

Figure 5: Screenshot of query interpretations

- the keyword match objects can be connected via various relationships and form various interpretations.

As the user often has some particular search intention in mind, ExpressQ actively involves the user in the query evaluation process. In particular, if a keyword is associated with more than one tag, the user is offered the opportunity to choose the tag(s); if the Query Interpreter constructs more than one query pattern, the user is again allowed to select his/her intended query pattern and retrieve the corresponding answers. This interactive approach has the advantage of systematically leading the user to obtain answers that satisfy his/her search intention. This approach also gives the user insight into how the query is interpreted by the system and the results that can be expected.

Recall the query {Project Employee Green Brown}. Suppose the user issues this query to find the project that involves both the employees Green and Brown. Since the keyword Brown can refer to an employee named Brown or a department at Brown street, the ExpressQ shows the possible matches for the user to choose from (see Figure 4). If the user selects Brown as referring to an employee name, then ExpressQ will show how the employees Brown and Green can relate to a project in terms of query patterns (see Figure 5). Note that these patterns are ordered by their ranking scores.

Another feature of ExpressQ is it depicts the query interpretations and answers in human natural language to facilitate users' understanding. For instance, the tag $T_{14} = (\text{Employee, Name, Brown})$ in Table 1 is described as "Brown refers to an employee with name matching Brown". The query pattern P_1 in Figure 5 is represented as a tree annotated with the semantics of objects and relationships. The root of the tree denotes the output object while the leaves denote the restrictions on the output object. The meaning of this pattern is to "Find the projects that involve the employee with name matching Green and involve the employee with name matching Brown". Thereby, the user can easily identify the intended query pattern by the tree structure, and verify its meaning by the description. After the user selects a query pattern, ExpressQ retrieves the answers and represents them according to the corresponding search intention. Figure 6 shows the screenshot of the interface which displays the answers w.r.t. the query pattern P_1 in Figure 5.

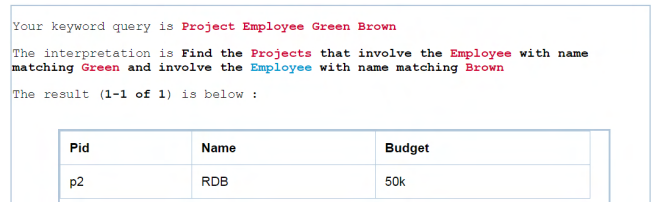


Figure 6: Screenshot of answers retrieved

4. DEMONSTRATION

In our demonstration, we will present a web-based browsing interface of ExpressQ, which communicates with the main Java based server. The system is available at <http://expressq.comp.nus.edu.sg>. We intend to show the use of ExpressQ against a number of real application scenarios such as the IMDB database (www.imdb.com), and the ACM Digital Library (dl.acm.org).

The demonstration will consist of two parts. First, we will run a number of sample keyword queries against these sources. We will demonstrate how ExpressQ exploits the semantics of objects/relationships in the database and utilize the keywords that match meta-data in the query to infer the search intention for these queries. Next, the user will be free to run their own queries. We will demonstrate how ExpressQ interactively leads the user to retrieve the intended answers effectively. During the query processing, the system will present different interpretations of the queries. The user will be able to choose the ones he/she is interested in, and thus obtain the answers that satisfy the search intention.

Through this demonstration, we will highlight three key-points to the audience. First, the interpretation of the user's search intention is critical to keyword search over relational database. This requires the keyword search system to be knowledgeable about the semantics of objects and relationships in the database. Second, keywords that match the meta-data are helpful to infer the search intention of the user since they provide the context of subsequent keywords in the query. Third, the presentations of the query interpretations and query answers are important to facilitate user understanding and subsequent interaction with the system.

5. REFERENCES

- B. Aditya, G. Bhalotia, S. Chakrabarti, A. Hulgeri, C. Nakhe, P. Parag, and S. Sudarshan. BANKS: Browsing and keyword searching in relational databases. In *VLDB*, 2002.
- M. Kargar, A. An, N. Cercone, P. Godfrey, J. Szlichta, and X. Yu. MeanKS: Meaningful keyword search in relational databases with complex schema. In *SIGMOD*, 2014.
- F. Li and H. V. Jagadish. Usability, databases, and HCI. *IEEE Data Eng. Bull.*, 35(3):37–45, 2012.
- F. Li and H. V. Jagadish. NaLIR: An interactive natural language interface for querying relational databases. In *SIGMOD*, 2014.
- Y. Luo, W. Wang, and X. Lin. SPARK: A keyword search engine on relational databases. In *ICDE*, 2008.
- Z. Zeng, Z. Bao, T. N. Le, M. L. Lee, and W. T. Ling. ExpressQ: Identifying keyword context and search target in relational keyword queries. In *CIKM*, 2014.
- Z. Zeng, Z. Bao, M. L. Lee, and T. W. Ling. A semantic approach to keyword search over relational databases. In *ER*, 2013.