

Napredne baze podataka

Spojnost



•API funkcije

- Da bi aplikacija komunicirala s bazom podataka, potrebno je određeno "sredstvo". To sredstvo su **API funkcije** za pristup bazama podataka.
- Sama baza podataka nije ništa više nego "skladište" koje sadrži odgovarajuću reprezentaciju, odnosno apstrakciju procesa iz stvarnog života.
- Da bi podaci koji se čuvaju u bazi postali informacija od ikakvog značaja, potrebna je aplikacija koja zna kako transformirati te podatke u oblik koji krajnji korisnik želi vidjeti.
- Sredstvo pomoću kojeg će aplikacija "pričati" s bazom podataka naziva se **database API**.

•API funkcije

- Prema kronološkom redoslijedu njihovog pojavljivanja najbitniji API-ji su:
 - Open Database Connectivity (**ODBC**),
 - Object Linking and Embedding Database (**OLE DB**),
 - Java Database Connectivity (**JDBC**) i
 - ActiveX Data Objects (**ADO**).
- Autori svih ovih API-ja su Microsoft i JavaSoft.
 - Microsoft koristi i termin Universal Data Access (UDA) kao krovni termin koji označava ODBC, OLE DB, ADO i Remote Data Service (RDS)

•API funkcije

ODBC

- ODBC kao najstariji de jure standard ujedno je i najrašireniji.
- Microsoft i JavaSoft prilikom izrade novijih standarda vodili su računa o njihovoj kompatibilnosti s ODBC.
- Novi API-ji imaju objekte ili komponente koje putem **ODBC adaptera** pristupaju SQL bazama podataka.
- Microsoft strategija budući je pomak prema OLE DB i ADO tehnologijama, dok JavaSoft želi pomak prema Javi i JDBC standardima.

• API funkcije

ODBC

- Aplikacije za upravljanje bazama podataka *pozivaju funkcije u ODBC sučelju*, a ODBC putem svojih upravljačkih programa za baze podataka aplikaciji *vraća podatke iz baze*
- Ovi upravljački programi analogni su upravljačkim programima za pisače:
 - Kada npr. program za obradu teksta nešto ispisuje, sve naredbe za ispis upućuju se operacijskom sustavu koji upotrebljava odgovarajući upravljački program za željeni pisac.
 - Isto tako, kada aplikacija za obradu podataka želi pristupiti bazi, ona ODBC-u upućuje zahtjev za podacima, a ODBC koristi odgovarajući upravljački program za pristup toj bazi podataka.

•API funkcije

OLE DB

- OLE DB je, u neku ruku, nova verzija ODBC-a.
- OLE DB nadograđuje ODBC tako što na jednako uniforman način na koji to radi ODBC s relacijskim bazama, omogućava *pristup heterogenim izvorima* podataka i vrši njihovu prividnu homogenizaciju, pojednostavljujući tako rad s podacima.
- Microsoft strateški vodi OLE DB kao novu tehnologiju za pristup svim vrstama podataka i kao takav će u budućnosti u potpunosti potisnuti ODBC.

•API funkcije

JDBC

- JDBC je ekvivalent ODBC tehnologije namijenjen upotrebi prilikom razvoja aplikacija u Java programskom jeziku.
- Koristi sličnu arhitekturu zasnovanu na API pozivima i upravljačkim programima za razne vrste sustava za upravljanje bazama podataka (DBMS).
- Sun (kreator Java platforme) je u JDBC ubacio i takozvani **JDBC-ODBC "most"**
 - specijalni upravljački program koji aplikaciji koja koristi JDBC omogućava pristup ODBC izvorima podataka
- Budući da je većina baza podataka u trenutku pojave JDBC već imala gotove ODBC upravljačke programe, ovo je značilo da Java aplikacije mogu pristupati svim takvim bazama, iako JDBC upravljački programi za njih nisu (bili) dostupni.

•Vrste JDBC drivera

- **Type 1: *JDBC-ODBC bridge plus ODBC driver***
- Prvo prevodi JDBC API pozive u Microsoft ODBC pozive te ih predaje ODBC driveru. ODBC *driver* specifičan za operacijski sustav mora biti instaliran na računalu koje koristi ovaj tip JDBC-a.
- Performanse su slabe zbog dodatne transformacije poziva.
- Ne podržava sve mogućnosti JDBC sučelja jer ga ograničavaju mogućnosti ODBC *driver*-a.
- Ovaj tip *drivera* važan je jer omogućuje korištenje JDBC-a sa svakim SUBP-om za kojeg postoji ODBC driver, a za kojeg nije, ili nije još, razvijen JDBC *driver* tipa 3 ili 4.

•Vrste JDBC drivera

- **Type 2: *Native-API, partly Java driver***
 - Prevodi JDBC API pozive u pozive klijent programskih sučelja koja su specifična za SUBP (*DBMS-specific client API calls*: Oracle API, Informix API itd. pozivi).
 - Kao i u slučaju *drivera* tipa 1 i ovdje se zahtijeva instalacija izvršnog programskog koda (npr. Informix CLI) koji je specifičan za operacijski sustav. Performance su bolje nego za tip 1.
- **Type 3: *JDBC-Net pure Java driver***
 - Transformira JDBC pozive u DBMS nezavisne pozive koji se predaju srednjem sloju (*middle-tier server*). Srednji sloj transformira dobivene pozive u pozive specifične za pojedini DBMS.
 - Nedostatak se uglavnom svodi na to da proizvođač srednjeg sloja mora osigurati mogućnost rada sa svakim DBMS-om posebno.

•Vrste JDBC drivera

- **Type 4: *Native-protocol, pure-Java driver***
- Transformira JDBC pozive direktno u pozive koje koristi specifični DBMS. Java klijent i DBMS komuniciraju direktno. *Driver* je optimiziran za rad sa specifičnim SUBP. Za svaki SUBP potreban je poseban *driver*.
- Zaključak:
- Uporaba *drivera* kategorije 3 ili 4 je preporučena. Kategorije 1 i 2 su međurješenja koja se mogu koristiti u nedostatku *drivera* tipa 3 ili 4. Kategorije 3 i 4 podržavaju važne prednosti Java tehnologije, npr. "automatsku instalaciju": *download drivera* zajedno s *appletom* koji ga koristi.

•API funkcije

DAO

- Prvi objektni model je DAO (Data Access Objects), koji je sastavni dio VisualBasica.
- Originalno zamišljen kao set objekata za pristup MS Access bazama podataka preko MS Jet sustava za upravljanje podacima
 - Jet je prvi Microsoft DBMS na kojem je zasnovan Access
- DAO također omogućava pristup ODBC izvorima podataka.
- DAO je lakši za korištenje od standardnih ODBC API funkcija, ali ne pruža isti nivo kontrole kao ODBC API

•API funkcije

ADO

- ADO je nova tehnologija iz Microsofta čiji je cilj zamijeniti DAO kao standardni objektni model za pristup bazama podataka.
- Programiranje pomoću ADO modela ne pruža toliko kontrole kao direktan rad s OLE DB pozitivima, ali je ADO dizajniran tako da je broj slojeva između aplikacija i baze podataka sveden na apsolutni minimum, što mu daje znatno bolje performanse.
- Prednost ADO modela zasnovanost je na ActiveX tehnologiji, što ga čini upotrebljivim u svim jezicima koji podržavaju tehnologiju, uključujući i skriptne jezike poput VBScripta i JavaScripta, što OLE DB ne može postići.

• Baze podataka na web stranicama

- Danas su najčešće dvije tehnologije koje se koriste za razvoj dinamičkih web stranica:
 - ASP / MS SQL
 - PHP / MySQL
- Osnovna razlika je u platformi servera:
 - Microsoft server
 - Linux (UNIX) server

• Windows vs. LINUX

ASP vs. PHP

- Iako se za razvoj web aplikacija mogu koristiti mnogi skriptni i programski jezici (Perl, Python, C), većina programera odlučuje se na PHP ili ASP.
- ASP će kvalitetno raditi jedino na Windows platformi, iako postoje Apache moduli koji ASP-u donose podršku i za Unix.
- ASP stranice najčešće su napisane u VBscriptu, što je uobičajeni, ali ne i jedini način pisanja ASP aplikacija.
- S druge strane, PHP postoji i za Windows, čija je funkcionalnost donekle smanjena (neke funkcije, doduše rijetke, radit će samo na Unix/Linux verziji Apache web servera).

•Koja tehnologija?

- Za učenje PHP-a, Apache za Windowse ili PHP modul za PWS (Personal Web Server) ili IIS (Internet Information Server) poslužit će sasvim dobro.
- Programeri s iskustvom u Visual Basicu, ASP doslovno neće morati učiti.
- Programeri s iskustvom u C-u ili Perlu, lakše će savladati PHP.

•Spajanje na bazu podataka

ASP

- **DSN (Data Source Name) metoda spajanja na bazu u ASP:**
 - Set baza = Server.CreateObject ("ADODB.Connection")
 - Baza.Open "DSN=naziv_dsn; UID=username;PWD=password"
- **DSN-less metoda spajanja na Accessovu bazu podataka u ASP-u:**
 - Set baza = Server.CreateObject ("ADODB.Connection")
 - Baza.Open "DRIVER={Microsoft Access Driver (*.mdb) }; DBQ=" & Server.MapPath ("naziv_baze.mdb") & "; UID=username; PWD=password"

•Spajanje na bazu podataka

ASP

- **DSN-less metoda spajanja na Microsoft SQL bazu podataka u ASP-u:**
 - Set baza = Server.CreateObject ("ADODB.Connection")
 - Baza.Open "DRIVER={SQL Server}; Server=ime_servera; Database=ime_baze; UID=username; PWD=password;
- **OLEDB metoda spajanja na Access bazu podataka u ASP-u:**
 - Set baza = Server.CreateObject ("ADODB.Connection")
 - Baza.Open "Provider=Microsoft.Jet.OLEDB.4.0; Data Source=" & Server.MapPath ("naziv_baze.mdb") & "; User ID=admin;Password=;"

•Spajanje na bazu podataka

PHP

- **DSN metoda spajanja na bazu podataka u PHP-u:**
 - `$baza = odbc_connect ('SYSTEM_DSN' , 'username' , 'password');`
- **DSN-less metoda spajanja na Access bazu podataka iz PHP-a (Windows):**
 - `$baza = new COM ("ADODB.Connection");`
 - `$konekcija = "DRIVER= {Microsoft AccesscDriver (*.mdb) }; ";`
 - `$konekcija`
`.="DBQ="baza_podataka.mdb;uid=username:pwd=password;";`
 - `$baza->open ($konekcija);`
- **DSN-less metoda spajanja na MySQL vezu preko ASP-a:**
 - `Konekcija = "driver={MySQL}; server=host; uid=username`
`pwd=password;database=baza"`
 - `Set baza = Server.CreateObject ("ADODB.Connection")`
 - `Baza.open Konekcija`

•Spajanje na bazu – MySQL-PHP

- PHP osigurava funkcije za specifični rad s MySQL - počinju uvijek s *mysql_*
- Spajanje na MySQL bazu iz **mysql_**PHP-a (Unix):
 - `$link = mysql_connect ("host", "username", "password")` or die ("Neuspjesno spajanje na server");
 - `Mysql_select_db ("baza",$link)`

• Postavljanje upita PHP-Mysql

- Funkcija **mysql_query** koristi se za postavljanje upita
- `resource mysql_query (string $query [, resource $link_identifier])`
- `$upit=mysql_query("SELECT * FROM racun");`
- Ista funkcija koristi se i za INSERT i UPDATE te zadavanje naredbi MySQL poslužitelju tipa SET CHARACTER ...
- *\$upit* vraća resurse dobivene naredbom - to može biti set rezultata ili jedan rezultat
- **Mysql_error()** vraća poruku greške ako se dogodila, a ako se greška nije dogodila, `mysql_error()` neće vratiti ništa.

• Dohvat podataka

- **Mysql_num_rows**

- Vraća broj rezultat (n-torki) koje je vratila naredba zadana s `mysql_query`

- **Mysql_result**

- Dohvaća rezultat: zadaje se broj n-torke i naziv atributa
- Može se još dodatno rezultat prebaciti u polje (broj dimenzija odgovara dimenziji rezultata)

•Primjer

- Pronaći sve zapise u tablici *račun* kod kojih je saldo veći od 100. Tim zapisima postaviti saldo na 0.

```
$link1=mysql_connect($baza_ip,$baza_login,$baza_pass);
mysql_query("SET CHARACTER SET 'cp1250'", $link1);
mysql_selectdb($baza);
$obav=mysql_query("SELECT * FROM racun;");
if(mysql_num_rows($obav)!=0){
    for($i=0;$i<mysql_num_rows($obav);$i++){
        $saldo=mysql_result($obav,$i,"saldo");
        if($saldo>100) {
            $b=mysql_result($obav,$i,"br_rac");
            $obav1=mysql_query("UPDATE racun SET saldo=0 WHERE br_rac=".$b);
        }
    }
}
```

•Primjer 2

- Ubacivanje datoteke u bazu podataka: prvi.php
- `echo "<form action=drugi.php method=post enctype=multipart/form-data>";`
- `echo "Ubaci datoteku:<INPUT type=hidden name=MAX_FILE_SIZE value=25000000 />";`
- `echo "<INPUT class=logiraj size=30 type=file name=objekt />";`
- `echo '<INPUT class=logiraj type=submit value="Izvrši akciju">';`
- `echo "</form>";`

• Primjer 2

- Drugi.php // data u tablici je tipa blob
- //ubaci opis u bazu
- `$file_tip= $_FILES["objekt"]["type"];`
- `$file_vel=$_FILES["objekt"]["size"];`
- `$file_name=$_FILES["objekt"]["name"];`
- `$data=addslashes(file_get_contents($_FILES["objekt"]["tmp_name"]));`
- `$sql = "INSERT INTO tbl_files ";`
- `$sql .= "VALUES ('$data', ";`
- `$sql .= "'$file_name', '$file_vel', '$file_tip')";`
- `$result = mysql_query($sql, $link1);`
- `$nesto=mysql_error();`

• Primjer 2

- Čitanje iz baze bloba
- `$result = "SELECT * FROM tbl_files WHERE ...";`
- `$data = mysql_result($result, 0, "bin_data");`
- `$name = str_replace("", "_", mysql_result($result, 0, "filename"));`
- `$size = mysql_result($result, 0, "filesize");`
- `$type = mysql_result($result, 0, "filetype");`
- `header("Content-type: $type");`
- `header("Content-length: $size");`
- `header("Content-Disposition: inline; filename=$name");`
- `header("Content-Description: PHP Generated Data");`
- `echo $data;`

•VB6

- Spajanje na bazu

Function load()

On Error GoTo ja

```
conn = "Driver=MySQL ODBC 3.51 Driver;Server=" + ipbaze +  
";UID=login;PWD=lozinka;database=razredbeni;"
```

```
Set baza = New ADODB.Connection
```

```
baza.Open conn
```

```
pitaj sql, "SET CHARACTER SET 'cp1250'"
```

```
Exit Function
```

ja:

```
MsgBox "Nemate dozvolu za spoj na bazu razredbeni ili ne radi mreža"
```

```
End
```

```
End Function
```

•VB6

- Postavljanje upita

```
Public Function pitaj(s As ADODB.Recordset, koko As String)
```

```
    Set s = New ADODB.Recordset
```

```
    s.CursorLocation = adUseClient
```

```
    s.CursorType = adOpenKeyset 'adOpenDynamic
```

```
    s.LockType = adLockOptimistic
```

```
    s.Open koko, baza, , , adCmdText
```

```
End Function
```

```
Public Function pitaj_blob(s As ADODB.Recordset, koko As String)
```

```
    Set s = New ADODB.Recordset
```

```
    s.Open koko, baza, adOpenStatic, adLockOptimistic
```

```
End Function
```

•VB6

- Otvaranje baze

```
Public baza As ADODB.Connection  
Load
```

- Postavljanje upita

```
Public sql1,sql2 As ADODB.Recordset  
Pitaj sql1, "SELECT * FROM racun);  
FOR i =1 to sql1.recordcount  
    If sql1("saldo")>100 THEN  
        Pitaj sql2, "UPDATE racun SET saldo=0 WHERE br=" & str(sql1("br"))  
    ENDIF  
    sql1.movenext  
NEXT i
```