PAPER STRUCTURE

What do I want to sell in this paper?

The aim of this paper is to “sell” an approach for data integration – based on reusability functions – that is more efficient in terms of performance and reduces the integration cost while delivering high quality results.

Structure:

1. **Introduction**
   Introduce the context, current limitations on integration approaches, how do we plan to address these limitations and objectives.
2. **Illustrative example**
   Present an example in order to understand the problem and use the same example to the experiments
3. **Approach**
   Present the approach. Describe our formalism, query taxonomy, algorithms and reusability functions.
4. **Experiments**
   Show experiments and results comparing the reusability scenario with the traditional one that process the entire integration.
5. **Related work**
   Introduce the works related to query rewriting (service composition), their limitations, and where our approach differ compared to them.
6. **Conclusion**
   Conclude the paper and present future works.

Experiments: considering that we have an incoming query Q, create the following databases for our history:

<u>DB1:</u> 100 queries that can be completely reused when Q is equivalent or equivalent and less restrict.

<u>DB2:</u> 100 queries that can be partially reused when Q is equivalent and including at least one requirement more restrict.

<u>DB3:</u> 100 queries that can be completely reused, but Q should be applied to a projection algorithm when Q is contained in the queries of the history, and the requirements of Q is equivalent or less restrict.

<u>DB4:</u> 100 queries that can be partially reused, but Q should be applied to a projection algorithm when Q is contained in the queries of the history, and at least one requirement of Q is more restrict.

<u>DB5:</u> 100 queries that can be completely reused, but Q should be applied to a small rewriting algorithm to complete the query when Q contains the queries of the history, and the requirements of Q is equivalent or less restrict.

DB6: 100 queries that can be partially reused, but Q should be applied to a small rewriting algorithm to complete the query when Q contains the queries of the history, and at least one requirement of Q is more restrict.

For all the examples, vary also the number of services in the database from 50 to 500 services, and the number of abstract service in the query from 1 to 5.