TclRAL:
A Relational Algebra for Tcl

Andrew Mangogna

13th Annual Tcl Conference
October 11-13, 2006

Goals

- To provide relation values as native Tcl objects.
- To provide a rigorous and complete set of relational operators over the relation values.
- To provide variables to hold relation values and a useful set of integrity constraints on the values those variables may hold.
- To serve as a framework for exploring the use of relational concepts in Tcl programming.

Overview

- Structure
  - New Data Types
- Operations
  - New Commands
- Integrity
  - Relation Variable Constraints

I will, in fact, claim that the difference between a bad programmer and a good one is whether he considers his code or his data structures more important. Bad programmers worry about the code. Good programmers worry about data structures and their relationships.

-- Linus Torvalds

The Tuple Type

- Tuple is the base aggregate type in the extension
- Tuples have a heading
  - Attribute Name
  - Attribute Type
- ```
  set t {Tuple {Name string Age int}\
       {Name Fred Age 35}
  ```
  ```

The Relation Type

- Relation values consist of heading and body
  - Heading is like Tuple heading
  - Body is a set of Tuples
- Relation values are subject to identity constraints.
- ```
  set r {
      Relation
      {Name string Breed string Weight double}
      Name
      {{Name Rover Breed retriever Weight 15.0}
       {Name Buffy Breed poodle Weight 5.3}}
  }
  ```
  ```

Relation Variables

- TclRAL defines a separate variable space to hold significant relation variables
- Relvar variable space is hierarchical using the same conventions as ordinary Tcl variables
- Relvars are also mirrored as standard Tcl variables.
- Relvars can have referential integrity constraints applied to their values.

Relation Operations

- 38 Relation operators as options to the `::ral::relation` command
  - Set operations
    - union, intersection, difference, comparison
  - Selection operations
    - restrict, project, eliminate
  - Join operations
    - join, times, divide, semijoin, semiminus
  - Computational operations
    - summarize, extend, rank
  - Linkage to other Tcl data types
    - ordinary variables, array, dict, matrix

Rank Example

```
set OWNER {Relation {
    OwnerName string Age int
    } OwnerName {
      {OwnerName Alice Age 30}
      {OwnerName Sue Age 24}
      {OwnerName Mike Age 50}
    }
}
relformat [relation rank $OWNER Age AgeRank]
+=========+---+-------+
|OwnerName|Age|AgeRank|
|string   |int|int    |
+=========+---+-------+
|Alice    |30 |2      |
|Sue      |24 |1      |
|Mike     |50 |3      |
+=========+---+-------+
```

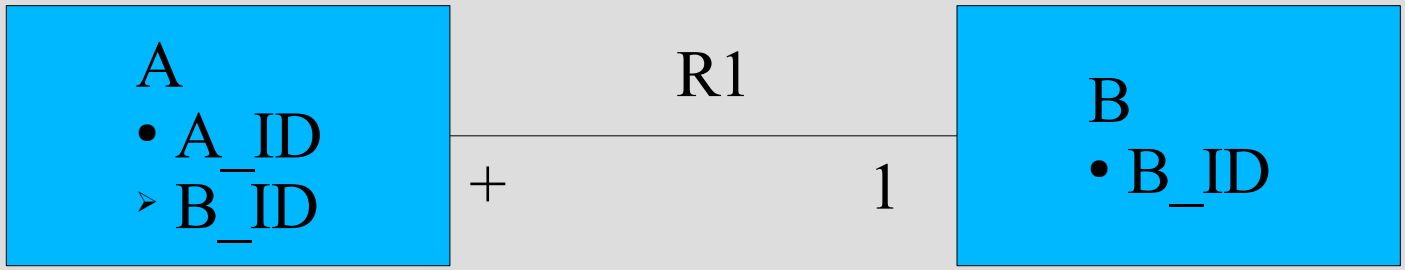
Constraints

- TclRAL enforces declarative referential integrity constraints
- Constraints are of three varieties
  - Association Constraints
  - Partition Constraints
  - Correlation Constraints

Association Constraints

- Association Constraints define traditional referential constraints.
- Attributes in one relation refer to attributes in another relation.
- References are to identifiers.
- Multiplicity and conditionality can be specified.

Associations

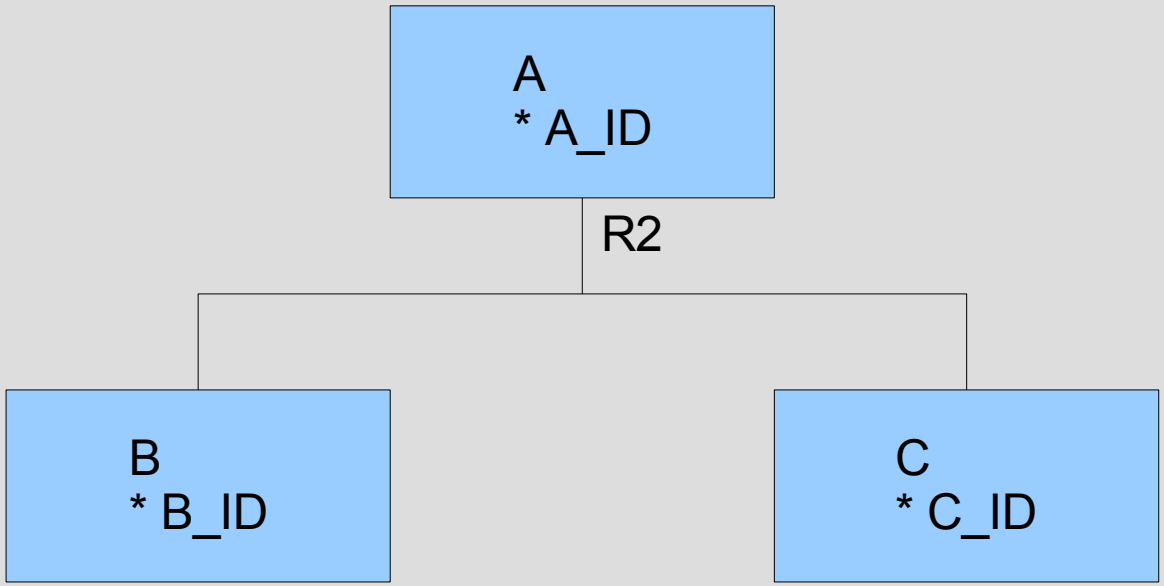


relvar association R1 A B_ID + B B_ID 1

Partition Constraints

- Partition Constraints define a set of relations that are complete and disjoint sub-sets of some super-set relation.
- Every tuple in every sub-type refers to a tuple in the super-set
- Every tuple in the super-set is referenced by exactly one tuple in some sub-set.

Partitions

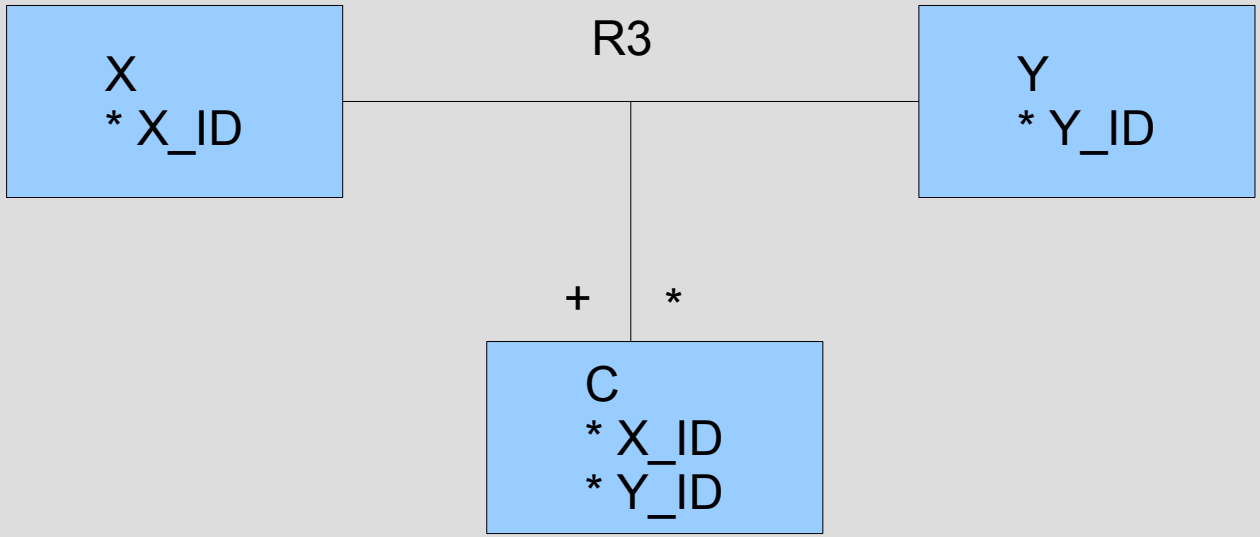


relvar partition R2 A A_ID B B_ID C C_ID

Correlation Constraints

- Correlation constraints define an integrity constraint between two relvars that is mediated by a third relvar.
- The correlating relvar references the two other relvars in the constraint.
- Often arise in a many-to-many situation.

Correlations




```
relvar correlation C X_ID + X X_ID Y_ID * Y_ID
```

What TclRAL is not!

- Not a Database Management System
- Not SQL based
- No NULL's or three valued logic
- No transparent persistence

What's Next

- Relvar tracing
- Cascading update and delete
- Default values and system assigned identifiers
- Procedural constraints
- Transparent persistence
- Continuing improvements in the code base

Summary

- TclRAL extends Tcl to naturally integrate relational ideas
  - No new syntax
  - Native Tcl object values
- TclRAL removes the conceptual mismatch between relational design approaches and our favorite programming language