Report date 19/10/2024

Paper report:

The missing fields are due to the nature of the paper that doesn't implement nor evaluate any tools or instruments.

**Overview:**

The paper explains the concept of lineage in databases. I've fully read the paper document provided during the course on Data Preparation, is consist of the introduction only. I've also taken a closer look at the provenance semiring definition as suggested by Stefan.

The paper introduces and mention the need for a provenance system.

Three provenances are analyzed:

* Why provenance:  
  It's often referred to as the lineage. Why prov task is to keep track of which tuples took part in generating the output (a tuple itself).A set of tuples that gave origin to the output tuple is referred to as the witness, witnesses are a subset of the input database records that is sufficient to ensure that a given output tuple appears in the result of a query. For each tuple there may be several witnesses, we may have multiple sets of tuples that generate the same output tuple, this may be for instance due to SELECTION operator usage that transforms several distinct tuples into multiple equal tuples. Since there may be multiple witnesses the why provenance focuses on the witness basis which consist of a compact representation of all the witnesses. Provenance is sensitive to distinct but equivalent queries: queries achieving the same semantic but through different syntax may return different provenance for a tuple. For this reason the minimal witness basis is used, which is invariant under equivalent queries.
* How provenance:  
  This kind of provenance not only describes which tuples are involved in the output creation but also in which manner. It is based on the Provenance Semiring set of functions that includes set K, identity over + which is 0, identity over ∙ which is 1, and + and ∙ . It follows all the commutative semiring constraints, so we can do "math" to generate how provenance.  
  K is the set of tuples, ∙ is used when joining tuples and + is used when both the tuples in the operation can be there (such as when you do unions or when you make two tuples the same by selection fields in which they share the values). The paper also describe how each operator affects the how provenance equation. This provenance is more specific than why provenance and all the components of how provenance are also all the components of the why provenance, vice versa is not true tho.
* Where provenance:  
  Where provenance tracks where a specific data comes from in terms of which cell of which tuple. It describes where a piece of data is copied from and in a relational database scenario it is a cell in a tuple. In this case equivalent queries may return different results. The where provenance of a cell consists in locations that can be found in tuples of the why provenance of the tuple that contains the cell. Not all the tuples in the why provenance will be relevant to the where provenance. Where provenance is also used to forward annotations on data when forwarding and using informations from an annotated source dataset.

The paper also explains eager and lazy provenance computations and gives an introduction to the notation used in the rest of the paper.

**Authors:**

J. Cheney, L. Chiticariu and Wang-Chiew Tan

**Year:**

2009