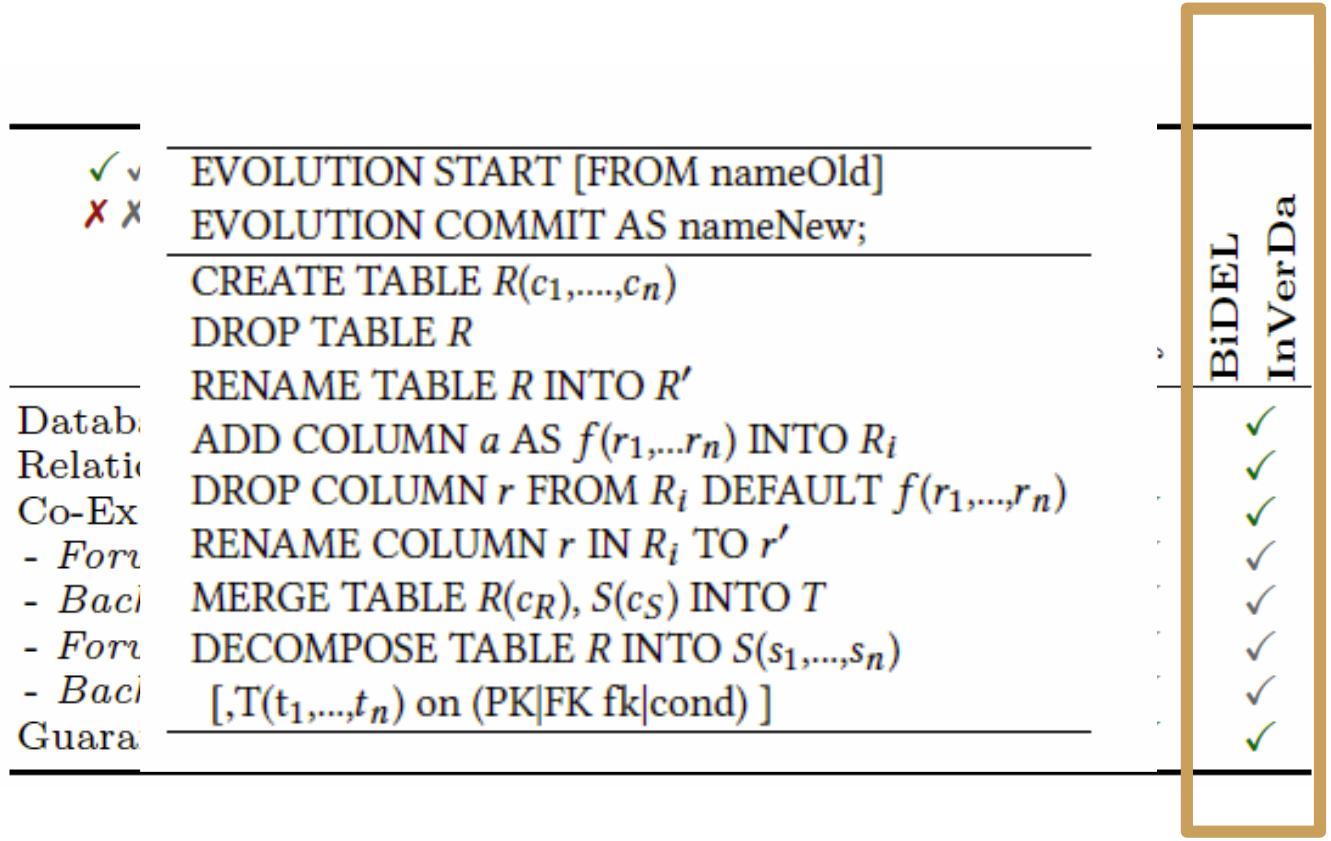


IDEALLY

- We would like tools enabling automatic data migration
- Tools enabling the understanding of what has been changed!
  - Identify modelling choices of the past
  - Do not reinvent the wheel, avoid the same mistakes again!

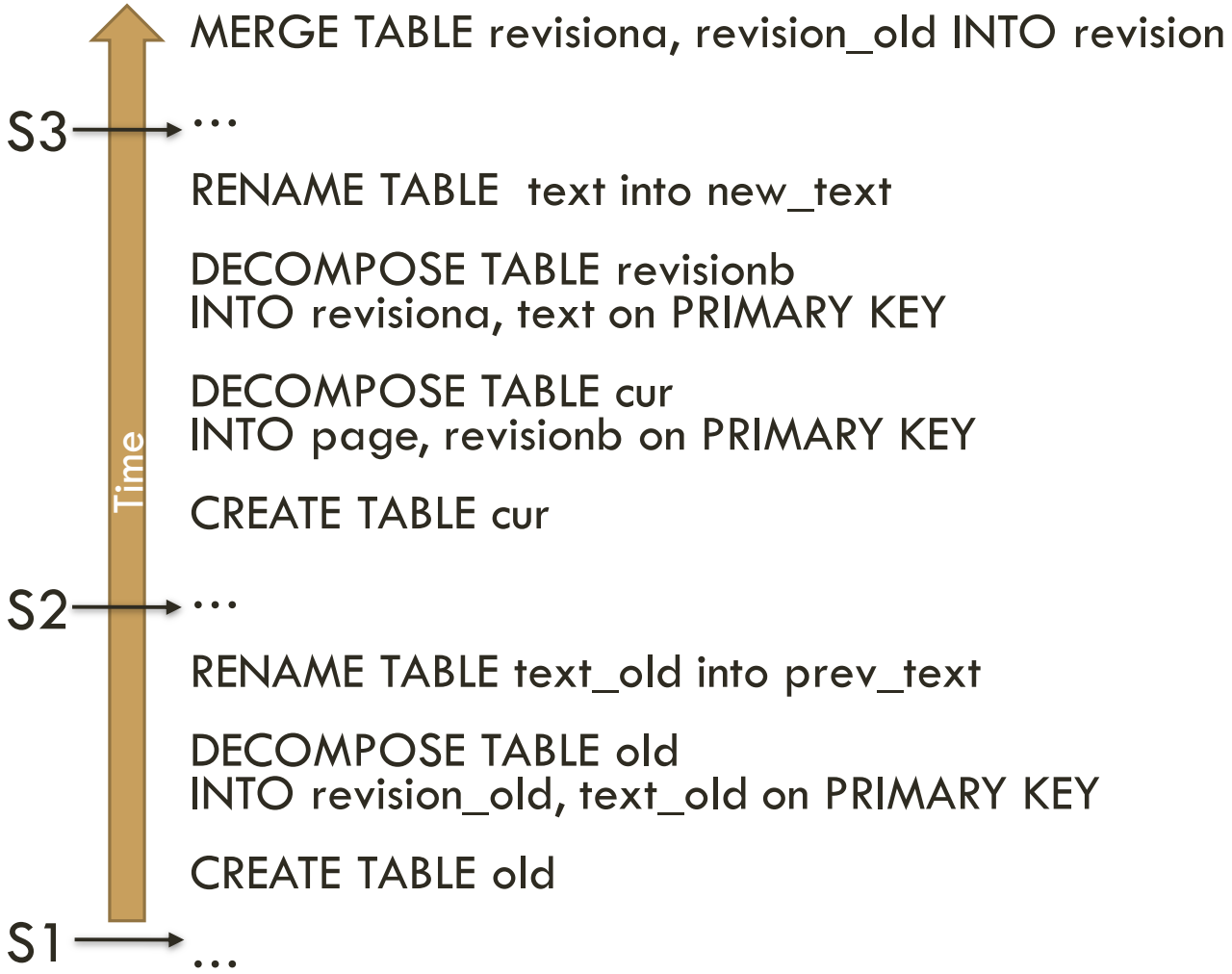
PROPOSED SOLUTIONS FOR SCHEMA EVOLUTION

- Model Management approach
  - tools to match, diff, merge and extract mappings between schema versions
- Multiple tools offer Schema Modification Operations (SMOs)
  - implemented for data migration and for rewriting past queries to work in the new versions



	BiDEL	InVerDa
EVOLUTION START [FROM nameOld]		
EVOLUTION COMMIT AS nameNew;		
CREATE TABLE $R(c_1,....,c_n)$		
DROP TABLE R		
RENAME TABLE R INTO R'		
ADD COLUMN a AS $f(r_1,...r_n)$ INTO R_i	✓	✓
DROP COLUMN r FROM R_i DEFAULT $f(r_1,...,r_n)$	✓	✓
RENAME COLUMN r IN R_i TO r'	✓	✓
MERGE TABLE $R(c_R)$, $S(c_S)$ INTO T	✓	✓
DECOMPOSE TABLE R INTO $S(s_1,...,s_n)$ [,T($t_1,...,t_n$) on (PK|FK fk|cond)]	✓	✓

RUNNING EXAMPLE



ARE WE DONE YET?

Give the SMOs to the users!

Which SMO introduced table Revision?

In which schema version Revision was introduced?

Which is the sequence of SMOs that introduced Revision?

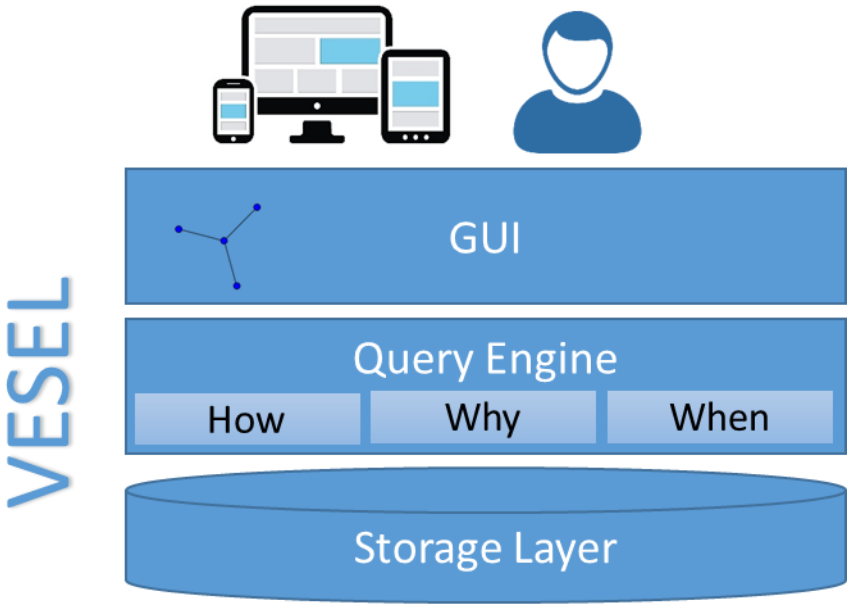
EVOLUTION START; CREATE TABLE user (user_id INT, //autoincrement?? user_name TEXT, user_rights TEXT, //blob user_password TEXT, user_newpassword TEXT, user_email TEXT, user_options TEXT, user_touched TEXT); CREATE TABLE user_newtalk (user_id INT, user_ip TEXT, cur_user_text TEXT, cur_timestamp TEXT, cur_restrictions TEXT, cur_counter TEXT); CREATE TABLE old (old_id INT, old_namespace INT, old_title TEXT, inverse_timestamp TEXT);CREATE TABLE archive (ar_namespace INT, ar_title TEXT);CREATE TABLE links (l_from TEXT, l_to INT);CREATE TABLE brokenlinks ss_total_views INT, ss_total_edits INT, ss_good_articles INT); CREATE TABLE ipblocks (ipb_address TEXT, ipb_user INT, ipb_by INT, ipb_reason TEXT, ipb_timestamp TEXT); CREATE TABLE image (img_name TEXT, img_size INT, img_description TEXT, img_user INT, img_user_text TEXT, img_timestamp TEXT); CREATE TABLE oldimage (oi_name TEXT, oi_archive_name TEXT, oi_size INT, oi_description TEXT, oi_user INT, oi_user_text TEXT, oi_timestamp TEXT);CREATE TABLE random (ra_current INT, ra_title TEXT); CREATE TABLE recentchanges (rc_timestamp TEXT, rc_cur_time TEXT, rc_user INT, rc_user_text TEXT, rc_namespace INT, rc_title TEXT, rc_comment TEXT, rc_minor INT, rc_bot INT, rc_new INT, rc_cur_id INT, rc_this_oldid INT, rc_last_oldid INT);CREATE TABLE watchlist (wl_user INT, wl_namespace INT, wl_title TEXT);CREATE TABLE math (math_inputhash TEXT, math_outputhash TEXT, math_html_conservativeness INT, math_html TEXT, math_mathml TEXT);CREATE TABLE searchindex (si_page INT, si_title TEXT, si_text TEXT);EVOLUTION COMMIT AS v01284;EVOLUTION START FROM v01284;DROP TABLE random;EVOLUTION COMMIT AS v01308;EVOLUTION START FROM v01308;CREATE TABLE interwiki (iw_prefix increment";DB "CREATE SEQUENCE increment STAR v01640;EVOLUTION START FROM v01640;CREAT hitcounter (hc_id INT);EVOLUTION COMMIT AS v recentchanges;ADD COLUMN rc_moved_to_title TE ipblocks;EVOLUTION COMMIT AS v02473;EVOLU COLUMN l_from_tmp IN links TO l_from;CREATE TABLE imagelinks (il_from INT, il_to TEXT);DROP COLUMN lcc_title FROM linkscc DEFAULT """";DROP TABLE imagelinks;EVOLUTION COMMIT AS v02687;EVOLUTION START FROM v02687;ADD COLUMNuser_real_name TEXT AS """" TO user;EVOLUTION COMMIT AS v03204;EVOLUTION START FROM v03204;CREATE TABLE querycache (qc_type TEXT, qc_value INT, qc_namespace INT, qc_title TEXT);EVOLUTION COMMIT AS v03513;EVOLUTION START FROM v03513;CREATE TABLE objectcache (keyname TEXT, value TEXT, exptime DATE);EVOLUTION COMMIT AS v03516;EVOLUTIONSTART FROM v03516;CREATE TABLE categorylinks (cl_from INT, cl_to TEXT, cl_sortkey TEXT, cl_timestamp DATE);EVOLUTION COMMIT AS v03582;EVOLUTION START FROM v03582;ADD COLUMN rc_ip TEXT AS """" TO recentchanges;EVOLUTION COMMIT AS v04073;EVOLUTION START FROM v04073;CREATE TABLE blobs (blob_index TEXT, blob_data TEXT);EVOLUTIONCOMMIT AS v04083;EVOLUTION START FROM v04083;CREATE TABLE validate (val_user INT, val_title TEXT, val_timestamp TEXT, val_type INT, val_value INT);EVOLUTION COMMIT AS v04321;EVOLUTION START FROM v04321;ADD COLUMN val_comment TEXT AS """" TO validate;EVOLUTION COMMIT AS v04332;EVOLUTION START FROM v04332;ADD COLUMN rc_id INT AS"nextval('incre v04619;EVOLU TEXT);EVOLUTIO user_options, use TOuser;EVOLUT COMMIT AS v05541;//EVOLUTION DROP v05541;EVOLUTION START FROM v05532;CREATE TABLE group (group_id INT, group_name TEXT, group_description TEXT);CREATE TABLE user_groups (user_idINT, group_id INT);EVOLUTION COMMIT AS v05648;EVOLUTION START FROM v05648;RENAME COLUMN user_id IN user_rights TO ur_uid;RENAME COLUMN user_rights IN user_rights TO ur_rights;RENAME COLUMN user_id IN user_groups TO ug_uid;RENAME COLUMN group_id IN user_groups TO ug_gid;EVOLUTION COMMIT AS v06057;EVOLUTION START FROM v06057;RENAME COLUMN ur_uidIN user_rights TO ur_user;RENAME COLUMN ug_uid IN user_groups TO ug_user;RENAME COLUMN ug_gid IN user_groups TO ug_group;EVOLUTION COMMIT AS v06058;EVOLUTION START FROM v06058;ADD COLUMN group_rights TEXT AS """" TO group;EVOLUTION COMMIT AS v06072;EVOLUTION START FROM v06072;ADD COLUMN user_emailauthenticationtimestamp TEXT AS '"0"' TO user;

VESEL

A visual approach enabling the active exploration of schema evolution

Answers the following provenance questions

- How: Which was the operation that introduced a specific table or a specific column?
- When: At which version the specific table/column was introduced?
- Why: Which was the sequence of operations that led to the introduction of a specific table/column?



AN AFFECTING SCHEMA MODIFICATION OPERATION

$\delta_d(s)$ $\delta_a(s)$

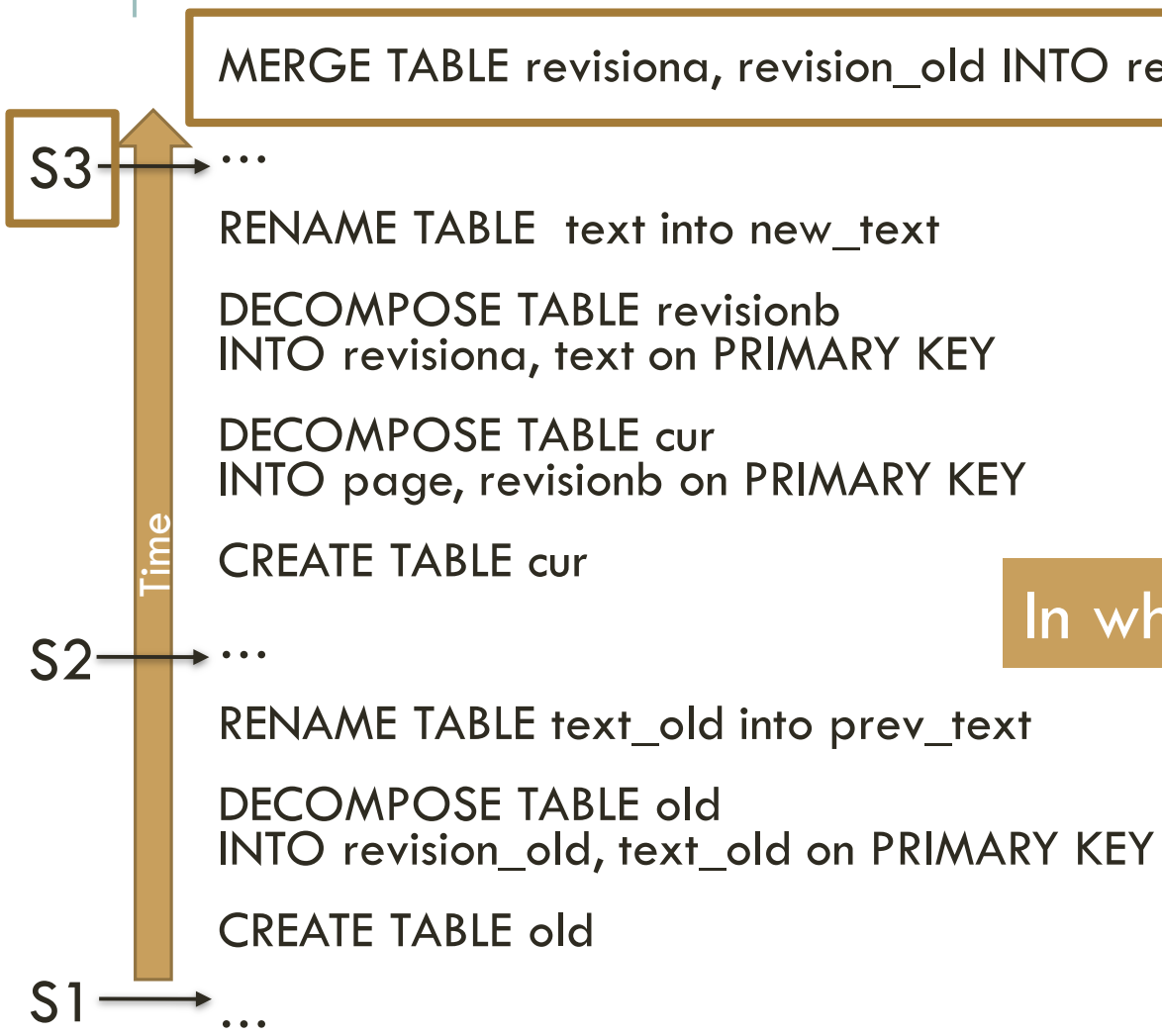
s: MERGE TABLE revisiona, revision_old INTO revision

- Let $\Delta^{Sk, Sm}$, (k < m) be the SMOs between versions S_k and S_m
- An SMO s in $\Delta^{Sk, Sm}$ affects a table/field t, if t in S_m and t in $\delta_a(s)$
- It captures the way a table/column was introduced between two schema versions

ANSWERING PROVENANCE QUERIES

- How: Which was the operation that introduced a specific table or a specific column?
  - Identify the affecting SMO scanning the delta log once
- When: At which version the specific table/column was introduced?
  - Identify to which version this affecting SMO belongs

AN EXAMPLE – ANSWERING HOW/WHEN QUERIES



Which SMO introduced table Revision?

In which schema version Revision was introduced?

ANSWERING PROVENANCE QUERIES

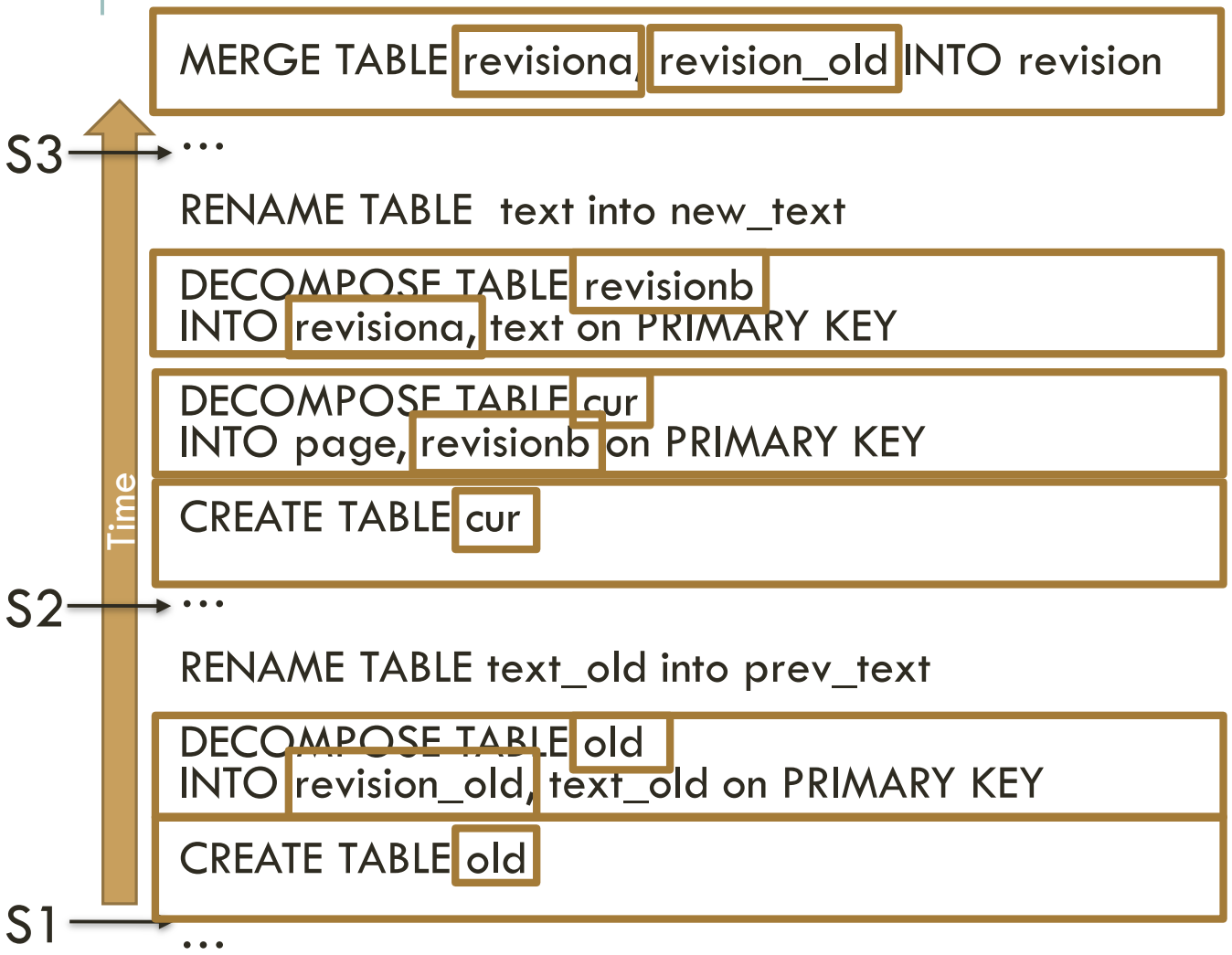
- Why: Which was the sequence of operations that led to the introduction of a specific table/column?

CHANGE SEQUENCE

A change sequence for a schema modification operation **s** in $\Delta^{Sk, Sm}$ denoted by CS^s, is the **minimal** sequence of schema modification operations in $\Delta^{Sk, Sm}$ such that

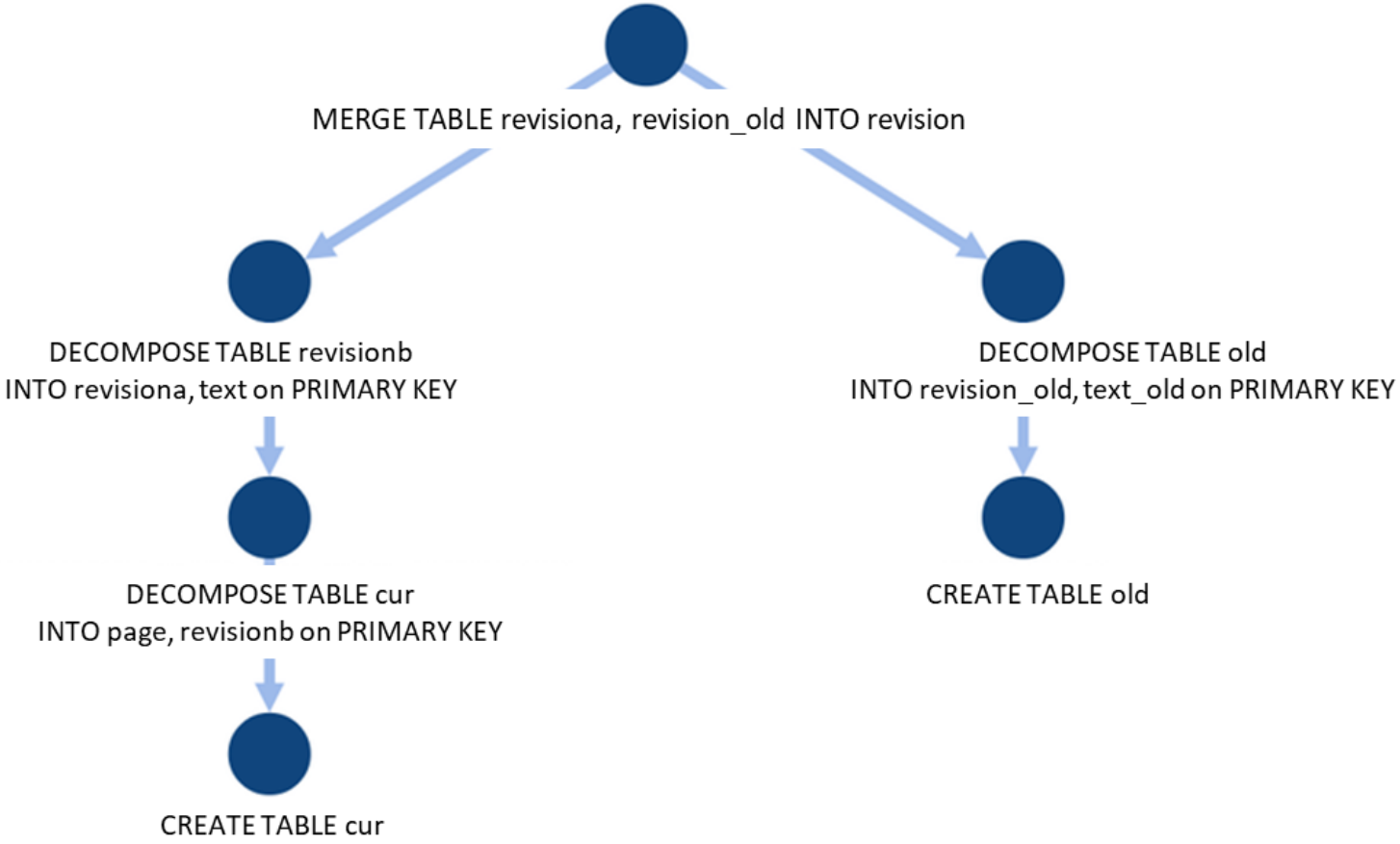
- **s** in CS^s
- when CS^s is applied to S_k we get the fields/tables participating in **s**
- one cannot remove any of the SMOs in the change sequence, and still when applied to the version S_k you get the fields/tables participating in **s**

AN EXAMPLE – ANSWERING WHY QUERIES



Which is the sequence of SMOs that introduced Revision?

THE CHANGE SEQUENCE (TREE)



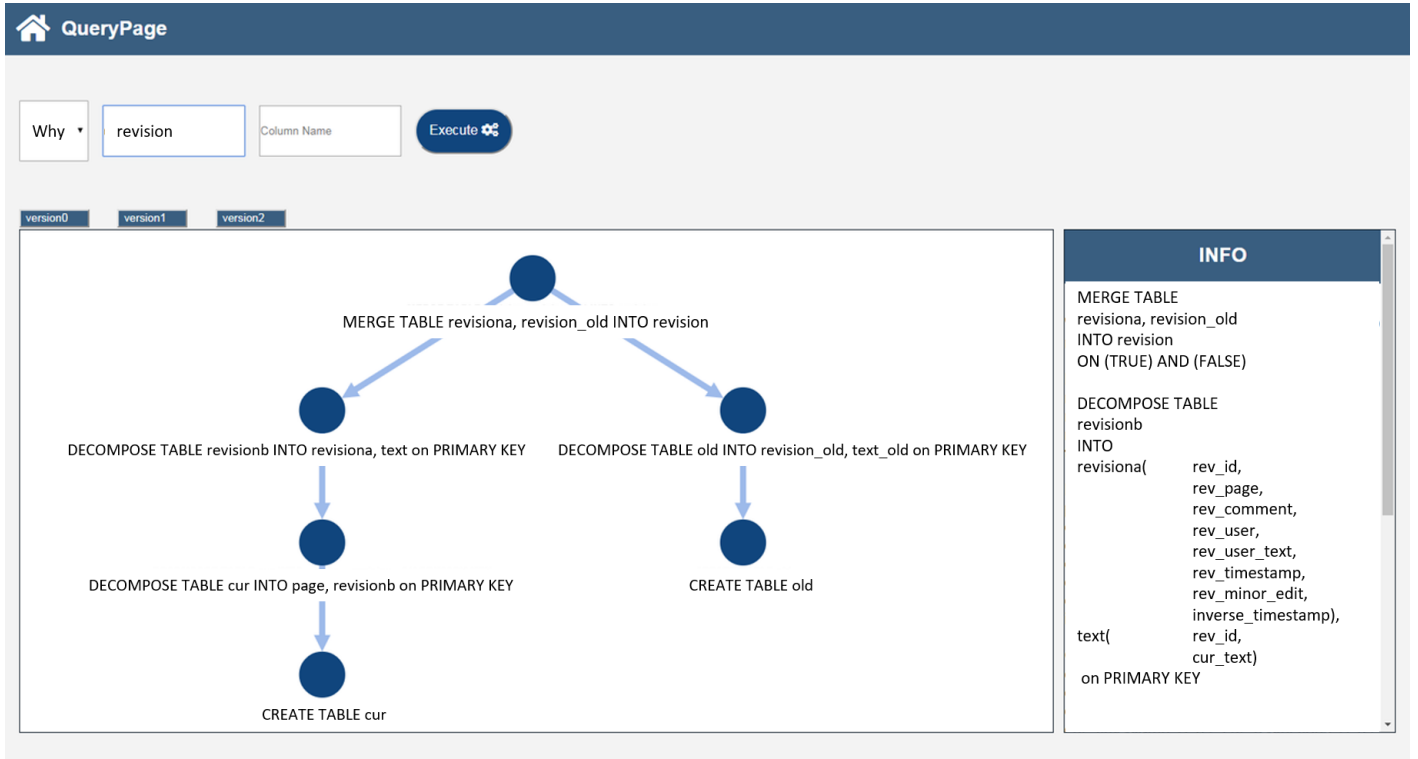
INTERESTING PROPERTIES

- A change sequence for the BiDEL language if it exists it is unique
- The change sequence is reversible and bidirectional

- This means we could get the evolution steps for a table/column of a past schema version till we reach the most recent version by providing as input the $\Delta^{Sm, Sk}$ to our algorithms – constructed by directly inversing $\Delta^{Sk, Sm}$

DEMONSTRATION

Wikimedia containing more than 170 schema versions with the corresponding SMOs



CONCLUSIONS

- VESEL is a novel system enabling the visual exploration of multiple schema versions, able to visualize the identified change sequence
- Advantages
  - Scalability
  - Simplicity
- Future Work
  - Investigate other SMO languages as well
  - Provide also statistical information on the evolution
  - Study the effects of visualizing schema evolution in collaborative/multi-user distributed environments
  - What about conflicts?

QUESTIONS ?