

ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА ПРОЕКТИРОВАНИЯ SQL ЗАПРОСОВ, ОРИЕНТИРОВАННАЯ НА ПОЛЬЗОВАТЕЛЕЙ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ

В.В. Балтрушайтис

Рассматриваются проблемы создания инструментальных средств программирования, ориентированных на пользователей с ограниченными возможностями здоровья. Для этой группы пользователей важной характеристикой таких средств является объем вводимой информации. Для частичного решения данной задачи в статье предлагается создать интеллектуальную систему проектирования SQL запросов полного цикла. Систему предполагается базировать на модифицированной модели базы данных, введя в нее дополнительные категории, на базе которых будет создана база знаний (правил) SQL запросов.

The problems of developing the programming software tools aimed at disabled users are considered. The volume of input information is an important feature of these tools for this category of users. In order to partially solve the problem in question, the paper suggests developing an intelligence design system of full cycle SQL queries. The system is expected to be based on a modified database model supplied by additional categories, which will form the basis for the SQL query knowledge (rules) base.

1. ВВЕДЕНИЕ

В последнее время значительное внимание уделяется созданию инструментальных средств, ориентированных на пользователей с ограниченными возможностями по зрению [1, 2, 3].

Частичное решение этой проблемы в плане создания средств автоматизированного проектирования SQL запросов предложено в работе [2]. Вариант решения этой проблемы предложен в такой системе, как ACCESS [4]. Но графический интерфейс, используемый при этом, непригоден для пользователей с ОВЗ.

Реализация предложенного метода с использованием базы знаний и аппарата логического вывода представлена в работе [3], эффективность такого подхода и позволяет определить пути его дальнейшего развития. В процессе реализации осуществлено автоматизированное проектирование только раздела FROM SQL запроса, но работы с пользователями выявили необходимость автоматизации проектирования структуры результата запроса.

2. МОДЕЛЬ БАЗЫ ДАННЫХ

В работе [1] предложена модель базы данных реляционного вида, ориентированная на представление не только данных, но и знаний. Принципиальным отличием этой модели от известной модели «сущность – связь» является наличие дополнительных категорий модели: отношения между сущностями (связи), набор связанных сущностей, а также название проекта



запроса. В результате введения этих категорий в модель базы данных логического вида ее можно представить в виде четырех видов таблиц:

- наборов;
- отношений;
- сущностей;
- проектов.

Структуры этих таблиц имеют следующий вид.

Проект

Проект	ИД набора	Полное название проекта

Наборы

ИД набора	№ отношения	ИД левой сущности	ИД правой сущности	Структура результата выполнения запроса

Отношения

ИД левой сущности	ИД правой сущности	ИД поля 1 левой сущности	ИД поля 2 левой сущности	Операция реляционной алгебры (РА)	ИД поля 1 правой сущности	ИД поля 2 правой сущности

Сущности

ИД сущности	ИД поля					
	ИДП1	ИДП2	ИДП3	ИДП4	ПР конец записи
						*

3. МОДЕЛЬ БАЗЫ ЗНАНИЙ

В таблицах «Проект», «Наборы», «Отношения» размещена информация с правилами проектирования запросов, а в таблице «Сущность» — идентификаторы всех полей, им принадлежащих и используемых для организации интерфейса.

В модели предложено правила представлять в виде продукции следующего вида [5, 6]:

Правило 1:

IF (\exists IS1 Л ISp=null) then execute SELECT (IS1)

Правило 2:

IF \exists IN (\exists (IS1 Л ISp)) then execute IOIN (IN, 1, NR)

Правило 3:

IF \exists IN (ISp = NULL) then execute UNION (IN, 1, NR)

Текстовое изложение этих правил дает более детальное представление о выполняемых действиях:

Правило 1.

Если существует некоторая сущность S с идентификатором ISl из которой необходимо выбрать записи по условию или без него, а ISp = NULL, то выполнить присоединяемую процедуру SELECT (ISl).

Правило 2.

Если существует некоторый набор отношений с идентификатором IN, в котором определены идентификаторы левой сущностей, то выполнить присоединяемую процедуру JOIN с циклической обработкой всего набора отношений в диапазоне от первого до последнего (NR).

Правило 3.

Если существует некоторый набор сущностей одинаковой структуры с идентификатором IN, в котором идентификатор правой сущности равен ПУСТО, то выполнить присоединенную процедуру UNIX для всех сущностей заданного набора.

4. АЛГОРИТМ ВЫВОДА РЕШЕНИЙ

Алгоритмы вывода решений базируются на следующих основных положениях.

- Информационная часть правил размещена в четырех таблицах БД («Проекты», «Наборы», «Отношения», «Сущности»).
- Операторная часть правил размещена в присоединенных процедурах.

В процессе вывода решений необходимо:

- по заданному идентификатору проектируемого документа найти набор отношений обработки, выполнение которого приведет к решению;
- по заданному идентификатору документа определить тип и структуру проектируемого запроса и выполнить проектирование структуры документа (состав полей);
- по заданным идентификаторам связываемых сущностей найти условия соединения сущностей и осуществить проектирование раздела FROM со всеми его структурными компонентами (идентификаторами сущностей, операциями реляционной алгебры и условиями соединения таблиц);
- выполнить проекты множества запросов SELECT с последующим их объединением с помощью операторов UNION.

В качестве основного механизма вывода используется алгоритм прямого вывода в ширину:

- по заданному ключу связи между сущностями условия их соединения;
- по заданному составному ключу набора отношений и идентификатора сущности, а также полей, выбираемых из этой сущности в результате.

Найденное решение во всех случаях является окончательным и единственным. Количество проходов выполняется в цикле проектирования запроса, равно количеству соединяемых или объединяемых таблиц, уменьшенному на единицу.

5. ПРИМЕР РЕАЛИЗАЦИИ ПРЕДЛОЖЕННОЙ МОДЕЛИ И АЛГОРИТМА ПРОЕКТИРОВАНИЯ ЗАПРОСА

Все сказанное проиллюстрируем примером логической базы данных (БД) (рис. 1).

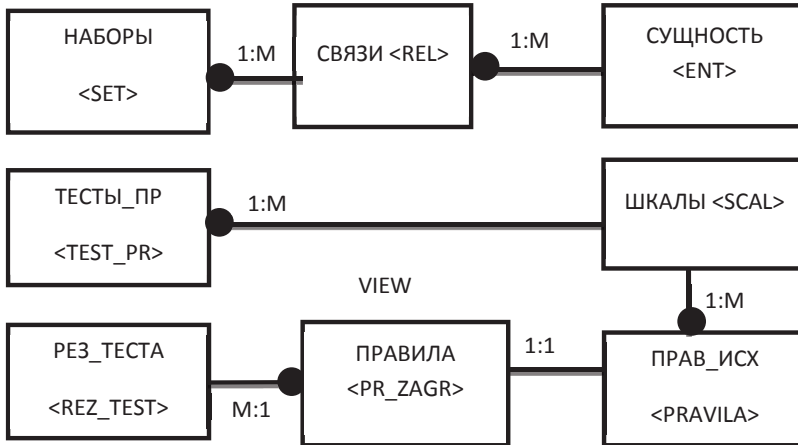


Рис.1. Фрагмент базы данных системы психологического тестирования ПСИ.

На рисунке представлен фрагмент базы данных системы психологического тестирования ПСИ, состоящий из двух структурных компонентов:

- 1) базы данных с результатами тестирования;
- 2) базы знаний, используемой для обработки результатов тестирования [7].

К первой части относятся таблицы БД:

<TEST_PR>, <SCAL>, <PRAVILA>, <PR_ZAGR>.

Ко второй части таблицы – <ПРОЕКТ>, <SET>, <RELATION>, <ENTITY>.

Вторая часть представляет собой Мета-данные и содержат факты и правила, необходимые для решения основной задачи системы – обработки результатов тестирования.

В приведенном фрагменте базы данных содержится следующая информация

1. таблица <TEST_PR>:

- полное название теста – NAIME,
- идентификатор теста – IDTEST,
- программная структура, используемая при обработке этого теста – IDSTR;

2. таблица <SCAL>:

- идентификатор программной структуры – IDSTR,
- идентификатор элемента – IDEL,
- идентификатор шкалы – IDSC,
- формат элемента – FORMAT;

3. таблица <PRAVILA>:

- идентификатор теста – IDTEST,
- идентификатор правила – IDPRAV,
- идентификатор шкалы – IDSC,
- предикат – PRED1,
- предикат – PRED2,
- эталонные значения параметров – ZN1, ZN2,
- операторы выхода из правил – VIHOD,

– операторы вывода результатов – VIVOD;

4. таблица <PR_ZAGR> представляет собой загрузочный вариант таблицы <PRAVILA>, в которой произведена замена идентификаторов шкал на идентификаторы программных структур и их элементов.

Проиллюстрируем работу предложенного алгоритма на практическом примере проектирования SQL запросов к базе данных психологического тестирования, приведенной в таблицах ниже.

Проект

Проект	ИД набора	Полное название проекта
Проект набора правил	SET_PR	Проект набора правил обработки результатов тестирования

Наборы

ИД набора	№ отношений	ИД левой сущности	ИД правой сущности	Структура результата выполнения запроса
SET_PR	1	TEST_PR	SCAL	IDTEST, IDSTR,
	2	SCAL	PRAVILA	IDSC, IDEL,
	3	PRAVILA	PR_ZAGR	IDPRAV, PRED, ZN, VIH, VIV

Отношения

ИД левой сущности	ИД правой сущности	ИД поля 1 левой сущности	ИД поля 2 левой сущности	Операция реляционной алгебры (РА)	ИД поля 1 правой сущности	ИД поля 2 правой сущности
TEST_PR	SCAL	IDTEST		JOIN	IDTEST	
SCAL	PRAVILA	IDTEST	IDSC	JOIN	IDTEST	IDSC
PRAVILA	PR_ZAGR	IDPRAV	GRUP	JOIN	IDPRAV	GRUP

Сущности

ИД сущности	ИД поля									
	ИДП1	ИДП2	ИДП3	ИДП4	ИДП5	ИДП6	ИДП7	ИДП8	ИДП9	К.з.*
TEST_PR	IDTEST	IDSTR	NAIM	*						
SCAL	IDSTR	NEL	IDSC	IDEL	FORMAT	*				
PRAVILA	IDTEST	IDPRAV	GRUP	SC	PRED	ZN	VIH	VIV	*	
PR_ZAGR	IDTEST	IDPRAV	GRUP	IDSTR	IDEL	PRED	ZN	VIH	VIV	*

* К. з. – конец записи.

Предположим, что перед нами поставлена задача выполнить проект создания таблицы правил обработки результатов тестирования в терминах идентификаторов программного уровня. В исходной таблице <PRAVILA> эти правила представлены в терминах идентификаторов пользовательского уровня. Решение такой задачи может потребоваться, например, после редактирования базы данных.

Для выполнения проекта пользователю системы необходимо выбрать вариант проекта из таблицы <PROECT> и запустить систему. Возможен вариант, когда требуется уточнить условия проекта, однако в данной случае это не нужно.

В результате решения получим SQL следующего вида.

```
SELECT TEST_PR. IDTEST, SCAL, SCAL. ISC, IDEL, PRAVILA. IDPRAV, GRUP,  
       PRED, ZN, PR_ZAGR. VIHOD, VIVOD  
FROM TEST_PR LEFT OUTER JOIN SCAL ON TEST_PR. = SCAL. IDTEST  
       AND TEST_PR. IDSTR = SCAL. IDSTR  
LEFT OUTER JOIN PRAVILA ON IDTEST = PRAVILA. IDTEST  
       TEST_PR LEFT OUTER JOIN SCAL ON TEST_PR. = SCAL. IDTEST  
       AND IDSTR = PRAVILA. IDSTR  
LEFT OUTER JOIN PR_ZAGR ON IDPRAV = PR_ZAGR. IDPRAV  
       AND IDSC = PR_ZAGR. IDSC  
WHERE  
       ORDER BY IDTEST, IDPRAV, GRUP;
```

6. ЗАКЛЮЧЕНИЕ

Преимущества предложенного подхода заключаются в следующем:

- выполняется автоматизированное проектирование не только раздела FROM, но и раздела структуры результата;
- повышается качество проекта за счет системного представления структуры базы данных;
- снижается трудоемкое программирование SQL запросов, что особенно важно для пользователей с нарушениями зрения.

ЛИТЕРАТУРА

1. Балтрушайтис В.В. Модель базы данных, ориентированная на использование в практических задачах психодиагностики // Моделирование и анализ данных: научный журнал. 2007. № 3. С. 97–100.
2. Балтрушайтис В.В. Поискный процессор базы данных семантического вида, ориентированный на пользователей с ограниченными возможностями // Моделирование и анализ данных: научный журнал. 2009. № 4. С. 72–82.
3. Балтрушайтис В.В. Интеллектуальная инструментальная система проектирования алгоритмов и программ, ориентированная на пользователей с ограниченными возможностями здоровья // Моделирование и анализ данных: научный журнал. 2011. № 1. С. 153–163.
4. Михеева В., Харитоновна И. Microsoft Access 2003. Наиболее полное руководство. СПб.: БХВ-Петербург, 2002. 1040 с.
5. Поспелов Г.С., Поспелов Д.А. Искусственный интеллект — прикладные системы. М.: Знание, 1985. 48 с.
6. Поспелов Д.А. Моделирование рассуждений. Опыт анализа мыслительных актов / Д.А. Поспелов. М.: Радио и связь, 1989. 184 с.
7. Стернин М.Ю., Шепелев Г.И. Метод представления знаний в интеллектуальных системах // Новости искусственного интеллекта. 2003. № 4 (58).

Работа поступила 29.11.2012