USING SEMANTICS IN DATABASE SYSTEMS

Pedro R. Acosta[1]*, Fredy Campos*[1]*, José L. Castillo*[3]*, José P. Miguel*[2]

Universidad Nacional de Ingeniería, Perú[1], Universidad Peruana Cayetano Heredia, Perú[2], Universidad de Alcalá, Spain[3]
e-mails: pacosta@uni.edu.pe; f.campos@ieee.org; jluis.castillo@uah.es; jose.miguel@upch.pe
Perú, Spain

Abstract: This paper aims to make explicit the incorporation of semantics or intention in Databases. We introduce the concept of Semantic Comment which is incorporated into the elements of the Database. Syntax diagrams are established for the common operations of Selection, Projection and Join. In practice the Semantic Comment incorporates the meaning of all the elements in a Database System. The Semantic Comment will help in the creation of ontology starting from a Relational Database System. The table storing the Semantic Comments will be helpful for the optimization in answering new queries.

Key words: semantic comment, database, semantic predicates, syntax, queries.

1. INTRODUCTION

At present it is common to speak of very large amounts of information available on the Internet. This can be a problem in terms of processing time and to improve their meaning or semantics, which gives rise to the issue of semantic web, where the main domains are the Databases (DB) and Artificial Intelligence techniques (AI). We introduce the Semantic Comment (SC) as a tool close to users to manage this abundance of information. By definition the SC is a string of characters that establishes the meaning of the associated element being this either a file name, description of a field of a table in a database, etc. In [12], the author uses semantic segments with a mathematical structure; although it is not entirely clear the capture of the semantic meaning.

Regarding DB, the search times can be improved on specific topics using results already stored as basic information for search algorithms. At present the relational model is the most widely used database model, but its weakness is in capturing the semantics of the database. Achieving capture the semantics within the DB will answer queries efficiently to end users, and even allow the development of high-level interfaces that can facilitate better communication.

This paper proposes that semantics and consultation must be described together making them easier to use and manage. This is accomplished through the Semantic Comment which is associated with the query operations.

According to [9] the general semantic model emerged as a computational model for the study of how communication works with natural language. The fundamental objective is to define, in abstract form, what can be applied to both natural and artificial agents, so as to achieve a model that defines the natural language communication. Also is used to design more accurate schemes compared to the data model, because the DB only has a limited knowledge about the meaning of stored information. Regarding AI agents they are provided with a better understanding of the data and easier access to it.

The other important concept related with semantics is ontologies [3]. Ontology is the study of entities or things that exist in reality. Depending on the topic, ontology can take many forms, but necessarily includes a vocabulary of terms and specifications about their meanings; these specifications contain the interrelationships between concepts in the field being studied. From [14] there is a great relationship between ontology a database and the relational model can be represented as ontology. The problem is that the relational model lack of semantic content.

In [15] the author use the semantic to make an intelligent image indexation using semantic and the search engine use ontology allowing a more efficient method for the recuperation of information. In this paper the SC is the element that will be incorporated to a DB. The SC will provide the semantic meaning of the element in the DB and will facilitate the answer to future queries.

In article [9] we find a study about the semantic Databases identifying the motivation the developers had for the creation of several models. The problem has always been similar and is to incorporate aspects of the real world in the model and the need for greater interaction with the user. The author acknowledges the logical complexity and mainly the problems interpretation, because consultations with an approach to natural language, ambiguities can occur. In this paper, with the addition of SC we propose to minimize the problem of ambiguity and facilitate the consultation process.

This article is organized as follows. We begin with a review of the state of art related to semantic databases in Section 2. In Section 3 we propose semantic comments as a way to add semantic meaning to the databases. In Section 4 we describe the syntax and semantics of a query to a database and Section 5 presents storing semantic query as a table and how it is used.

2. ANTECEDENTS AND PREVIOUS WORKS

In [1] the authors propose a semantic model to capture the semantic location for customers to access hypertext. This has been explored in Semantic Cache Memory (SCM) schemes (see [2]), used to process queries whose answers are partially contained in its Cache Memory (CM). The paper applies the same concept for hypertext-based navigation Database on the web, using the concept of reference location. He advocates a semantic model that takes into account the semantics of data in hypertext. Experimentally they create a database called HiperDB and generate workloads that have different references locations. These workloads are used for the analysis of SMC schemes for hyper-linked data.

In [3] ontology is proposed for the description of fixed networks models. This is due to the fact that ontologies are formal models of knowledge in a particular domain and are composed of classes that represent concepts that define the logical relationships that connect these concepts. The authors propose the first version of its ontology Computational Neuroscience. This concept can also be applied to define ontology for DB.

Currently and according to [4] there is a breakthrough in the development of technology oriented towards the creation of automated machines, such as the availability of thousands of robots and their various capacities. However robots communication is difficult, because it requires an increasingly rich and deep semantics. In [4] the author emphasizes the fact that you need to display the language and its meaning as a complex adaptive system that is constantly changing and is shaped by language users to meet their needs within the ecological environment they face. This means that semantics must be in a process of constant evolution and the blocks of meanings and their use in communication cannot remain static.

In [5] a methodology is proposed to show how a Semantic Web (SW) can meet a lot of requirements through characteristics of Knowledge Discovery (KD). This methodology consists of three phases: 1) Selecting an appropriate logical formalism, which allows the use of AI tools that can make inferences in documents of the SW. 2) Identification of some key aspects that must be taken into account for a SW and 3) the selection of a specific inference apparatus that meets the criteria established in phase 2). For the main part of their methodology they develop an interface for the KD, which is a web application, which implements the decisions and criteria for the three parts.

In [6] a hierarchical method is described for images which are semantically similar to be grouped together. Viewed through a series of general semantic characteristics related images with semantic same characteristics are grouped in a hierarchical top down structure. The method requires semantic descriptions of images, and each image is grouped into a "semantic bag", which consists of a set of descriptions related to the image.

In [7] the authors discuss problems of word clusters in text mining. They propose a text clustering algorithm based on semantic sequences.

In [8] text clustering is considered as an important aspect of Text Mining (TM). The text clustering algorithms based on distance or similarity often have great complexity and algorithms that are not based on distance are not suitable for dynamic or changing documents. Thus the authors propose a text clustering algorithm based on semantic sequences. The experiments performed indicate that the precision of their algorithm is better than other text clustering algorithms.

3. PROBLEMS AND PROPOSED SEMANTICS

Revised articles emphasize the use of semantics for different applications, such as image processing, text search, knowledge discovery, data processing in hypertext, etc. The central issue is the semantic interpretation of the information.

The relational model is currently the most widely used model of DB, even if its weakness is in capturing the semantics of a DB. One solution is to describe it separately, making difficult to administer and use. DB designers have to convert real world data structures to a low-level language that requires an extra level of indirection.

The other problem is that new applications demand data models that work with complex relationships, with significant limitations and management of large-scale data. For example Bioinformatics and Computer Aided Design are evolving with great intensity, making that data stored in a database system to have astronomical increases. The relational model would not be appropriate for the management of such a demand.

The Semantic Model of DB (SMDB) dates from 1981 [11] and was developed to solve the relational model problems described above. The SMDB incorporates much of the semantics but not in its entirety and focuses more on the perception of the real world problems and the relationships between them.

In this paper we introduce the Semantic Comment which is defined using predicates and strings. The SC incorporates data containing semantic meaning. The SCs are available for all elements of the DB. This will be explained in the following sections using practical examples.

4. SYNTAXES AND SEMANTIC OF A QUERY IN A DB

To answer a query the following tasks must be done:

a) An analysis of the syntax and semantic (parsing).
b) Optimization: Is the enumeration of the execution plans and the selection of the one that minimizes the processing cost.
c) Execution of the chosen plan and answer to the query is send to the user.

In this section we establish the main elements of the syntax and the semantic of a query language for a DB, developing the semantic analysis given in a).

The syntax analysis includes lexical analysis aimed at identifying the lexical elements in the query text. Next the parsing which goal is to review the syntax and assure the grammatical correctness of the query.

In this paper we focus on the semantics validation of the query. Semantic Comments are used to achieve this goal. The intention of the query is made explicit when the semantic component is incorporated into the syntactic definition. This allows us to perform tasks such as verifying the validity of the tables, fields, queries, operators, etc.

4.1 Structure of the Proposed Semantic Commentary

Table 1 shows the basic relation we will be using to illustrate the SC concept. They are data taken from an academic environment.

Table 1. Tables with their fields

Name of Table	Name of the fields
STUDENT	*Scod*, *Sname*, *Sage*, *Ccod*, *Sfaculty*, *Scred*
COURSES	*Ccod*, *Cname*, *Cnumcred*, *Cinstructor*
NOTES	*Ccod*, *Scod*, *Nnote*

Based on [12], consider the DB, $D = \{R_i\}$ a set of basic relations or tables. For each basic relation A_i denote the set of attributes or fields.

The SC is constructed using a set of predicates, which are used to capture the semantic meaning of all the components of a DB. We introduce the following definitions and types for a SC.

Definition 1: Simple Predicate P, consists of a sentence reflecting the intention of each element in the database. For example, the STUDENT table has the associated simple predicate “Students general information table”. Similarly the *Sage* field has the associated simple predicate “Student age”.

Definition 2: Compare Predicate P, where $P = a$ op c, a is an attribute of a basic relation, op $\in \{\leq, <, \geq, >, =\}$, c is a value domain or a constant. For example for the STUDENT table we have $P = Scred \geq 50$.

This definition does not consider the different operator ($\neq$) because their impact is minimal adding unnecessary complexity to the proposal.

Definition 3: Join Predicate P, where $P = (R_i.a_m = R_j.a_n)$, R_i and R_j are basic relations, a_m and a_n are attributes for the join operation. For example consider the tables STUDENT and NOTES, we have P = (STUDENT.*Scod* = NOTES.*Scod*). This predicate is expressed as a new relationship, joining tables STUDENT and NOTES by the student code.

Definition 4: Semantic Predicate P is defined as a Compare Predicate or a Join Predicate.

Definition 5: Semantic Segment. Given a Data Base $D = \{R_i\}$, a Semantic Segment, S, is a 4-upla $< S_R, S_A, S_P, S_C >$, where $S_R = \{Ri_1, Ri_2, \ldots, Ri_k\}$, $S_A \subseteq A_1 \cup A_2 \cup \ldots \cup A_n$; $S_P = P_1 \vee P_2 \vee \ldots \vee P_n$, where each P_j is a conjunction of simple predicates and $S_C = \pi S_A \sigma S_P(Ri_1 \times Ri_2 \times \ldots \times Ri_k)$, σS_P is the join operation using the predicate S_P and πS_A is the projection over S_A which gives the answers to the query.

For example, if the query is given by:

SELECTION (*Sname*, *Ccod*)
FROM STUDENT, COURSES
WHERE STUDENT.*Ccod* = COURSES.*Ccod* AND STUDENT.*Sage* > 18
In this case the Semantic Predicate is given by:
$S = < S_R, S_A, S_P, S_C >$ where
$S_R = \{\text{STUDENT, COURSES}\}$;
$S_A = \{Sname, Ccod\}$;
$S_P = (\text{STUDENT}.Ccod\text{=COURSES}.Ccod) \wedge (\text{STUDENT}.Sage > 18)$
$\sigma S_P(Ri_1 \times Ri_2 \times \ldots \times Ri_k)$, is
$\sigma(\text{STUDENT}.Ccod\text{=COURSES}.Ccod) \wedge$
$(\text{STUDENT}.Sage > 18)\ (\text{STUDENT} \times \text{COURSES})$ and
S_C contains the query result tuples.

4.2 Incorporation of the Semantic to Query Language Commands of a DB

Assuming all the queries made to a DB are a combination of operations of selection, projection and join, we establish the syntax with the addition of semantics to each.

SELECTION: operation to select a set of records from one of the tables in the database, meeting certain requirements. The syntactic structure is represented in Figure 1.

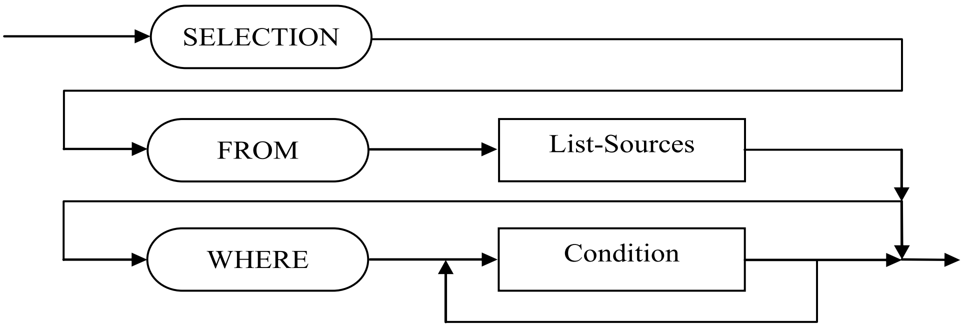

Figure1. Syntactic structure for SELECTION

The semantics will be included in **List-Source**, using the SC, which must appear automatically when mentioned. The SC is created upon the definition of the source, which is typically represented by a table. The syntax for the **List-Source** is shown in Figure 2.

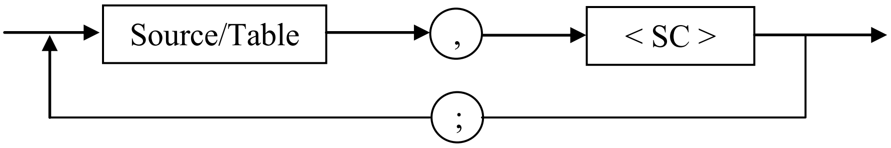

*Figure 2. Syntactic structure for **List-Sources***

PROJECTION: operation which aims to consider some fields from a table of the database. The syntactic structure is represented in Figure 3.

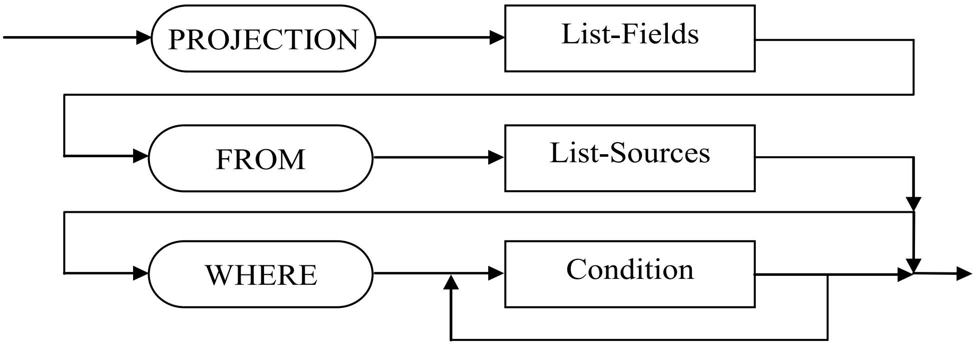

Figure 3. Syntactic structure for PROJECTION

The semantics will be included in **List-Fields**, using the SC, which must appear automatically when the field is mentioned. The SC is created upon the definition of the field. The syntax for the **List-Fields** is shown in Figure 4.

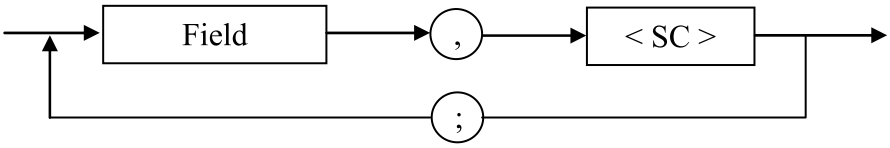

Figure 4. Syntactic structure for ***List-Fields***

JOIN: operation that aims to select a set of records from multiple tables in a database, considering certain fields. The syntactic structure is represented in Figure 5. The syntactic structures of **List-Sources** and **List-Fields** are shown in Figures 2 and 4.

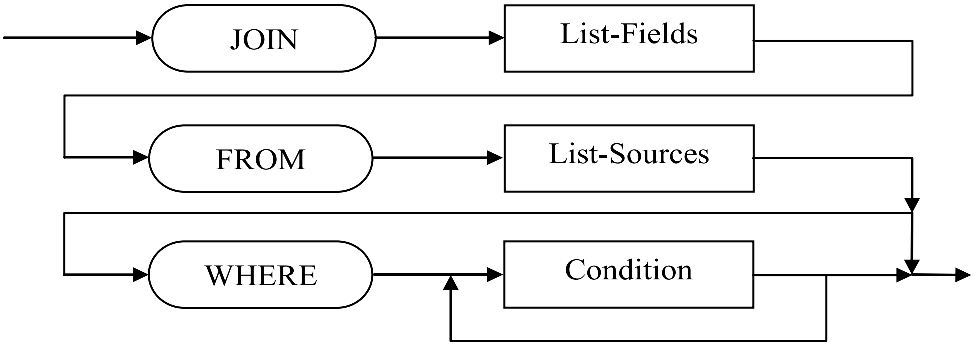

Figure 5. Syntactic structure for JOIN

4.3 Semantic Tables

The issues discussed in sections 4.1 and 4.2, reflect the construction of semantic tables representing simple predicates, compound predicates, queries and their corresponding meaning, as described in [13]. These tables are in expansion and their used will be for query processing.

The examples given in Section 4.1 allow as the construction of Tables 2 and 3.

Table 2. Simple Predicates

No.	**Name**	**Semantic meaning**
1	*Scod*	Student Code / STUDENT table
2	*Ccod*	Course Code / COURSES table
3	Sage	Age of the student / STUDENT table
4	*OS*	Operating System
5	*FIIS*	Faculty of Industrial and Systems Engineering
6	*Scred*	Credits approved
…	…	…

Table 3. Semantic query

No.	**Query**	**Semantic meaning**
1	SELECTION *Sname*, *Sage* FROM STUDENT WHERE *Sage* >18	Extracts information from the STUDENT table and displays the name and age of all students, whose age is greater than 18 years.
2	SELECCIÓN *Sname*, *Sfaculty* FROM STUDENT WHERE *Efacultad*=”FIIS” AND *Scred* > 50	Extracts information from the STUDENT table and displays the name and faculty of all students, whose faculty is FIIS and have approved more than 50 credits.

5. STORAGE OF QUERIES AS A SEMANTIC TABLE AND ITS USE

In this section queries expressed by semantic segments are stored in Table 4 named Semantic segments. This table includes results of the queries and therefore is related to the semantic query table (Table 3). This is the key which will be used in future queries.

The structure of Table 4 contains the semantic segment and includes a field named **No**. This field is the same that appear in the Table 3 and the field named S_{TS}

represents the date and time that the query was performed. The field S_{TS} is used for maintenance considerations, such as deciding the pages to replace in an algorithm such as LRU (Least Recently Used). The field S_T is used to relate semantic query and indicates the number of the query with which it interacts. In the absence indicates independence.

Take for example the tables defined in section 4.1 (see Table 1) and consider the following queries:

Query 1:
SELECTION *Sname*
FROM STUDENT
WHERE *Sage* > 18
Query 2:
SELECTION *Ccod*, *Cname*
FROM COURSES
Query 3:
SELECTION *Sname*, *Cname*
FROM STUDENT, COURSES
WHERE (STUDENT.*Ccod* = COURSES.*Ccod*) AND (COURSES.*Ci* = ”Acosta”)

The Semantic segments are represented in Table 4.

Table 4: Semantic segments

No.	S_R	S_A	S_P	S_C	S_{TS}	S_T
1	{STUDENT}	{*Sname*}	*Sage* > 18	{4, 5, 6}	T_1	
2	{COURSES}	{*Ccod*, *Cname*}	True	{3, 7}	T_2	
3	{STUDENT, COURSES}	{*Sname*, *Cname*}	(STUDENT.*Ccod* = COURSES.*Ccod*) AND (COURSES.*Cinstructor* = ”Acosta”)	{10}	T_3	

When performing a new query, this is translated into a semantic segment which we denote by $Q = <Q_R, Q_A, Q_P, Q_C>$. This semantic segment is compared with the stored semantic segments in Table 4, to determine coincidence. If the entire semantic segment coincides with one of the stored it means that is the same query and the problem is solved.

However in many cases the query can be answered in part by a semantic segment or not be answered. These situations are illustrated in Figure 6.

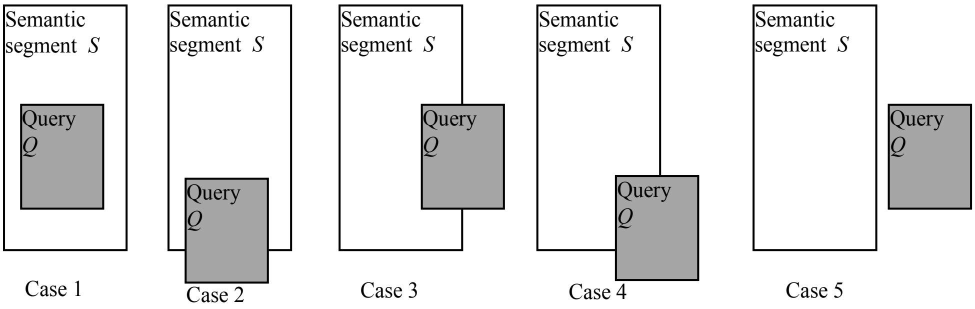

Figure 6. Possible relationships between a query and a semantic segment

Case 1: Total contention. The query Q is fully contained in segment S, meaning that $Q_A \subseteq S_A$. To answer the query we do operations of projection or selection on the results. For example if the query is:

Query 4:
SELECTION *Sname*
FROM STUDENT
WHERE *Sage* > 20

The following row is added to Table 4.

Table 5. Semantic segment generated for Query 4

No.	S_R	S_A	S_P	S_C	S_{TS}	S_T
4	{STUDENT}	{*Sname*}	*Sage* > 20	{ 5, 6}	T_4	1

To obtain the values of S_C a selection of the established pages for the original query S_1 has been done using $\pi Q_A \sigma Q_P(S_{1C})$.

Case 2: Horizontal Partitioning. This means that part of the query can be answered but there are some rows or records missing on the semantics table. To answer the query the result obtained is expanded. For example if the query is:

Query 5:
SELECTION *Sname*
FROM STUDENT
WHERE *Sage* > 15

The following row is added to Table 4.

Table 6. Semantic segment generated for Query 5

No.	S_R	S_A	S_P	S_C	S_{TS}	S_T
5	{STUDENT}	{*Sname*}	*Sage* > 15	{4, 5, 6, 7}	T_5	1

This means $Q_A \subseteq S_{1A}$, $S_{1P} \subset Q_{1P}$. To construct the corresponding S_{5C} to this new query we join query 6 to the original e.g. $S_{1C} \cup S_{6C}$ where query 6 is:

Query 6:

SELECTION *Sname*
FROM STUDENT
WHERE (*Sage* > 15 and *Sage* ≤ 18)

Case 3: Vertical Partitioning. This means that $Q_A \subseteq S_A$ is not true. In this case some fields have to be added to the stored result from the tables they come from. Consider the set $B = \mathrm{Q}_A - S_{xA}$, which contains the attributes of the query Q that are not contained in the semantic segment x. Then the missing elements can be expressed by $\pi B \sigma Q_P(R_1 \times R_2 \times \ldots \times R_n)$. For example if the query is:

Query 7:

SELECTION *Sname*, *Sfaculty*, *Scred*
FROM STUDENT
WHERE *Sage* > 18

The set $B = \{Sfaculty, Scred\}$ with $x = 1$.

The following row is added to Table 4.

Table 7. Semantic segment generated for Query 7

No.	S_R	S_A	S_P	S_C	S_{TS}	S_T
7	{STUDENT}	{*Sname*, *Sfaculty*, *Scred*}	*Sage* > 18	{4, 5, 6}	T_7	1

In Table 7, pages 4, 5 and 6 of the set S_C, contains the original values of the query 1, with the fields mentioned in S_A for which we use a join operation.

Case 4: Mixed Partitioning. It happens when you have to add both rows and columns. This means that $Q_P \subseteq S_p$ y $Q_A \not\subset S_A$. In this case we have to do the operations done in cases 2 and 3. For example, the following query:

Query 8:

SELECTION *Sname*, *Sfaculy*, *Scred*
FROM STUDENT
WHERE *Sage* > 15

The following row is added to Table 4.

Table 8. Semantic segment generated for Query 8

No.	S_R	S_A	S_P	S_C	S_{TS}	S_T
8	{STUDENT}	{*Sname,* *Sfaculy,* *Scred*}	*Sage* > 15	{4, 5, 6, 7}	T_8	1

Case 5: Not contained. In this case we have a new query that generates a new record in Table 4. For example if the query is:

Query 9:

SELECTION *Sname,Sage*
FROM STUDENT
WHERE *Sfaculty=" Faculty of Industrial and Systems Engineering"*

The following row is added to Table 4.

Table 9. Semantic segment generated for Query 9

No.	S_R	S_A	S_P	S_C	S_{TS}	S_T
1	{STUDENT}	{*Sname, Sage*}	*Sfaculty=" Faculty of Industrial and Systems Engineering"*	{9,10, 11}	T_9	

6. CONCLUSIONS

In this paper we propose the incorporation of semantic aspects to a DB using the SC. A SC contains the meaning for each element of a DB and is available for several uses. All the SCs are stored in tables using diverse types of predicates. The queries are stored in a table (in our example is Table 3) and for each query a semantic segment is created containing relevant information about its semantic. The semantic segments are stored in Table 4. Table 4 has fields that allow the establishment of relationships with other Tables. New queries will use the stored semantic segments to get the answers in a more efficient way.

The SC, as we propose, can be used at various levels (fields, queries, etc.) to provide semantic information to different users (simple users, programmers, DB administrators, etc.) to make explicit the intention. However, this may lead to overload information and impose some burden on the programming part but all can be offset by the quality of the final information and the time to get it. The SC will be always available through Tables (see Tables 2, 3, and 4) and hence the semantic meaning will be at hand any time.

Finally, we conclude that SC increment the semantically related content of a DB and hence will facilitate the construction of related ontology. This is left for future work.

REFERENCES

[1] Cheng Kai y Kambayashi Yahiko. A Semantic Model for Hypertext Data Caching, S. Spaccapietra, S.T. March, and Y. Kambayashi (Eds*.): ER 2002, LNCS Springer-Verlag Berlin Heidelberg* 2503, pp. 276–287, 2002.

[2] Li Luo, Konig-Ries Birgitta, Pissinou Niki, Makki Kia. Strategies for Semantic Caching, *Database and Expert Systems Applications. Lecture Notes in Computer Science*. Volume 2113, 2001, pp 284-298.

[3] Le Franc Yann, Davison Andrew, Gleeson Padraig, Imam Fahim, Kriener Birgit, Larson Stephen. Computational Neuroscience Ontology: a new tool to provide semantic meaning to your models. *From Twenty First Annual Computational Neuroscience Meeting*: CNS 2012. Decatur, GA, USA. 21-26 July 2012.

[4] Steels Luc, J. Bach and S. Edelkamp. Why We Need Evolutionary Semantics. *Springer-Verlag Berlin Heidelberg* (Eds.): KI 2011, LNAI 7006, pp. 14–25, 2011.

[5] Koutsomitropoulos Dimitrios, Fragakis Markos, Papatheodorou Theodoros. A Methodology for Conducting Knowledge Discovery on the Semantic Web. *Studies in Computational Intelligence. Springer-Verlag Berlin Heidelberg* (SCI) 14, 95–105 (2006).

[6] Chen Na and Prasanna Viktor. Semantic Image Clustering Using Object Relation Network. *Springer-Verlag Berlin Heidelberg*. S.-M. Hu and R.R. Martin (Eds.): CVM 2012, LNCS 7633, pp. 59–66, 2012.

[7] Mikhailov V. and Emel'yanov M. Semantic Clustering and Affinity Measure of SubjectOriented Language Texts. *Pattern Recognition and Image Analysis*. ISSN 10546618, Vol. 20, No. 3, pp. 376–385. Pleiades Publishing, Ltd., 2010.

[8] Feng Zhonghui, Shen JunyF, Bao Junpeng. An Incremental Algorithm of Text Clustering Based on Semantic Sequences. *Wuhan University Journal of Natural Sciences*. Vol. 11 No. 5 1340-1344, 2006.

[9] Shiguihara Pedro. Un breve panorama sobre las Bases de Datos Semánticas. *Escuela de Informática, Universidad Nacional de Trujillo*, 2011.

[10] Shaul Dar, Michael J. Franklin, Bjon T. Jonsson, Divesh Srivastava, Michael Tan. Semantic Data Caching and Replacement, *In Proc. Of: Intl. Conf. on Very Large Database (VLDB)*, Bombay, India, pages 330-341, Setember 1996.

[11] Hammer M., McLeond D. Database Description with SDM: A Semantic Database Model, *TODS* 6(3): 351-386, 1981.

[12] Qun Ren, Margaret H. Dunham. Semantic Caching and Query Processing, *Technical Report*: *Department of Computer Science and Engineering* 98-CSE-04. Southern Methodist University, Dallas, TX, USA.

[13] Kumar P.M., Vaideeswaran J. Semantic based Efficient Cache Mechanism for Database Query Optimization, *International Journal of Computer Applications* (0975 – 8887). Volume 43– No.23, April 2012.

[14] Martinez-Cruz C., Blanco I., Vila M. Ontologies versus relational databases: are they so different? A comparison. *Artif Intell Rev* 38:271–290, 2012.

[15] Osman T., Thakker D., Schaefer G. Semantics-Based Intelligent Indexing and Retrieval of Digital Images – A Case Study. Emergent Web Intelligence: *Advanced Information Retrieval* in Advanced Information and Knowledge Processing, pp 117- 134, Springer-Verlag London Limited 2010.

Information about the authors:

Pedro R. Acosta. Is Assistant professor in the Department of Basic Sciences, Faculty of Industrial and System Engineering, Universidad Nacional de Ingeniería, Lima, Perú. His main research interests are in the fields of: Data Base Systems, Fuzzy Logic, Parallel Programming, Knowledge extraction based on Neural Network and Data Mining.

Fredy Campos. Is an electronic engineer working as Senior Advisor in Networking and Comunications. His main research interests are in the fields of: Artificial Intelligence, Networking Real Time Systems and Data Base Systems.

José L. Castillo. Is Full Professor in the Department of Computer Science, Escuela Técnica Superior de Ingeniería Informática – ETSII, Universidad de Alcalá, Madrid, Spain. His main research interests are in the fields of: Information retrieval, Knowledge extraction based on evolutionary learning and Neural Network, Data Mining, Documentation, Software Engineering and the application of artificial intelligence techniques to the software cost estimation, Learning and teaching innovation with Web 2.0.

José P. Miguel. Is Professor in the Department of Basic Sciences, Faculty of Sciences, Universidad Peruana Cayetano Heredia, Lima, Perú. His main research interests are in the fields of: Data Base Systems, Software Engineering and Programming Real Time Systems.

Received: 12 January 2013
Revised: 27 February 2013