Close and Loose Associations in Keyword Search from Structural Data

Johanna Vainio
University of Tampere
Kanslerinrinne 1
FI-33014 University of Tampere, Finland
s.johanna.vainio@staff.uta.fi

Marko Junkkari
University of Tampere
Kanslerinrinne 1
FI-33014 University of Tampere, Finland
marko.junkkari@uta.fi

Jaana Kekäläinen
University of Tampere
Kanslerinrinne 1
FI-33014 University of Tampere, Finland
jaana.kekalainen@uta.fi

ABSTRACT

Keyword search over structural data enables users to seek information from databases without knowing the structure of data or mastering actual query languages like SQL. In a keyword query, data items or text attributes are matched to the keywords and the result of a query is typically a set of graphs consisting of connected tuples. The result should be ranked which means that the text attributes and connections must be scored and combined. Typically, the length of a connection is the main criterion in ranking the connections, i.e. shorter connections are scored higher than longer ones. The length of a connection is usually based on the foreign key references but their direction has received less attention. At the conceptual level, cardinality constrains correspond to foreign key references or their combination. In the present paper, we investigate the effect of the combinations of cardinality constrains on the result of a keyword search. We find that the combination of cardinality constraints indicates how close the association between keywords is. We also show that the Minimal Total Joining Network of Tuples (MTJNT) principle loses semantic connections or fragments the results of a keyword search from relational databases.

CCS Concepts

• **Information systems** • **Information systems ~ Relational database model** • **Information systems ~ Entity relationship models** • *Information systems ~ Database query processing*

Keywords

Keyword queries over structured data; associations in relational databases; cardinality constraints; ER model.

1. INTRODUCTION

Keyword search enables end-users to search data from relational databases without knowledge of the syntax of a query language or the structure of the data. However, keyword search involves ambiguity and raises new challenges. Traditionally, ambiguity is associated with the nature of keyword search, i.e. matching search keys to document contents is more or less fuzzy. In the context of structured data, the nature of relationships among entities and text attributes may also affect different kinds of semantic interpretations. Namely, entities may be associated with each other via different kinds of relationships. In the present paper, we first study which kinds of settings the conceptual associations of a semantic data model serve for the connections of entities. Then, we analyze their roles in keyword search in relational databases.

In information retrieval, keyword search finds documents that contain all or some of the keywords and ranks the documents according to the statistical properties of their words. There is no need to solve how documents containing the keywords are connected. In context of relational databases, keyword search can be used to find the top ranked connections of tuples that contain all or some of the keywords. To produce ranking, tuples that contain a keyword are retrieved, and connections between these tuples are produced. A connection of tuples may, for example, be a minimal total joining network of tuples (MTJNT) [4] or Steiner tree [1] [2]. There are also different approaches how to rank the produced connections. Ranking can be based, simply, on the number of joins of a connection or attribute, tuple or edge level scores or combinations of them. [6, 7, 8] Different connections may contain different amount of information and different interpretations, even between the same keyword tuples. Therefore, the shortest connection is not always the best; a longer path may be more appropriate [5, 6]. We draw this conclusion by analyzing the closeness and looseness of conceptual association. This dimension is based on the cardinality constrains that appear in the connections of entities.

2. CONCEPTUAL ASSOCIATIONS

Semantic data models are conceptual methods for representing concepts and the relationships among them. The ER model is the most common semantic data model and its principal primitives are the entity type, attribute, and relationship (type) between the entity types. A relationship involves a cardinality constrain that may be 1:1, 1:N, N:1 or N:M. A constraint determines how many instances are participating in the relationship at the extensional (instance) level. Let the ER schema of Figure 1 illustrate this. The schema is a fragment from [3] but no attributes are represented. The example contains four entity types (DEPARTMENT, EMPLOYEE, DEPENDENT and PROJECT) and four relationships among them. In the example, several employees may work for a department and an employee works in one department. An employee may have several dependents and a dependent has one employee as a guardian. Furthermore, an employee may work on several projects and a project may have several employees. Finally, a department may control several projects and a project is controlled by a single department.

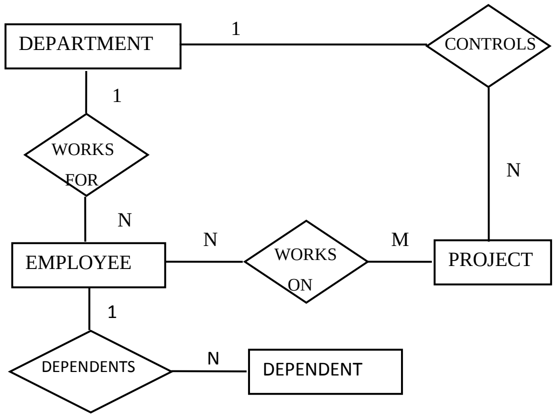


Figure 1: ER- schema

Figure 1 illustrates that an employee and a department may be in association in two ways. First, an employee works for a department and second, (s)he works on a project controlled by the department. The first alternative involves one relationship and the second two relationships, i.e. the first path is shorter but the longer one contains more information because it also determines in which project the employee is working. This is an essential issue in keyword search from structural data where the result connections should be ranked. In other words, if we want to emphasize access to more information a longer connection should be ranked before shorter connections. However, usually longer connections are lower in a result list or not in the results list at all. This is justified, because longer connections entail more ambiguity than shorter ones and may lose associations between entities. However, the level of the ambiguity a connection involves can be examined, and thus, decreased and controlled. Next we investigate how the level of the ambiguity can be determined based on the cardinality constraints of the ER-model.

An entity type involves a set of entities whereas relationships determine how the entities can be connected to each other. In the present paper, we conceptualize that a close connection between entities means that they are associated with each other unambiguously through their relationships. Table 1 contains a sample of immediate and transitive relationships between entity types. Relationships 1 and 2 represent a situation where two entity types and the corresponding entities are connected immediately. In the immediate relationships, there is no ambiguity in the semantics of the connections, i.e. the corresponding entities are closely associated to each other.

A transitive relationship contains more than one immediate relationships, i.e. the corresponding entities are connected to each other via a middle entity. Transitive relationship 3 consists of two immediate relationships both having the cardinality constrains 1:N. In other words, for one department there are several employees and for each employee there may be several dependents, but not vice versa. This means that there is a transitive 1:N relationship between the entity types department and dependent. In other words, the connection is (inverse) functional. We interpret both inverse functional connections, only 1:N relationships, and functional connections, only N:1 relationships, as functional. This is because a connection can be represented in both directions, i.e. the connection 3 in Table 1 can be represented from dependent to department (dependent ⟨N:1⟩ employee ⟨N:1⟩ department) as well. A functional relationship may also contain 1:1 relationships. Therefore, we define that if $\langle\langle X_1,Y_1\rangle,\ldots,\langle X_n,Y_n\rangle\rangle$ represents the cardinality constraints of a transitive relationships such that $\forall i \in \{1, \ldots, n\}$ holds that $X_i = 1$ or $\forall i \in \{1, \ldots, n\}$ holds that $Y_i = 1$ then the relationships is functional.

In general, the immediate relationships and transitive functional relationships determine a close connection between entities at the extensional level.

Table 1. Relationships and their cardinalities in the ER schema

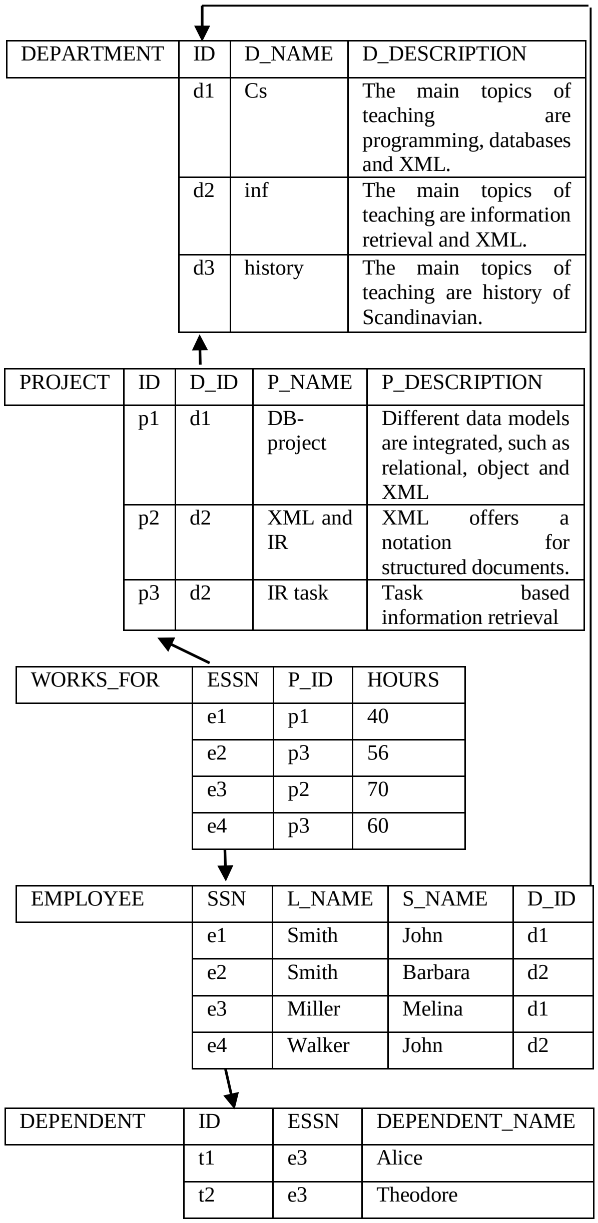
	Relationship	Cardinality
1	department – employee	department ⟨1:N⟩ employee
2	project – employee	project ⟨N:M⟩ employee
3	department – employee – dependent	department ⟨1:N⟩ employee ⟨1:N⟩ dependent
4	department – project – employee	department ⟨1:N⟩ project ⟨N:M⟩ employee
5	project – department – employee	project ⟨N:1⟩ department ⟨1:N⟩ employee
6	department – project – employee – dependent	department ⟨1:N⟩ project ⟨N:M⟩ employee ⟨1:N⟩ dependent

Transitive relationship 4 consist of 1:N and N:M relationships respectively. This means that one department can be associated with employees through several projects. In other words, employees that work on a project controlled by a department may or may not work in the department. For the reason that there are two kinds of semantic interpretations, this kind of transitive relationship may cause a loose connection at the extensional level. Transitive relationship 5 contains two immediate relationships having cardinality constraints N:1 and 1:N respectively. This is called a transitive N:M relationship because several entities of the start entity type may be connected to several entities of the end entity type via a middle entity. This kind of relationship causes a more ambiguous interpretation on the connection of entities. Namely, an employee is associated with a project although s(he) may not work on it, i.e. an employee is associated with every project a department is controlled. Therefore, a transitive N:M relationships may also cause a loose connection between entities. In general, let $\langle\langle X_1,Y_1\rangle,\ldots,\langle X_n,Y_n\rangle$, where $X_1 \neq 1$ and $Y_n \neq 1$, represent the cardinality constraints of a transitive relationships, then the relationship is N:M transitive. Connection 6 contains three immediate relationships. The first relationship possesses the 1:N constraint and the last 1:N constraint. However, this is not transitive 1:N relationship because it contains a transitive N:M relationship as a part of it. Therefore, it allows loose connections at the extensional level.

Next we demonstrate close and loose connections at the database level and their effects on keyword search in relational databases.

3. ASSOCIATIONS IN RELATIONAL DATABASES

Roughly speaking, an ER-schema is implemented in relational databases such that for each entity type a relation is implemented. For each 1:N relation a foreign key is inserted to the N-site. For each N:M relationships a middle relation is formed. This relation contains the foreign keys from both the participating entity types (relations in RDB). A foreign key constraint is typically represented as an arrow from a foreign key to the related primary key. The database schema and database instance of Figure 1 is represented in Figure 2. Attributes are now represented.



DEPARTMENT	ID	D_NAME	D_DESCRIPTION
	d1	Cs	The main topics of teaching are programming, databases and XML.
	d2	inf	The main topics of teaching are information retrieval and XML.
	d3	history	The main topics of teaching are history of Scandinavian.

PROJECT	ID	D_ID	P_NAME	P_DESCRIPTION
	p1	d1	DB-project	Different data models are integrated, such as relational, object and XML
	p2	d2	XML and IR	XML offers a notation for structured documents.
	p3	d2	IR task	Task based information retrieval

WORKS_FOR	ESSN	P_ID	HOURS
	e1	p1	40
	e2	p3	56
	e3	p2	70
	e4	p3	60

EMPLOYEE	SSN	L_NAME	S_NAME	D_ID
	e1	Smith	John	d1
	e2	Smith	Barbara	d2
	e3	Miller	Melina	d1
	e4	Walker	John	d2

DEPENDENT	ID	ESSN	DEPENDENT_NAME
	t1	e3	Alice
	t2	e3	Theodore

Figure 2. Database schema and instance

A keyword search typically focuses on attribute values. A keyword may match the whole attribute value or a word in a text attribute. Let us consider a sample keyword search

Smith XML

"Smith" matches two first employees whereas "XML" matches two projects and two departments. Connections 1 – 7 in Table 2 represents some of the connections for the keyword query "*Smith XML*" in the RDB in Figure 2.

John Smith is associated with XML through different connections. The shortest and the longest connections are between an employee and a department as shown in Table 1. John Smith is also associated with XML through the project by the connections having two steps (connections 2 and 3 in Table 2). However, WORKS_FOR is a middle relation and the length of the connection would be one if the conceptual schema were followed. In other words, in conceptual approach middle relations should not be taken into account when calculating the length of a connection.

Table 2. Connections in the RDB and lengths of the connections in the RDB and the ER

	connection	length in RDB	length in ER
1	d1$^{(XML)}$ – e1$^{(Smith)}$	1	1
2	p1$^{(XML)}$ – w_f1 – e1$^{(Smith)}$	2	1
3	p1$^{(XML)}$ – d1$^{(XML)}$ – e1$^{(Smith)}$	2	2
4	d1$^{(XML)}$ – p1$^{(XML)}$ – w_f1 – e1$^{(Smith)}$	3	2
5	d2$^{(XML)}$ – e2$^{(Smith)}$	1	1
6	p2$^{(XML)}$ – d2$^{(XML)}$ – e2$^{(Smith)}$	2	2
7	d2$^{(XML)}$ – p3 – w_f2 – e2$^{(Smith)}$	3	2
8	d1 – e3 – t1$^{(Alice)}$	2	2
9	d2 – p2 – w_f3 – e3 – t1$^{(Alice)}$	4	3

In a schema (intensional) level, connections 1 and 2 have a close association and connections 3 and 4 have a loose association between the entities. However, in an instance level, also connections 3 and 4 have a close association between the entities. The connections can be read as follows:

1) "employee e1$^{(Smith)}$ works for department d1$^{(XML)}$"
2) "employee e1$^{(Smith)}$ works on a project p1$^{(XML)}$"
3) "employee e1$^{(Smith)}$ works for department d1$^{(XML)}$, that controls project p1$^{(XML)}$"
4) "employee e1$^{(Smith)}$ works on project p1$^{(XML)}$, that is controlled by department d1$^{(XML)}$.

In this case employee e1 works on project p1 as associated in connection 3 and employee e1 works for department d1 as associated in connection 4, but this cannot generally be assumed without investigating other connections. This is illustrated next.

The closest and longest association between Barbara Smith and XML relates to the description of her department because she works in a department that matches XML (connections 5 and 7 in Table 2). It is worth noting that Barbara is also associated with project p2 in connection 6 although she does not work in it. This is because the connection contains N:1 and 1:N relationships. In other words this connection gives broader interpretation and project p2 and employee e2 (Barbara Smith) are in a loose association.

If the rank of connections 1 - 7 were based on the length of the connection in RDB, the best connections are 1 and 5 and the worst connections are 4 and 7. If the length of the ER-model were followed and the close associations were emphasized, the best connections are 1, 2 and 5 and the worst connections are 3 and 6. In the latter approach connections 4 and 7 have a better rank because they do not lose the close association (in the schema level), i.e. the employee works in the department and in the project the connection includes. Connections 8 and 9 in Tables 2 and 3 correspond to relationships 5 and 6 in Table 1. Connection 8 has a close association and connections 9 has a loose association between entities in both the schema and instance levels.

A commonly used approach to form connections is Minimal Total Joining Network of Tuples (MTJNT) [4] or Steiner tree [1] [2]. In the MTJNT approach every keyword exists in at least one tuple of the joining network. It is not possible to remove any tuple from the joining network without losing MTJNT. The MTJNT approach returns minimally connected tuples that still contain every keyword. This approach can lose some meaningful tuples that are associated to keyword queries and MTJNTs. In the previous example connections 3, 4, 6 and 7 are lost, if the MTJNT approach were followed.

Table 3. Connections and relationships of the connections in the RDB

	Connection	Connection with relationships
1	$d1^{(XML)}$ – $e1^{(Smith)}$	$d1^{(XML)}$ ⟨1:N⟩ $e1^{(Smith)}$
2	$p1^{(XML)}$ – w_f1 – $e1^{(Smith)}$	$p1^{(XML)}$ ⟨1:N⟩ w_f1 ⟨N:1⟩ $e1^{(Smith)}$
3	$p1^{(XML)}$ – $d1^{(XML)}$ – $e1^{(Smith)}$	$p1^{(XML)}$ ⟨N:1⟩ $d1^{(XML)}$ ⟨1:N⟩ $e1^{(Smith)}$
4	$d1^{(XML)}$ – $p1^{(XML)}$ – w_f1 – $e1^{(Smith)}$	$d1^{(XML)}$ ⟨1:N⟩ $p1^{(XML)}$ ⟨1:N⟩ w_f1 ⟨N:1⟩ $e1^{(Smith)}$
5	$d2^{(XML)}$ – $e2^{(Smith)}$	$d2^{(XML)}$ ⟨1:N⟩ $e2^{(Smith)}$
6	$p2^{(XML)}$ – $d2^{(XML)}$ – $e2^{(Smith)}$	$p2^{(XML)}$ ⟨N:1⟩ $d2^{(XML)}$ ⟨1:N⟩ $e2^{(Smith)}$
7	$d2^{(XML)}$ – p3 – w_f2 – $e2^{(Smith)}$	$d2^{(XML)}$ ⟨1:N⟩ p3 ⟨1:N⟩ w_f2 ⟨N:1⟩ $e2^{(Smith)}$
8	d1 – e3 – $t1^{(Alice)}$	d1 ⟨1:N⟩ e3 ⟨1:N⟩ $t1^{(Alice)}$
9	d2 – p2 – w_f3 – e3 – $t1^{(Alice)}$	d2 ⟨1:N⟩ p2 ⟨1:N⟩ w_f3 ⟨N:1⟩ e3 ⟨1:N⟩ $t1^{(Alice)}$

The association of the keyword query in connection 4 is already implicitly visible for the user in connections 1 and 2. However, in that case we have to assume that the user browses through these two answers and discovers the association from answers. Further, it is not always the case that the association is implicitly visible in the other returned associations as is the case in connection 7.

4. DISCUSSION AND CONCLUSIONS

We have investigated the effects of the types of connections on the results of keyword queries over structural data. We considered how cardinality constraints affect the ranking of query results. We noticed that cardinality constrains can be utilized to infer the looseness of an association. A loose association gives a more extensive result for a keyword query because entities (tuples) are associated to each other through a more general entity or several entities. The closeness of a connection at the extensional level can partly be inferred from the cardinality constraints of the ER model. Immediate and transitive functional relationships ensure the close connection between the corresponding entities. Instead, other combinations allow close or loose connections between participative entities. For example, in a transitive N:M relationship several entities may be connected to each other through a more general entity and the semantics of the relationship is vague. However, further studies are needed for investigating how our findings could be utilized in ranking the result connections. One criterion could be the number of transitive N:M relationships in a connection. A more precise approach could be achieved by analyzing the actual number of participating entities (tuples) in a database instance.

We also proposed that the length of connections should be based on the relationships at the conceptual level because the N:M relationship corresponds to a conceptual relationship. Moreover, 1:N or N:1 relationship can be implemented by a middle relation. By using conceptual relationships the length of connections does not depend on implementation issues of this kind.

The results of a keyword search may produce several paths between tuples and they should be ranked based on their assumed relevance. One widely used indication has been the length of the path, i.e. the shortest paths are typically assumed to be more relevant than longer paths. However, longer paths may contain more information than shorter paths and shorter paths may chop a semantic connection between entities or text attributes/documents. Therefore, there should be an alternative where the user could select longer paths, if s/he is interested in larger context of matched values or documents.

5. REFERENCES

[1] Aditya, B., Bhalotia, G., Chakrabarti, S., Hulgeri, A., Nakhe, C., Parag, P. and Sudarshan, S. BANKS: Browsing and Keyword Searching in Relational Databases. In *Proceedings of the 28th International Conference on Very Large Data Bases*. VLDB '02. VLDB Endowment, 2002, 1083-1086.

[2] Bergamaschi, S., Guerra, F. and Simonini, G. Keyword Search over Relational Databases: Issues, Approaches and Open Challenges. In Ferro, N. ed. *Bridging Between Information Retrieval and Databases: PROMISE Winter School 2013. Revised Tutorial Lectures.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, 54-73. 2002, 1083-1086.

[3] Elmasri, R. and Navathe, S. B. *Fundamentals of Database Systems, Fourth Edition.* Addison-Wesley Longman Publishing Co., Inc, Boston, MA, USA, 2003.

[4] Hristidis, V. and Papakonstantinou, Y. Discover: Keyword Search in Relational Databases. In *Proceedings of the 28th International Conference on Very Large Data Bases*. VLDB '02. VLDB Endowment, 2002, 670-681.

[5] Kargar, M., An, A., Cercone, N., Godfrey, P., Szlichta, J. and Yu, X. MeanKS: Meaningful Keyword Search in Relational Databases with Complex Schema. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data.*. SIGMOD '14. ACM, New York, NY, USA, 2014, 905-908.

[6] Kargar, M., An, A., Cercone, N., Godfrey, P., Szlichta, J. and Yu, X. Meaningful keyword search in relational databases with large and complex schema. In Anonymous *2015 IEEE 31st International Conference on Data Engineering*. 2015, 411-422.

[7] Li, L., Petschulat, S., Tang, G., Pei, J. and Luk, W. Efficient and Effective Aggregate Keyword Search on Relational Databases. Int.J.Data Warehous.Min., 8, 4 (oct 2012), 41-81. DOI=10.4018/jdwm.2012100103.

[8] Zeng, Z., Bao, Z., Lee, M. L. and Ling, T. W. Towards An Interactive Keyword Search over Relational Databases. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, New York, NY, USA, 2015, 259-262. .