STUDIA INFORMATICA
Nr 1/2(7) Systemy i technologie informacyjne 2006

Efficiency of Technology of Access to Databases

Andrzej Barczak[1], Artur Mościcki[1]
[1] Institute of Computer Science, University of Podlasie,
ul. Sienkiewicza 51, 08-110 Siedlce, Poland

Abstract: In this paper the evolution of technologis of access to databases is presented. For definite sysytems of databases' management (databases servers) and technologis of access, there was made an experiment, enabling to estimate the avarage of time of answering (realizeing), definite types of questions- what means eficiency of each technologis.

Keywords: databases, ODBC, BDE, ADO

1. Evolution of database access technology

When in early 90's database systems like *dBase, Paradox, Clipper* and *Fox Pro* became popular, there was a need of creating mechanism of access to data kept in different bases from level of different languages. It is maine reason of creating *Query By Example (QBE)* – first technology of access to databases. *Query By Example* is relatively friendly for user, made by IBM, used in many informatic systems, technic of creating questions for databases. It consists in filling empty record with looked for sequence of signs suitable for records' structure in database for example: "Brussel" in field "**City"** or "Nowak" in field **"Name".** The result of question is list of all records conteining definite sequence of signs in field.

Query By Example makes an convertion of user's question into formal database's question. Due to this user can realize even sophisticated questions to database without knoulage of formal methods. Craeting questions in *Paradox* using *QBE* containes of folowing stages:

- choice of database's charts, wich are connected with questions,
- defineing conditions of linking charts(for questions embraceing more then one chart),
- choice of atrybutes belonging to outcome chart (in standard Answer Chart),
- specification of criterions of data selection (optionaly),
- specification of calculation made on data from chart (optionaly),
- installation of questions,
- conservation questions (optionaly) in Answer Chart.

Additionaly we can choose setings of the Answer chart, for example to save it on diferent name. Answer chart is temprorar answer table, which is saved by

Paradox in user's private catalog. By loading of question, table is overloaded, and by shouting down of *Paradox* it is dilited. To save the table, you only have to change it's name after question. It is also possible to set extrotion of Answer chart's name before loading question in a maine dialog window of tha table.

After *Query By Example* there was created many mechanisms of access to databases such as: *DAO (Data Access Object), OLE.DB (Object Linking and Embedding DataBase)* or *ODBC (Objects DataBase Connectivity)*.

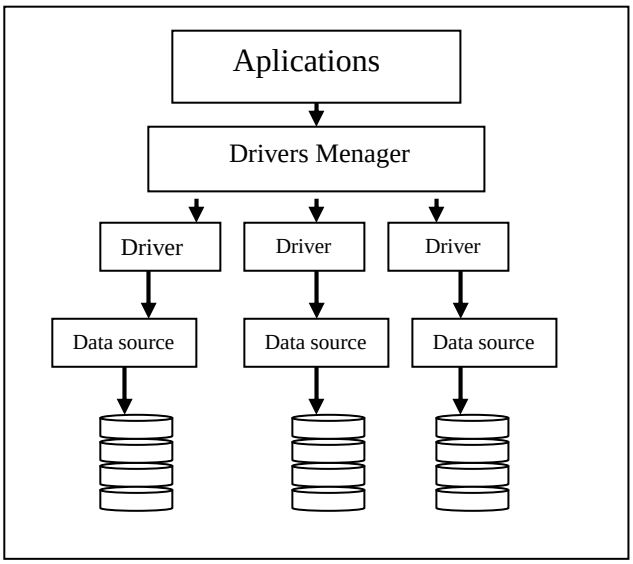
One of the first mechanisms of access to databases, which was competition for *Query By Example*, was ***DAO technology.*** That is the technology, which as first one used objects of access to bases. It consists of definite sequence of objects, making it able to operate small databases like dBase, MS Access end so on in Windows operating system.

Next technology was ***ODBC*** (*Open DataBase Connectivity*). ODBC is an interface anabling programs to conect with databases' managing system. This is an API interface (*Application Programming Interface*) independent from programing language, operating system and database. This standart was inroduced in September 1992 by *SQL Access Group* as an alternative for *Query By Example* and *DAO* and is used till nowadeys. Representatives of some companys producing both software and hardware, were working for *SQL Access Group (SAG)* on defineing universal method of access to data, for purpuse of facilitating client–server software. Microsoft used results of SAG's work for creating so called *call-level interface,* named *Open DataBase Connectivity (ODBC).*

ODBC defines low leveled sequence of functions, anabling client's and server's aplications change of data and passing instruction without having information about client's and server's implementation. It concerns any instructions made in client-server aplications, even when client and server works on diferent computers or software/hardware platforms.

Architecture of ODBC interface consists of 4 elements (fig. 1):

- ***aplication*** making specifickactivity of processing using SQL questions to get and store data indespencible for process;
- ***driver manager*** – in form of DLL library (*Dynamic Link Library*), which has task of making aveliable aplications of defined database driver;
- ***driver*** usualy in form of DLL library, called out by driver manager element making function of ODBC interface. It also sent to data sources SQL demants, and outcomes to aplication. If it is needed, driver can modyfie being done SQL questions in order to adapt them to character of destination data's source (for example adapting SQL dialect to defined databases' managing system.)
- ***data source*** – moust often system of databases' menagment.



Picture 1. ODBC architecture

Wille ODBC interface was being created, in Microsoft there was made an conception of COM (*Component Object Model),* which can be used in every programic enviroment of Win32. The result of create of COM opjects was contol with dll and OLE.DB[1] technology.

OLE.DB is Microsofts's mechanism are used not only to get acess to SQL base but also any ather data sources. Aplications can use OLE.DB to direct access to data, or by OLE.DB can call out ODBC, to get access to database by ODBC.

OLE.DB is very dificult to implementate, that is why technologies of higher level, although they are using OLE.DB to conect to databases have been created. Their implimentation has been simplifaiced. Example of such a technology is ADO (*ActiveX Data Objects*). In OLE.DB like in DAO clases, objects and methots of conecting to databases has been used. OLE.DB is still being inmproved , that is why now it is one of key technology.

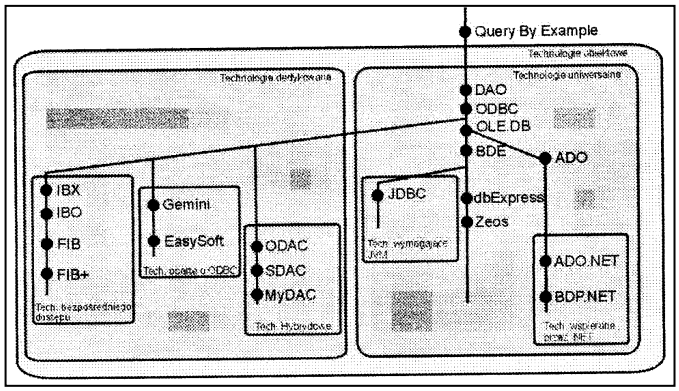
After success of OLE.DB and ODBC enother technology of databases' access came into being. However the bigest boom of access technology began when enviroments ***Borland Delphi*** and ***Borland C++ Builder*** ocured on the market. For tis enviroments there was created many specific technologis of access to databases: ***IBX*** (*InterBase Express),* ***IBO*** (*InterBase Objects),* ***FIB*** (*Free InterBase),* ***ODAC*** (*OracleDataAccessComponents),* ***SDAC*** (*SQL Server Data Components*), ***MyDAC*** *(MySQL Data Access Components),* ***Gemini, EasySoft, BDP:NET*** (*Borland Data*

[1] Boduch A. *Delphi 7. Ćwiczenia zaawansowane*. Helion Publisher, Gliwice 2003

Provider for .NET), ***BDE*** (*Borland Data Engine)*. Also Microsoft and Sun created technology of access for their programistic tools. That is origin of ***ADO*** and ***JDBC*** *(Java DataBase Connectivity)*. ADO technology was adopted also by ather firms, that is why we can use ADO for example in Borland Delphi.

Picture 2 presents evolution of access to databases with takeing into account characteristic fitures each technologis. We schould notice that with the developmet of technologis of access to databases their device and specification ocure.

There exists, and probably will ocure new, universal technologis based on objective conception, enabling access to diferent databases. However through process of evolution technologis developed which must be turned on on virtual machine or in turning on enviroment (.NET Framework). Thanks to that their implementation and posibility of moveing to an ather computer is much easier.



Picture 2. Evolution of technologie of access to databases

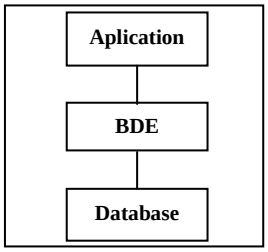
When Borland Delphi ocured, technologis dedicated to one database server started to be constructed. Among them there are such a technologis as:

- technology of direct access to databases,
- technologis based on ODBC mechanisms,
- hybryd technologis, which besides specific mechanisms of access to databases can also use ODBC sources.

After some time it was observed that ODBC and OLE.DB are not efficient enough, and new mechanism of access – ***BDE*** *(Borland DataBase Engine)* was created. BDE was created by cooperation of Borland, Microsoft, IBM, HP and Oracle. After it ocured, it was basic technology of access, nowadeys it is displaced by athers like ADO or dbExpress. The advantages of BDE in comper with ODBC is higher speed and easy service, whreas the disadvantage is limited ability of movement. It is a result of a fact that BDE dose not serve databases' managing system on it's oun, but use indeirect program-

SQL Links. Nowadeys Borland has resined from this technology, but it is possible to use it for conecting aplications with databases.

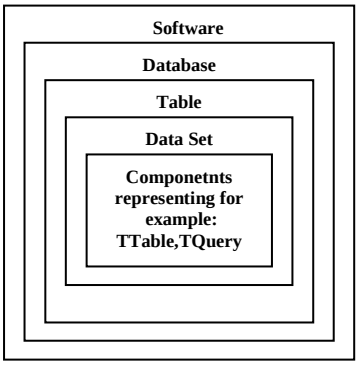
BDE in compare with presented technologis, is a seqence of DLL librarys and tools mediaiting in conecting to databases. By dint of that there is no need of manpulating on functions of defined database, suffice only option accessale in BDE. Picture 3 showes dependence beetwin aplication and BDE.



Picture 3. Relations between aplication and DBE mechanism

In BDE conception of ***database engine*** apeared first time. Every databases managing system have got own API interface comunicating with database. Rearly or even never happens that two computers has got the same interface. Due to BDE we do not have to know detailes of implementation of such databases – conection with for example Oracle or InterBase is the same. Special drivers in form of SQL Links' programs make translation of questions BDE for questions specific for particular databases managing system. Due to such a solution aplication can conect with any database without knowing detailes of implementation.

Also in BDE oun records' bufor in form of dataset ocured. Definition of set was formulated as structure of colums (group data of the same type) and lines, which are intermediarys beetwen compoent of data service (in Delphi this is for example component of TTable class) and visual components, which are used to present content of chart (DBGrid). Instructions of levels of BDE are presented on picture 4.



Picture 4. Relations between levels in BDE

Definition of database was formuled in BDE as: special catalog in which enother objects like charts can egsist.

Unfortunately BDE is not ideal solution. It was projected, when simple database system based on flat files like Paradox or dBase was dominaiting. Nowadeys relational databases, served by SQL language, cooperating with software in architecture client-server are used most often.Instaling BDE on rented internet servers is very often imposile becous of server mansging firm's justifable fears connected with instalisation of system services on their server. Dispaite updaiting BDE (for instance ading service of such mechnisms like object-relation model), a lot of fitures of this engine is steal attached to their paradox's roots. Good example is the way of maping numeric fields for data's types. In BDE this activity creates problems of agreement at cooperation of Delphi code with SQL servers.

Nowadeys BDE technology is fitted to small local projects, where security and cohesion of data is not importand.

Successes of ODBC and BDE made Microsoft to elaborate new mechanism of access to databases, which is easier to implementate then OLE.DB. This is origin of technology of access to databases coled ***ADO*** (*ActiveX Data Object).* ADO like OLE.db is based on COM objects (*Component Object Model*).

Idea, wich coused creating of ADO, assumed access to database without knowings its inside structure. However diference between BDE and ADO is significant. BDE is ordinary component, whereas ADO is indirect layer and technology.

Service of all common capabilities means establischment of the lowest common denominator for all databases, becous ADO assume the same service for databases' sytems (with diferent funcionality).

Like I mentioned ADO is an indeirect lier. This technology was created to facilitate usieng OLE.DB. Apart from access to databases, by ADO you can gain access also to Excel's files, Lotus' files, HTML and more data sources, what is innovative and very usefull idea.

With apperance of .NET platform, Microsoft introduced to ADO some changes. Nowadeys the "new" ***ADO.NET*** is accessable. It could be divaiced for:

- system of access to data,
- system storing data.

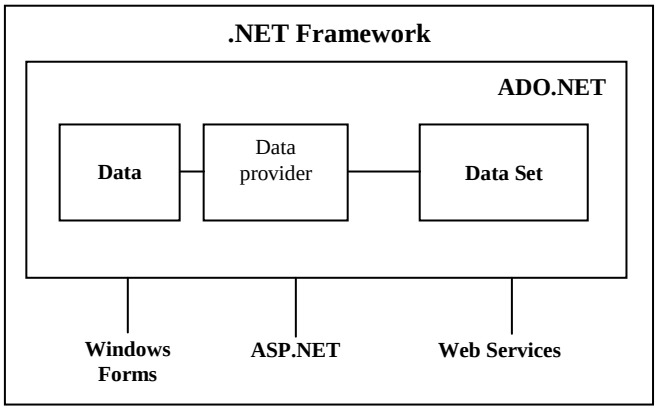
All interfaces characteristic for .NET ware placed in space of *System.Data* names. Classes connected with storing data are DataSet, DataTable, DataRow and DataColumn. In contrast with to mechanisms of storing data, which are in classes, access to data take place with use of inerface. There are 5 mechanisms of access to data, which are in the space, in the .NET:

- System.Data.SQLClient,
- System.Data.OleDb,
- System.Data.ODBC,
- System.Data.SQLServerCE,

– System.Data.OracleClient.[2]

Simplificed ADO.NET architecture whose maine componentsare: DatSet and Data Provider is presented on picture 5. Source of data can be in physical data base or XML file. Data Provider conects and send orders, wheraes DataSet represents data in memory.

Data source is most often database – local or puted in the server, rarely - XML file, Excel Lotus and so on ADO.NET deliver components, which makes operation on server, send SQL questions and resive results of actions.



Picture 5. Architecture ADO.NET

Data provider is equivalent of driver, whose task is conecting with database or reading proper file, hideing detailes of implementation.Ado.NET provides service of MS SQl Server databases, Oracle and OLE.DB technology. However Borland is makieng avialable ancillary mechanism – Borland Data Provider ***(BDP)***, which enable easy conection with anccilar data sources for instance InterBase databases or DB2.

DataSet in ADO.NET is mnemonic image of data, tables and relations. ADO.NET workes with data in procedure of disconected access to databases. That means that all operation on data are not physicly refflected in table – user hase only such a impression. To reall update data, aditional functions schould be programmed.

In ADO data conections between data sources and transporting applications ware in form of records set (RecordSet), regardless of fact that they came from one table or more. In ADO.NET records set hes been changed into data set. Data which are in data set, are inclouded into tables (*DataTable*). If aplication redemand from ADO.NET data from two tables, this data are given back in for of two tables, which

[2] Vide: Boduch A.: *Delphi 8.NET. Kompendium programisty.* Helion Publisher, Gliwice 2004.

are in data set. Relations between tables wich are in data set, are represented by instance of *DataRelation* classe[3].

Except from presented differences between ADO and ADO.NET there is one more feature differentiate this two technologis – platform .NET Framework which is reqiered to activate ADO.NET aplication. .NET Framework is activating enviroment reqiered to actvate in Microsoft Windows system programs created in .NET technology (for example in C# language, Borland Delphi 2005, VB.NET). It means, that .NET Framework is something like Java virtual machine. However there is essential difference between virtual machine and activaiting enviroment.: virtual machine is used to activate Java aplications on a lot off system's platform, whereas .NET Framework is used to activate aplications written in many languages but on one platform MS Windows. .NET Framework implicates comon data's types,varriables and mechanisms of access, which can be used in different languages addapted to .NET.

I have mentioned before about BDP mechanism simplifying connection with data sources. When ADO.NET appeared, new sources providing data for .NET ware created. They were caled *Borland Data Provider for .NET (BDP.NET). BDP.NET* is package of components corresponding with CLX (*Component Library X-Platform for Cross Platform)*, broadening functionality of ADO.NET and seemplifying using databases (for example InterBase) in .NET.

BDP.NET is code managied by .NET, but it was produced by Borland. It couses the need of adding some components to a system, on which aplication will be activated. In case of using ADO.NET there is not such a need – library System.Data.dll is in Net Framework. However in case of Delphi, BDP.NET it is proper and obligate reason, if there is need of connecting with InterBase or DB2 using ADO.NET.

Success of ADO technology coused craeting by Borland, new and much better then BDE technology of access to databases. That is origin of ***dbExpress technology***, which is avialable both in Delphi and in Kylix. This is fast and much more flexible technology than BDE and ODBC. This technology uses efficient drivers, that is why it provides one of the fastest accesses to information storred in database. Specialy made components of this group have got universal character, they are avialable also for Linux platform. However this universalism prejuges about rather poor reportoire of abillities of manipulaiting data.

The bases of dbExpress architecture are drivers for different systems of databases managment. Every of this drivers implementates set of interfaces annabling access to data specific for server. Cooperation between aplication and this drivers are organised by components of DataCLX group, which function similar to BDE components, but are divested of useless ballast.

[3] See Boduch A.: *Delphi8.NER. Kompedium programisty*. Helion Publisher, Gliwice 2004.

One of most importand featers of dbExpress architecture (not ocuring in any ather technology), making it so eficient, is one directed character of data sets. Resualts of questions are stored in one directed courcores. Such a coursores make it able to move from one record to the next one, but, disable to come back to arterior.This feature couse that it is impossible to use DBGird components for presenting data. However such a treatment couse that all the operations connected with data procesing are made much faster.[4]

Onedirectional character of data sets prejudge about luck of bufering records, which is usefull onely by dobledirectional navigation and data processing.

Enother limitation is luck of abillity to use Last() and Prior() methods of TdataSet class. From navigating methods only First() and Next() can be used. It is also imposible to modyfie data (becouse of luck of editing bufors). However ther is possibility to edit records using enother components (TclientDataSet, TSQLClientDataSet). Onedirectional data sets do not realize filtering – as mechanism refering to many records, it would need polirecord bufering. Onedirectional data sources do not service reviev fields. It means that there are many restrictions connected with this technology, however due to them the efficiency of access to data (speed) is much higher then in ODBC.[5]

On the countrary then enother technologis, dbExpress dose not spend server's resources for need of questions connected with metadata, or enother aditional orders, duering realizong user's reqirements. dbExpress also does not need so big clients resources as BDE, becouse onedirectional sets don't need cacheing data. In dbExpress definitions of metadata are proccesed by interfaces implementated in DLL library. dbExpress dose not generate aditional questions connected for instance with : navigateing through data set or reading fields storeing data of BLOB type. Only questions generated by user's aplication get into server. This not only results in biger efficiency of makin aplications, but also makes proces of creating aplication easier.[6]

Enother very importand advantage of dbExpress technology is usefulness for Delphi and for Kylix.This technology uses CLX components, so after compilaiting by Kylix compilator, aplication usieng can be activated in Linux. Access to interplatform databases systems, such as Oracle, MySQL or InterBase is becoming possible.

Using BDE, ADO or dbExpress we are using engine independent from server. It is possible to switch server used by aplication, but it is not easy. If it is needed the aplication use only one databases server, we should take care that databases engine service defined data API server, which will be used. It couse that programs will not be moved to enother database server, but indeirect layer is avoided, we make by funkcionality, and on speed. Most often we do not create such an API, but we look

[4] Mościki A., Kruk I.: *Oracle 10g i Delphi. Programowanie baz danych.* Helion Publisher, Gliwice 2006.
[5] www.ai.komisauto.pl
[6] jw.

for components, which will reconstruct API and suit to Delphi and to architecture and its class. Great example of such components is InterBase Express (IBX). Aplications created with use of this technology work beter and faster, what is more they make it possible to make most of featers of specific server – InterBase or Firebird – becouse only to this databases servers we can geine access with use of IBX[7]. IBX is not technology, whose aime was to replace or rival with ather technology. It was created to, cooperate with defined database – InterBase.

Using InterBase Experess we avoid indeirect layer in form of for example BDE and we communicate straight with client software InterBase or Firebird. IBX components avialable in two types of Delphi, are fully created in Delphi and are specialaised for InterBase and Firebird servers. It has got a lot of advantages.

Due to removal indeirect layer like for example BDE, between aplication and InterBase server, coating for executing questions is minimalised and time of comunication with server became extremely short.

IBX takes diferent solution then ather technologis – it is specialised technologie dedicated only for InterBase and Firebird. Due to this it can use this InterBase's functions, which are rear in diferent servers.]

Furthermore in IBX ocured many classes unknown before . Most importand of them – this, which dose not have their equivalents in technologis presented before are:

- TIBExtract – classe needed, when we have to generate SQL code, which will be used to create objects in database. By objects of this classe, we can gane access to codes creating systemic objects in databases;
- TIBSQLVar – classe used to to annable access to fields as a result of question made by user;
- TIBInputRawFile – classe anabling reading fles in such format as outside files of InterBase databases are saved;
- TIBConfingService – classe anabling seting databases paramets;
- TIBBackupService – classe anabling creating spare copies of database;
- TIBRestoreService – classe annabling rerun database on base of spare copie;
- TIBValidationalService – classe anabling checking properitin of databases and used to dealing with problems with transactions;
- TBInstall, TIBIUninstall – classes anabling installing and uninstalling InterBase server.

Becouse components and classes of IBX are written wholy in Delphi, they are compiled in exe files. There is no need off dystibuteing a lot of DLL files, like in BDE. Becouse aplication only containes of fille, which could be done, there is no need of craeting sohisticated installing programes. There is also no need of threat that somoene could distroy configuration (like in BDE) – everything is under control of IBX.

[7] Cantu M. *Delhi 6*. MIKOM Publisher, Warszawa 2002 .

2 Technologies of access to databases of XXI century

After sukces of InterBase database and IBX tecnology, a lot of firms started to think about creating their own technology (of indirect access), enabling access to InterBase server. Such a technology would be alternative for IBX or BDE. In such circumstances at the beginning of XXI century two new technologis: FIB and IBO only anabling access to InterBase.

Technologies IBX, FIB and IBO are very symilar to each other. They differenitate with names of components, features and classes, which we have to use, to connect with databases. Another difference is efficiency. However except of this, there are not many difrerences between this technologois. In FIB and IBO there are not any classes, which does not have their equivalents in IBX technology. That is why FIB and IBO are not revolutionry, they are just another technologis, which are used to work with InterBase.

Second group of technologis dedicated to definite server are technologis, which are descended from ODBC. Very good examples of such technologis are Gemini and EasySoft. However this brench of evolution of technology of access to databases are becoming less importand.

After suckes of IBX, Core Lab Software Development decided to creat technology of access to databases dedicated to ather server than InterBase. This is origin of ODAC, SDAC and MyDAC technologis. Due to being hybryd technologis, they are revolutionary. It means that, they have oun mecanisms of access to databases, but they can also use ODBC sources defined in operating system.

IN ODAC, SDACK and MyDACK we can find some classes, which did not occurred in technologis presented before. Among new classes, the most importand are:

- TOraSmart (SmartQuery) – classe which is alternative for ToraQuery,
- TOraSQL – classe used to executingPL/SQL instructions, syntax procedures and so on,
- ToraNestedTable – classe used to sevice nset tables,
- ToraScript – classe used to execute sequences,
- ToraPackage – classe used to execute PL/SQL packages,
- ToraAlert – classe anabling sending data between sesions,
- TOraLoader – classe anabling fast loading data to databases,
- ToraErrorHandler – classe used to translating error reports,
- TBDESession – classe anabling integrating ODAC components with apliactions created in BDE technology,
- TConnectDialog – classe used to storing user's name, pasword and database's name,
- TVirtualTable – classe used to storing data set in memory.

With occurrence of, new, mobile, independent from platform and software's architecture programing language – Java, apeared ideas of creating new database interface, which would fullfil similar conditions to language's ones. Natural

consequence of such a decision was detachment from existing API technologis, especialy ODBC, and creating new technology, fully created in Java language. However, it does not mean that programists who had been using ODBC, had to change for copletely diferent course of thinging during creating database aplications. Quite the opposite – JDBC functionaly is similar to ODBC technology. Only abilities of interface had been increased and it had been adapted to Java's specific character.

Nowadeys JDBC is not only used directly, as unified technology of access to any database, but also as base for aplications/interfaces of higher level, which anable access to databases from higher level. Examples of such tasks are for example:

- SQLJ – mechanism characterizeied by "hiden" in Java code SQL language. This code is processed to get out from him proper SQL comends, executed by JDBC;
- Java Blend – mechanism anabling direct translation of table of relational database into Java's objects.

JDBC define level of agreement with SQL standart. However the maine guideline says that, every JDBC driver, must answer minimum ANSI SQL–92 version of SQL standart. Moreover, authors of interface ade enother guidelines, which grand followings vercions of JDBC standard (discused downwardes). Due to this guidelines SQL questions are sent to proper DBMS, regardless of ability to realize tham. When definet DBMS can not service order sent by JDBC, aplication generate define exception. Furthermore JDBC give access to information about DBMS and its characteristic features (settings, abilities). This information is sent to user as so caled ***metadata.***

JDBC driver is set of complicated classes (so caled Java's baitcode), which implementates all interfaces from java.sql pacage and impliments againe remaining classes. In many cases this classes are written only in Java. Implementation directly depends on databases management system, with which aplication using drivers is conecting. That is why every JDBC driver is used to communicate with definet database and can not be used in case of enother producents' databases.

Despite presented solution is simillar to techologis used in C++ or Delphi, in Java is becomeing importand. Java is in many cases is mobile language, drivers; producent have to creat only one package for every thersions of JDBC (in exception of JDBC-ODBC breages and Native-API partly-Java). In compare with mentioned interfaces for C ++ language, we use one binarium both in Linux and Windows.

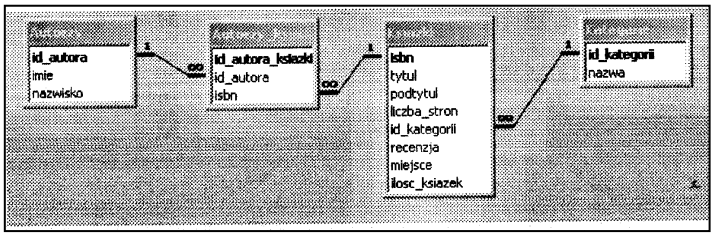
JDBC drivers craeted by individual producentsare in many cases free, and are complements of JDBC package. They can be downloaded from DBMS delivers' pages or from Sun's archives. However not ale drivers are free. Part of them is not spread in comertial way.

New reqirements, new databases systems and new programing languages couse creating new technologis of access to databases. That is why, we should expect, that in near feature new revolutionize technology will have been created.

Certainly, small companys, will be steal creating, new, les knowen, usually dedicated to one server technologis of access to databases.

3 Subject and range of researches

For needs of efficiency reserches, simple database wes created. Its structure is presented on picture 6.



Picture 6. Relations between databases

Data base containes of 4 tables: **Books, Categories, Authors, Authors~books.** The maine table, in this database, is **Book** table, which store informatine about ISBN numer, subtitels, numbers of pages, numer of cathegories, to which book belongs to, reviews, place of stoareg and number of examples stored in definet place. This table was used for research of eficiency of access technology.

Software method measureing time of makieng SQL questions was created in Borland Delphi enviroment to check efficiency of technology. This technology was used to measure efficiency of followieng technologies: ADO, ODBC, BDE, dbExpress, IBX, IBO, FIB, ODAC, SDAC, MyDAC, EasySoft and Gemini. We should notice that researches were made in MS Windows, which is not system of real time, that is why time of executieng question might be diferent on diferent computers and depends on many factors: the occupation of server, fast of server, amount of operational memory, loade of system and a lot more (table 1 presents software and hardware platform on which research was made). Number of actions trigered in the system also has influence on time of executieng questions, that is why dureing test all usseles proceses has been closed. Due to this differences between times of executting the same processes were very small, maximum vallue achieved only 0,01 seconds.

Table 1. Parameters of software and hardware on which aplications were tested

Hardware parameters	Software parameters
Processor: AMD Duron 800 MHz	OS: Windows XP Professional + SP 2
RAM: 384 MB DIMM	
HDD: Samsung 80 GB, 4 MB Cache	

Presented methode of measering time, make it able to ascertain, which from tested technologies of access to databases are more or less eficient. Due to coed showen below, we can measure eficiency of executeing questions of SELECT, INSERT, UP'DATE, DELETE type.

```
var
    Freq, TimeStart, TimeEnd : Int64;
    czas : double;
begin
    …
    if QueryPerformanceFrequency(Freq) then
    begin
        QueryPerformanceCounter(TimeStart);
            // czynność, której czas wykonania jest mierzony
            QueryPerformanceCounter(TimeEnd) ;
            czas := (TimeEnd-TimeStart)/Freq;
            RichEdit.Lines.Add ( 'Wykonanie zapytania SELECT zajęło:'
                                    +FloatToStr(czas)+' sekundy') ;
    end;
end;
```

This method is the most exact of few methodes of measing time, becouse it is based on tacts of procesor. At first frequency of processor is defined (line 6), than the value of profesor's tacts are readed before (line 8) and after executeing process. Next (line 11) time of executeing process is measered due to formula differnence of tactes divaised for frequency of processor. At laste time of executeing question is projecting in RichEdit controle.

Presented methode was used to measure time of executeing seven SQL questions(INSERT, DELETE, UPDATE and SELECT type), on every database. Two extream values were discarded and remaining 5 were used to measure the everage of time needed for executeing question of definet type.

Table 2 presents examples of questions, which was executed on databases and, whose eficency was measered. In first column of table ther are namesof questions used in next subections. In further part of article instead of hole name of question for exmple SELECT * FROM Books, name SELECT ocures.

Table 2. Examples of questions made on „Books" table, whiches eficiency was measered

Sign of the question	**Contents of the quetion**	**Description**
SELECT	SELECT *FROM Booksi	All data from ksiazki table are downloaded
INSERT	INSERT INTO books VALUES (""+ Editisbn.Text+"", '"+Edittitel.Text+"','"Editsubtitel.Text+"', '"+Editnumberofpages.Text+"', '"+DBLookupComboBoxidkategorii.Firld.As.String+"', '"+Memoreview.Text+"', '"+Editplaceofstorage.Text+"', '"+Editnumberofcopies.Text+')'	The new record is put, in which data are loweded from editoring fields and unrolleing lists.
Delete	DELETE FROM books WHERE isbn='"+isbn+"'	The record marked by user in DbGrid net is deleted.
UPDATE	UPDATE books SET isbn ='"Editisbnl.Text+"' titel='"+Edittitel.Text+"', subtitel='"+Editsubtitell.Text+"', number_of_pages='+Editnumberofpages.Text+', id_kategorii='+DBLookupComboBoxidkategorii.Field.AsString+', review='"+Memoreview.Text+"', place='"+Editplaceofstorage.Text+"', number_of_copies='+Editnumberofcopies.Text+' WHERE isbn='"+isbn+"'	All fields of of record, which was marked by user according to values inserted by user to text fields and unroling lists of form, are set.

Programist, author of database aplication, who wary about its efficiency, has to tke int consider both technological eficiency and database eficiency. In table 3 eficiency of researched technologis cooperating with databases is presented.

Table 3. Efficiency of chosen technolgis of databases menaging systems and technologis of access to databases

Databases system	Technology	Everage of processing questions in seconds (place in calssification)			
		SELECT	INSERT	LPDATE	DELETE
Oracle	ADO	0,0148 (20)	0,0103 (20)	0,0096 (18)	0,0102 (24)
	dbExpress	**0,0051 (5)**	0,0093 (19)	0,0076 (15)	0,0079 (21)
	ODAC	0,0063 (11)	0,0072 (16)	0,0072 (14)	0,0072 (18)
	BDE	0,0096 (17)	0,0069 (15)	0,0082 (16)	0,0079 (21)
	ODBC	0,0094(16)	0,0241 (25)	0,0249 (25)	0,0130 (25)
	EasySoft	0,0249 (23)	0,0317 (26)	0,0285 (26)	0,0196 (26)
InterBase	IBX	**0,0051 (5)**	**0,0015 (2)**	**0,0012 (1)**	**0,0008 (1)**
	ADO	0,0342 (25)	0,0029 (8)	0,0039 (9)	0,0024 (10)
	dbExpress	**0,0020 (2)**	**0,0017 (3)**	**0,0019 (3)**	**0,0015 (3)**
	BDE	0,0076 (14)	0,0023 (6)	**0,0022 (5)**	0,0019 (6)
	FIB	0,0058 (8)	**0,0011 (1)**	**0,0013 (2)**	**0,0008 (1)**
	IBO	0,0125 (18)	0,0035 (10)	0,0040 (10)	0,0020 (7)
	Gemini	0,0081 (15)	0,0028 (7)	0,0024 (6)	0,0020 (7)
SQL Server 2000	ADO	0,0163 (21)	0,0092 (18)	0,0132 (22)	0,0086 (23)
	SDAC	0,0059 (10)	0,0033 (9)	0,0029 (8)	0,0023 (9)
	dbExpress	**0,0022 (3)**	0,0052 (13)	0,0083 (17)	0,0060 (17)
	ODBC	0,0058 (8)	0,0051 (12)	0,0048 (12)	0,0050 (15)
DB2	ADO	0,0273 (24)	0,0087 (17)	0,0118 (21)	0,0074 (19)
	dbExpress	**0,0016 (1)**	0,0064 (14)	0,0098 (19)	0,0051 (16)
	ODBC	0,0072 (12)	0,0105 (21)	0,0114 (20)	0,0075 (20)
MySQL	MyDac	**0,0044 (4)**	**0,0016 (3)**	**0.0019 (3)**	**0,0018 (4)**
	ODBC	0,0072 (12)	**0,0019 (5)**	0,0024 (6)	**0,0018 (4)**
Access	ADO	0,0208 (22)	0,0039 (11)	0,0060 (13)	0,0047 (14)
	ODBC	0,0054 (7)	0,0141 (23)	0,0045 (11)	0,0035 (11)
Informix	ADO	0,0374 (26)	0,0144 (24)	0,0167 (24)	0,0041 (13)
	ODBC	0,0136 (19)	0,0115 (22)	0,0138 (23)	0,0035 (11)

4 Analise of resultes

By dint of resultes from table 5 we can answer a question, what database system and what technology of access to data should we use, to creat the most eficient aplication. Five of the most eficient technologis used to executeing diferent types of SQL questions are marked in table 5 by bolde. Table 5 presents, that the best solution for people who, wary about eficency of aplication, is chooseing InterBase server and IBX, FIB, dbExpress technology or MySQL server and MyDAC technology. Table confirms the opinion that, most eficient are technologys dedicated to one server (IBX, FIB and MyDAC service only one server of databases). It also confirms fact that InterBase and MySQL are the most eficients servers of databases. We should remember that InterBase is database server written in Delphi (due to this all layers of access to data like ODBC are omited), and dbExpress was created by Borland (producer of Delphi). That is why another proposal cames: **the bigest eficiency you get by using products and technologis of one comppany.**

Remaining tested technologis were created in ather languages, which use indirect leyers or are designed for less efiecence databases systems. It is a reason of advantage in fast executeing SQL instructions in IBX and FIB technologise. Acording to authores, it will be dificult to improve efiecence of IBX and FIB technologis. In respect of eficiency they are the best technologis working on the best server – InterBase.

Mentioned dbExpress technology is the most efficient technology from technologis analing access to many servers. db Express is technology of access to databases recommended by Borland. Times of executeing SQL questions received by this technology are realy very good, and times of executeing question of SELECT type are the best from all technologis.

Technology ADO fail in respect of eficiency. In case of ADO, modern mechsanisms created by Microsoft are enabling

easy way of executeing SQL instructions form level of few programing languages, for example from level of Delphi. However times of executeing questions are repeadly worse than times of executeing the same questions by ODBC technology.

Programist creating database technology very often has dylema, what technology of access to databases he should use. Should he use polyplatforme technology or monoplatform, serviceing a lo of database systems, or dedicated to one system?

There is no unambigious answer for this question . If he care about eficiency and speed of action and amplication is designed for MS Windows, the good solution is dbExpress technology and IBX, FIB, ODAC, SDAC, MyDAC technologies, which are dedicated to one database server that is why they are so eficient.

Bibliography

1. Bodueh **A.:** Delphi 7. Ćwiczenia zaawansowane. Helion Publisher, Gliwice 2003.
2. Boduch **A.:** Delphi 8 .NET. Kompendium programisty. Helion Publisher, Gliwice 2004.
3. Cantu M.: Delphi 6. MIKOM Publisher, Warszawa 2002.
4. Mościcki A., Kruk I.: Oracle 10g i Delphi. Programowanie baz danych. Helion Publisher, Gliwice 2006.
5. www.ai.komisauto.pl
6. www.openlinksw.com
7. www.wikipedia.pl