Annotated Relational Algebra System

Akshat Bisht
40053762

System Overview

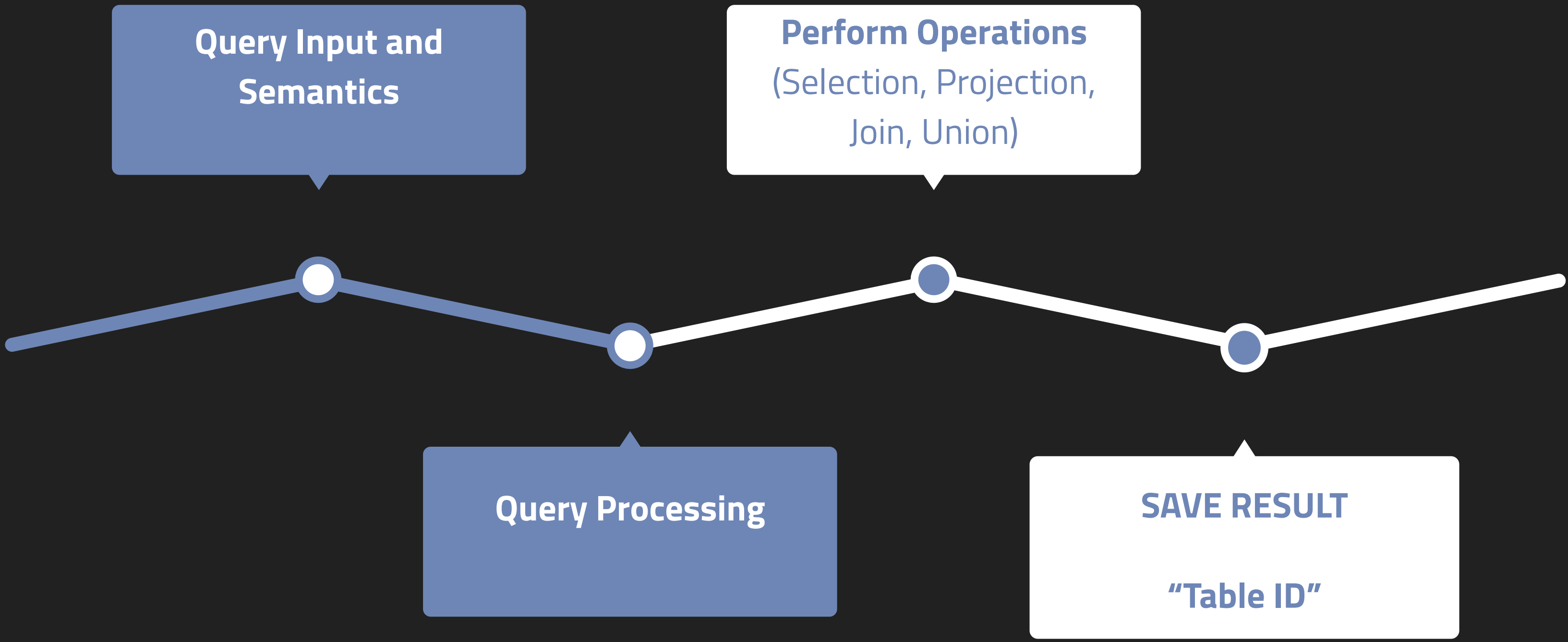
The system implements the work of Green et al[PODS 2007] on Provenance Semiring.

The Semantics implemented include : Standard, Bag, Polynomial, Probability and Certainty.

The Relational Algebra Operations implemented are : Selection, Projection, Natural Join and Union.

The system is coded with the Python and MySQL.

System Pipeline



Operation Syntax

1. **Select: S** [Column 1, column 2, ...] (Table name) |Condition 1^ Condition 2 & Condition 3|
2. **Project:** #[Column 1, column 2, ...] (Table name)
3. **Union: U** [Table name1] (Table name 2)
4. **Join: J** [Table name1] (Table name 2)

Query processing

Prefix Annotations

{} → delimiters for separation of operations.

Example query

Q(R)= $\pi_{AC}(\pi_{AB}(R) \bowtie \pi_{BC}(R) \cup \pi_{AC}(R) \bowtie \pi_{BC}(R))$

will be entered as:

Q(R)=
{{ #[A,C]({U[{ J [{ #[A,B](R) }] ({ #[B,C](R) }) }] ({ J [{ #[A,C](R) }] ({#[B,C](R) }) })}) }

Query Execution Order

- {#[A,C]({U[{J[{#[A,B](R)}]({#[B,C](R)})}]({J[{#[A,C](R)}]({#[B,C](R)})})})}
- {#[A,B](R)} → **Projection (Table1)**
- {#[A,C]({U[{J[*Table1*]({#[B,C](R)})}]({J[{#[A,C](R)}]({#[B,C](R)})})})}
- {#[B,C](R)} → **Projection (Table2)**
- {#[A,C]({U[{J[Table1](Table2)}]({J[{#[A,C](R)}]({#[B,C](R)})})})}
- {J[Table1](Table2)} → **JOIN (Table3)**
- {#[A,C]({U[Table3]({J[{#[A,C](R)}]({#[B,C](R)})})})}
- {#[A,C](R)} → **Projection (Table4)**
- {#[A,C]({U[Table3]({J[Table4]({#[B,C](R)})})})}
- {#[B,C](R)} → **Projection (Table5)**
- {#[A,C]({U[Table3]({J[Table4](Table5)})})}
- {J[Table4](Table5)} → **Join (Table6)**
- {#[A,C]({U[Table3](Table6)})}
- {U[Table3](Table6)} → **Union (Table7)**
- {#[A,C](Table7)}
- {#[A,C](Table7)} → **Projection (Table8)**
- Table8 → **Final Query Table**

Perform Operations

- There are 4 operations (Selection, Projection, Join, Union) that are implemented.
- Each operation executes different algorithms based on the semantics that are selected.
- The Combinations are:

Standard	Bag	Polynomial	Probability	Certainty
Std : Projection	Bag: Projection	Pol: Projection	Pol: Projection	Cer: Projection
Std: Selection	Bag: Selection	Pol: Selection	Pol: Selection	Cer: Selection
Std: Join	Bag: Join	Pol: Join	Pol: Join	Cer: Join
Std: Union	Bag:Union	Pol: Union	Pol: Union	Cer: Union

Projection Operation

The Projection operation is used to Project a set of mentioned Columns and their tuple values.

The Projection Operation removes all the duplicates and update the Semantics values as required.

Let us take an example database and see how the projection operation changes all the Semantics values.

Initial Database and the Semantics

category_id	StandardSem	BagSem	PolynomialSem	ProbabilitySem	CertaintySem
6	1	6	B1	0.07	0.64
6	1	9	B2	0.77	0.14
6	1	7	B3	0.65	0.79
6	1	9	B4	0.92	0.52
6	1	3	B5	0.66	0.24
6	1	8	B6	0.54	0.63
6	1	1	B7	0.73	0.43
6	1	3	B8	0.01	0.28
5	1	9	B9	0.87	0.09
4	1	9	B10	0.31	0.6

Applying Projection: $\pi_{category\_id}$

category_id	StandardSem	BagSem	PolynomialSem	ProbabilitySem	CertaintySem
4	1	9	(B10)	0.31	0.6
5	1	9	(B9)	0.87	0.09
6	1	46	(B1+B2+B3+B4+ B5+B6+B7+B8)	0.99	0.79

The use of Projection function is used to the different operations (Select and Union) to make sure that we get rid of all the duplicate values and update the semantics appropriately .

Implementation Of Projection Function

Standard Semantics: All the tuples which have a semantic value of 0 are eliminated and the remaining tuples that have same values are group together with semantic value 1.

Bag Semantics: An addition is performed between all the multiplicities of the tuples with same values.

Polynomial Semantics: An addition is performed among all the semantic variables of the tuples with the same values.

Probability Semantics: All the tuples with common values have their probabilities combined using the formula 1- (1-P(A)) * (1-P(B)).

Certainty Semantics: All the tuples with common values are group together and the maximum certainty value among them are selected.

Join Operation

Standard Semantics: The cross product of 2 tables are taken and all their Semantic values are multiplied to with each other. All the tuples that do not satisfy the join conditions have their Semantics values updated to 0. The tuples with Semantics values as 0 are removed from table and the final table is saved.

Bag, Probability and Certainty have their joins performed as natural join and have all their semantics multiplied with their respective tuples they are joined to.

Inference from Project

We can infer from after completing this project that there is a common algebraic structure to all the different semantics for positive algebra operations.

Using definitions 3.1 and 3.2 from Green et al [PODS 2007] we can define all the positive algebra operations using a similar algebraic structure.

1. (B,⋀,⋁,false,true) → Set Semantics
2. (N,+,.,1,0) → Bag Semantics
3. (PosBool(B),⋀,⋁,false,true) → C-tables
4. (P(Ω),U,∩,Ø) → Event tables(Probability Semantics)
5. (P(X),U,U,Ø,Ø) → Why provenance

Polynomials for Provenance is useful because it helps us find out the how-provenance which overcomes the limitations of the why-provenance.

A possible extension of this project is the incorporation of Difference and Duplication Elimination Operator in positive relational algebra to get a full relational algebra on K-relations.

Extension of Project

To include the difference(RA^+_k (\)) and duplication(RA^+_k (δ)) removal operations in our positive relational algebra operations we need to define a query language for both of them separately and together (RA^+_k (\,δ)) as well.

For the difference operation:we introduce the monus operator $\ominus$ which allows us to extend RA^+_k (\).

When the monus operator is introduced, we make an assumption that K is naturally ordered and for each pair of elements x,y ∈ K, the set {z ∈ K| $x \leqslant y \oplus z$} has a smallest element.

A semiring K with monus is called m-semiring.

Monus behaves like set difference K_B (Set Semantics),K_{prob}; truncated minus sign K_N m-n if m>n otherwise 0; For C tables the monus operation is only valid under closed world semantics.

For Ctables the monus operator can be used as follows: $ø1 \ominus ø2 = ø1 \land \neg ø2$

Extension of Project

The duplicate elimination operation: For K_N it is common to include the duplicate elimination operator in the query language.

But to include it in general to K-relations, we restrict our attention to $K=(K,\oplus,\otimes,0,1)$ that are finitely generated.

Finitely generated mean, every element in K can be written as a finite sequence of sums and products of a finite set of elements k1,k2...km in K called generators of K (Gen(K)).