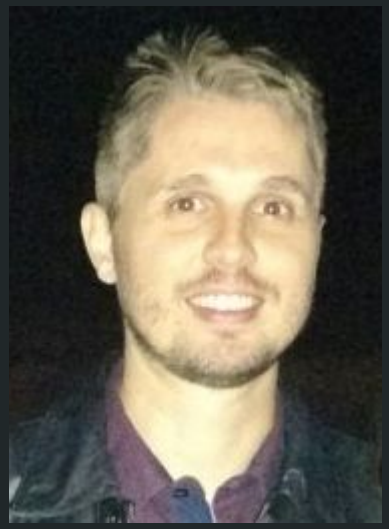
Invariant Control in Eventually Consistent Databases.

Paulo Arion Flores and Frank Siqueira



Paulo Arion Flores

System Analyst (CIASC)

MSc Candidate in Computer Science (UFSC)

BSc in Computer Science (UFSC)

pauloarion@gmail.com

https://www.linkedin.com/in/paulo-arion-flores-9b692313

Summary

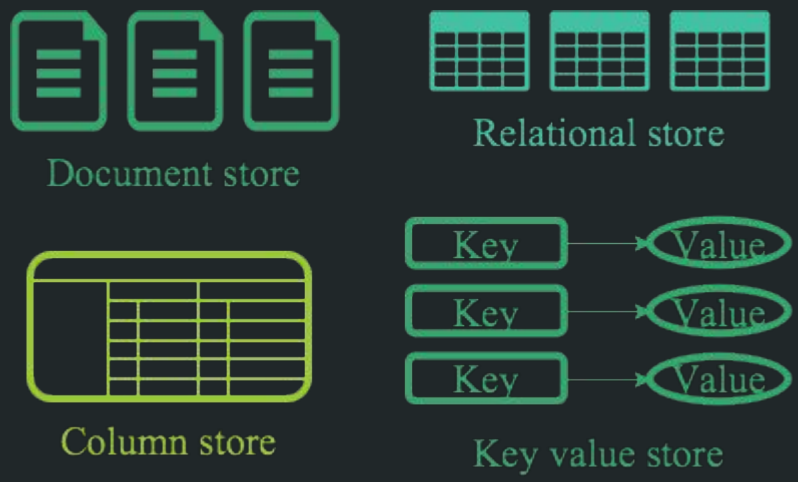
- ❏ Context
- ❏ Objectives
- ❏ Related Work
- ❏ Solution
- ❏ Example
- ❏ Evaluation
- ❏ Conclusion

Context

- Distributed Systems.
- Replicated databases.
- Consistency.
- ACID
- CAP

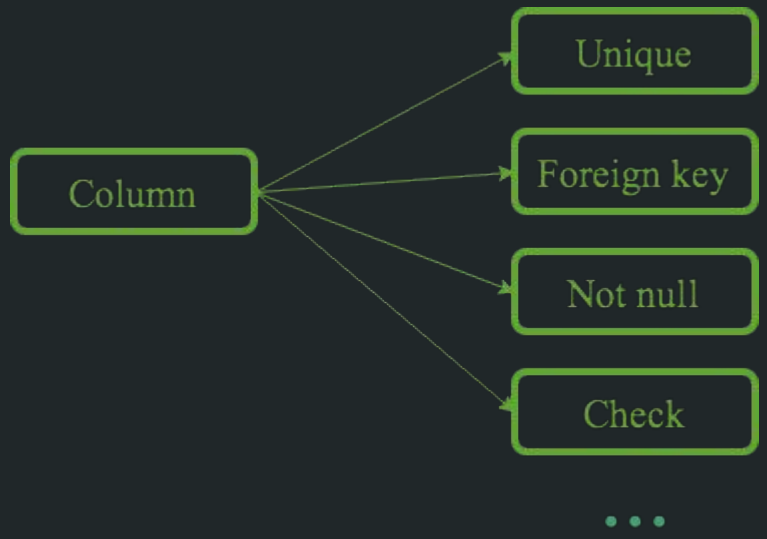
NoSQL

- Relational
- NoSQL
- NewSQL
- Data model
- Developers



Invariants

- Many approaches
- RDT
- Intermediate languages
- FOL
- Consistency levels
- Integrity constraints
- Multi-variable



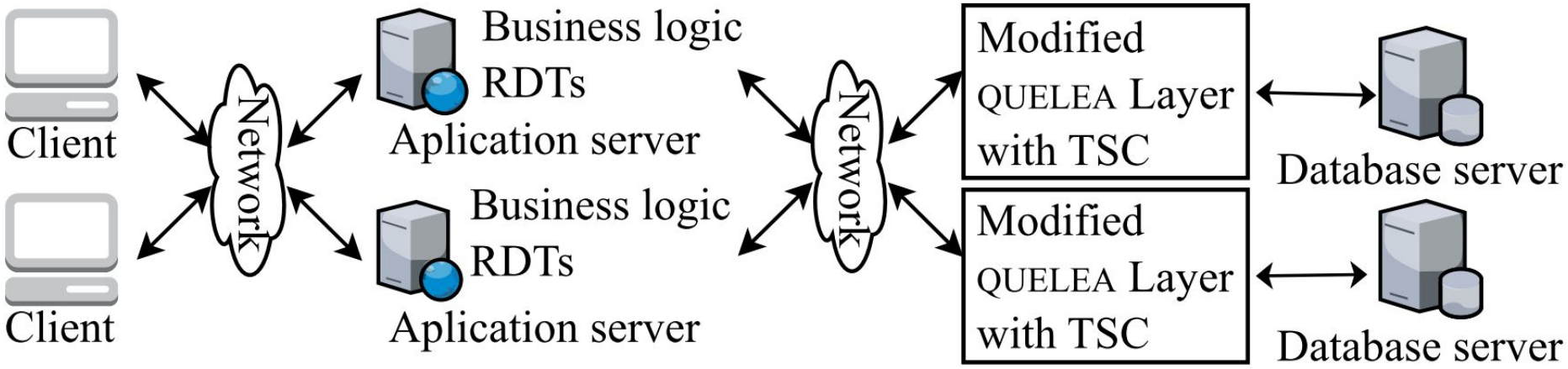
Objectives

- Replicated DDBS consistency.
- Integrity Constraints using RDT and FOL contracts.
- Multi-variable RDT control mechanism.

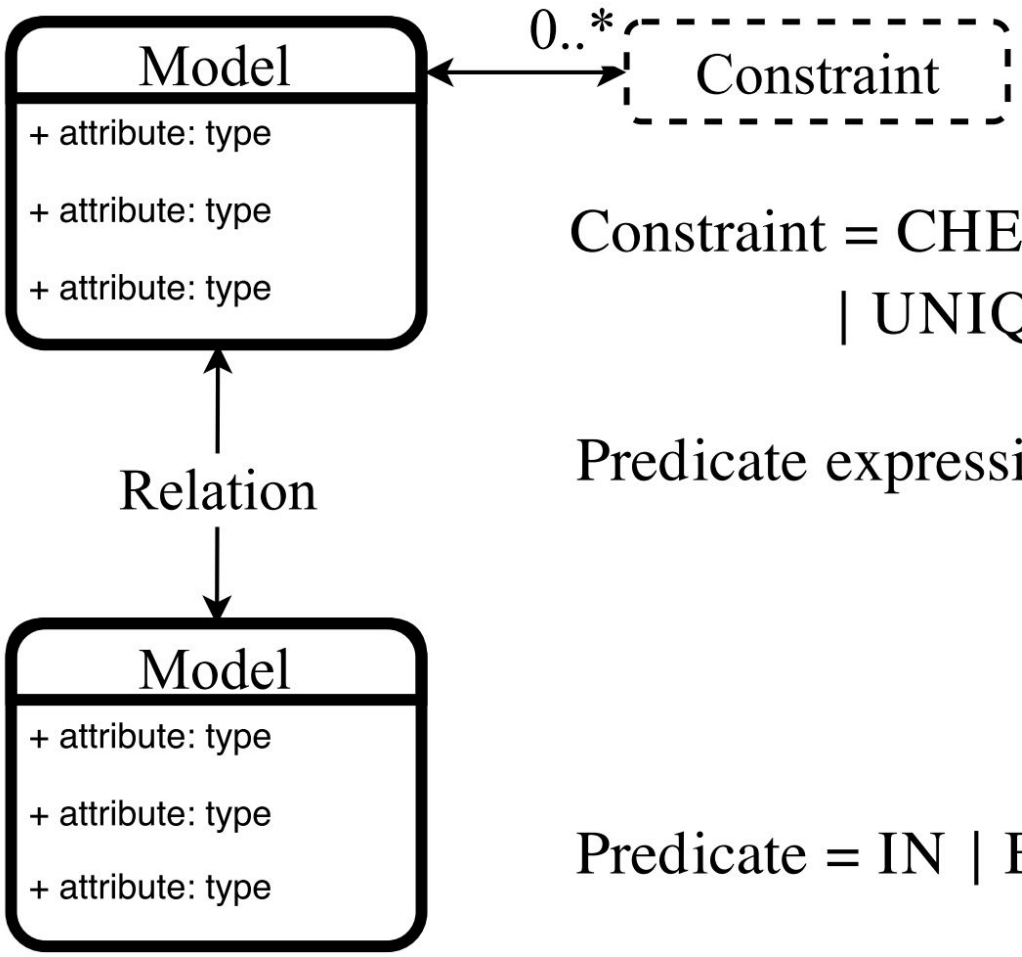
Related work

	Invariants	Consistency	Database
RedBlue (Li et al)	States	Red/Blue	Any
Indigo (Balegas et al)	Hoare logic	Reservations	Any
I-Confluence (Bailis et al)	States	Coordination-free	Any
CISE (Gotsman et al)	RDT	Hybrid	Any
SIEVE (Li et al)	States/CRDT	Red/Blue	SQL
QUELEA (Sivaramakrishnan et al)	RDT/FOL	Contracts	Any
Homeostasis protocol (Roy et al)	Symbolic tables	LR-slices and treaties	Any

Solution


Client
Client
Network
Business logic
RDTs
Aplication server
Business logic
RDTs
Aplication server
Network
Modified
QUELEA Layer
with TSC
Modified
QUELEA Layer
with TSC
Database server
Database server

Invariant Description



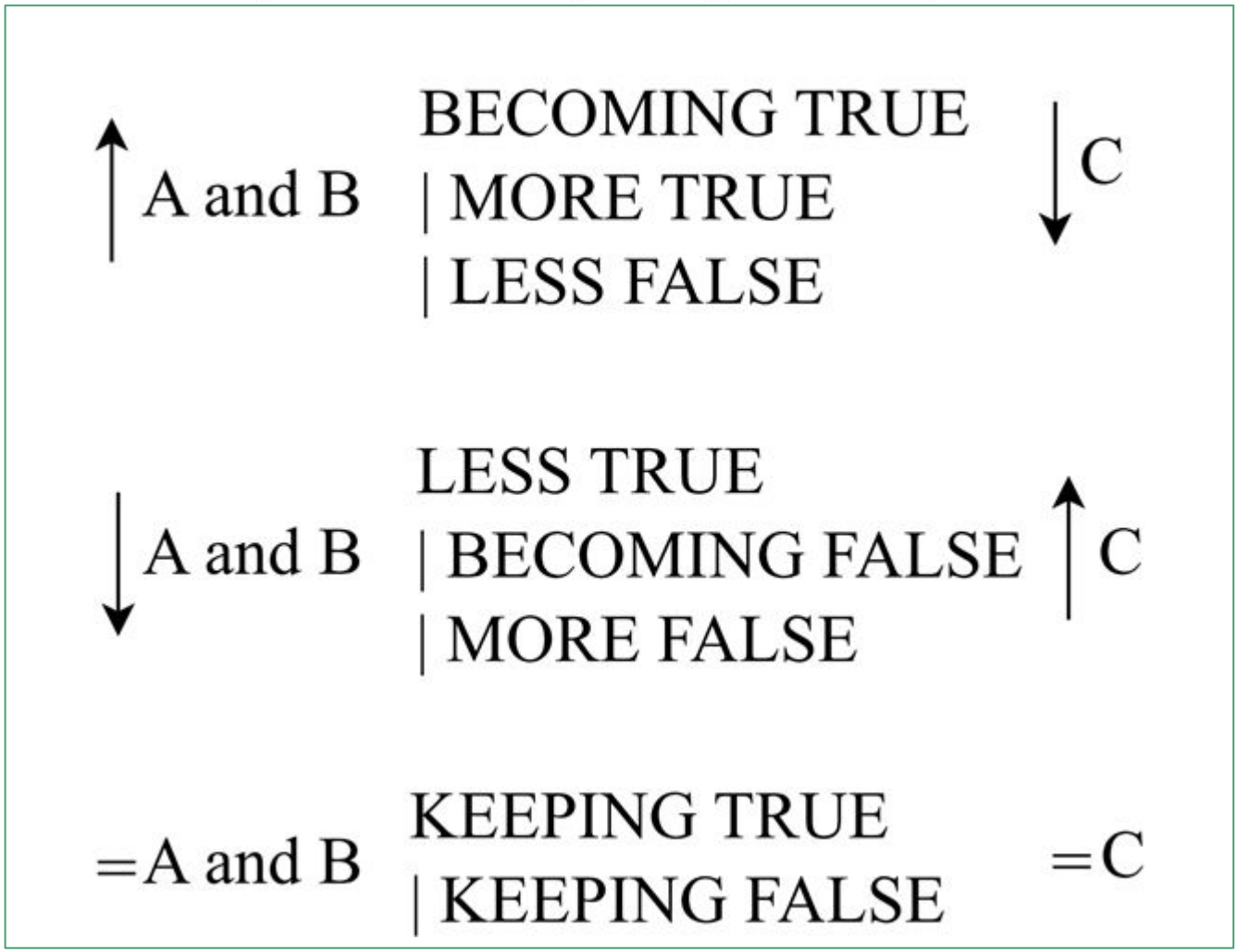
Constraint = CHECKON | NOTNULL
| UNIQUE | REFERENCES

Predicate expression = Predicate
| Predicate ∨ Predicate
| Predicate ∧ Predicate
| Predicate → Predicate

Predicate = IN | BETWEEN | FUNCTION

Transition states consistency

$(A + B > 10) \rightarrow (C < 50)$



RDT Generation

NOT NULL → No RDT

UNIQUE → Simple RDT

REFERENCES → INSERT_PRIMARY_KEY | REMOVE_PRIMARY_KEY | INSERT_FOREIGN_KEY | REMOVE_FOREIGN_KEY → RDT

CHECK ON → BECOMING TRUE | LESS TRUE | MORE TRUE | KEEPING TRUE | BECOMING FALSE | LESS FALSE | MORE FALSE | KEEPING FALSE → RDT

RDT Consistency

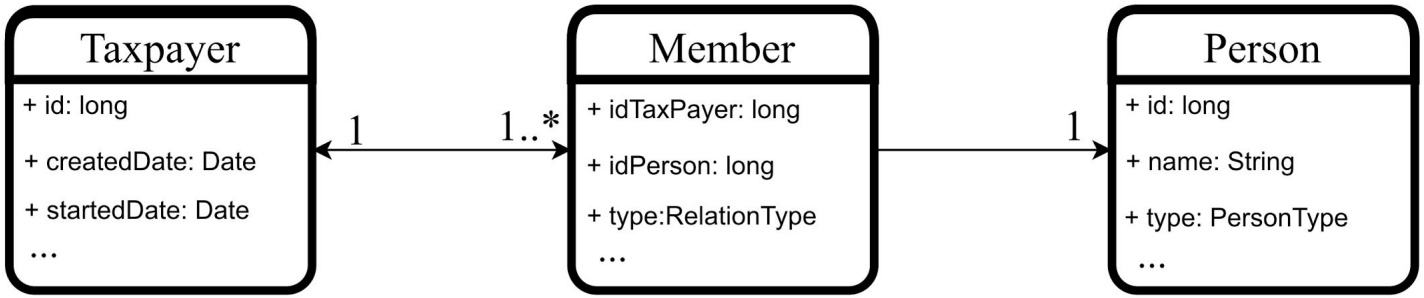
	P	$P \rightarrow P$	$P \wedge P$	$P \vee P$
Becoming true	Rejected	Both	Rejected	Eventual
More true	Eventual	Eventual	Eventual	Eventual
Less true	Strong	Both	Strong	Strong
Keeping true	Eventual	Eventual	Eventual	Eventual
Becoming false	Rejected	Both	Rejected	Strong
More false	Rejected	Eventual	Rejected	Eventual
Less false	Rejected	Both	Rejected	Eventual
Keeping false	Rejected	Eventual	Rejected	Eventual

QUELEA

- Language
- FOL
- RDT
- Haskell
- Cassandra layer

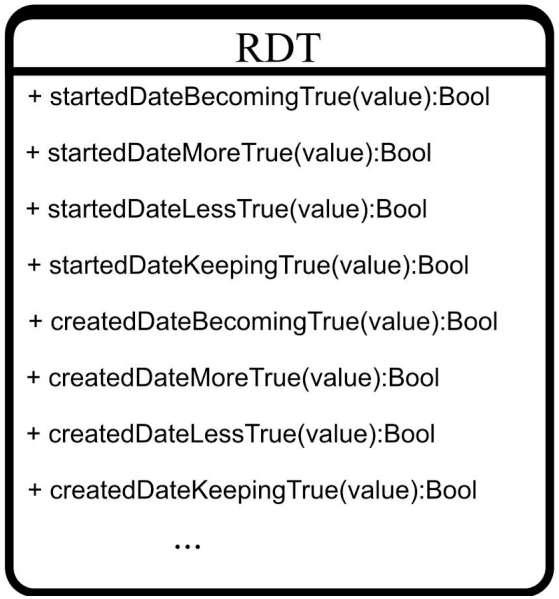
Authors: KC Sivaramakrishnan, Gowtham Kaki, and Suresh Jagannathan

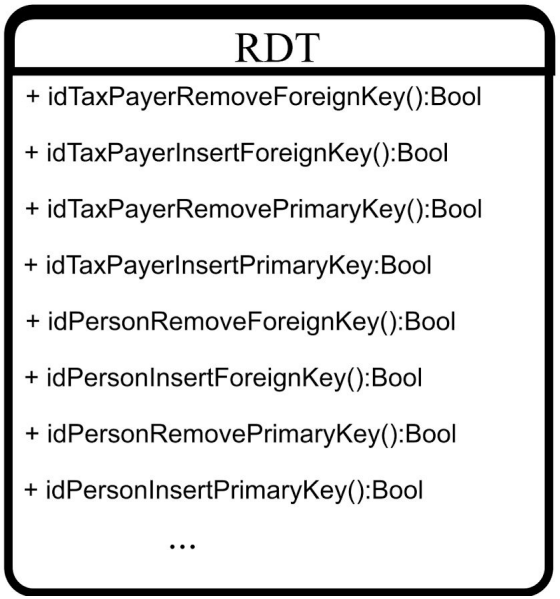
Example



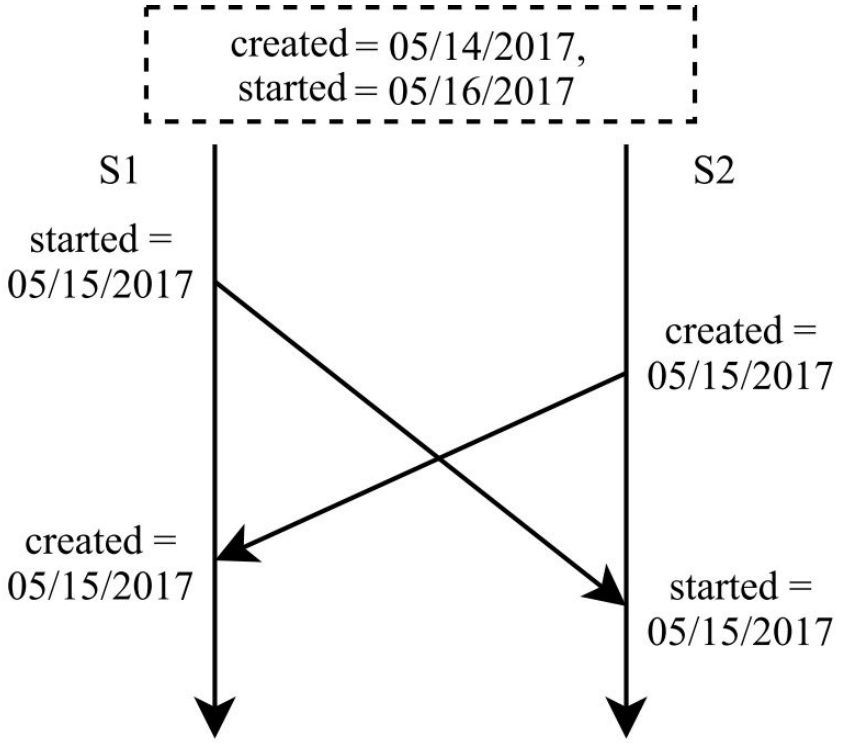
Constraint: Started date > Created date

Member references: Member model contains 1 taxpayer and 1 person.

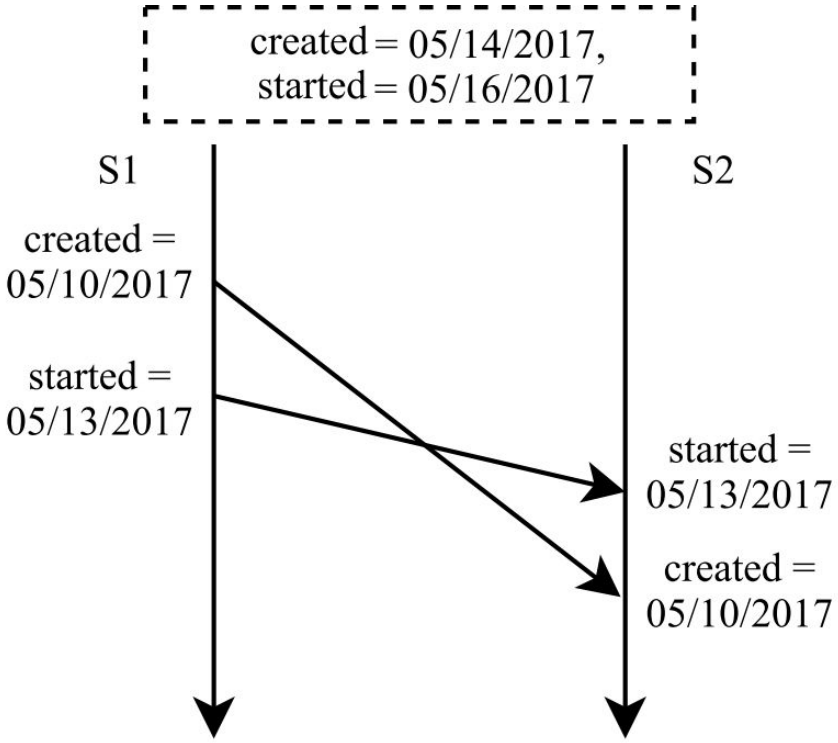




Anomalies



Less true updates conflict.



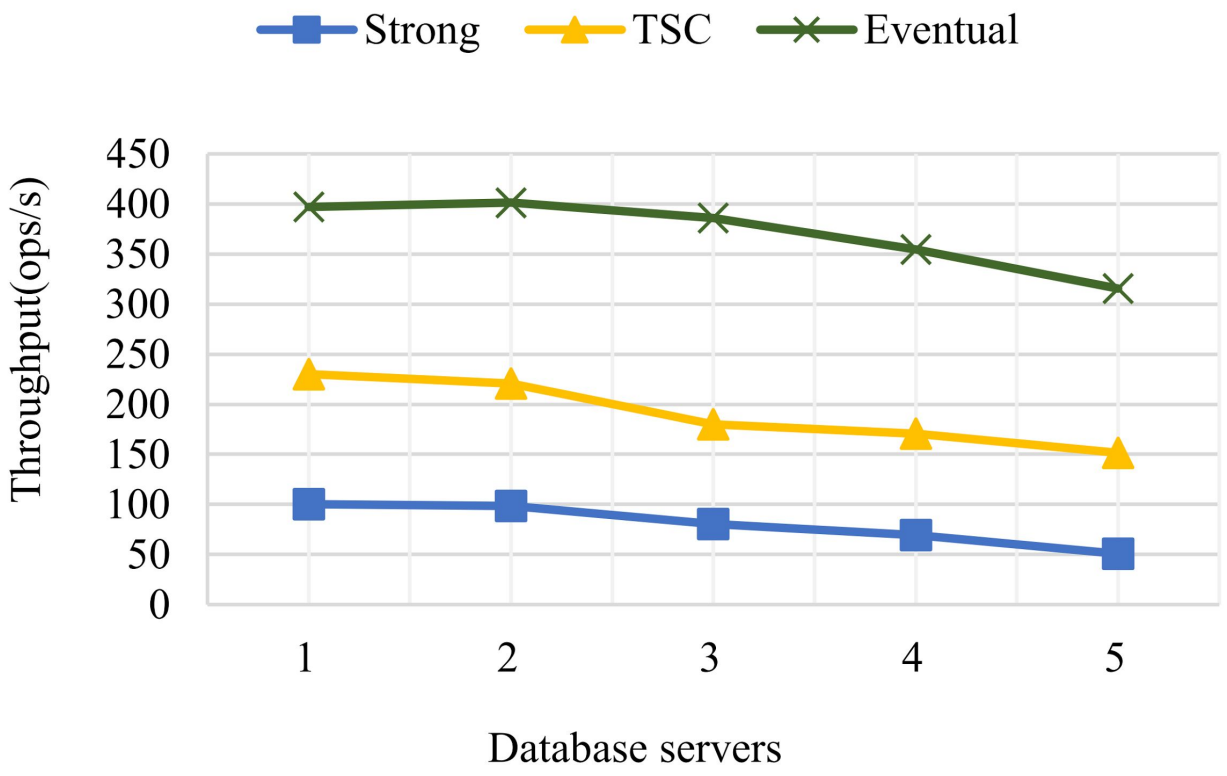
Less true update after more true update.

Example using QUELEA

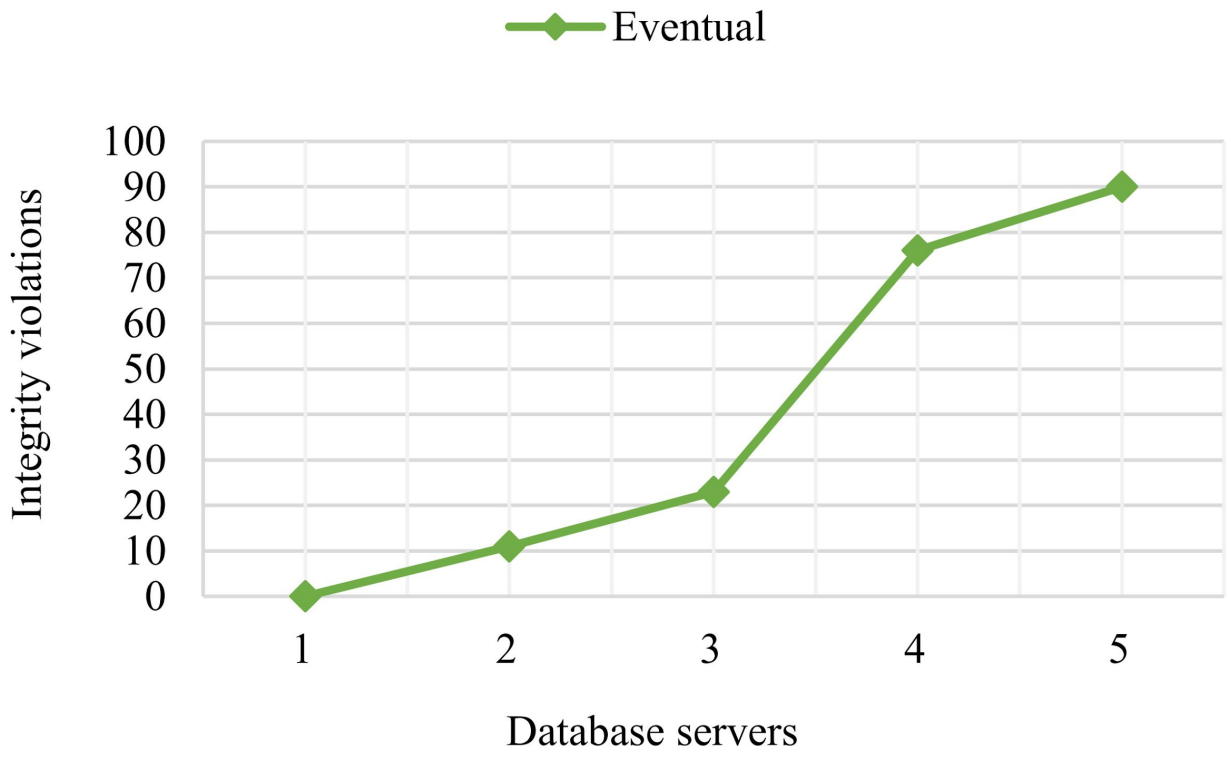
```
startedDateLessTrue :: [ConstraintRDT] -> (UTCTime, UTCTime) -> Resab Bool
startedDateLessTrue ctxt (value, t) =
    if (isValid value)
    then (True, Just $ StartedDateLessTrue_ value t)
    else (False, Nothing)

startedDateLessTrueContract :: Contract Operation
startedDateLessTrueContract x = forall_ $ \a ->
                                    liftProp $ (vis a x ∨ vis x a ∨ appRel SameEff x a)

select :: Contract Operation
select x = forallQ_ [startedDateMoreTrue, startedDateKeepingTrue] $ \a ->
           forallQ_ [createdDateMoreTrue] $ \adep ->
           forallQ_ [createdDateMoreTrue, createdDateKeepingTrue] $ \b ->
           forallQ_ [startedDateMoreTrue] $ \bdep ->
              liftProp $ ((vis adep a ∧ vis a x) ⇒ (vis adep x)) ∧
              ((vis bdep b ∧ vis b x) ⇒ (vis bdep x))
```

Evaluation - Inequation

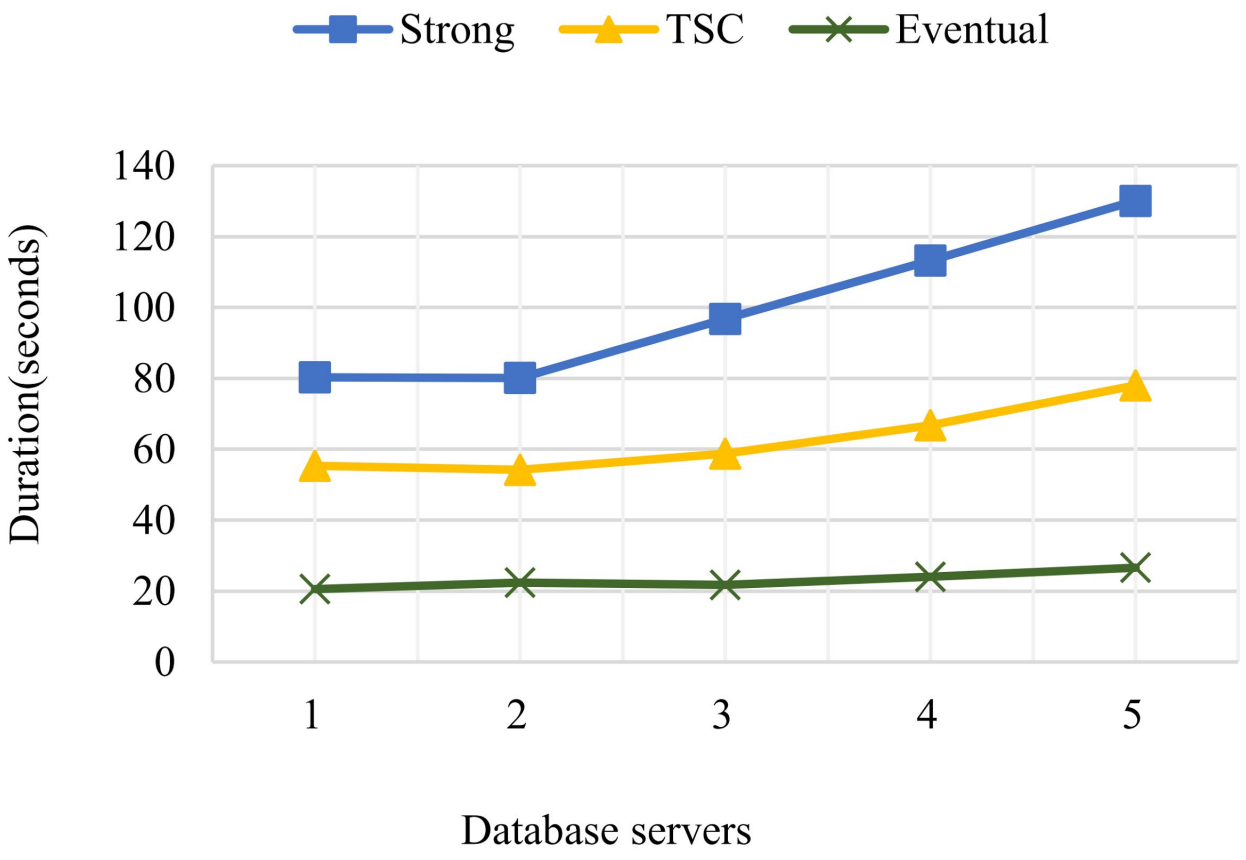
Evaluation - Inequation



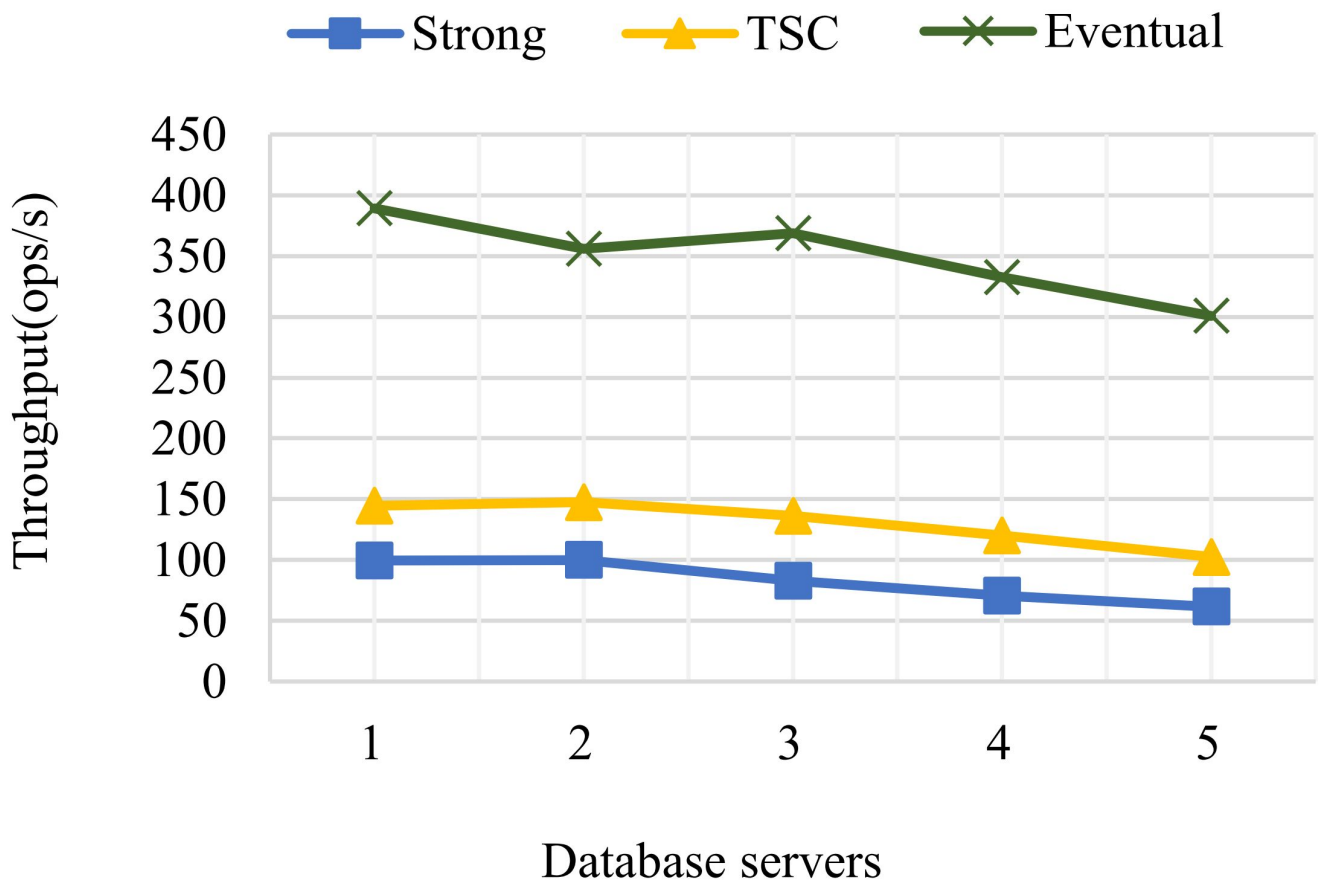
Evaluation - Inequation



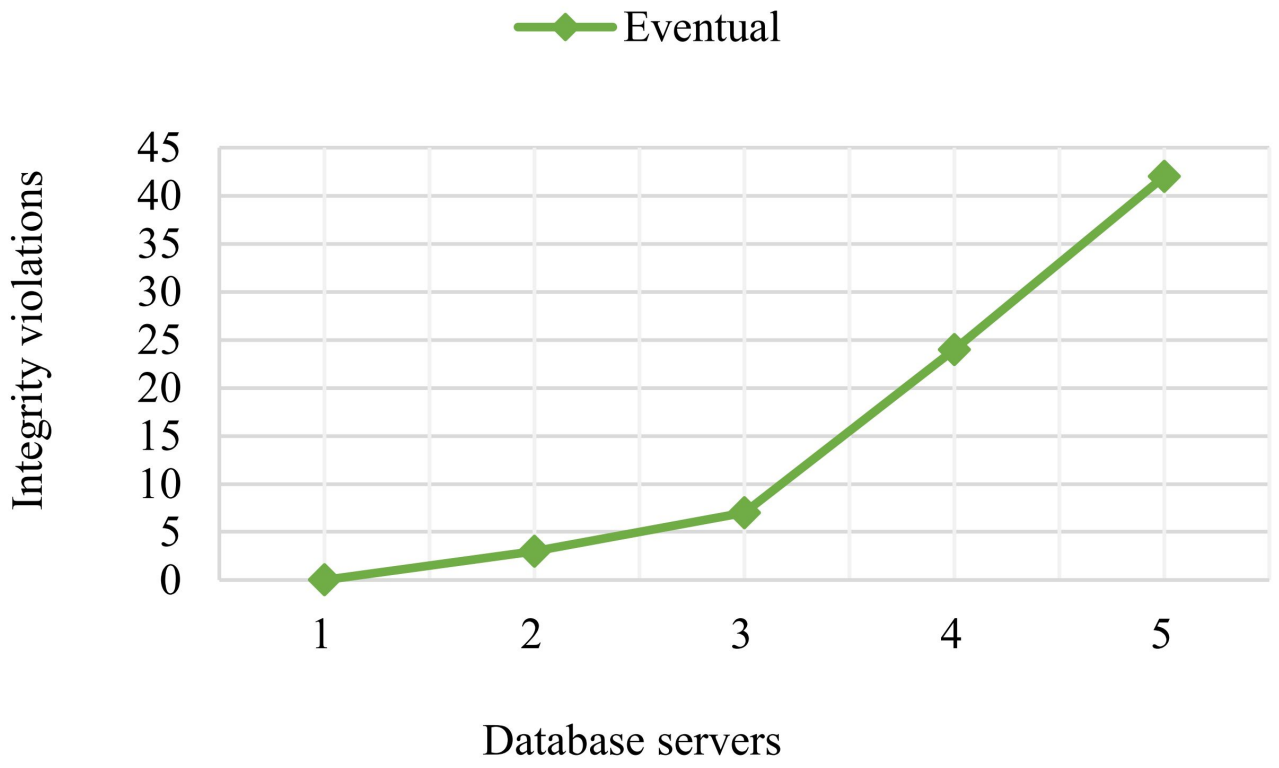
Evaluation - Foreign key



Evaluation - Foreign key



Evaluation - Foreign key



Conclusion

- RDT for expressions
- Semantic and syntax
- Variable share in RDT
- Cache
- Framework

Questions