

DATU BĀZES TEHNOLOĢIJU PIEEJAMĪBAS ATTĪSTĪBAS ANALĪZE AUGSTĀKĀ LĪMEŅĀ VALODĀS

Abstract

Datu bāzes ir aptvērušas praktiski visu nozaru programmatūras. Lielākajā daļā programmatūrās tiek izmantotas datu bāzes. Nopietna problēma ir programmaprodukta mijiedarbība ar datu bāzēm. Šim mērķim tiek pielietotas dažādas tehnoloģijas un standarti, tādas kā ODBC, OLE DB, ADO.NET, LINQ, Hibernate, JDBC, Nhibernate, ORM u.c. Programmaprodukta izstrādei, apmācību mērķiem ir interesanti salīdzināt un izpētīt datu bāzes piekļuves tehnoloģijas tendences.

Darbā tiek analizētas un salīdzinātas vairākas datu bāzes piekļuves tehnoloģijas pie to datiem. Lielākā uzmanība tiek pievērsta jaunākām piekļuves tehnoloģijām, to attīstības analīzei, kā arī salīdzināšanai. Ar tehnoloģiju salīdzināšanu tiek iegūtas pārskatāmākas katras tehnoloģijas priekšrocības un trūkumi. Izvērtējot tehnoloģijas ir jāņem vērā arī kritēriji, kas noteiktā situācijā spēj palīdzēt izvēlēties kādu no esošām datu bāzes pieejas tehnoloģijām. Ir izveidots informatīvais mācību resurss (vietne), kā palīgmateriāls apmācībā.

Atslēgas vārdi: datu bāze, pieejamības tehnoloģijas, attīstības analīze, pieejamības tehnoloģiju attīstības tendences.

Ievads

Mūsdienu dzīve nav iedomājama bez efektīvas pārvaldes. Svarīga nozīme ir informācijas apstrādes sistēmām, kas lielā mērā ir atkarīga no efektivitātes jebkuram uzņēmumam vai iestādei. Šādai sistēmai vajadzētu:

- Nodrošināt kopējo un / vai sīki detalizēto darba ziņojumu saņemšanu par darba rezultātiem;
- Ļauj viegli identificēt tendences galveno rādītāju noteikšanu;
- Nodrošināt informācijas saņemšanu, bez būtiskas kavēšanās;
- Veikt precīzu un pilnu datu analīzi.

Informācijas nesējs tradicionālā veidā ir papīrs. Reālā uzņēmumā, protams, eksistē mapītes ar dokumentiem. Lai piekļūtu pie vajadzīgās informācijas, ir jāatver kāda no atvilktnēm, no tās jāizņem attiecīgā mapīte un tajā jāatrod konkrētais dokuments. Informācija uz papīra parasti tiek izvadīta datorizdrukas veidā. Meklēšana un apkopošana bieži vien tiek veikta, pārkopējot materiālus no dažādām izdrukas lappusēm uz vienas. Datu analīzei un kārtējā pārskata izveidošanai palīgā ņem kalkulatoru vai elektronisko tabulu. Datorizētā datu bāze nav nekas cits kā automatizēta datu aizpildīšanas, uzglabāšanas un apskatīšanas sistēma. Datorizētās datu bāzēs informācija glabājas noteiktā formātā. Šādās datu bāzēs var uzglabāt jebkura veida informāciju – no vienkāršas teksta (piemēram, adrese, tālruņa numurs u.tml.) līdz sarežģītai datu struktūrai (zīmējumi, skaņa un video). Šos datus var vienkārši apskatīt un izdrukāt, izmantojot tādas datu bāzu objektus, kā, piemēram, formas un pārskati.

Tēmas aktualitāti nosaka vairāki apsvērumi un galvenie no tiem :

- viegli, ērti pielietojama
- ietaupa laiku citu uzdevumu veikšanai
- plaši izmantot dažādu nozaru uzņēmumos un iestādēs

Šī darba izstrādāšanas metode balstās uz literatūras analīzi, kā arī uz statistisko datu apkopojumu un praktisko analīzes atspoguļojumu informatīvajā vietnē.

Datu bāzes Pieejas Tehnoloģiju attīstības analīze

Datu bāzes pieejas tehnoloģijas ir izstrādātas, lai nodrošinātu vienotu, standartizētu valodu un pārvades ceļu starp pakalpojumu sniedzēja (jebkura veida) datiem un patērētāju (pasūtītājs, lietotājs). Taču gadu gaitā, dažas tehnoloģijas bija vairāk veiksmīgas savos uzdevumos kā citas. Šajā rakstā gribētu iepazīstināt datu bāzu tehnoloģijām, lai palīdzētu attīstīt un izmantot šīs tehnoloģijas.

Ausmas laikmetā datubāzes izstrādātājiem ir pietiekami zināt tikai tās datu bāzes, kuras viņi izmanto. Tomēr, datu bāzes un to tehnoloģijas ir attīstījušās diezgan strauji - no relāciju datu bāzēm līdz ne relāciju informāciju glabātuvēm, tādām kā elektroniskais pasts un failu sistēmas. Datu bāzu pilnveidošana, iet kopsolī ar straujajām pārmaiņām tehnikā. Un ar parādīšanos klient-serveru un daudzlīmeņu arhitektūras izstrādātājiem, nākas tikt galā ar visām daudzlīmeņu datu bāzu tehnoloģijām. Vairums izstrādātāju patērēja gadus, lai izzinātu kādu no datu bāzes pieejas tehnoloģijām. Izsekojot datu bāzu tehnoloģiju attīstību, vienkāršāk ir izvēlēties piemērotāko tehnoloģiju un optimizēt to saviem mērķiem, uzdevumiem. Organizēt piekļuvi pie datu bāzēm mūsdienu programmēšanas valodā nesastāda nekādu grūtību. Un pašas programmēšanas valodas izstrādātāji visvairāk novērtē pēc tipa un iebūvētām iespējām piekļuvei pie datu bāzēm, ērtībai un saskarnes pilnībai.

ODBC

Microsoft Atvērto savienojumu datu bāzes (ODBC). ODBC - zema līmeņa, augstas veiktspējas saskarne neobjektīvā C valodā, kas ļauj pielikumiem iegūt pieeju dažāda veida datu bāzu vadības sistēmām un kurš ir izstrādāts priekš relāciju datu noliktuvēm. Pielikums būs neatkarīgs no jebkuras DBVS, no kuras viņš iegūst pieeju datiem. Pielikuma lietotāji var pievienot, kā programmnodrošinājuma komponentu draiveri, kas mijiedarbojas ar saskarni starp pielikumiem un noteiktām DBVS. Neatkarība realizējas ar starpposmu bibliotēku palīdzību, kas iekļauj sevī specifisku kodu dotajai DBVS. Vecs nozares standarts, relāciju datu pārvaldīšanai, izmantojot SQL vaicājumu sintaksi caur neizmērāmiem datu avotiem. ODBC tehnoloģija nodrošina papildus līmeņa izveidi starp pielikumu un izmantojamo DBVS.

ODBC pakalpojumi nodrošina informācijas izvēles saņemšanu no vaicājuma pielikumiem un tulko to adresējamās datu bāzes kodola valodā, lai varētu piekļūt pie glabājamās tajā informācijas. Pamatnozīme ODBC ir abstrahējamam pielikumam sastāvā, atkarībā no serveru datu bāzes kodola īpatnībām ar kuru tas īsteno mijiedarbību. ODBC tehnoloģiju viens no plusiem ir pielikuma izstrādes vienkāršība, ar augsta līmeņa abstraktā interfeisa pieeju pie praktiski jebkura tipa svarīgākajiem DBVS datiem. Galvenais ODBC tehnoloģiju trūkums ir saistīts ar vaicājumu translācijas nepieciešamību, kas samazina piekļuves ātrumu pie datiem. (Li)

OLEDB

Mūsdienu uzņēmējdarbības standarts piekļuvei pie datu bāzēm. Ja datu bāzēm nav OLE DB pakalpojuma sniedzēja, tad OLE DB pakalpojumu sniedzējs izmanto ODBC draiverus, lai parādītu kaut vai kādu ODBC datu avotu OLE DB lietotājam. OLE DB eksistē tamdēļ, lai atvieglotu piekļuvi datu bāzēm, failu sistēmām, ziņu glabātuvēm, direktoriju pakalpojumiem, kā arī darba dokumentāciju. *OLE DB* pamatojas uz *ODBC* veiksmi, nodrošina atvērto standartu tam, lai tas iegūtu piekļuvi pie visa veida datiem. Atšķirībā no ODBC, bet neuzliek noteiktus ierobežojumus vaicājumu sintaksei vai dotām datu struktūrām. Šai tehnoloģijai ir objektorientēta saskarne dažāda veida programmēšanas valodām, savietojamām ar *COM*. OLE DB tehnoloģija ir būvēta uz ODBC pamata un paplašina to līdz komponentu arhitektūrai, kura nodrošina ar augsta līmeņa saskarni piekļuvei pie datiem. (Li)

ADO

ADO sastāv no šiem vairākiem augsta līmeņa objektiem kā savienojums, ieraksts, komanda, kļūda, lauks. Tā nodrošina augstas klases programmējamo modeli, kurš tiks atbalstīts un uzlabots ar laiku. Kaut tas un nenodrošina tādas iespējas, kā kodēšana pa tiešo OLE DB vai ODBC. ADO ir vienkārša apmācībā un izmantošanā un var tikt izsaukta no tādām scenāriju valodām. (Андрей Колесов, 1999)

ADO.NET

ADO.NET tas ir revolucionārs uzlabojums salīdzinājumā ar tradicionālo ADO pie sadalīto pielikumu izveides. ADO.NET klašu virkne, kuras nodrošina pakalpojumus programmētājam piekļuvei pie datiem .NET. ADO.NET nodrošina funkcionalitāti izstrādātājiem, rakstošiem vadības kodu līdzīgu funkcionalitātei, piešķirtai saviem izstrādātājiem COM ADO. ADO.NET nodrošina nepretrunīgu piekļuvi pie datu avotiem,

tādiem kā SQL serveris, kā arī datu avotiem, piešķirti caur OLE DB un XML. Produkts ir ar augsta līmeņa saskarnes lietojumprogrammu programmēšana, kura ir domāta vāji savienotiem, daudz ķēžu pielikumiem, strādājošiem internetā un atbalstoši bezvadu piekļuve pie datu bāzēm. Lietotāja pielikumi kopā ar datu izmantošanu, var izmantot ADO.NET, lai savienotos ar tā datu avotiem un iegūtu, apstrādātu un atjaunotu datus, kurus tie satur. ADO.NET atdala piekļuvi pie datiem no datu manipulācijas diskrētajās komponentēs, kuras var tik izmantotas atsevišķi vai tandēmā. ADO.NET iekļauj datu pakalpojuma sniedzēja platformu .NET, lai savienotos ar datu bāzi, izpildot komandas un iegūstot rezultātu. (Microsoft, 2011)

LINQ

Integrētā vaicājumvaloda. Kam sākumā jābūt SQL- standartizēta pieprasījuma valoda dažādiem datu avotiem. Microsoft projekts, kurš pievieno sintaksi vaicājumvalodā, atgādinošu SQL, programmēšanas valodu platformā .NET Framework . Jēdzienu kopumu, kurus ievieš LINQ, sākotnēji pārbaudīja pētnieciskajā projektā Microsoft C# . LINQ iekļauj sevī standarta šablonus vaicājumu izveidei un datu atjaunošanai. Tehnoloģija var būt paplašināta potenciāli jebkura tipa datu noliktuvju atbalstam. Visual Studio 2008 ietver sevī LINQ piegādātāju salikumu, kas ļauj izmantot LINQ ar platformas .NET Framework kolekcijām, SQL Server datu bāzēm, ADO.NET datu kopumiem un XML dokumentiem. LINQ ir paplašinātais valodu kopums, atbalstošs datu veidā vaicājumu formēšanu, drošam pēc tipiem. Pieprasāmie dati var būt attēloti XML formā (LINQ vaicājumi pie XML), datu bāzēm (ADO.NET ar LINQ atbalstu, kurā ieiet LINQ to SQL, LINQ to DataSet, LINQ to Entities), objektiem (LINQ to Objects) utt. (Microsoft, LINQ: .NET Language-Integrated Query, 2011)

NHibernate

ORM risinājums portēšanai no *Java* uz *Microsoft.NET* platformu. Tā ir atvērtā koda bezmaksas bibliotēka, kura tiek izplatīta tāpat kā Hibernate ar *GNU Lesser General Public License* licenci. *NHibernate* ļauj attēlot biznesa loģikas objektus relāciju datu bāzēm. Pēc uzdotā *XML* apraksta jēgu un savienojumus. *NHibernate* automātiski izveido SQL vaicājumus ielādei un objektu saglabāšanai. *NHibernate* ir .NET ports *Hibernate* bibliotēkas populārā Java platformā. *NHibernate* arī nodrošina datu vaicājumu un izguves līdzekļus. *NHibernate* ģenerē SQL komandas un atbrīvo izstrādātāju no manuālas datu kopas apstrādes, saglabājot pielikumu pārnesamu lielākajai daļai SQL datu bāzēs, ar datu bāzes pārnesamības varbūtību ļoti maziem ražīguma zaudējumiem. (Community, *NHibernate*)

Hibernate

ORM ir Java valodas Hibernate tehnoloģija, kura ne tikai rūpējas par savienojumu starp Java klasēm un datu bāzu tabulām. Tā arī sniedz līdzekļus automātiskai vaicājumu veidošanai un datu iegūšanai un var zināmi samazināt izstrādes laiku, kurš parasti tiek tērēts uz SQL un JDBC koda aprakstīšanu ar rokām. Hibernate ģenerē SQL izsaukumus un atbrīvo izstrādātāju no liekas rezultāta iegūtas datu komplektus apstrādes un objektu konvertēšanas, saglabājot pielikumu portējamu visās SQL datu bāzēs. Hibernate ir Java valodas bibliotēka, domāta objektrelāciju projicēšanas uzdevumu risināšanai (*object-relational mapping* - ORM). Tā ir bezmaksas programmatūras nodrošinājums ar atvērto kodu (*open source*), kuru izplata *GNU Lesser General Public License* nosacījumos. Dotā bibliotēka dot viegli izmantot karkasu (*framework*), lai varētu attēlot objektorientētu datu modeli tradicionālās relāciju datu bāzes. Izstrādātājs var izmantot Hibernate kā klašu un tabulu sistēmas projektēšanas procesā „no nulles”, tā arī dara ar jau eksistējošo datu bāzi. Hibernate risina ne tikai uzdevumus Java klašu savienojumos ar datu bāzes tabulām (un Java datu tipiem ar SQL datu tipiem), bet arī piešķir līdzekļus automātiskai ģenerācijai un tabulu kopu atjaunošanai. Vaicājumu izveidi un iegūto datu apstrāde var zīmīgi samazināt izstrādes laiku, kas parasti tiek tērēts uz SQL un JDBC koda aprakstīšanu. Hibernate automatizē SQL vaicājumu ģenerāciju un atbrīvo izstrādātāju no liekas rezultātā iegūtās datu komplektiem apstrādes un objektu pārveidošanas, maksimāli atvieglojot pielikumu pārnesei (portēšanu) uz jebkurām SQL datu bāzēm. (Community, Hibernate)

ORM

ORM ir programmēšanas tehnoloģija, kura saista datu bāzi ar objektorientēto programmēšanas valodu koncepciju, izveidojot „virtuālo objektu datu bāzi”. Eksistē komerciālā un brīvi pieejamās šīs tehnoloģijas realizācija. Objektorientētā programmēšanā objekti programmā tiek attēloti kā objekti no reālās pasaules (piem. adrešu grāmatiņa). Problēmas jēga ir objektu pārveidošana formā, kurās tie var būt tikt saglabāti failos vai datu bāzēs un kuras viegli var tikt iegūtas ar īpašību saglabāšanu un attiecības starp tiem. Šie objekti tiek saukti „glabājamie”(*persistent*). Vēsturiski eksistē vairākas pieejas šī uzdevuma risināšanai. Ir izstrādātas vairākas paketes, risinošas nepieciešamību objektu pārveidošanā glabāšanai relāciju datu bāzēs. Dažas paketes risina šo problēmu, dodot iespēju klašu bibliotēkām, spējošām izpildīt tādas pārveidošanas automātiski. (Wikimedia Foundation, 2011)

Darba analīzes attēlojums ir viens no jaunākiem informācijas iegūšanas veidiem. Darbs attīsta iztēli un konceptuālo uztveres spēju. Precīzas konstrukcijas un interaktīvas lietotnes ļauj intuitīvi uztvert uzdevumā iekļauto ideju. Individuālais darbs ar kādu no mācību resursiem nostiprina teorētiski apgūtās likumsakarības. Dažādu uzdevumu risinājumu kombinēšanai ar vispārīgām metodēm strukturē domāšanu. Īpaši izstrādāti mācību moduļi sniedz precīzu izpratni par datu bāzes likumiem un to pielietojumiem. Mācību līdzekļi viegli un ātri palīdz iegūt jaunu informāciju, kā arī to izmantot apgūstot dažādus kursus, šodien dod lielu efektu, jo elektroniskie mācību līdzekļi sniedz jau gatavu rezultātu. Jāakcentē arī tas, ka mūsdienās praktiski viss ir datorizēts, un tas arī atvieglo mācību procesu un ne tikai to. Un tas, savukārt, ir svarīgi ne tikai studentiem, skolēniem, bet arī pasniedzējiem.

Bibliogrāfija

1. Community, J. (bez datuma). *Hibernate*. Ielādēts 2010. gada 28. 10 no Relational Persistence for Java & .NET: <http://www.hibernate.org/>
2. Community, J. (bez datuma). *NHibernate*. Ielādēts 2010. gada 26. 10 no Relational Persistence for Java & .NET: <http://community.jboss.org/wiki/NHibernateForNET>
3. Li, W.-M. (bez datuma). *Эволюция технологий доступа к данным*. Ielādēts 2010. gada 13. 10 no <http://masters.donntu.edu.ua/2006/fvti/zubritskaya/library/articles/3.htm>
4. Microsoft. (2011). *ADO.NET*. Ielādēts 2011. gada 20. 1 no MSDN Library: <http://msdn.microsoft.com/en-us/library/h43ks021.aspx>
5. Microsoft. (2011). *LINQ: .NET Language-Integrated Query*. Ielādēts 2011. gada 25. 1 no MSDN Library: <http://msdn.microsoft.com/en-us/library/bb308959.aspx>
6. Wikimedia Foundation, I. (2011. gada 26. 1). *ORM*. Ielādēts 2011. gada 1. 2 no Wikipedia: <http://ru.wikipedia.org/wiki/ORM>
7. Андрей Колесов, О. П. (1999). *DAO, ADO, RDO...* Ielādēts 2010. gada 1. 9 no доступ к данным с помощью технологии ADO: <http://www.visual.2000.ru/develop/ms-vb/cp9907-2/msado1-1.htm#5>