

СРЕДА ЗА РАБОТА С БАЗИ ОТ ДАННИ

Румен Тодоров, Даниела Орозова

Бургаски свободен университет
Бул. „Сан Стефано“ 62, Бургас 8000
e-mail: orozova@bfu.bg

Резюме: В този доклад се представя създадената програма „SQL Flat“, която представлява развойна среда за проектиране и реализиране на бази от данни. В предлаганата среда потребителят може да използва богат набор от средства за администриране, създаване и обработка на релационни бази от данни. Предлагат се средства за развиване на средата в обучаваща, адаптивна към потребителя развойна среда.

Ключови думи: Релационни бази от данни, Адаптивна развойна среда, XML файл, T-SQL код.

1. Въведение

Създадена е развойна среда за работа с бази от данни [4], която предоставя различни средства за създаване и манипулиране на данни, съхранявани и управлявани от SQL Server чрез езика SQL. Потребителят има на разположение техники, чрез които може да зарежда данни, да ги променя, записва и изпълнява команди. Интерфейсът, който програмата предоставя, е интуитивен, логически подреден и групиран в няколко основни секции:

- общи ленти с инструменти;
- секция по създаване и работа с SQL програмен код;
- секция по визуализация на таблици от базата от данни;
- секция с опционални инструменти.

Всяка отделна секция има своето предназначение и предлагана на потребителя различна функционалност. Общите ленти с инструменти предоставят основни методи за създаване, зареждане, търсене на бази от данни, както и изпълнение на команди върху тях. Секцията по създаване и работа с програмен код предоставя на потребителя различни опции по форматирането и архивирането на изходния код, написан на T-SQL [9]. Зареждането на таблиците, включени в текущо отворената база от данни, както и различни инструменти за обхождане, търсене и промяна са налични в секцията по визуализация на таблици от базата данни [7]. SQL Flat предлага различни допълнителни настройки към инструментите за работа с бази от данни, които са част от секцията по опционални настройки. В нея могат да се зададат различни „фини“ настройки относно низа за свързване със сървъра, настройки по заявки и транзакции.

Програмният продукт е създаден с Microsoft Visual Studio, използвайки езика C++. Версията на езика, която се използва в Visual Studio .NET е C++/CLI [2]. Това е разширена версия, която осъществява съвместимост между стандартния C++ и обкръжението .NET [10]. Използваната технология за достъп до база от данни е ADO.NET [5], а доставчикът на данни е SQL Server. Визуално SQL Flat се основава на Windows Forms. Технически създаденият софтуер се основава на най-новите и актуални технологии, реализирани от Microsoft. Програмата е проектирана за работа изключително върху операционни системи на Microsoft Windows XP и Windows 7.

2. Общи ленти с инструменти

Общите ленти с инструменти предоставят на потребителя фундаментални средства за работа с програмата: основно за създаване и зареждане на бази от данни. Те са независими от методите, предлагани от останалите секции. На фиг. 1 е показана лентата с инструменти, предлагаща тези методи.



Фигура 1.

Създаването на нова база от данни се осъществява, чрез бутона Create New Database, с предварително зададено име в текстовото поле. Зареждането на вече съществуваща база се извършва с бутона Open Database. Предварително потребителят трябва да дефинира символния низ за връзка към сървъра, като явно укаже името на базата за осъществяване на връзка. Низът се дефинира в секцията по задаване на допълнителни опции, в частта Connection String. Ако операцията по зареждане на базата от данни е приключила успешно, таблицата в секцията по визуализиране на данните се попълва с информация относно зарежданата база. Такава информация е каталогът, схемата, имената на различните таблици и техния тип. На фиг. 2 е показано как изглежда типичен екран на програмата след успешно изпълнение на операцията.

TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE
master	dbo	spt_fallback_db	BASE TABLE
master	dbo	spt_fallback_dev	BASE TABLE
master	dbo	spt_fallback_usg	BASE TABLE
master	dbo	spt_monitor	BASE TABLE
master	dbo	spt_values	BASE TABLE
master	dbo	MSreplication_op...	BASE TABLE

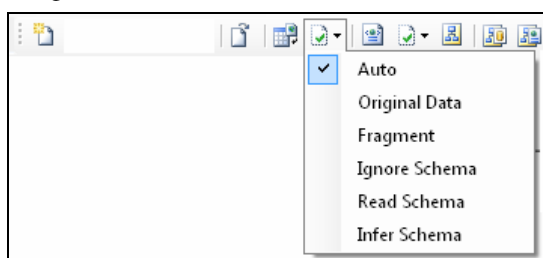
Фигура 2.

В показания пример се зарежда база от данни с име Master, а низът за връзка е показан във фрагмента от фиг. 3.

Server=localhost; Database=Master; Integrated Security=SSPI

Фигура 3.

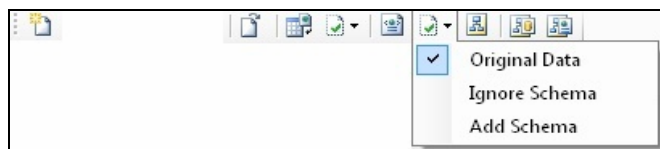
SQL Flat предлага възможност за зареждане и на бази от тип XML. Бутонът, изпълняващ тази функция е Open XML Data. Възможно е да се зададат и опции за начина, по който ще се зареждат XML данните от източника. Опционалните възможности са показани във фиг. 4.



Фигура 4.

Чрез тази функционалност, потребителят може да избере, каква част от данните да се прочетат и визуализират. Има възможност да се зареди само схемата от източника, да се игнорира тази схема и да се използват само данните в текущата схема или да се форматираат текущите данни според схемата на източника.

Програмата SQL Flat позволява данните да бъдат записани в XML файл. Това се осъществява, чрез бутона Save XML Data. Потребителят би могъл да зададе допълнителни опции, свързани с начина на съхранение на данните, показани на фиг. 5. В различните случаи се записват схемата или данните.



Фигура 5.

Предоставена е възможност да се определи типът на източника на данни – дали това е SQL Server или XML файл. Това се извършва чрез бутоните Set SQL Data Source и Set XML Data Source. При избор на XML източник се влиза в режим, при който не могат да се изпълняват заявки и транзакции.

Полезна лента с инструменти е показаната на фиг. 6. Бутоните позволяват отваряне на връзка към база от данни и възможното и затваряне.



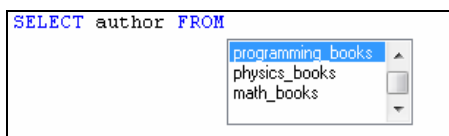
Фигура 6.

вият редактор поддържа динамично и синтактично оцветяване на ключови на думи и фрази от езика. Всички запазени думи на T-SQL са включени в речник. Изображение с работата на редактора е показано на следващата фигура.

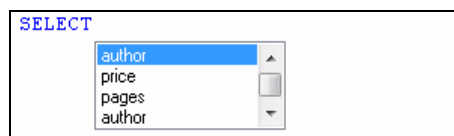
```
-- създава нова база от данни
CREATE DATABASE library
-- създава таблици
CREATE TABLE math_books (author char(50), price int, pages int)
CREATE TABLE physics_books (author char(50), price int, pages int)
CREATE TABLE programming_books (author char(50), price int, pages int)
-- вмъква записи в таблица
INSERT INTO math_books VALUES ('some_author', 22, 202)
INSERT INTO programming_books VALUES ('some_author', 20, 800)
```

Фигура 10.

Редакторът притежава функционалност, улесняваща потребителя в много отношения. По време на въвеждането на кода може да се изведе списък с всички таблици, съдържащи се в текущо заредената база от данни, както и колоните към тях. Местоположението им зависи от координатите на курсора в редактора. На фиг. 11 и фиг. 12 се показва начинът, по който се извеждат списъците с таблици и колони. Примерът използва предварително създадена база от данни с име library, която съдържа таблици с имена math_books, physics_books и programming_books. Всяка таблица съдържа определени колони: author, price и pages.



Фигура 11.



Фигура 12.

Допълнителни инструменти могат да променят режима на работа на текстовия редактор, като се забрани синтактичният разбор на програмния код и оцветяването на ключовите думи. Това става възможно чрез командата Syntax Coloring. Ако тази опция е активна, разпознаването и промяната на цвета на запазените думи на езика се извършват по време на въвеждането на кода. Налична е опция, която позволява разборът и оцветяването да се изпълняват единствено при преминаване на нов ред. Бутонът, позволяващ това, е Line by Line Highlight. Потребителят може да използва функционалност, която анализира програмен код, който не е форматиран в съответните правила, и установява необходимите промени по него, като например оцветяване. Бутонът Coloring изпълнява тази задача.

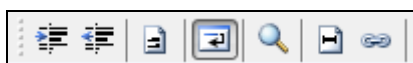
Като защита срещу нежелани промени в програмния код може да се приложи заключване на целия документ с опцията Lock Document, която забранява неговата редакция. Възможно е да се забрани промяната само на отделни фрагменти от кода, а останалата част да може свободно да се редактира. Защитата на избран фрагмент се осъществява с командата Lock Code. Средата предоставя и възможност за архивиране на текущия код в текстовия редактор. По този начин се съхранява негово копие, което е възможно да се възстанови при необходимост в работата.

Една от основните ленти с инструменти предоставя на потребителя средства за зареждане и запис на файлове с програмен код. Допустимите формати са: обикновен текстов (.txt) и богат текстов формат (.rtf). На фиг. 13 е показано изображение на бутоните. Особеност тук е едновременната поддръжка на различни и независими фрагменти, съдържащи изходен код. Те се съхраняват в динамичен масив. Потребителят може да работи само върху един от тези фрагменти в определено време, изпълнявайки кода или прилагайки друг тип функционалност. Има възможност за добавяне на нови елементи с код към масива или тяхното изтриване. При обхождането на масива от кодови фрагменти е необходимо потребителят да го актуализира, като добавя направените промени. Това се осъществява чрез бутона Update. Освен това, може да се избере ниво на защита на информацията, съхранена във файла, като се използва метод за криптиране на данните. Това дава известна сигурност на програмния код.



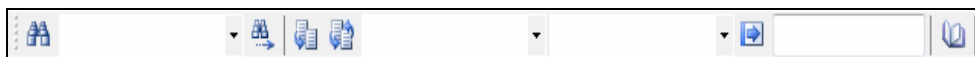
Фигура 13.

На фиг. 14 е показана лентата с инструменти, която съдържа команди за подравняването на фрагменти от програмния код (бутони Indent и Outdent), големината на шрифта (бутон Zoom) и определяне дали текстовият редактор ще тълкува интернет адресите като хипервръзки или като обикновен текст.



Фигура 14.

Средата предлага възможности за търсене на определени думи или фрази в програмния код, както и замяната им с други. За тази цел се използват бутонът Find. Бутонът Go To се използва в случаите, в които потребителят желае курсорът да се премести директно на определен ред от програмния код в текстовия редактор. Bookmark задава отметка към определена позиция в текста на програмния код и позволява на потребителя да се върне към нея, когато е необходимо.



Фигура 15.

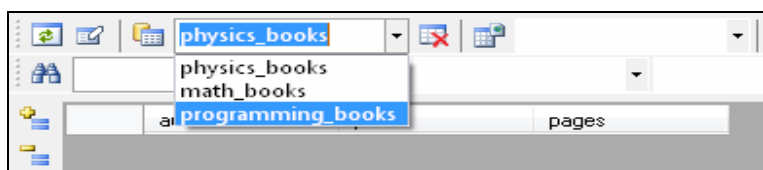
4. Секция по визуализация на таблици от базата данни

Тази секция се използва за визуално представяне на таблиците и данните в тях. Могат да бъдат заредени таблици, независимо от техния източник (SQL Server или XML). Лентите с инструменти предоставят различни команди, улесняващи потребителската работа при зареждането на отделните таблици от базата и тяхното обработване. Основната лента с команди е показана на фиг. 16:



Фигура 16.

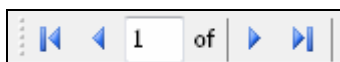
Тя осигурява зареждането и изтриването на таблици независимо от източника им. Бутонът **Select Table** визуализира таблица, избрана от падащия списък, който се обновява всеки път при зареждане на нова база от данни. Резултат от работата на бутона е показан на фиг. 17. Демонстрира се избирането на една от трите съществуващи таблици в текущо заредената база от данни.



Фигура 17.

Командата **Reload Table** зарежда отново текуща таблица. Използва се в случаите, когато потребителят е направил определени промени (независимо по какъв начин – чрез заявка или транзакция). Бутонът **Unbound Table** премахва зависимостта между обекта, който визуализира дадена таблица, и самата таблица. Съществува възможност за изтриване на таблица, принадлежаща на базата от данни с помощта на бутона **Delete Table**.

Програмата предоставя средства за обхождане на отделните записи, благодарение на командите от лентата с инструменти показана на фиг. 18.



Фигура 18.

Позволява се както последователно преминаване, така и директно прескачане към първия или последния запис в таблицата. Потребителят бива информиран за номера на реда, който е маркиран в момента. Поради големите обеми данни, които може да съдържа дадена таблица, е наложително да има средства за търсене, замяна и маркиране на клетки или цели записи. На фиг. 19 е показан начинът, по който изглеждат командите, осигуряващи тази функционалност.



Фигура 19.

Бутонът **Find** намира всички съвпадения на думата, посочена от падащия списък. Търсенето се извършва в цялата и само в текущата таблица. Ако има намерени съвпадения, те се отбелязват чрез промяна на фоновия цвят на съответната клетка, като се използва лесно открояващият се жълт цвят. На фиг. 20 е показан примерен резултат от тази команда: таблицата, в която се търси, е `math_books`, а търсеният низ е `201`.

author	price	pages
some_author ...	20	200
some_author ...	21	201
some_author ...	22	202

Фигура 20.

В случаите когато е нужно премахване на фоновото оцветяване, се използва бутонът **Clear Highlighting**. Има възможност за създаване на отметка върху дадена клетка и за връщане към нея в по-късен етап от работата на потребителя. Командата е с име **Bookmark**.

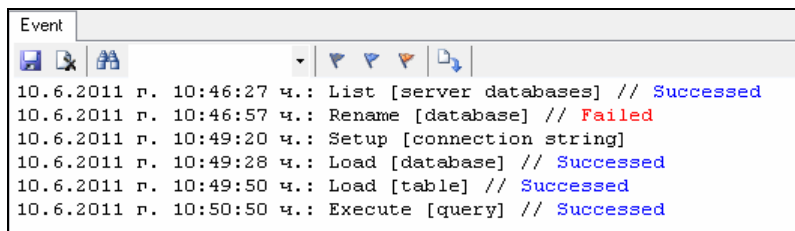
Съществува възможност потребителят да разреши или забрани промяна на размерите на колоните и редовете или да промени подредбата им във визуалната представа на таблицата. Тези промени не влияят на данните. Налични са опции, които определят дали може да се добавят или премахват записи от таблицата. Това важи единствено в случаите, когато източникът на данни е XML файл. Възможно е заключването на цялата таблица с цел да се ограничат непозволените промени в нея.

5. Секция с опционални инструменти

Секцията с опционални инструменти предоставя на потребителя различен набор от методи както за наблюдение на процесите, извършващи се в програмата, така и за допълнителна настройка на някои от тях. Секцията е разделена на части, всяка от които съдържа инструменти, подпомагащи процеса на изграждане на база от данни. Те са организирани в отделни табове относно:

- архивиране на действията на потребителя;
- извеждане на информация, относно процеса на компилация;
- извеждане на информация, относно възникнали грешки;
- извеждане на резултата от изпълнението на XML заявка;
- установяване на свойствата на свързващият низ;
- създаване на нов файл с база от данни;
- задаване на свойства на заявка;
- задаване на свойства на транзакция;
- хронологично съхраняване на операциите от текстовия редактор и работа с клип-борда;
- работа с календар.

Средата за работа предоставя възможност за хронологично записване на действията от страна на потребителя. На фиг. 21 е даден пример за това.



Фигура 21.

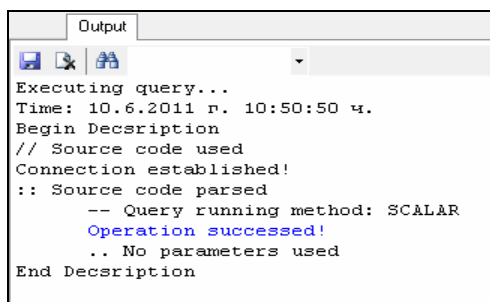
Информацията, която се съхранява, е: датата на изпълнение на съответната команда, часът, типът на операцията, която се извършва, върху какво се изпълнява тази операция и резултатът от нея. Програмата разделя типовете операции на два основни вида.

- операции с особена важност;
- стандартни операции.

Към първия вид се причисляват тези команди, които са от критична важност за проектирането на базата от данни. Най-често чрез тях се правят промени в сървъра или в текущо заредената база. В много от случаите тези промени са трудно обратими като изтриване на база от данни или изтриване на таблици.

Вторият вид команди са тези, които информират потребителя за настъпило събитие. Програмата предлага и възможности за филтриране на записите според техния тип. Чрез командите Show All, Show Only Succeeded и Show Only Failed се избира кой тип операции да бъдат показани: дали това да са само успешно приключилите, само неуспешно приключилите или всички, независимо от техния резултат. Поради големия обем от събития, които могат да настъпят, съществува възможност за търсене на определени думи или фрази в хронологията. Бутонът предоставящ тази функционалност, е Find. Удобна опция при работа е записването във файл на потребителските действия.

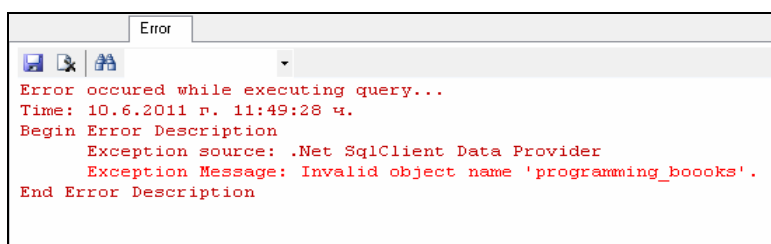
Извеждането на информация относно компилацията е от изключителна важност по време на работа със средата. Предлага се на потребителя средство за описание на процеса на изпълнение. На фиг. 22 е показан пример за това.



Фигура 22.

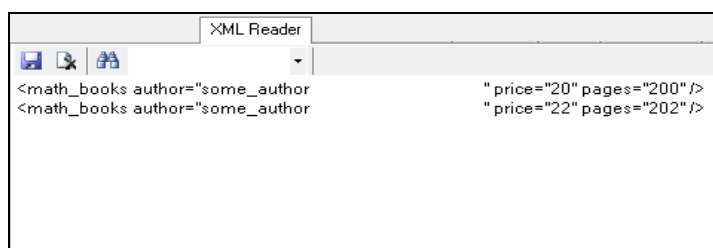
Данните, които се извеждат след изпълнение на команда, съдържат информация относно нейния тип, датата и час на изпълнение, указва се дали програмата е в

режим на изпълнение на целия код или само на фрагмент от него, посочва се дали е установена връзка със сървъра, типа на изпълнение, резултата, и дали са използвани параметри. Тук отново са на разположение стандартните опции за записване на файл, съдържащ описанието на процеса по изпълнение и търсенето на думи в текста. Поради възможността от възникване на грешки при изпълнението на заявки или транзакции, средата предлага средство за информиране на потребителя, относно настъпилата грешка. Примерен екран е показан на следващата фигура.



Фигура 23.

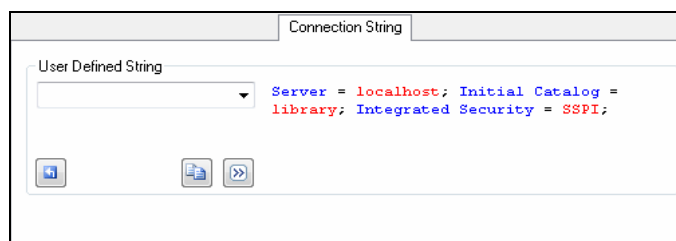
Извежда се информация за типа на командата, датата и часът на изпълнението ѝ, източникът на грешката, както и краткото нейно описание. Потребителят отново има на разположение опции за записване на описанието и търсене на съвпадащи низове. При изпълнение на XML заявки резултатът се съхранява в таб XML Reader (фиг. 24).



Фигура 24.

Потребителят има възможност да запише файла със съдържанието на XML заявката в .xml или .xsd файлови формати.

Низът за свързване със сървъра е важна част от процеса за създаване и работа с база от данни. На фиг. 25 и фиг. 26 са показани примери относно задаването му.



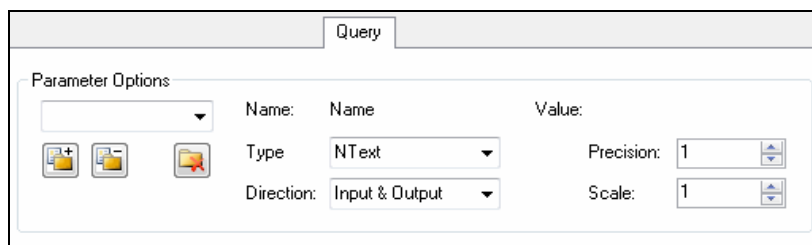
Фигура 25.

Падащият списък **User Defined String** съдържа всички запазени думи при дефинирането на свързващ низ. Потребителят, който няма голям опит или не познава добре синтаксиса може последователно да избира ключови думи и да им задава стойности в текстовото поле. Резултатът е синтактично обработен и се визуализира в друго текстово поле отстрани. Има опции, които копират символния низ в клип-борда на Windows. На фиг. 26 е показан друг подход за дефинирането му. В този случай се предполага, че потребителят познава добре структурата на низа и знае значението на необходимите му запазени думи.



Фигура 26.

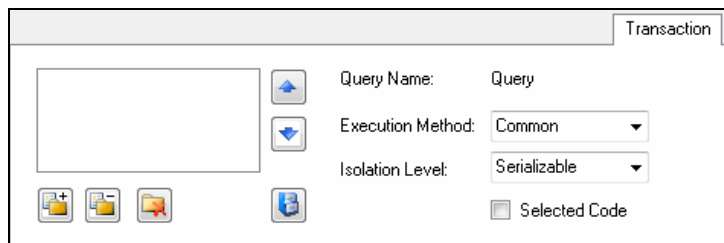
Задаването на аргументи е от голямо значение при работа с параметрични заявки. Програмата предлага опции, чрез които се посочват различните свойства на отделните параметри. Такива могат да бъдат името на параметъра, типът на стойностите, които той може да приема, точността и стойността му. На фиг. 27 е показан екранът при задаването на тези свойства.



Фигура 27.

Множеството от параметри се съхранява в динамичен масив, като към него могат да се добавят нови и да се премахват съществуващи параметри.

Работата с транзакции, както и техните аргументи, са представени в таб на име **Transaction**. Чрез него потребителят може да създаде самостоятелна транзакция от различни заявки, да зададе логическо име на всяка една от тях, да определи типа и поредността им на изпълнение. Важно е да се отбележи възможността за указване на нивото на изолация на транзакцията.



Фигура 28.

При последователното изграждане на една транзакция е възможно да възникне момент, в който е необходима промяна на някоя от заявките, участващи в колекцията. Бутонът Extract се използва за извличане на заявка и показването ѝ в редактора на програмен код, което позволява да се направят нужните промени.

Последният таб от секцията с опционални инструменти предоставя функционалност на календар. Тази услуга е полезна поради честата нужда от работа с дати в базите от данни. На фиг. 29 е показано изображение на календара в представяната среда.

юни 2011 г.						
пон	вт	ср	четв	пет	съб	нед
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

Фигура 29.

6. Развитие на средата

Програмата SQL Flat представлява развойна среда за работа с релационни бази от данни. Инструментите, предлагани от нея, предоставят на потребителя разнообразни техники и похвати в процеса на работа. Програмата представлява автономен продукт, чрез който може да се администрира сървър с бази от данни, могат да се създават и редактират такива бази от данни.

Съществува широк набор от възможности за развитие на продукта. Възможно е добавянето на допълнителни функции, които да улесняват работата на потребителя. Такива функции могат да бъдат предварително дефинирани шаблони на различни често използвани команди като заявки, съхранени процедури, тригери, изгледи и др. Това значително ще съкрати времето за създаване на програмния SQL код, както и ще помогне на начинаещите потребители. Внедряване на система за графично проектиране на таблици и взаимовръзки между тях също представлява перспектива за развитие на средата. Съществува възможност за добавяне на допълнителна функционалност, която да позволи свързването към бази от данни, различни от SQL Server. Такива могат да бъдат източници на данни като Oracle, OLE DB, ODBC и други, което значително разширява сферите на приложение на средата. В допълнение би могло да са добави модул, който да следи за атаки от тип „SQL Injection“.

Доразвиването на програмната среда в обучаваща предоставя възможност на потребителя за експериментиране и решаване на проблемни ситуации, които помагат за изпълнение на задачата и самонаблюдаване на обучението.

Всяка обучаваща система поддържа в различна степен трите основни модела: модел на предметната област, педагогически модел и потребителски модел. Освен това се осигуряват възможности за тяхното непрекъснато разширяване, надграждане и усъвършенстване. Предпоставки, които налагат използването на такива среди за решаване на различните задачи, са:

- нарастването на обема на данните, които трябва да се поддържат и обработват и усложняването на структурата на съхраняваната информация;
- развиването на големи, интегрирани бази от данни и получаването в резултат на много данни, но малко информация за аналитика;
- недостигът на време и ресурси за анализиране на всички връзки, които могат да са информативни;
- откриването на закономерности в данните, които могат да служат за разбиране и прогнозиране на дейностите на обучаемите.

Тенденцията е към създаване и развитие на съвременни обучаващи среди като адаптивни системи, поддържащи високо ниво на интерактивност и колаборативност. Интересът на младите хора към новото, тяхното любопитство и комуникативност могат успешно да се използват в процеса на тяхното обучение.

Литература

- [1] Азълов, П., *Бази от данни – релационен и обектен подход*, Техника, 1991.
- [2] Наков, С., колектив, *Програмиране за .Net Framework*, Изд. „Българска асоциация на разработчиците на софтуер” София, 2004.
- [3] Пенева, Ю. *Бази от данни*, Изд. „Регалия 6” София, 2004.
- [4] Тодоров, Р. *Развойна среда за релационни бази от данни*, Дипломна работа, специалност ИКН, Бургаски свободен университет, 2011.
- [5] ADO.NET Architecture
[http://msdn.microsoft.com/en-us/library/27y4ybxw\(v=vs.80\).aspx](http://msdn.microsoft.com/en-us/library/27y4ybxw(v=vs.80).aspx)
- [6] Connolly, T., C. Begg. *Database Systems*, 7th Ed., Addison-Wesley, 2000.
- [7] Date, C. J. *An Introduction to Database Systems*, 7th Ed., Addison-Wesley, 2000.
- [8] Elmasri, R., S. Navathe. *Fundamentals of Database Systems*. Addison-Wesley, 2000.
- [9] SQL-92 Standart
<http://www.contrib.andrew.cmu.edu/~shadow/sql/sql1992.txt>
- [10] Шриприя, К. С. *Microsoft Visual C++ .NET Професионални проекти*, София, Duo Дизайн, 2003.