Model sustava za poučavanje jezika SQL

Tunjić, Marina

Undergraduate thesis / Završni rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split, University of Split, Faculty of science / Sveučilište u Splitu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:166:444833

Rights / Prava: Attribution 4.0 International/Imenovanje 4.0 međunarodna

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Repository of Faculty of Science

zir.nsk.hr

UNIVERSITY OF SPLIT

dabar

DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U SPLITU
PRIRODOSLOVNO MATEMATIČKI FAKULTET

ZAVRŠNI RAD

MODEL SUSTAVA ZA POUČAVANJE JEZIKA SQL

Marina Tunjić

Split, rujan 2016.

Sadržaj

Uvod 2

1. Sustavi e-učenja 3

2. Baze podataka 6

2.1. Arhitektura i shema baze podataka 6

2.1.1. Entiteti, atributi, veze 7

2.2. Relacijski model 8

3. SQL upitni jezik 10

3.1. Upiti u relacijskoj bazi podataka 10

3.1.1. Tipovi podataka 12

3.1.2. Funkcije u SQL Upitu 13

4. ADDIE model 15

4.1. Analiza 15

4.2. Oblikovanje 17

4.3. Razvoj 20

4.4. Implementacija 21

4.5. Vrednovanje 26

Zaključak 27

Literatura 28

Sažetak 29

Summary 30

Uvod

Okruženje suverenog nastavnog procesa sve je više pod utjecajem informacijske i komunikacijske tehnologije pa tako e-učenje postaje sve raširenији način obrazovanja. Razvoj internet tehnologija rezultira povećanim brojem korisnika interneta, nastaje i sve veća potreba za učenjem preko interneta.
„E-učenje pokriva veliki skup primjena i procesa kao što su: učenje temeljeno na Web-u, učenje temeljeno na računalu, prividne razrede i digitalnu suradnju“. (Kaplan-Leiserson)

Kod kreiranja nastavnog materijala za e-obrazovanje, postavlja se pitanje kako prikazati i grupirati željeni nastavni sadržaj (lekcije, riješene primjere, zadatke za vježbu i dr.), odnosno na koji će način biti oblikovan taj obrazovni proces. Ideja ovog završnog rada je iskoristiti prednosti suvremenog i dinamičnog oblika učenja, odabrati jedan od postojećih modela instrukcijskog dizajna koji se koriste za modeliranje sustava za učenje i realizirati sustav za poučavanje SQL jezika. SQL jezik je iznimno važan prilikom rada s podacima u relacijskoj bazi podataka, njegovo razumijevanje i upotreba neophodna je svakom programeru.

Ovaj završni rad organiziran je u četiri poglavlja. U prvom poglavlju definirani su pojmovi e-učenje i inteligentni tutorski sustavi, općenito oblikovanje nastavnog sadržaja u e-učenju korištenjem modela. Također, očekivanja prilikom razvoja sustava prema modelu. Drugo poglavlje posvećeno je bazama podataka. Nadalje, treće poglavlje govori o SQL upitnom jeziku za koji je napravljen sustav za poučavanje. Završni dio je realizacija projekta, model sustava za poučavanje SQL-a prema procesnom modelu ADDIE za oblikovanje nastave u e-učenju.

Commented [M1]:

1. Sustavi e-učenja

U uvodnom dijelu naglasak je na potrebi korisnika za obrazovanjem putem interneta. Prema tome, oblikovanje sustava koji služe za učenje je u stalnom razvoju. Postoji jako mnogo definicija koje govore što je to zapravo e-učenje, dalje u ovom poglavlju objašnjene su vrste isporuke nastavnog sadržaja, vrste sustava, te modeli koji se koriste prilikom izrade samog sustava.

E-učenje predstavlja isporuku nastavnog sadržaja obuke ili obrazovnog programa pomoću elektroničkih sredstava. Ono obuhvaća upotrebu računala ili nekog elektroničkog uređaja radi ostvarivanja obuke, obrazovanja ili pristupa nastavnom sadržaju za učenje. Može uključivati brojne tehnike i opremu i obično podrazumijeva isporuku nastavnih sadržaja online u internet ili intranet okruženju, kao i u offline okruženju pomoću CD-ROM ili DVD medija. [1]

Prema tehnologiji za isporuku nastavnog sadržaja, e-učenje možemo podijeliti u dvije kategorije:

- ***Asinkrono*** je ono učenje kod kojeg se interakcija učitelja i učenika događa povremeno i pri tome njihove aktivnosti nisu po vremenu sinkronizirane. Dohvat i isporuka nastavnih sadržaja moguća je na bilo kojem mjestu, u bilo kojem vremenu i napredovanje s vlastitim tempom (eng. any place, any time, and self paced). [2]

- ***Sinkrono*** e-učenje podrazumijeva interakciju učitelja i učenika u načelu u realnom vremenu, aktivnosti su po vremenu sinkronizirane i odvijaju se po unaprijed dogovorenom scenariju na unaprijed dogovorenim mjestima. Na ovaj način se formiraju prividne učionice (eng. Virtual classroom). Ovo je razlog zašto sinkroni sustavi zadržavaju gotovo sve atribute koji obilježavaju tradicionalni predavački model poučavanja. [2]

Svaki sustav e-učenja bi trebao zadovoljiti temeljne funkcije nastavnog procesa kako bi bili potpuni. Osnovne funkcije nastavnog procesa su učenje, poučavanje, testiranje i vrednovanje. Dakle, svaki sustav e-učenja trebao bi osim isporuke nastavnih sadržaja imati i testiranje i vrednovanje znanja učenika. Također, upravljanje ili na višoj razini vođenje procesa i poučavanja i administriranje sudionika (učenika, učitelja, stručnjaka područnog znanja). Ove funkcionalnosti utemeljene su na tradicionalnim načelima učenja. [3]

Posebna klasa asinkronih sustava e-učenja su inteligentni tutorski sustavi (ITS). Oni predstavljaju napredno okruženje učenja i poučavanja prilagođeno aktualnoj razini znanja učenika. ITS su generacija računalnih sustava namijenjeni potpiri i poboljšanju procesa učenja i poučavanja u odabranom područnom znanju, uvažavajući pri tom individualnost onoga tko uči i tko se poučava. Uvažavanje individualnosti onoga tko se poučava je zapravo poučavanje po modelu jedan-na jedan (eng. one-to-one tutoring) koje kad se odvija pod okriljem ljudskog tutora je dokazano uspješno i smatra se najučinkovitijim načinom odvijanja nastavnog procesa. [3]

Da bi sustav bio inteligentan mora se ponašati inteligentno tj. raspolagati sa znanjem i biti u mogućnosti dati informacije o učenikovom znanju u bilo kojoj fazi procesa učenja. Također, sustav može biti inteligentan iako ne donosi zaključke o učenikovom znanju, ali se to ne smatra tutorskim sustavom nego alatom s određenim mogućnostima dijagnosticiranja.

Prilikom postavljanja nastavnog sadržaja za učenje na web-u kroz neki sustav za učenje potrebno je odabrati pedagoški okvir koji će se koristi prilikom oblikovanja tijeka poučavanja. Moguće je primijeniti nekoliko različitih modela poučavanja. Kada govorimo o e-obrazovanju većina svih modela temelji se na sličnim elementima, od kojih su najčešći:

- definiranje ciljeva on-line tečaja;
- oblikovanje plana rada na tečaju;
- izbor obrazovnog sadržaja i određivanje njegovog slijeda i strukture;
- odlučivanje kako će biti prikazani nastavni materijali i koje će metode poučavanja biti primijenjene. [4]

Postoji mnoštvo modela za oblikovanje nastave u e-učenju, međutim većina najkorištenijih se može podijeliti na:

- **Procesne (fazne) modele** – oni opisuju kako se proces oblikovanja nastave treba odvijati i što u pojedinoj fazi treba postići. Ovdje spadaju ADDIE model, model Dicka i Carreya, ubrzano prototipiranje, Gangeov model devet koraka, Salomonovih pet faza e-modeliranja;

- **Modele koji se odnose na procese i ishode učenja** – opisuju kako pridobiti i motivirati učenika, kako kvalitetno postaviti ciljeve učenja i kako vrednovati rezultate. To su primjerice ARCS model (engl. *Attention, Relevance, Confidence, Satisfaction*), minimalistički model, model konstrukcije interpretacija, Bloomova taksonomija, Laurillardov konverzacijski okvir, Khanov okvir osam komponenti i dr. [5]

U ovom završnom radu posvetit ćemo se ADDIE modelu, najpopularnijem modelu za oblikovanje nastavnog sadržaja. Korištenjem ovoga modela oblikovati će se sustav za poučavanje SQL-a.

2. Baze podataka

Osnovna ideja tehnologije baze podataka je da sve aplikacije koriste zajedničku kolekciju podataka (bazu podataka), osim toga i vrsta rada s podacima na posredan način koristeći softver posrednik između aplikacije i podataka. Baze podataka čine sastavni dio informacijskog sustava.

Informacijski sustav prikuplja, pohranjuje, čuva, obrađuje i isporučuje važne informacije za organizaciju i društvo, tako da budu dostupne i upotrebljive za svakog tko ih želi koristiti, uključujući poslovodstvo, klijente, osoblje i ostale. Aktivni je društveni sustav koji može koristiti suvremenu informacijsku tehnologiju. [6]

„Baza podataka je skup međusobno povezanih podataka, pohranjenih u vanjskoj memoriji računala. Podaci su istovremeno dostupni raznim korisnicima i aplikacijskim programima. Ubacivanje, promjena, brisanje i čitanje podataka obavlja se posredstvom posebnog softvera, takozvanog sustava za upravljanje bazom podataka (DBMS-a) “ [7]

Baze podataka predstavljaju višu razinu rada s podatcima, omogućuju fizičku neovisnost podataka, logičku neovisnost podataka, fleksibilnost pristupa podacima, čuvanje integriteta, zaštitu neovlaštenog korištenja, brz i fleksibilan pristup podacima, istovremeni pristup podacima, mogućnost podešavanja i kontrole.

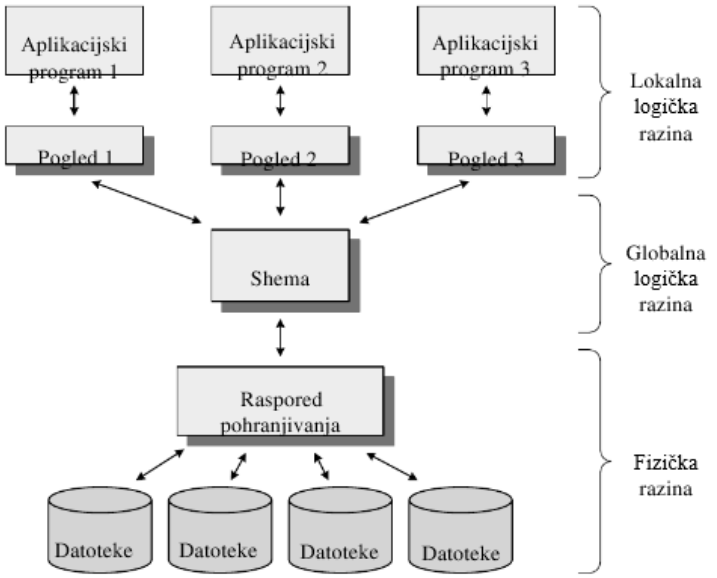
Programski sustav koji upravlja spremanjem, pronalaženjem i sortiranjem podataka naziva se DBMS (Data Base Management System). DBMS pouzdano obavlja sve operacije u ime klijenta.

2.1. Arhitektura i shema baze podataka

Arhitektura baze podataka sastoji se od tri razine apstrakcije.

- ***Fizička razina*** odnosi se na raspored podataka i fizički prikaz na jedinicama vanjske memorije, vidljiva je samo sistemskim programerima.
- ***Globalna logička razina*** se odnosi na logičku strukturu cijele baze. Shema je naziv za opis globalne logičke definicije. To je tekst ili dijagram koji definira logičku strukturu baze te je u skladu sa zadanim modelom i vidljiva je samo administratoru.

- ***Lokalna logička razina*** se odnosi na logičku predodžbu o dijelu baze koji koristi pojedina aplikacija i vidljiva je korisnicima ili aplikacijskom programeru. Opis lokalne logičke definicije naziva se pogled.



Slika 1.1 Arhitektura baze podataka [7]

Nakon zadane sheme DBMS automatski generira potrebni raspored pohranjivanja i fizičku bazu. DBMS prevodi korisničke zahtjeve za podacima s lokalne logičke razine na globalnu te ih dalje realizira na fizičkoj razini. Prva faza oblikovanja baze podataka je konceptualno oblikovanje gdje stvaramo shemu baze koja je sastavljena od entiteta, atributa i veza.

2.1.1. Entiteti, atributi, veze

- „Entitet je nešto što postoji i što se u stvarnom svijetu može identificirati.“
- Atribut je svojstvo entiteta ili veza. Svaki entitet je opisan skupom atributa
- Veza je odnos između entiteta. Ona se definira na razini tipova entiteta. Veze mogu biti 1:1 (jedan naprama jedan), 1:n (jedan naprama više), n:1 (više naprama jedan), m:n (više naprama više). [7]

Entiteti u projektu:

1. T_ZAPOSLENICI

   Entitet T_ZAPOSLENICI sadrži atribute ID_ZAPOSLENIK, IME, PREZIME, KONTAKT, PLACA, SEF. Sadrži podatke o više zaposlenika.

2. T_ODJEL

   Entitet T_ODJEL sadrži atribute ID_ODJEL i NAZIV_ODJEL. Ova tablica sadrži podatke o nazivima odjela.

3. T_ZAPOSLENICI_ODJEL

   Entitet T_ZAPOSLENICI_ODJEL sadrži atribute ID, ID_ODJEL, ID_ZAPOSLENIK, DATUM_POČETKA_RADA, DATUM_ZAVRŠETKA_RADA. ID_ODJEL i ID_ZAPOSLENIK su strani ključevi i povezuju entitete T_ZAPOSLENICI I T_ODJEL.

Nakon što su utvrđeni entiteti, atributi i veze između njih, crta se dijagram koji povezuje navedene elemente. ER dijagram (engl. Entity relationship) entiteti veze je dijagram koji služi u opisivanju odnosa između entiteta u bazi podataka. Sadržava atribute koji predstavljaju značajke entiteta i ključeve koji predstavljaju atribute koji jednoznačno identificiraju entitete.

2.2. Relacijski model

Druga faza oblikovanja baze podataka je stvaranje relacijske sheme baze. U njoj su entiteti i veze među entitetima pretvoreni u relacije. Relacijski model je krajem 60-tih godina 20. stoljeća zasnovao Edgar Codd. Danas se većina DBMS-ova koristi tim modelom. Ciljevi relacijskog modela su:

- omogućiti nezavisnost podataka
- dati teorijske temelje za konzistentno semantičko postupanje s podacima
- omogućiti razvoj skupovno orijentiranih jezika za obradu podataka
- dati bogat model podataka za opis i obradu jednostavnih i kompleksnih podataka. [6]

Relacijski model zahtijeva da se baza podataka sastoji od relacija (skupa pravokutnih tablica). Svaka relacija ima svoje ime. Jedan stupac relacije sadržava vrijednost jednog atributa tj. atribut je imenovani stupac relacije.

Relacijski model baze podataka prikazuje logičku raspodjelu podataka u bazi podataka, on se ne bavi problemom fizičkog smještaja podataka u bazi podataka. Relacijski model podataka većinom je normaliziran do treće normalne forme.

Normalizacija je postupak prevođenja jedne ili skupa relacija iz niže u višu normalnu formu. Pritom se služi operacijama projekcije i prirodnog spajanja. [6]

Postoji 6 vrsta normalnih formi.

Prva normalna forma (1NF) se zasniva na tome da nema ponavljajućih grupa, one se izdvajaju u zasebnu tablicu. Relacija sadrži samo jednu vrijednost koja ne smije biti složena. Definiran je primarni ključ koji ima uvijek jedinstvenu vrijednost. Svi neključni atributi funkcijski su zavisni o ključu relacije. Vrijednost atributa unutar relacije mora biti jednostruka i jednostavna.

Druga normalna forma (2NF) definira se uz pomoć potpune funkcijske zavisnosti. Svi neključni atributi su u potpunosti zavisni o ključu relacije (primarnom ključu).

„Relacija se nalazi u trećoj normalnoj formi (3NF) ako ni jedan neključni atribut nije tranzitivno zavisan o bilo kojem ključu relacije. Svaki neključni atribut mora zavisiti o ključu, cijelom ključu i ni o čemu osim o ključu“. [6]

3. SQL upitni jezik

SQL (engl. Structured Query Language) je referentni jezik za relacijske baze podataka. Razvijen je 70-tih godina u IBM Research Laboratory at San Jose – California. Temeljen je na relacijskoj algebri i predikatnom računu. Standardiziran preko standarda ANSI (American National Standards Institute - Američki nacionalni institut za standarde) i ISO(International Organization for Standardization - Međunarodna organizacija za normizaciju).

Omogućuje korisniku da bez nekog većeg poznavanja i razumijevanja SUBP-a pregledava podatke. Glavna karakteristika je neproceduralnost, opisuje što se želi dobiti kao rezultat, ali ne i kako se do tog rezultata dolazi, nije potrebno definirati korake (procedure) izvođenja. SQL je u stalnom razvoju. Na početku je je bio veoma jednostavan, blizak korisniku (engl. User friendly), no danas se može reći da je kompleksan i proceduralno/deklarativan jezik.

3.1. Upiti u relacijskoj bazi podataka

SQL je jednostavan u korištenju, na višem nivou apstrakcije neko neki drugi programski jezici omogućuje rad sa relacijskom bazom podataka. U upitu se određuju podatci koje će upit sadržavati. Prilikom kreiranja upita susrećemo se s nekoliko ključnih riječi, a to su :

instrukcije – glagoli koji definiraju nekakvu akciju koja se treba izvršiti (SELECT),

kvalifikatori - ograničenja opseg vrijednosti entiteta (WHERE),

klauzule - prilagođavaju akciju instrukcije (ORDER BY),

operatori – služe da bi poredali vrijednosti (=,<,>).

Zaključno sa SQL – 92 standardom SQL naredbe jezika podijeljene su u tri kategorije:

DML (Data Manipulation Language) Naredbe za traženje, umetanje, izmjene i brisanje podataka. Primjeri naredbi ove kategorije su:

- SELECT - prikazuje sadržaj relacijske baze.

  Oblik: `SELECT Ime FROM T_ZAPOSLENICI [WHERE PLACA < 8000]`

- UPDATE - izmjenjuje vrijednost retka tablice, važno je paziti da se vrijednost svakog atributa može mijenjati samo jednim izrazom.

  Oblik: `UPDATE tablica SET atribut=izraz [WHERE uvjet]`

- DELETE - izbacuje retke u tablici, unaprijed zadani kriteriji, biše samo sadržaj tablice ne i samu tablicu.

  Oblik: `DELETE [FROM] tablica [WHERE uvjet]`

- INSERT - dodaje redak postojećoj tablici.

  Oblik: `INSERT INTO naziv_tablice VALUES (vrijednosti atributa)`

DDL (Data Definition Language) Naredbe za definiciju objekata (tablica,indeksa,pogleda) relacijske baze podataka.

- CREATE TABLE - kreiranje tablice baze podataka. Kod kreitanja treba navesti ime relacije i za svaki stupac tablice, ime, tip, dužinu i jesu li NULL vrijednosti dozvoljene.

  Oblik: `CREATE TABLE ImeTablice (atribut tip [not null],...)`

- CREATE VIEW - kreiranje „pogleda“ – virtualne tablice koja uopće ne zauzima memorijski prostor, nema vlastite podatke. Ažuriranje preko pogleda može se vršiti samo kad je pogled definiran samo nad jednom tablicom i ako sadrži sve stupce NOT NULL.

  Oblik: `CREATE VIEW ImePogleda[(atributi)] SELECT atr FROM...`

- ALTER TABLE - izmjena definicije tablice, dodavanje novog stupca tablici.

  Oblik: `ALTER TABLE tablica ADD (atribut tip [, atribut tip])`

- DROP TABLE - izbacivanje tablice iz baze podataka, i sadržaj i tablicu.

  Oblik: `DROP TABLE tablica`

DCL (Data Control Language) Naredbe za upravljačke funkcije

- GRANT - dodjeljivanje prava korištenja tablice drugim korisnicima, sva prava bi bila ALL, svim korisnicima PUBLIC

- REVOKE - oduzimanje prava korištenja tablice od drugih korisnika
- COMMIT - spremanje promjena u bazi podataka
- ROLLBACK - ukidanje svih promjena od zadnje COMMIT naredbe

SQL:1999 standard razvrstava SQL naredbe u 7 kategorija. Razlog tome je uvođenje novih koncepata i potreba preciznijeg grupiranja naredbi. Definirane su sljedeće kategorije: naredbe za shemu baze podataka, naredbe za podatke, naredbe za transakcije, naredbe za kontrolu, naredbe za konekcije, naredbe za sesije, naredbe za dijagnostiku. Osim predefiniranja raspodjele naredbi po kategorijama, SQL:1999 određuje i više načina korištenja SQL-a. Osnovna dva načina korištenja SQL-a su direktno ili interaktivno korištenje i povezivanje SQL-a sa nekim klasičnim programskim jezicima („ugrađeni" SQL). [8]

3.1.1. Tipovi podataka

Svaki stupac u tablici baze podataka prilikom kreiranja zahtjeva ime i tip podatka. U sljedećoj tablici su navedeni i opisani neki od najčešćih tipova podataka:

Tabela 1 [9]

CHARACTER(n)	String sa definiranom duljinom n
VARCHAR(n) ili CHARACTER VARYING(n)	String varijabilne duljine, maksimalna duljina je n
BOOLEAN	Pohranjuje TRUE ili FALSE vrijednosti
INTEGER	Cijeli broj pohranjen u 4 byte-a
SMALLINT	Cijeli broj pohranjen u 2 byte-a
DECIMAL(p,s)	Decimalni tip podatka , m=ukupan broj znamenki, n=broj znamenki iza decimalne točke

FLOAT(p)	Prostor
REAL	Approximate numerical, mantissa precision 7
FLOAT	Approximate numerical, mantissa precision 16
DOUBLE PRECISION	Approximate numerical, mantissa precision 16
DATE	Pohranjuje godinu, mjesec i dan
TIME	Pohranjuje sat, minute i sekunde
INTERVAL	Pohranjuje informacije o trajanju – količini vremena

3.1.2. Funkcije u SQL Upitu

Osim prikaza vrijednosti koje su memorirane u tablicama baze podataka, SQL ima nekoliko funkcija koje se koriste za dobivanje izvedenih informacija. Obično te funkcije se zovu agregatne funkcije. Najznačajnije agregatne funkcije su:

AVG (atribut) – izračunava srednju vrijednost

SUM (atribut) – izračunava ukupnu vrijednost

MIN (atribut) – nalazi minimalnu vrijednost

MAX (atribut) – nalazi maksimalnu vrijednost

COUNT (*) – broji ukupan broj redaka tablice

SQL naredbe mogu sadržavati i aritmetičke izraze sastavljene od imena stupaca i konstantnih vrijednosti povezanih aritmetičkim operatorima ('+', '-' , '*' , ' / ').

GROUP BY operator ima zadatak da omogući grupiranje vrsta u rezultirajućoj tablici na osnovu identičnih vrijednosti**.**

Primjer u kodu:

```
SELECT T_ODJEL.NAZIV_ODJEL, COUNT(*) AS 'NAZIV ODJELA'
```

```
FROM T_ZAPOSLENICI INNER JOIN T_ZAPOSLENICI_ODJEL ON
T_ZAPOSLENICI.ID_ZAPOSLENIK=T_ZAPOSLENICI_ODJEL.ID_ZAPOSLENIK
INNER JOIN T_ODJEL ON T_ZAPOSLENICI_ODJEL.ID_ODJEL =
T_ODJEL.ID_ODJEL

GROUP BY T_ODJEL.NAZIV_ODJEL

ORDER BY 'NAZIV ODJELA' DESC
```

Kod 1

NAZIV_ODJEL	NAZIV ODJELA
Odjel upravljanja kvalitetom	2
Računovodstvo	2
Odjel razvoj proizvoda	1
Odjel vođenja projekata	1

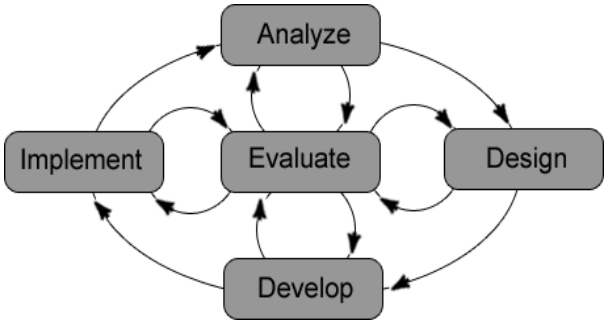
Slika 1

Pored navedenih najznačajnijih agregatnih funkcija, SQL:1999 uvodi i funkcije EVERY,ANY i SOME koje su definirane nad logičkim (BOOLEAN) tipovima podataka. No, osim ove tri navedene funkcije SQL:1999 standard sadrži 4 tipa funkcija za obradu pojedinačnih redova: [8]

- Numeričke funkcije
- Tekstualne funkcije
- Datumske funkcije
- Intervalne funkcije

4. ADDIE model

Za izradu sustava korišten je ADDIE (engl. Analysis, Design, Development, Implementation, Evaluation) model za oblikovanje nastavnog sadržaja u e-obrazovanju. ADDIE procesni model se sastoji od 5 faza: analize, oblikovanja, razvoja implementacije i vrednovanja. U svakoj od ovih pet faza jasno je definirano što se treba napraviti.



Slika 2 – ADDIE model [10]

4.1. Analiza

Analiza je prva faza ADDIE modela, u kojoj se definiraju: ciljevi koji se žele postići, za koga je namijenjen sadržaj, što je potrebno naučiti, što se podrazumijeva da se zna i određene nastavne strategije.

Oblikovanje nastave:

- U procesu oblikovanja sudjeluje student koji se eventualno konzultira sa mentorom.
- Svrha ovog projekta je omogućiti studentima lakše učenje SQL jezika, sustav je na hrvatskom jeziku, te je napravljen po uzoru na w3schools tutorijal. Sadržaj sustava su SQL naredbe koje omogućuju korisnicima unos, brisanje, mijenjanje i pretraživanje baze podataka.
- Isporuka nastavnog sadržaja se odvija unutar Internet preglednika (Google Chrome, Mozilla Firefox, Internet Explorer) na klijentskom računalu.

Polaznici:

- Svi koji žele naučiti upitni jezik SQL,

- Preporučuje se razumijevanje Baza podataka koje su opisane u 2. Poglavlju, iako nije potrebno za učenje SQL jezika.

Ciljevi:

- Cilj sustava za poučavanje je upoznati studente sa osnovnim tipovima naredbi, te korištenje jednostavnih i složenih upita nad bazom podataka.

Nastavni sadržaj i isporuka nastavnog sadržaja:

- Sadržaj su SQL naredbe koje omogućuju korisnicima unos, brisanje, mijenjanje i pretraživanje baze podataka.
- Koristit će se ASP.NET web forme koje omogućuju objektno orijentiranu izradu mrežnih aplikacija. Koriste se serverske kontrole kao vizualni elementi na formi koje uzrokuju automatsko generiranje HTML koda i olakšava pisanje pozadinskog koda. Isporuka nastavnog sadržaja se odvija unutar Internet preglednika na klijentskom računalu, mrežne aplikacije se izvršavaju na poslužitelju.

  Aplikaciji će se moći pristupiti putem Interneta i mogu je koristiti tri vrste korisnika:

  -Administratori (imaju mogućnost korištenja linka Upravljanje bazom podataka, gdje je na jednom mjestu moguće upotrijebiti sve upite, administratori također istodobno mogu biti i korisnici)

  -Korisnici (imaju mogućnost isprobavanja svih upita koji se nalaze na stranici)

  -Anonimni korisnici (nemaju mogućnost isprobavanja upita, osim za naredbe SELECT i INNER JOIN, mogu samo vidjeti objašnjenja, te se također imaju mogućnost registrirati da postanu član ili logirati se ako su već od prije član)

  Prijavljenim korisnicima i administratorima će se u gornjem lijevom kutu vidjeti opcija za odjavu iz sustava, pored njihovog korisničkog imena. Autentifikacija se postavlja u izborniku WebSite odabirom ASP.NET Configuration gdje stvaramo korisnike i definiramo njihove uloge odnosno kojim stranicama mogu pristupiti.

Nastavne strategije:

- Smanjiti uvodna izlaganja, izbjegavati nepotrebnu teoriju, te što više se usmjeriti praktičnim i primjenjivim aktivnostima. Što brže savladavanje praktično primjenjivih znanja i vještina uz zadatke koji trebaju dozvoljavati polaznicima da samostalno zaključuju i napreduju

Strategije ocjenjivanja:

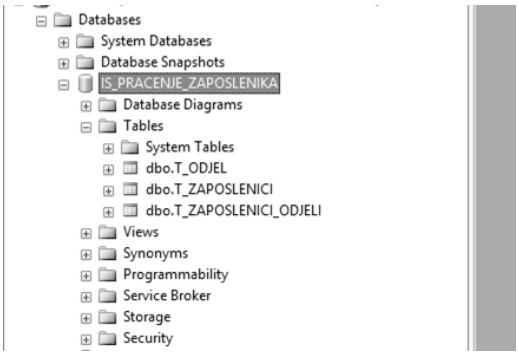
- Sustav je usmjeren polaznicima kao pomoć pri samostalnom učenju stoga nije predviđeno ocjenjivanje.

4.2. Oblikovanje

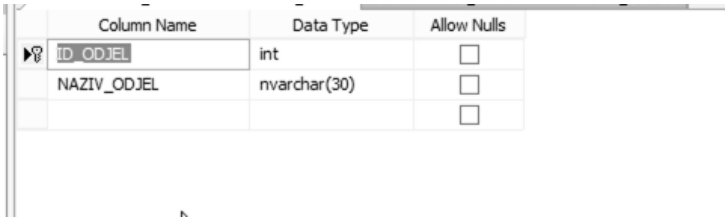
Druga faza, nakon analize, je oblikovanje. U ovoj fazi bi se trebalo utvrditi na koji način će se postići ciljevi koje smo definirali u prethodnoj fazi. Određuje se i vrijeme koje je potrebno za razvoj i alat koji ćemo koristiti.

Raspored rada:

- Vrijeme potrebno za izradu same aplikacije ovisi o znanjima i sposobnostima osobe koja ih izrađuje, konkretno za izradu ovoga sustava trebalo je mjesec dana sa mogućnošću manjeg odstupanja. Alat koji se koristio pri izradi je Microsoft Visual Studio Professional 2012 za izradu aplikacije i Microsoft SQL Server EXPRESS 2008 za izradu bazi podataka. Struktura baze:



Slika 3 – Struktura baze podataka iz projekta



	ID_ODJEL	NAZIV_ODJEL
▶	1	Računovodstvo
	2	Odjel vođenja projekata
	3	Odjel razvoj proizvoda
	4	Odjel upravljanja kvalitetom
*	NULL	NULL

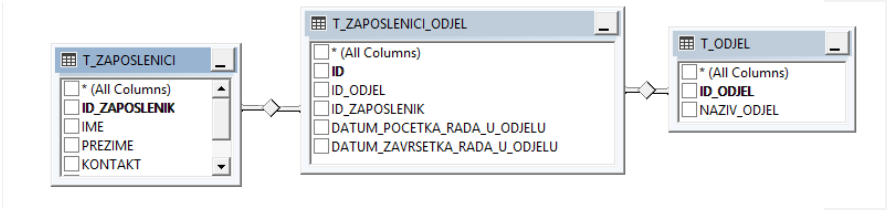
Slika 4 – Tablica T_ODJEL

	ID_ZAPOSLENIK	IME	PREZIME	KONTAKT	PLACA	SEF
▶	1	Pero	Perićić	pero.peric@gma...	10569,56	da
	2	Marko	Markić	markomarkic@y...	7896,45	ne
	3	Tina	Tinić	tinictina@gmail.c...	7895,45	ne
	4	Dino	Dinić	dinodinic@yaho...	10789,67	da
	5	Ana	Anić	aanic@gmail.com	4567,39	ne
	6	Ana	Dinić	adinic@gmail.com	10600,59	da
	7	Teja	Dinić	tejadinic@gmail....	5678,36	ne
	8	Goga	Gogić	ggogic@gmail.com	78946,45	da
*	NULL	NULL	NULL	NULL	NULL	NULL

Slika 5 – Tablica T_ZAPOSLENICI

	ID	ID_ODJEL	ID_ZAPOSLENIK	DATUM_POCETKA_RADA_U_ODJELU	DATUM_ZAVRSETKA_RADA_U_ODJELU
▶	1	1	1	2006-04-13	NULL
	2	1	5	2010-01-01	NULL
	3	2	2	2009-02-01	2010-08-31
	4	3	2	2010-09-01	NULL
	5	4	4	2000-03-01	NULL
	6	4	3	2005-05-01	NULL
*	NULL	NULL	NULL	NULL	NULL

Slika 6 – Tablica T_ZAPOSLENICI_ODJEL



Slika 7 – Baza podataka

- Vrijeme korištenja ovog sustava za poučavanje ovisi o polazniku koliko želi utrošiti vremena na pojedine zadatke, te naravno o njegovim prethodnim znanjima. Otprilike za prolazak kroz sve upite potrebno je nekoliko sati.

Zadatci:

- Polaznici bi trebali zadovoljiti ciljeve koji su utvrđeni u fazi analize, a za ostvarivanje ovih ciljeva uz kratak opis pruža se mogućnost direktne interakcije sa bazom i opisani SQL upiti.

Pisana priprema:

- Prije svega polaznici bi trebali proučiti što ih očekuje, u sustavu se nalaze SQL Uvod i kratak opis naredbi pod linkom SQL Sintaksa.

SQL je računalni jezik za rad sa skupovima činjenica i vezama među njima. Relacijski programi za baze podataka, primjerice Microsoft Office Access, koriste SQL za rad s podacima. Za razliku od drugih računalnih jezika, SQL nije teško čitati i razumjeti, čak i ako ste početnik. Kao i mnogi drugi računalni jezici, SQL je međunarodni standard koji priznaju standardizacijske ustanove, primjerice ISO i ANSI.

Tipovi naredbi unutar SQL-a:

1. DQL – Data Query Language (Upiti za prikazivanje podataka relacijske baze)

2. DDLS – Data Definition Language Statements (Upiti za definiciju objekata relacijske baze)

3. DCLS – Data Control Language Statements (Upiti za upravljanje objektima relacijske baze)

4. DMLS – Data Manipulation Language Statements (Upiti za upravljanje podacima relacijske baze)

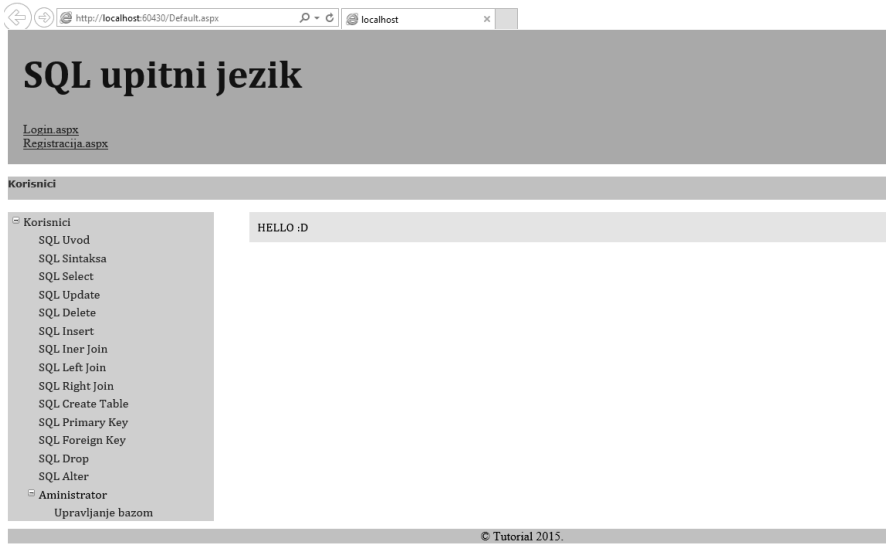
Slika Uvod

U uvodnom dijelu osim kratkog upoznavanja sa nastavnim sadržajem, bitno je motivirati korisnika, važno je naglasiti kako SQL nije teško razumjeti čak ni početnicima.

Nakon definiranja specifičnih ciljeva i očekivanja od korisnika, naveden je i kratak opis sadržaja, pa prelazimo na fazu razvoja.

4.3. Razvoj

U trećoj fazi ADDIE modela se razvija preostali dio nastavnog sadržaja. Ono na što treba obratiti pozornost u ovoj fazi je kvalitetan razvoj koji će biti pregledan polaznicima, dakle stvaranje početne stranice i organizacija linkova. Sama organizacija stranice omogućuje korisniku dostupnost bilo kojem dijelu sadržaja u bilo koje vrijeme. Na sljedećoj slici je prikazana početna stranica iz projekta.



Slika 8 – Sučelje aplikacije

Početna stranica jednaka je svim korisnicima. S lijeve strane se nalazi izbornik koji omogućuje određene akcije. Administratori i korisnici se prijavljuju u sustav koristeći svoje korisničko ime i lozinku. Nakon logiranja, ovisno o ulozi, korisnik može isprobavati SQL naredbe. Temelj svake mrežne aplikacije, prvi korak svakako je MasterPage. To su predlošci stranica pomoću kojih je moguće definirati fiksni sadržaj, te prostor za umetanje ostatka sadržaja. Prednost ovakvog pristupa je što se u slučaju izmjene predloška automatski mijenja izgled svih podstranica (engl. Content Page). Podstranice mogu sadržavati proizvoljan sadržaj, a koriste se za popunjavanje praznina odnosno onih dijelova koji su ostavljeni za prilagodbu unutar MasterPage-a. Na mjestu praznina nalazi se kontrola ContentPlaceHolder koju nije moguće koristiti na običnim stranicama. Na njegovo mjesto se ubacuje različiti

sadržaj na svakoj podstranici. S MasterPage-om se povezuje preko ID-a. U MasterPage-u koristimo kontrolu TreeView kojoj je izvor podataka SiteMapDataSource. TreeView nam služi za kretanje u aplikaciji, pomoću njega šetamo po podstranicama. Izvor SiteMap-u je XML (engl. EXtensible Markup Language) dokument Web.sitemap u kojemu su navedene putanje stranica. [11]

Pregled učinjenog, okruženje i izrada materijala:

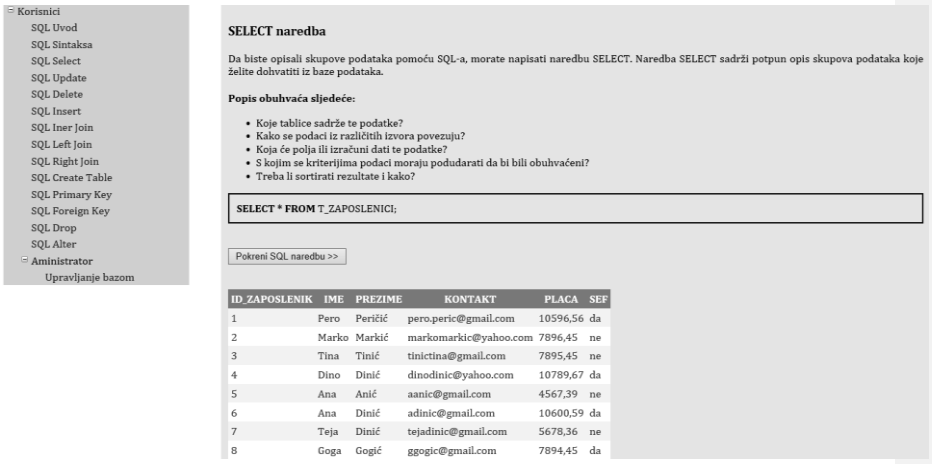
- Aplikacija je izrađena po uzoru na w3schools tutorjial, gdje možemo pronaći i tutorijal za SQL
- Uz pregled koraka u fazi analize i oblikovanja potrebno je utvrditi materijale potrebne za konačnu izradu. Dakle, glavni cilj je bio naučiti korisnika SQL upitni jezik.
- Za spajanje na bazu nam služi ConnectionString koji se dodaje u web.config datoteku i definira na koju bazu podataka je spojena web aplikacija. Aplikacija SQL sustava gotovo u svakom trenutku koristi bazu podataka, bilo to u dodavanju novih podataka, brisanju i mijenjanju starih, te za pretraživanje. Veza s bazom podataka se u .NET frameworku može stvoriti na više načina, eksplicitno koristeći SqlDataSource ili korištenjem ADO.NET objektnog modela koji ima 2 ključne komponente, a to su DataSet i .NET DataProvider. ADO.NET je objektno orijentirani skup biblioteka koji omogućuje interakciju s izvorima podataka, s njim pristupamo izvoru podataka. DataSet je komponenta na kojoj se temelji disconnected arhitektura ADO.NET-a, što znači da se podaci dohvaćaju iz izvora podataka neovisno o kojem se izvoru podataka radi. DataProvider – skupovi biblioteka koje se grupiraju prema načinu komunikacije s izvorom podataka. ADO.NET model služi kao most između ASP.NET objekta i pozadinske baze podataka. [11]

4.4. Implementacija

Općenito je implementacija je faza koja se odnosi na izgradnju programske podrške. Svaka SQL web forma izgleda različito, i korišteni su drugačiji pristupi u oblikovanju.

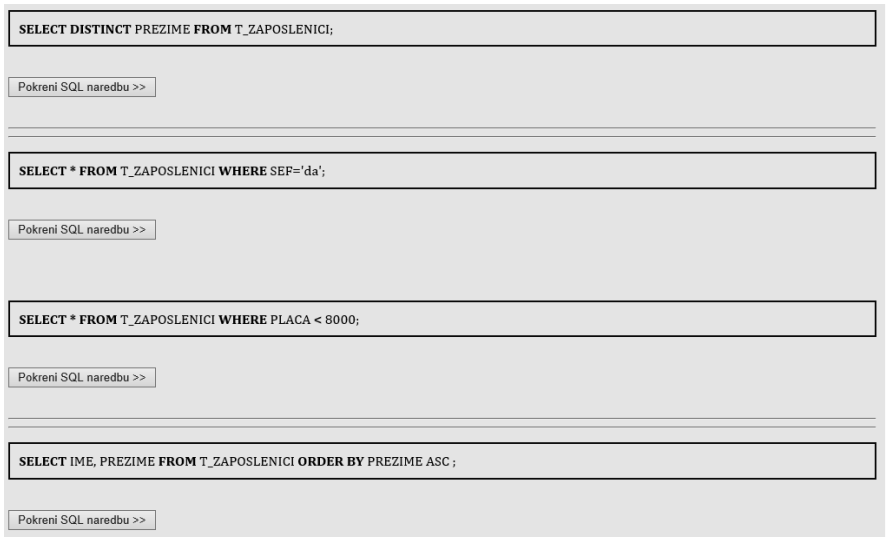
Kod stvaranja SQL SELECT web forme dodani su Button, SqlDataSource i GridView kontrola. Za kontrolu SqlDataSource potrebno je odabrati podatke koje će kotrola sadržavati.

Odabirom gumba 'Pokreni SQL naredbu', dodaje se pozadinski kod koji će omogućiti da GridView kontrola prikaže podatke koji se nalaze unutar SqlDataSource kontrole.



Slika 9 – SELECT naredba

Na slici je prikazan primjer naredbe SELECT. Pritiskom na dugme Pokreni SQL naredbu, u GridView-u nam se prikaže tablica rezultat upita, svi zaposlenici iz tablice T_ZAPOSLENICI. Ostali SELECT upiti:



Slika 10 – Ostali primjeri SELECT naredbe

UPDATE naredba

SQL naredba UPDATE služi za mijenjanje podataka u jednom ili više redaka u tablici. Mogu se promijeniti svi redovi ili samo podskup redaka, na temelju uvjeta.

```
UPDATE T_ZAPOSLENICI
SET IME = @IME, PREZIME = @PREZIME, KONTAKT = @KONTAKT,
PLACA = @PLACA, SEF = @SEF
WHERE ID_ZAPOSLENIK = @ID_ZAPOSLENIK;
```

	ID ZAPOSLENIK	IME	PREZIME	KONTAKT	PLAĆA	ŠEF
Edit	1	Pero	Peričić	pero.peric@gmail.com	10596,56	da
Edit	2	Marko	Markić	markomarkic@yahoo.com	7896,45	ne
Edit	3	Tina	Tinić	tinictina@gmail.com	7895,45	ne
Edit	4	Dino	Dinić	dinodinic@yahoo.com	10789,67	da
Edit	5	Ana	Anić	aanic@gmail.com	4567,39	ne
Edit	6	Ana	Dinić	adinic@gmail.com	10600,59	da
Edit	7	Teja	Dinić	tejadinic@gmail.com	5678,36	ne
Edit	8	Goga	Gogić	ggogic@gmail.com	7894,45	da
Edit	11	dadafa	sdas	,kweje@fsds.ds	33333	da

Slika 11 UPDATE naredba

- Kako bi korisnici mogli provjeriti svoj napredak dovoljno je ubaciti nekoliko zadataka prilikom učenja. Primjer zadatka za naredbu INNER JOIN uz prethodno objašnjenje naredbe:

INNER JOIN

Normalizirana baza je ona u kojoj su podatci "razbijeni" na više malih tablica, sve s namjenom eliminacije ponavljajućih podatak, uštede prostora, poboljšanja performansi i povećanja integriteta podataka. Ovo također podrazumijeva da se podatci nalaze "posvuda". Upravo se iz tog razloga za dohvaćanje podataka iz više tablica koristi JOIN *Clause.*
Unutarnji spojevi (eng. *inner joins*) su spojevi koji se najčešće koriste. Ti spojevi povezuju dvije tablice na temelju vrijednosti u kojima se podudaraju.

Potrebno je ispisati tablicu koja sadrži Ime i Prezime zaposlenika, Naziv odjela u kojem je zaposlen, te Datum početka rada u odjelu i Datum završetka rada.

Rješenje:

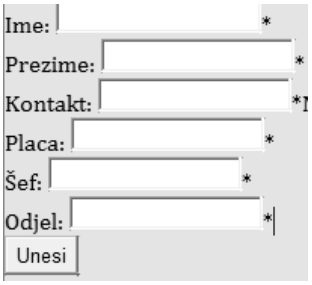
```
SELECT T_ZAPOSLENICI.IME, T_ZAPOSLENICI.PREZIME, T_ODJEL.NAZIV_ODJEL,
T_ZAPOSLENICI_ODJELI.DATUM_POCETKA_RADA_U_ODJELU, T_ZAPOSLENICI_ODJELI.DATUM_ZAVRSETKA_RADA_U_ODJELU
FROM
T_ZAPOSLENICI
INNER JOIN
T_ZAPOSLENICI_ODJELI ON
T_ZAPOSLENICI.ID_ZAPOSLENIK = T_ZAPOSLENICI_ODJELI.ID_ZAPOSLENIK
INNER JOIN
T_ODJEL ON T_ZAPOSLENICI_ODJELI.ID_ODJEL = T_ODJEL.ID_ODJEL
```

Slika 12

IME	PREZIME	NAZIV ODJELA	DATUM POČETKA RADA	DATUM ZAVRŠETKA RADA
Pero	Peričić	Računovodstvo	13.4.2006. 0:00:00	
Ana	Anić	Računovodstvo	1.1.2010. 0:00:00	
Marko	Markić	Odjel vođenja projekata	1.2.2009. 0:00:00	31.8.2010. 0:00:00
Marko	Markić	Odjel razvoj proizvoda	1.9.2010. 0:00:00	
Dino	Dinić	Odjel upravljanja kvalitetom	1.3.2000. 0:00:00	
Tina	Tinić	Odjel upravljanja kvalitetom	1.5.2005. 0:00:00	

Slika 13

Web forma Upravljanje bazom, koja je dostupna samo administratoru sadrži TextBox polja koja služe za unos podataka u bazu. Za TextBox-ove su upotrebljene dvije spremljene procedure jer se podaci o odjelu nalaze u zasebnoj tablici. Pri unosu podataka izvršava se određena validacija, odnosno kontrolira se jesu li uneseni podaci ispravni. Nakon što unesemo odgovarajuće podatke potvrđujemo unos na botun Unesi. Tada se pokreće metoda btnUnesi_Click kojom se vrši povezivanje na bazu podataka pomoću SqlConnection.



Slika 14- Unos podataka

U ovom projektu korišten je DataSet koji se dodaje u folder AppCode. U DataSet-u su napravljene spremljene procedure koje olakšavaju unos podataka u bazu. Unos podataka se odvija pomoću spremljenih procedura. Za unos nam služe polja TextBox, DropDownList, a izvor podataka joj je ObjectDataSource, tablice Zaposlenici i Odjel iz baze podataka.Poziva

se spremljena procedura kojom se podaci spremaju u bazu, učitavaju se uneseni podaci na stranici koji se spremaju i procedura se izvršava i prekida se veza s bazom. Korištene su If-else petlje jer su neki podaci u bazi navedeni kao Null vrijednost, što znači da nije obavezan njihov unos.

SQL DELETE stranica sastoji se od TextBoxa u koji je predviđen unos neke SQL izjave, gumba „Pokreni“ i GridView na kojem se prikazuju rezultati.

```
protected void Button1_Click(object sender,
EventArgs e)
    {
        SqlConnection cmd = new
SqlConnection(ConfigurationManager.AppSettings["con
nStr"]);
        try
        {
           string str = TextBox1.Text;
            int index = str.IndexOf("DELETE");
            if (index >= 0)
            {
                SqlCommand xp = new SqlCommand(str,
cmd);
                cmd.Open();
                xp.ExecuteNonQuery();
                cmd.Close();
            }
            else
            {
                MessageBox.Show("Upit mora
sadržavati ključnu riječ DELETE");
            }

        }
```

```
            catch
            {
                GridZaposlenik.DataSource = null;
                GridZaposlenik.DataBind();
                Label1.Text = "Pogrešan unos";

            }
        }
    }
```

Kod 2

Prethodni kod služi za dohvaćanje SQL naredbe iz TextBox-a. Svi upiti moraju sadržavati ključnu riječ DELETE.

4.5. Vrednovanje

Vrednovanje je faza koja treba biti zastupljena kroz sve četiri prethodne faze. Formativno u cilju identifikacije mogućih pogrešaka. Uglavnom se odnosi na ocjenu aktivnosti učenja i poučavanja, te rezultat učenja pojedinog korisnika. Na aplikaciji je provedeno alfa testiranje, te je omogućeno slanje maila administratoru u slučaju otkrivenih grešaka. Sumativno vrednovanje bi bilo ocjenjivanje uspješnosti koje također sadrži pregledavanje svih prethodno izvršenih formativnih vrednovanja. [12] Obzirom da nije predviđeno ocjenjivanje u ovom stadiju sustava budući ciljevi ovog sustava su unaprijediti sustav sa provjerom znanja kojim će se moći vrednovati stečeno znanje.

Zaključak

Aplikacija za poučavanje SQL-a ostvarena je kao mrežna aplikacija. Prije izrade same aplikacije modelirana je i implementirana baza podataka. Mrežna aplikacija je ostvarena u ASP.NET (Framework 4.0) tehnologiji, a pozadinski kod pisan je u C# programskom jeziku, HTML-u, JaviScript, te SQL jeziku. SQL jezik ne zahtijeva puno vremena za shvaćanje (sigurno manje nego ostali jezici). Koristi se objektno orijentirani jezik jer alat predstavlja relacijsku bazu podataka objektno. Provedeno je samo alfa testiranje, tijekom izrade na razvojnom poslužitelju Microsoft Visual Studio 2012, te nakon instalacije na lokalni poslužitelj je testirana u web preglednicima Google Chrome, Mozilla Firefox i Internet Explorer. Aplikacija omogućuje pretraživanje baze, unošenje novih podataka u bazu, brisanje i uređivanje, prijavu i registraciju korisnika. Omogućuje sigurnu i efikasnu manipulaciju podacima u relacijskoj bazi podataka. Baza i aplikacija su napravljene tako da se mogu nadograditi za neke nove ideje.

Literatura

[1] D. STOCKLEY, E-learning Definition and Explanation, 2003..

[2] M. Rosić, Zasnivanje sustava obrazovanja na daljinu unutar informacijske infrastrukture (magistarski rad), Zagreb, 2000.

[3] S. Stankov, Inteligentni sustavi: teorija i primjena, Split: PMF Split, 2010.

[4] S. Anđelić, „Instrukcijski dizajn u e-obrazovanju“,.

[5] G. B. i. dr., „Taksonomija modela instrukcijskog dizajna i IMS LD standard“.

[6] M. Varga, Baze podataka, Zagreb, 1994.

[7] R. Manger, Baze podataka, Zagreb, 2012.

[8] »http://www.link-university.com/lekcija/Standardni-upitni-jezik---SQL/5505,« [Mrežno].

[9] »http://www.w3schools.com/sql/sql_datatypes_general.asp,« [Mrežno].

[10] »http://www.asls.com.au/addie-tell-me-more/,« [Mrežno].

[11] D. H. J. LIBERTY, Programiranje ASP.NET.

[12] S. S., „Inteligentni tutorski sustavi: teorija i primjena”, Split: PMF Spli, 2010.

[13] D. C. „. timline“, „ADDIE timline“, 1995..

Sažetak

U ovom završnom radu se govori općenito o e-učenju, bazama podataka, relacijskom modelu, SQL-u i modelima koji se koriste prilikom kreiranja programa za poučavanje. SQL je jezik koji služi za upravljanje podacima koji se nalaze u bazi podataka. Jednostavan je za korištenje jer relacije koje se kreiraju jednom naredbom odmah su dostupne, a svi podaci i rezultati instrukcija se prikazuju tablicama. Za lakše razumijevanje teksta priložen je kod. Nadalje, pisano je o ADDIE modelu. Ovaj model, instrukcijskog dizajna, korišten je u dizajniranju i razvoju edukacijskog programa za poučavanje SQL jezika. Prikazano je kako se mogu postavljati različiti upiti nad različitim tipom podataka koristeći jedinstvenu sintaksu koja je ugrađena u programske jezike kao što je C#. Microsoft Visual Studio 2012 je korišten program prilikom izrade mrežne aplikacije i pisanja SQL upita.

Summary

This document provides general facts about e-learning, databases, relational model, structured query language (SQL) and models that are used when creating programs for teaching. SQL is programming language for retriving and storing informations and all kind of database manipulation. It is easy to learn and use, because the relations created with one command are immediately available. All data and instruction results are displayed by tables. For easier understanding of the topics covered by this paper it includes some parts of code and examples from project. Furthermore, it covers the ADDIE model. “ADDIE” stands for Analyze, Design, Develop, Implement and Evaluate. This model, instructional design method, is used in designing and developing educational and training program for teaching SQL. It shows how different data domains can be queried by using uniform syntax which is integrated into programming languages such as C#. Microsoft Visual Studio 2012 tools were used when creating network applications and writing SQL queries.