

Programiranje 6

Rad s bazama podataka II. (1/2)

1. Korištenje programskog koda za dizajn i izmjenu sadržaja baze podataka

Potrebno je izraditi aplikaciju koja će otvoriti novu tablicu u bazi podataka HR.mdb pod nazivom email_adrese, te će za svim zaposlenicima dodijeliti email adrese oblika *iprezime@tvrka.hr* gdje je *i* prvo slovo imena a *prezime* prezime zaposlenika.

Za potrebe izrade nove tablice u bazi podataka korištenjem koda potrebno je koristiti programske elemente iz modula «Microsoft DAO 3.6 Object Library» (Naredba Project → DataAccess Project... → References...)

1. Otvorite postojeći Data Project od prethodnog zadatka – Rad s bazama podataka I.!
2. Na polaznu formu sa strane dodajte dugme pod nazivom cmdEMailGenerator. Na dugmetu treba biti natpis: «Generiraj e-mail adrese».
3. Pritiskom na dodano dugme treba se pokrenuti izvođenje procedure koja će:
 - a) u bazu HR.mdb dodati novu tablicu sa sljedećim poljima (tip podatka) : redni broj (integer), ime (text), prezime(text), email(text)
 - b) zatim će pretraživanjem tablice Zaposlenici za sve postojeće zaposlenike formirati email adresu te je pohraniti u novonapravljenu tablicu email_adrese.
4. Za početak potrebno je definirati sve objekte koji će biti potrebni za dodavanje tablice i izmjenu podataka:
 - a) koristite naredbu za deklariranje varijabli/objekata DIM
 - b) objekt db predstavljat će bazu podataka (tip objekta DAO.Database)
 - c) objekt tbl predstavljat će novu definiciju strukture tablice (tip objekta DAO.TableDef)
 - d) objekt rstEMail predstavljat će podatke iz tablice email_adrese, a objekt rstZaposlen podatke iz tablice Zaposlenici (tip podatka DAO.RecordSet)
5. Drugi korak, nakon deklariranja objekata, je postavljanje polaznih vrijednosti tj. inicijalizacija objekata. Za tu potrebu koristi se naredba Set.

- a) Prvo otvaramo postojeću bazu HR. mdb korištenjem metode OpenDatabase kao i svojstva path naše aplikacije kojom ćemo utvrditi lokaciju mape u kojoj se nalazi baza:

```
Set db = OpenDatabase(App.Path & "\HR.mdb")
```

- b) Zatim stvaramo novi objekt za definiciju strukture tablice korištenjem naredbe CreateTableDef pod nazivom email_adrese.

```
Set tbl = db.CreateTableDef("email_adrese")
```

6. Treći korak je formiranje strukture nove tablice kreiranjem (metoda .CreateField) i dodavanjem stupaca (metoda .Append svojstva Fields) u definiciju tablice:

```
tbl.Fields.Append tbl.CreateField("Redni broj", dbInteger)
```

Na isti način dodajte i preostala polja (ime, prezime, email), samo za opis tipa podatka koristite dbText.

Programiranje 6

Rad s bazama podataka II. (2/2)

7. Ovako definiranu strukturu tablice treba sačuvati u bazu podataka gdje se nalaze i sve ostale definicije postojećih tablica (kolekcija objekata TableDefs i metoda .Append):

```
db.TableDefs.Append tbl
```

Nakon ovoga objekt tbl više nije potreban ba ga je moguće ukloniti: `Set tbl = Nothing`

8. S obzirom da je sada prisutna nova tablica u bazi podataka moguće je puniti je podacima o e-mail adresama. Stoga treba inicijalizirati objekte za manipuliranje zapisima/ RecordSet-ovima Zaposlenici i email_adrese:

```
Set rstZaposlen = db.OpenRecordset("Zaposlenici")  
Set rstEmail = db.OpenRecordset("email_adrese")
```

9. E-mail adresu treba dodati svakom zaposlenom pa je potrebno koristiti petlju `do...loop`. Adresu treba sastaviti korištenjem imena i prezimena zaposlenog te je pohraniti u tablicu `email_adrese`.

- a) Prije pokretanja petlje treba pronaći prvi zapis u tablici Zaposlenici. Za potrebe brojanja zaposlenih uvodi se varijabla `i`:

```
rstZaposlen.MoveFirst  
i = 1
```

- b) Unutar `do... loop` petlje potrebno je prvo preuzeti podatke o imenu i prezimenu te sastaviti email adresu:

```
Do  
ime = rstZaposlen.Fields!ime  
prezime = rstZaposlen.Fields!prezime  
adresa = Left(ime, 1) & prezime & "@tvrtka.hr"
```

- c) Zatim dobivene podatke sačuvati u novi zapis tablice `email_adrese` (skraćeni oblik desno):

<pre>rstEmail.AddNew rstEmail.Fields![Redni broj] = i rstEmail.Fields!ime = ime rstEmail.Fields!prezime = prezime rstEmail.Fields![E-mail] = LCase(adresa) rstEmail.Update</pre>	<pre>With rstEmail .AddNew .Fields![Redni broj] = i .Fields!ime = ime .Fields!prezime = prezime .Fields![E-mail] = LCase(adresa) .Update End With</pre>
--	---

- d) Nakon toga prelazi se na idućeg zaposlenog, a ukoliko više nema zaposlenika izlazi se iz `do...loop` petlje:

```
rstZaposlen.MoveNext  
i = i + 1  
Loop Until rstZaposlen.EOF
```

10. Na kraju je samo potrebno zatvoriti sve tablice i bazu te obavijestiti korisnika o obavljenom poslu:

```
rstZaposlen.Close  
rstEmail.Close  
db.Close  
MsgBox "E-mail adrese su dodijeljenje!"
```