УДК 004.043

ГЕНЕРАЦИЯ SQL-ЗАПРОСОВ В ЗАДАЧЕ СОГЛАСОВАНИЯ ДАННЫХ ЭЛЕКТРОННОГО ДОКУМЕНТА И ТАБЛИЦ БАЗЫ ДАННЫХ

С.Ю. Марулин

Одесский национальный политехнический университет
65044, Одесса, проспект Шевченко, 1.
Тел. 8 (048) 779 7566
stasfoot@mail.ru

В работе описывается методика автоматизированной генерации *SQL*-запросов, позволяющих согласовывать данные электронных документов и соответствующих таблиц реляционной базы данных с целью актуализации информационного пространства организации. Методика представленная в работе позволяет значительно сократить число ручных операций создания *SQL*-запросов.

The paper describes a method of automated generation of *SQL*-queries to making data relevant electronic documents and relational database tables for actualization the information space organization. Method which describe in this paper allows considerably to shorten the number of hand operations of creation *SQL*-query.

Введение

Для связанной и эффективной роботы всех структурных подразделений большой организации, которая использует информационную систему (ИС) для поддержки производственных процессов, необходимо решать задачу согласования разнородных структур данных с единым хранилищем – базой данных (БД). В основном вся информация уже хранится в виде электронных документов (ЭД) различных форматов и чем быстрее исходные данные из ЭД будут перенесены в БД, тем эффективнее будет управление. Такой процесс согласования называется schema matching (SM) [1] и позволяет установить однозначные информационные потоки между ЭД и БД ИС.

Обеспечить процесс согласования данных из разных источников можно с использованием ETL-технологии [2], который включает три этапа: извлечение – Extract, преобразования – Transform и загрузку – Load данных, показанных на рис. 1.

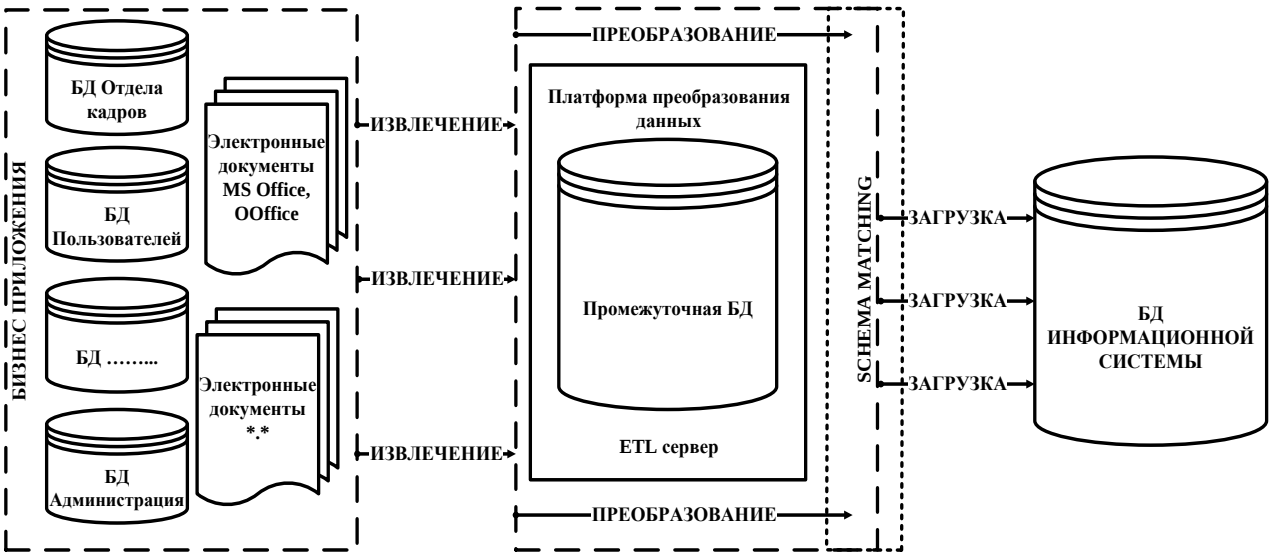

Рис. 1. Типичная схема *ETL* процесса большой организации

В работе [3] представлена ETL технология переноса содержимого ЭД в БД, где согласование структуры ЭД и таблиц БД осуществляется через создание модифицированной объектной модели ЭД с учетом словаря предметной области (СПО) и словаря БД (СБД). Однако, представленная ETL технология не реализует этап SM, поэтому цель работы – создание связи между ЕД и БД в виде SQL-запросов, автоматизирующей процесс согласования ЭД и БД.

Задачу SM необходимо решать при переходе из этапа преобразования к этапу загрузки (рис. 1), используя алгоритмы SM, представленные в работе [1]. Наиболее распространенным является синтаксический подход, основанный на вероятности совпадении названий – алгоритмы Левенштейна, N-грамм, SoundEx. Наряду с син-

таксисом в схемах данных используется семантическая составляющая (формат и типы данных, допустимые значения) а также терминологические отношения (синонимы, гиперонимы, гипонимы), что требует использование дополнительных структур (словари, онтологии). При структурном подходе к согласованию используются графовые алгоритмы [4], учитывающие взаимное расположение элементов в схемах данных. Оптимальным подходом считается комбинированный алгоритм [1]. Однако предложенные подходы не учитывают конкретные условия (предметную область) в которых выполняется процедура согласования и которые значительно влияют как на сам процесс согласования, так и на выявление точек согласования.

Объектная метамодель ЭД

Таким образом, точкой согласования со стороны системы электронного документооборота является информация ЭД, а со стороны БД модели SQL-запросов четырех типов: insert, select, update, delete.

Предложено представлять любой ЭД табличной структуры в виде объектной метамодели – ОМЭД модели. ОМЭД модель представлена в виде объединения двух элементов, образующих множество узлов:

$$\text{ОМЭД}=\text{head}\cup\text{body}=\{n_i\},$$

где head – заголовочная часть ЭД; body – содержательная часть ЭД; n_i – объект ОМЭД. Под объектом следует понимать информационную область (ИО), которая в исходном ЭД имеет вид объединенных ячеек или вид ячеек границы которых определяются наличием линий разграфки. Каждый объект ОМЭД характеризуется кортежем характеристик:

$$n_i = < pn, n, v, ix, l, m >,$$

где pn – номер узла родителя; n – номер узла; v – значение узла; ix – индекс значения узла v в словаре предметной области (СПО); l – уровень схожести значения узла v с эталоном в СПО; $m \in \{0, 1, 2, 3\}$ – метка принадлежности узла v к определенному типу ($0, 1, 2$ – статический, критериальный, динамический тип, соответственно).

В дальнейшем метка принадлежности узла ЭД к определенному типу позволяет устанавливать соответствия между блоками данных в ЭД и генерируемыми SQL-запросами.

Определение структурной зависимости ИО ОМЭД-модели

Для восстановления структуры ЭД появляется необходимость в обнаружении зависимости ИО одна от другой (1).

$$OI_{pn} \in OI_n \wedge OI_{pn} \notin OI_n. \quad (1)$$

Для определения такой зависимости разработана функция – GetParentNode:

GPN (*inRange As Range, compRange As Range*).

Функция, в качестве входного параметра, принимает массив координат $x1$, $y1$, $x2$, $y2$ всех ячеек, которые образовывают ИО и их порядковый номер n и значение координат $x1_i$, $y1_i$, $x2_i$, $y2_i$ текущей ячейки и ее порядковый номер n_i .

Функция *GPN* возвращает значение порядкового номера родительской *ИО* – pn, а в случае если такого значения не найдено возвращает 0. Значение $ИО_{pn} = 0$ свидетельствует о зависимости $ИО_n$ от самого ЭД. В табл. 1 показано результат работы функции *GPN*.

Таблица 1. Значения зависимости дочерней и родительской ИО

Значение информационной области	n	$y1$	$x1$	$y2$	$x2$	pn
форма	1	1	9	1	9	0
с-21	2	1	10	1	10	0
навчальний рік	5	5	1	5	1	4
семестр	7	5	9	5	9	0
напрям підготовки	8	6	1	6	1	5
курс	10	6	9	6	9	7
группа	13	7	9	7	9	10
ас091	14	7	10	7	10	11
відомість обліку успішності №	15	9	3	9	3	0
дисципліна	17	10	1	10	1	0

Для определения значений показателей уровня схожести и индекса лексем ОМЭД-модели разработано три функции: FuzzyMATCH, GetIndex, *MaxFuzzu* .

Функция FuzzyMATCH – использует методику нечеткого сравнения двух лексем [5] и возвращает коэффициент совпадения в диапазоне от 0 до 1, где 0 – лексемы полностью не совпадают, 1 – лексемы совпадают на 100%.

Функция MaxFuzzy – возвращает максимальный коэффициент совпадения двух лексем. З целью определения максимального коэффициента текущая лексема сравнивается со всеми лексемами в СПО. Определение максимального коэффициента должно удовлетворять (2).

$$\forall ОМЭД.v \; \exists ОМЭД.l = MaxFuzzu\left(ОМЭД.v_i, СПО.w_j\right). \tag{2}$$

Функция GetIndex – возвращает значение индекса лексемы из СПО уровень схожести которой максимальный с лексемой из ОМЭД-модели ЭД (3).

$$\forall ОМЭД.v \;\; \exists \; ОМЭД.ix = GetIndex\left(MaxFuzzu\left(ОМЭД.v_i, СПО.w_j\right)\right). \tag{3}$$

В табл. 2 представлен пример показателей созданной ОМЭД-модели ЭД.

Таблица 2. Значение показателей ОМЭД-модели ЭД

ОМЭД.v	*ОМЭД.n*	*ОМЭД.pn*	*ОМЭД.ix*	*ОМЭД.l*
форма	1	0	56	0,55
с-21	2	0	56	0,40
інститут комп'ютерних систем	4	0	22	1,00
навчальний рік	5	4	32	1,00
семестр	7	0	51	1,00
напрям підготовки	8	5	35	1,00
6050103	9	6	1	0,04
курс	10	7	29	1,00
группа	13	10	12	1,00
ас091	14	11	52	0,13
відомість обліку успішності №	15	0	37	0,80
1282	16	0	3	0,10
дисципліна	17	0	15	1,00

Типы блоков данных ЭД и правила их разметки

Статический тип данных (S) (4) – данные, которые постоянно присутствуют в ЭД и определяются на основании эталонных фраз из СПО:

$$ЭД^{(S)} \sim СПО\,. \tag{4}$$

Представим правила разметки узлов ОМЭД в терминах кванторов существования.

Правило № 1. $\forall \; ОМЭД.m \in СПО \rightarrow ОМЭД.m = 1$, где 1 – статический узел ЕД.

Словесное описание алгоритма:

- в цикле по всем узлам ОМЭД-модели ЭД определить наличие значение узла в СПО;
- если сходство найдено, то пометить узел как статический, иначе перейти к следующему узлу.

СПО имеет вид множества значений, каждое из которых характеризует одну запись предметной области –

$$Voc=\{voc_i\}\; Voc=\{voc_i\}, \text{ где } \; voc_i =< w, mw, intr, ix, dbr >,$$

где w – эталонная лексема для сравнения; mw – возможные варианты написания лексемы; $intr$ – интерпретация лексемы; ix – индекс лексемы в словаре; dbr – связь лексемы с таблицами в БД. $dbr =< dbs, dbl >$ – словарь БД (СБД), множество записей, которые описывают связь таблиц одна с другой в БД ИС. $dbs = \{<r, a, t, v>\}$ – множество кортежей, которые ставят в соответствие узел ЭД n_i таблицу r, атрибут таблицы a и значение v, $t \in \{vh, int, dt\}$ – тип атрибута: строковый, число и дата соответственно; $dbl = \{< t1, t2, a1, a2 >\}$ – множество кортежей, которые определяют связь атрибута $a1$ таблицы $t1$ с атрибутом $a2$ таблицы $t2$.

Динамический тип данных (D) (5) – данные образующие регулярные структуры в ЭД соответствующих статических полей. Динамические данные определяют *SQL*-запрос типа *update, delete*:

$$ЭД^{(D)} = f\,(ЭД^{(S)}, СВД, СПО)\,. \tag{5}$$

Динамические узлы, дерева ЭД, образовывают регулярные структуры – *Regular*.

Regular – это структура, образованная появлением в ЭД ряда однотипных значений, которые принадлежат к значениям из СБД, между каждой парой которых нет отличных значении. Пример регулярной структуры показан на рис. 2.

Прізвище, ініціали студента
АНАНЧЕНКО Н.О.
АНДРЄЄВ П.П.
АРНАУТОВ О.В.

Рис. 2. Пример регулярной структуры в ЭД

В последствии регулярные структуры сливаются в одно шаблонное значение.

Правило № 2. $\forall_{\text{ОМЭД.m}} \notin$ СПО $\wedge \forall_{\text{ОМЭД.m}} \in$ Regular→ОМЭД.m=3, где 3– динамический узел ЕД.

Словесное описание алгоритма:

– для всех не помеченных узлов ОМЭД-модели ЭД выполнить поиск значений, которые отвечают значениям из СБД;

– для найденных узлов ОМЭД-модели выполнить поиск регулярных структур.

Для поиска регулярности необходимо сравнивать значения, таблицы и поля двух соседних улов ОМЭД-модели ЕД:

$$\text{ОМЭД.}\ v_i \in \text{СБД.}\, v_j\ \textit{and}\ \text{ОМЭД.}\, v_n \in \text{СБД.}\, v_k\ \textit{and}\ \text{СБД.}\, c_j = \text{СБД.}\ c_k\ \textit{and}\ \text{СБД.}\ t_j = \text{СБД.}\ t_k,$$

где v_n – соседнее значение v_i. Под соседним значением следует понимать узел ОМЭД-модели, который стоит правее или ниже узла v_i (рис. 3).

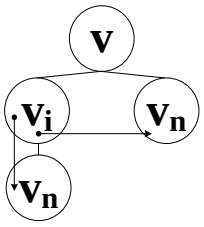

Рис. 3. Определение соседства узлов дерева электронного документа

Критериальный тип данных (С) (6) – задают ограничения на набор данных в ЭД и определяются на основании СБД и СПО. Критериальные данные отображаются в качестве условий на выборку после фразы *where* в *SQL*-запросах типа *select*:

$$ЭД^{(C)} = f\,(СБД,\ СПО). \tag{6}$$

Правило № 3. $\forall_{\text{ОМЭД.m}} \notin$СПО $\wedge \forall_{\text{ОМЭД.m}} \notin$ Regular $\wedge \forall_{\text{ОМЭД.m}} \in$СБД→ОМЭД.m=2, где 2– критериальный узел ЕД. Пример раскраски данных ЭД показан на рис. 4, а.

						Форма	С-21	
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ								
ІНСТИТУТ КОМП'ЮТЕРНИХ СИСТЕМ								
Навчальний рік	2011/2012					Семестр		
Напрям підготовки	6050103					Курс	3	
Спеціальність						Група	АС091	
	Відомість обліку успішності №		1282					Статические
Дисципліна	Конструювання Програмного Забезпечення					ІСП		Критериальные
Підсумкову оцінку виставив	Любченко Віра Вікторівна							Динамические
Поточний контроль здійснив	Жиро Лілія Сергіївна							
№ П/П	Прізвище, ініціали студента	№ інд. навч. плану	Підсумкова оцінка за шкалою				Підпис викладача	
			Стобальною	Чотири бальн.	ECTS			
1	АНАНЧЕНКО Н.О.		85					
2	АНДРЄЄВ П.П.		0					
3	АРНАУТОВ О.В.		30					

а – разбиение данных ЭД на три типа

	A	B	C	D	E	F	G	H	I
1								Форма	С-21
2	ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ								
3	ІНСТИТУТ КОМП'ЮТЕРНИХ СИСТЕМ								
4	Навчальний рік		$studyYear					Семестр	
5	Напрям підготовки		$code					Курс	$course
6	Спеціальність							Группа	$groupName
7			Відомість обліку успішності №		$vedomostNumber				
8	Дисципліна		$teachSubject					$controlType	
9	Підсумкову оцінку виставив		$firstTeacher						
10	Поточний контроль здійснив		$secondTeacher						
11									
12				№ інд. навч. плану	Підсумкова оцінка за шкалою				
13	№ П/П	Прізвище, ініціали студента			Стобальною	Чотири бальн.		ECTS	Підпис викладача
14	$stNum	$studName			$studMark		0	Не определено	

б – замена блоков данных условными обозначениями

Рис. 4. Разметка блоков данных электронного документа табличной структуры

Словесное описание алгоритма:

– для всех оставшихся не помеченных узлов ОМЭД-модели ЭД выполнить поиск значений, которые отвечают значениям из СБД;

– для всех найденных узлов ОМЭД-модели ЭД выполнить поиск узлов, которые не принадлежат Regular;

– все найденные узлы пометить как критериальные.

Генерация SQL-запросов

После разметки ЭД критериальные и динамические данные изменяются на условные переменные типа "$Name", где Name – имя условной переменной (рис. 4, б). Name формируется по правилу: в СБД ищется значение, соответствующее значению в текущем узле ЭД. Таблица и поле найденного значения подставляются в качестве значения Name:

$$(\forall_{\text{ОМЭД.m=2}} \wedge \forall_{\text{ОМЭД.m=3}}) \exists_{\text{СБД.dbr.db.r}}, \text{ где ОМЭД.v=СБД.dbr.dbl.v.}$$

Алгоритм генерации SQL-запроса типа Select.

SQL-запроса типа *Select* представлено как кортеж из трёх элементов:

$$SQL=<Sl, F, W>,$$

где $Sl=\{a_i\}$ – множество атрибутов фразы *select*; $F=\{t_i\}$ – множество таблиц фразы *from*; $W=\{w_i\}$ – множество условий фразы *where*.

1. Генерация выражения *Sl*. Для всех *n*, где *n.m=1*, сформировать фразу *Select*, где каждый атрибут *Sl* равен атрибуту *dbs.a* словаря *Voc* каждого n_i, который является наследником текущего *n*:

$$Select\{a_i=voc.dbr.dbs.a_j\}.$$

2. Генерация выражения *F*. Для всех *n*, де *n.m=1*, сформировать выражение *from*, где каждый атрибут *F* равен *dbs.r* словаря *Voc* каждый узел n_i, который является наследником текущего *n*:

$$from\{a_i= voc.dbr.dbs.r_j\}.$$

3. Генерация выражения *W*. Для всех *n*, где *n.m=1*, сформировать фразу *where*, где каждый атрибут *W*, формируется как ответ функцию *path(*dbd*)*+ условие. Условие сформировано для каждого текущего узла n_i, для которого название атрибута a_i равно названию соответствующего атрибута из *Voc:ai=voci.dbs.a*, а значение a_i равно конкретному значению узла $n_i.v$:

$$where\{path(dbr)+ a_i = n_i.v\}.$$

4. Сборка целостного *SQL*-запроса:

$$Select\{ a_i =voc.dbr.dbs. a_j \}||from\{ a_i =voc.dbr.dbs. r_j \}||where\{path(dbr)+ a_i = n_i.v\}.$$

Пример сгенерированного *SQL*-запроса типа *select* для критериального типа данных:

"select groupName from tableOfGroupName, tableOfStudyYear, tableOfCode, tableOfCourse where studyYear='$studyYear' and code='$code' and course='$course' + 'and'+ PATH(tableOfGroupName, tableOfStudyYear) + 'and' + PATH(tableOfGroupName, tableOfCode) + 'and'+ PATH(tableOfGroupName, tableOfCourse)".

Функция $PATH(t_i, t_j) =< t_i, \ldots, t_j >$ – выполняет поиск кратчайшего пути между таблицей t_i и таблицей t_j на основе алгоритма Дейкстры [6].

Пример сгенерированного *SQL*-запроса типа *update* для динамического типа данных:

"update tableOfVedomost_marks set studMark ='$studMark' where stNum=$stNum and studName=$studName"
+'and'+ PATH(tableOfVedomost_marks,tableOfstNum) +'and'+ PATH(tableOfVedomost_marks, tableOfstudName)"

Для определения количества операций автоматизированного создания *SQL*-запросов предложена комплексная методика определения этого количества, учитывающая не только этапы непосредственного создания запросов но предварительные этапы подготовки к процессу генерации.

Методика определения количества операций автоматизированного создания *SQL*-запросов

Для создания *SQL*-запросов администратору системы, выполняющего объединения двух информационных пространств – ЭД и таблиц БД информационной системы необходимо владеть знаниями в той предметной области, для которой создается запрос, а также знать структуру таблиц, которые обеспечивают обмен данными между конкретными ЭД и таблицами БД.

Таким образом, необходимо выполнять следующие операции:

– анализ структуры и содержание ЭД, с целью определения типов данных. Число операций ($|O_{ЭД}|$) зависит от количества найденных отдельных блоков динамического, критериального и статического типа;

– поиск регулярности и объединение данных. Число операций ($|O_{Reg}|$) зависит от содержимого ЭД и количества блоков данных, которые образовывают регулярные структуры в ЭД;

– анализ зависимых таблиц БД информационной системы. Число операций ($|O_{Ext}|$) зависит от числа таблиц, обслуживающих конкретный тип ЭД;

– сопоставление данных ЭД и таблиц БД. Число операций ($|O_{Match}|$) равно числу связей между таблицами БД и ЭД (количество *SQL*-запросов).

Расчет количества операций автоматизированного создания *SQL*-запросов предложено высчитывать по формуле:

$$O_{заг.} = |O_{ЭД}| + |O_{Reg}| + |O_{Ext}| + |O_{Match}|,$$

где $O_{заг.}$ – общее число операций, которое выполняется при генерации *SQL*-запросов.

Выводы

Предложенный подход автоматизированной генерации SQL-запросов был протестирован на смеси из 21 класса ЭД трех предметных областей и показал сокращение числа ручных операций согласования структур ЭД и таблиц БД на основе SQL-запросов в 9 раз.

1. *Rahm E., Bernstein P.A.* A survey of approaches to automatic schema matching // VLDB Journal. – 2001. – N 10. – P. 334–350.
2. *Vassiliadis P.* A survey of extract-transform-load technology. International Journal of Data Warehousing and Mining (IJDWM), 5(3):1-27, 2009.
3. *Alexander A. Blazhko., Marulin Stanislav, Kalashnikova Victoria* Data Exchange Technology Between Electronic documents and Relation Databases [Text] // Procs of 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), September 15–17, 2011, Prague,Czech Republic. – P. 624–628.
4. *Horst Bunke, Peter J. Dickinson, Miro Kraetzl*. Theoretical and Algorithmic Framework for Hypergraph Matching. ICIAP 2005: 463–470.
5. *Кунгурцев А.Б., Блажко А.А., Марулин С.Ю., Альсаффади Т.Д.* Алгоритмы сравнения однофразних текстов в технологии переноса содержимого электронных документов в реляционную БД [Текст] // Вестник Херсонского национального технического университета. – 2007. – № 4(27). – С. 308–311.
6. *Ибаа Сауд М.Р.* Информационная технология управления доступом к базам данных корпоративной информационной системы инструментальными средствами СУБД : дис. канд. тех. наук по информационным технологиям. – Одесский национальный политехнический университет. – Одесса, 2013. – 137 с.