

MySQL radionica

Thomas Pettit | Scott Cosentino

 kompjuter
biblioteka

 Packt

Izdavač:



**kompjuter
biblioteka**

Obalskih radnika 4a, Beograd

Tel: 011/2520272

e-mail: kombib@gmail.com

internet: www.kombib.rs

Urednik: Mihailo J. Šolajić

Za izdavača, direktor:

Mihailo J. Šolajić

Autori: Thomas Pettit

Scott Cosentino

Prevod: Biljana Tešić

Lektura: Miloš Jevtović

Slog: Zvonko Aleksić

Znak Kompjuter biblioteke:

Miloš Milosavljević

Štampa: „Pekograf“, Zemun

Tiraž: 500

Godina izdanja: 2022.

Broj knjige: 556

Izdanje: Prvo

ISBN: 978-86-7310-579-6

The MySQL Workshop

Copyright © 2022 Packt Publishing

ISBN 978-1-83921-490-5

Copyright © 2022 April Packt Publishing

All right reserved. No part of this book may be reproduced or transmitted in any form or by means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Autorizovani prevod sa engleskog jezika edicije u izdanju „Packt Publishing“, Copyright © April 2022.

Sva prava zadržana. Nije dozvoljeno da nijedan deo ove knjige bude reprodukovan ili snimljen na bilo koji način ili bilo kojim sredstvom, elektronskim ili mehaničkim, uključujući fotokopiranje, snimanje ili drugi sistem presnimavanja informacija, bez dozvole izdavača.

Zaštitni znaci

Kompjuter Biblioteka i „Packt Publishing“ su pokušali da u ovoj knjizi razgraniče sve zaštitne oznake od opisnih termina, prateći stil isticanja oznaka velikim slovima.

Autor i izdavač su učinili velike napore u pripremi ove knjige, čiji je sadržaj zasnovan na poslednjem (dostupnom) izdanju softvera. Delovi rukopisa su možda zasnovani na predizdanju softvera dobijenog od strane proizvođača. Autor i izdavač ne daju nikakve garancije u pogledu kompletnosti ili tačnosti navoda iz ove knjige, niti prihvataju ikakvu odgovornost za performanse ili gubitke, odnosno oštećenja nastala kao direktna ili indirektna posledica korišćenja informacija iz ove knjige.

CIP - Каталогизација у публикацији
Народна библиотека Србије, Београд

004.655.3SQL

ПЕТИТ, Томас

MySQL radionica / Thomas Pettit, Scott Cosentino ;
[prevod Biljana Tešić]. - Izd. 1. - Beograd: Kompjuter
biblioteka, 2022 (Zemun: Pekograf). - XXI, 694 str.:
ilustr.; 24 cm. - (Kompjuter biblioteka; br. knj. 556)

Prevod dela: The MySQL Workshop. - Tiraž 500. -
O autorima: str. III. - Registar.

ISBN 978-86-7310-579-6

1. Косентино, Скот [аутор]

а) Програмски језик „SQL“

COBISS.SR-ID 71553289

O AUTORIMA

Thomas Pettit je počeo da razvija softvere iz hobija. Od vozača kamiona postao je softverski programer, tako što je diplomirao razvoj softvera u svojoj 35. godini. Predavao je osnovne računarske veštine u Centru za obrazovanje odraslih u Melburnu dve godine, pre nego što je započeo svoju karijeru softverskog programera. Radio je za nekoliko vladinih agencija, uključujući agencije za odbranu, sprovođenje zakona i transport, ali i za velika i mala privatna preduzeća. Tokom svoje karijere bio je mentor nekim novim softverskim programerima i veoma uživa u pomaganju drugima u poboljšanju njihovih veština i unapređenju šansi za njihove karijere.

Scotty Cosentino je softverski programer i nastavnik koji se trenutno bavi računarskom bezbednošću. Intenzivno koristi programske jezike niskog i visokog nivoa u aplikacijama na nivou operativnih sistema i preduzeća. Voli da podučava i trenutno piše i kreira video-snimke o računarskoj bezbednosti i drugim temama programiranja. Razvio je obimnu biblioteku kurseva i podučavao je više od 45.000 studenata na kursovima izdavačkih kuća „Udemy, „Packt“ i „CodeRed“. Vodi blog na Mediumu i aktivan je na YouTubeu i LinkedInu, na kojima uživa u kreiranju sadržaja i interakciji sa studentima.

O RECENZENTU

Vlad Sebastian Ionescu je univerzitetski predavač sa doktoratom iz mašinskog učenja, ali radi i kao samostalni softverski inženjer. Ima više od 10 godina iskustva u nastavi informatike u različitim „ulogama“: bio je nastavnik u školi, privatni tutor, mentor na stažiranju, univerzitetski tehnički savetnik i predavač. Godinama koristi mnoge najsavremenije tehnologije u raznim oblastima, kao što su razvoj frontenda, dizajn i administriranje baze podataka, backend programiranje i mašinsko učenje.

„PACKT“ TRAŽI AUTORE KAO ŠTO STE VI

Ako ste zainteresovani da postanete autor za „Packt“, prijavite se na stranicu authors.packtpub.com. Saradujemo sa hiljadama programera i tehničkih profesionalaca da bismo im pomogli da podele svoje mišljenje sa globalnom tehničkom zajednicom. Možete da podnesete osnovnu prijavu, da se prijavite za specifičnu temu za koju tražimo autore ili da pošaljete neke svoje ideje.

Kratak sadržaj

PREDGOVOR	XV
DEO 1	
Kreiranje baze podataka	1
POGLAVLJE 1	
Osnovni koncepti	3
POGLAVLJE 2	
Kreiranje baze podataka	23
POGLAVLJE 3	
Korišćenje SQL-a za rad u bazi podataka	83
POGLAVLJE 4	
Izbor, agregacija i primena funkcija	129
DEO 2	
Upravljanje bazama podataka	157
POGLAVLJE 5	
Korelacija podataka u tabelama	159
POGLAVLJE 6	
Uskladištene procedure i drugi objekti	199

POGLAVLJE 7**Kreiranje klijenata baze podataka na platformi Node.js 233****POGLAVLJE 8****Upotreba podataka pomoću platforme Node.js 281****DEO 3****Ispitivanje baze podataka 329****POGLAVLJE 9****Microsoft Access - deo 1 331****POGLAVLJE 10****Microsoft Access – deo 2..... 371****POGLAVLJE 11****MS Excel VBA i MySQL – Deo 1 413****POGLAVLJE 12****MS Excel VBA i MySQL – Deo 2 465****DEO 4****Zaštita baze podataka 513****POGLAVLJE 13****Unošenje podataka u MySQL 515****POGLAVLJE 14****Manipulisanje korisničkim dozvolama 567****POGLAVLJE 15****Logičke rezervne kopije..... 587****DODATAK 609****INDEKS 685**

Sadržaj

PREGOVOR.....	XV
----------------------	-----------

DEO 1

Kreiranje baze podataka	1
--------------------------------------	----------

POGLAVLJE 1

Osnovni koncepti.....	3
------------------------------	----------

Uvod u bazu podataka.....	4
Arhitektura baze podataka.....	5
MS Access kao baza podataka.....	6
Sistem za upravljanje bazama podataka.....	7
RDBMS.....	8
Istraživanje MySQL-a	9
Tipovi podataka.....	10
Vežba 1.01 – Organizovanje podataka u relacionom formatu.....	11
Istraživanje MySQL arhitekture.....	12
MySQL slojevi	13
Aplikacioni sloj – veza sa klijentom.....	14
MySQL serverski sloj (logički sloj)	14
Sloj mašine za skladištenje (fizički sloj)	15
Mašine za skladištenje (InnoDB i MyRocks).....	16
ACID usklađenost.....	16
Modelovanje podataka	17
Normalizacija.....	19
Aktivnost 1.01 - Kreiranje optimizovane tabele za projekat zaposlenih	21
Rezime	22

POGLAVLJE 2**Kreiranje baze podataka 23**

Razvijanje baza podataka	24
MySQL Workbench GUI	24
Povezivanje GUI Workbencha sa MySQL-om	27
Vežba 2.01 – Kreiranje veze sa MySQL Workbench GUI-om.....	27
Pristup MySQL-u pomoću interfejsa sa komandne linije	32
Kreiranje baze podataka	32
Vežba 2.02 – Kreiranje baze podataka autoclub.....	33
Korišćenje Workbencha za dodavanje tabele	34
Uvoz objekata iz SQL datoteke skripta.....	40
Vežba 2.03 – Uvoz tabele iz SQL datoteke skripta	41
Indeksi MySQL tabele i spoljni ključevi	45
Indeksi	45
Vežba 2.04 – Kreiranje indeksa.....	46
Indeksi na više kolona	49
Spoljni ključevi	51
Vežba 2.05 – Kreiranje spoljnog ključa.....	53
Obrnuti inženjering baze podataka	58
Vežba 2.06 – Kreiranje EER modela iz baze podataka autoclub.....	61
Vežba 2.07 – Korišćenje EER dijagrama i direktnog inženjeringa za upravljanje modelom baze podataka	67
Vežba 2.08 – Unošenje izmena modela u proizvodnu bazu podataka pomoću Synchronize Modela	77
Aktivnost 2.01 – Modifikacija EER dijagrama, modela i baze podataka	81
Rezime	82

POGLAVLJE 3**Korišćenje SQL-a za rad u bazi podataka 83**

Uvod u rad u bazama podataka pomoću SQL-a	84
Upotreba podataka.....	85
Tipovi SQL iskaza	85
Pravljenje rezervnih kopija baza podataka	86
Vežba 3.01 – Pravljenje rezervne kopije baze podataka autoclub.....	88
Vraćanje baza podataka u prethodno stanje	89
Vežba 3.02 – Vraćanje baze podataka autoclub u prethodno stanje.....	91
Rad u SQL kodu za održavanje baze podataka.....	92
Kreiranje nove baze podataka	92
Vežba 3.03 – Kreiranje nove baze podataka	93
Kreiranje i modifikacija tabele.....	95
Vežba 3.04 – Kreiranje nove tabele	95
SQL upiti za kreiranje indeksa i spoljnih ključeva	99
Vežba 3.05 – Kreiranje tabele sa indeksima i spoljnim ključevima	100
Aktivnost 3.1 – Kreiranje tabele sa indeksima i spoljnim ključevima	105

Promena upita u tabeli	106
Vežba 3.06 – Modifikacija postojeće tabele	107
Dodavanje podataka u tabelu	110
Vežba 3.07 – Dodavanje jednog zapisa u tabelu members	110
Ažuriranje podataka u zapisu.....	113
Vežba 3.08 – Ažuriranje zapisa	113
Brisanje podataka iz tabela	116
Upiti drop.....	117
Objekti blob, datoteke i putanje datoteka	118
Vežba 3.09 – Datoteke i objekti blob	119
Datoteke i putanje datoteka.....	121
Vežba 3.10 – Datoteke i putanje datoteka.....	123
Aktivnost 3.2 – Dodavanje putanja datoteka slika u bazu podataka.....	126
Rezime	128

POGLAVLJE 4

Izbor, agregacija i primena funkcija 129

Uvod u upite za podatke.....	130
Upiti za tabele u MySQL-u	130
Vežba 4.01 – Upotreba jednostavnih upita	132
Filtriranje rezultata	133
Vežba 4.02 – Filtriranje rezultata	136
Primena funkcija na podatke.....	138
Matematičke funkcije	138
Funkcije za znakovne nizove	140
Funkcije za datum i vreme.....	142
Vežba 4.03 – Korišćenje funkcija	144
Agregiranje podataka	147
Vežba 4.04 – Agregiranje podataka	150
Iskazi case	152
Vežba 4.05 – Pisanje iskaza case.....	153
Aktivnost 4.01 – Prikupljanje informacija za turistički članak.....	154
Rezime	155

DEO 2

Upravljanje bazama podataka 157

POGLAVLJE 5

Korelacija podataka u tabelama 159

Uvod u obradu podataka u više tabela.....	160
Spajanje dve tabele	160
Slučajni unakrsni spojevi.....	163

LEFT JOIN i INNER JOIN	164
Vežba 5.01 - Spajanje dve tabele.....	168
Analiziranje podupita.....	169
Zavisni podupiti.....	170
Vežba 5.02 – Upotreba podupita	171
Uobičajeni izrazi tabela	173
Rekurzivni CTE.....	174
Vežba 5.03 – Upotreba CTE-a	178
Analiza performansi upita pomoću EXPLAIN-a	182
Vežba 5.04 - Upotreba EXPLAIN-a.....	189
Aktivnost 5.01 – Video-klub Sakila.....	195
Aktivnost 5.02 - Generisanje liste godina	197
Rezime	198

POGLAVLJE 6

Uskladištene procedure

i drugi objekti 199

Uvod u objekte baze podataka	200
Istraživanje različitih objekata baze podataka	200
Upotreba prikaza	201
Vežba 6.01 – Kreiranje mejling liste pomoću prikaza	201
Prikazi koji se mogu ažurirati	204
Aktivnost 6.01 – Ažuriranje podataka u prikazu.....	206
Upotreba korisnički-definisanih funkcija	207
Vežba 6.02 – Kreiranje funkcije	208
Upotreba uskladištenih procedura	211
Vežba 6.03 – Kreiranje uskladištene procedure	212
Vežba 6.04 – Uskladištene procedure i parametri.....	213
Upotreba ključnih reči IN, OUT i INOUT	215
Vežba 6.05 – IN i INOUT	216
Istraživanje okidača.....	221
Prednosti okidača	221
Nedostaci okidača.....	222
Ograničenja u okidačima	222
Vežba 6.06 – Okidači za sprovođenje poslovnih pravila.....	222
Upotreba transakcija	228
Vežba 6.07 – Implementacija transakcije	229
Rezime	231

POGLAVLJE 7

Kreiranje klijenata baze podataka na platformi Node.js 233

Uvod u upravljanje bazom podataka pomoću platforme Node.js.....	234
Najbolje tehnike za razvoj SQL klijenta.....	235
Instaliranje razvojnog MySQL servera	235
Kreiranje razvojnog MySQL servera	236

Pravljenje rezervne kopije pre unošenja izmena.....	237
Vraćanje baze podataka u prethodno stanje.....	239
Oporavak od slučajnog brisanja podataka.....	240
Vežba 7.01 – Bezbedno brisanje zapisa.....	241
JavaScript koji koristi Node.js.....	243
Podešavanje platforme Node.js.....	244
Početak rada na platformi Node.js.....	248
Osnove platforme Node.js.....	251
Vežba 7.02 – Osnovni izlaz u konzoli.....	253
Vežba 7.03 – Testiranje izlaza u pregledaču.....	254
Upisivanje izlaza u datoteke.....	256
Vežba 7.04 - Upisivanje u datoteku na disku.....	257
Povezivanje na MySQL.....	259
Vežba 7.05 – Povezivanje na MySQL server.....	261
Rešavanje grešaka u povezivanju.....	263
Modularizacija MySQL veze.....	267
Vežba 7.06 – Modularizacija MySQL veze.....	268
Kreiranje baza podataka pomoću platforme Node.js.....	270
Vežba 7.07 – Kreiranje nove baze podataka.....	271
Kreiranje tabela pomoću platforme Node.js.....	273
Vežba 7.08 – Kreiranje tabele u bazi podataka.....	274
Aktivnost 7.01 – Izrada aplikacije baze podataka pomoću platforme Node.js.....	276
Rezime.....	279

POGLAVLJE 8

Upotreba podataka pomoću platforme Node.js 281

Interakcija sa bazama podataka.....	282
Umetanje zapisa u Node.js aplikaciju.....	282
Vežba 8.01 – Umetanje zapisa u tabelu.....	284
Umetanje više zapisa.....	288
Vežba 8.02 – Umetanje više zapisa u tabelu.....	288
Umetanje pomoću više polja.....	292
Vežba 8.03 – Popunjavanje zapisa iz postojećih tabela.....	293
Ažuriranje zapisa tabele.....	296
Vežba 8.04 – Ažuriranje jednog zapisa.....	298
Aktivnost 8.01 – Višestruka ažuriranja.....	302
Prikaz podataka u pregledačima.....	304
Vežba 8.05 – Formatiranje podataka u veb pregledaču.....	307
ODBC veze.....	312
Tipovi DSN-ova.....	313
Šta je ovim DSN-ovima zajedničko?.....	315
Utvrđivanje da li su ODBC drajveri instalirani.....	315
Lokalne, LAN i udaljene ODBC veze.....	318
Lokalni ODBC (server je na vašem računaru).....	318
LAN ODBC.....	318
Udaljeni ODBC.....	318

Vežba 8.06 – Kreiranje LAN ili udaljene DSN/ODBC veze sa bazom podataka world_statistics	319
Kreiranje DSN datoteka/ODBC veza	322
Vežba 8.07 – Kreiranje DSN-a zasnovanog na datotekama /ODBC veze sa bazom podataka world_statistics	323
Aktivnost 8.02 – Kreiranje baze podataka o kupcima	326
Rezime	328

DEO 3

Ispitivanje baze podataka 329

POGLAVLJE 9

Microsoft Access - deo 1 331

Uvod u MS Access.....	332
Konfiguracije MS Access aplikacije baze podataka	333
Povećanje MS Access baze podataka u MySQL-u.....	335
Ručni izvoz MS Access tabela	340
Vežba 9.02 – Ručno povećanje tabele.....	341
Podešavanje svojstava polja.....	344
Vežba 9.03 - Ručna migracija tabela i podešavanje njihovih svojstava polja	345
Migracija pomoću „čarobnjaka“ (wizards).....	348
Vežba 9.04 – Korišćenje Workbench Migration Wizarda za povećanje tabele	349
Povezivanje sa tabelama i prikazima.....	361
Vežba 9.05 Povezivanje dobre MySQL tabele sa Accessom.....	362
Vežba 9.07 - Povezivanje problematične MySQL tabele sa Accessom.....	366
„Osvežavanje“ povezanih MySQL tabela	368
Aktivnost 9.01 - Povezivanje preostalih MySQL tabela sa MS Access bazom podataka	369
Rezime	370

POGLAVLJE 10

Microsoft Access – deo 2..... 371

Uvod u MS Access.....	372
Migracija MS Access aplikacije u MySQL.....	372
Prolazni upiti.....	372
Vežba 10.01 – Prosleđivanje (jednostavna SQL konverzija)	373
Aktivnost 10.01 – Konvertovanje statistike o polu i zapošljavanju	377
Pozivanje MySQL funkcija.....	379
Vežba 10.02 - Prosleđivanje (pozivanje MySQL funkcija)	379
Aktivnost 10.02 - Kreiranje funkcije i njeno pozivanje	380
Pozivanje MySQL uskladištenih procedura.....	382
Vežba 10.03 – Pozivanje MySQL uskladištene procedure.....	382
Aktivnost 10.03 – Kreiranje MySQL uskladištenih procedura i	

njihovo korišćenje u VBA-u	386
Korišćenje parametara	387
Parametrizovane uskladištene procedure	388
Vežba 10.04 – Parametrizovana uskladištena procedura (lista series)	388
Aktivnost 10.04 – Parametrizovana uskladištena procedura (lista series)	390
Vežba 10.05 – Uskladištena procedura sa više parametara (lista country)	391
Aktivnost 10.05 – Uskladištena procedura sa više parametara (lista date).....	395
Vežba 10.06 – Uskladištena procedura sa više parametara (unakrsni upiti)	396
Obrazac Bad Bits	401
Vežba 10.07 – Demonstracija obrasca Bad Bits.....	402
Nevezani obrasci	404
Drugi način da uklonite obrazac iz povezanih tabela.....	410
Vežba 10.08 – Uklanjanje svih povezanih tabela.....	410
Rezime	412

POGLAVLJE 11

MS Excel VBA i MySQL – Deo 1 413

Uvod u Excel	414
Vežba 11.01 – Podešavanje primera MySQL baze podataka	415
Istraživanje ODBC veze	417
Meni Developer	417
Vežba 11.02 – Aktiviranje kartice Developer i VBA IDE-a.....	417
Istraživanje Excel VBA strukture	421
Priprema Excel projekta.....	421
Vežba 11.03 – Kreiranje modula koda	422
Učenje o VBA bibliotekama	426
Vežba 11.04 – Referenciranje biblioteke	426
Vežba 11.05 – Umetanje radnih listova.....	429
Povezivanje sa MySQL bazom podataka pomoću VBA-a	431
Postavljanje okruženja	431
Vežba 11.06 – Promenljiva veze	432
Funkcije veze u VBA	434
Vežba 11.07 – Kreiranje funkcije veze	435
Čitanje podataka iz MySQL-a pomoću VBA-a.....	440
Vežba 11.08 – ReadGenreSales	440
Vežba 11.09 – Padajuća lista Genre	447
Funkcije automatskog pokretanja tokom otvaranja radne sveske.....	450
Vežba 11.10 – Funkcije automatskog pokretanja pri otvaranju radne sveske	451
Popunjavanje dijagrama.....	452
Popunjavanje grafikona – Prodaja žanrova.....	452
Vežba 11.11 – Učitavanje podataka iz dijagrama Genre Sales	453
Pokretanje koda za izmene u dokumentu	456
Vežba 11.12 – Detektovanje i upravljanje izmenama radnog lista	456
Aktivnost 11.01 – Kreiranje dijagrama (artist track sales)	461
Rezime	464

POGLAVLJE 12**MS Excel VBA i MySQL – Deo 2 465**

Uvod u MySQL veze	466
Povezivanje sa MySQL bazom podataka pomoću ODBC-a.....	466
Vežba 12.01 – Kreiranje funkcije DSN veze.....	467
Istraživanje generičkih funkcija za čitanje podataka	475
Vežba 12.02 – Generički čitač podataka	476
Kreiranje veza sa MySQL-om u Excelu	480
Vežba 12.03 – Kreiranje veze sa MySQL-om.....	480
Umetanje podataka pomoću plugina MySQL for Excel.....	484
Vežba 12.04 – Umetanje 25 najprodavanijih albuma izvođača	484
Ažuriranje podataka pomoću plugina MySQL for Excel	488
Vežba 12.05 – Ažuriranje MySQL podataka o zaposlenima.....	489
Prosleđivanje podataka iz Excela.....	494
Vežba 12.06 – Prosleđivanje podataka iz Excela u novu MySQL tabelu	494
Izvedene tabelle.....	498
Vežba 12.07 – Prodaja albuma.....	498
Aktivnost 12.01 – Izrada Excel dokumenta zasnovanog na MySQL-u.....	511
Rezime	511

DEO 4**Zaštita baze podataka 513****POGLAVLJE 13****Unošenje podataka u MySQL 515**

Uvod u pripremu podataka	516
Rad u X DevAPI-u.....	516
Primer X DevAPI-a.....	520
Upotreba MySQL Shella u X DevAPI-u	520
Vežba 13.01 – Umetanje vrednosti pomoću MySQL Shella u JS režimu.....	521
Umetanje dokumenata	523
Vežba 13.02 – Umetanje dokumenata u tabelu	525
Učitavanje podataka iz SQL datoteke	527
Vežba 13.03 – Učitavanje podataka iz SQL datoteke i prikaz tabela	528
Vežba 13.04 – Uvoz SQL datoteke pomoću MySQL Workbencha.....	530
Učitavanje podataka iz CSV datoteke	533
Format SELECT...INTO OUTFILE.....	533
Format LOAD DATA INFILE...INTO.....	534
Vežba 13.05 – Učitavanje podataka iz CSV datoteke	534
Učitavanje podataka iz JSON datoteke.....	536
Vežba 13.06 – Učitavanje podataka iz JSON datoteke	537
Upotreba CSV mašine skladištenja za izvoz podataka	539
Vežba 13.07 – Upotreba CSV mašine skladištenja za izvoz podataka.....	540
Upotreba CSV mašine skladištenja za uvoz podataka	542

Vežba 13.08 – Upotreba CSV mašine skladištenja za uvoz podataka	543
Pretraživanje i filtriranje JSON dokumenata	545
Vežba 13.09 – Pretraživanje kolekcija i filtriranje dokumenata	553
Korišćenje JSON funkcija i operatora za ispitivanje JSON kolona	556
Vežba 13.10 – Ispitivanje JSON podataka pomoću SQL-a	561
Upotreba generisanih kolona za upite i indeksiranje JSON podataka	563
Aktivnost 13.01 – Izvoz podataka izveštaja u CSV za Excel	565
Rezime	566

POGLAVLJE 14

Manipulisanje korisničkim dozvolama 567

Uvod u korisničke dozvole	568
Istraživanje korisnika i naloga	569
Kako da se povežete na MySQL pomoću skupa akreditiva	569
Zašto koristiti više korisničkih naloga?	570
Kreiranje, izmena i uklanjanje korisničkih naloga	570
Davanje dozvola	571
Provera korisnika	572
Vežba 14.01 – Kreiranje korisničkih naloga i davanje dozvola	575
Izmena korisničkih naloga	577
Flush privileges	578
Izmena dozvola	578
Vežba 14.02 – Izmjena korisničkih naloga i poništavanje dozvola	579
Korišćenje uloga	581
Vežba 14.03 – Korišćenje uloga za upravljanje dozvolama	582
Rešavanje problema pristupa	583
Aktivnost 14.01 – Kreiranje korisnika za upravljanje šemom world	585
Rezime	585

POGLAVLJE 15

Logičke rezervne kopije 587

Uvod u rezervne kopije	588
Učenje osnova pravljenja rezervnih kopija	588
Logička i fizička rezervna kopija	590
Tipovi vraćanja	590
Izvršavanje rezervnih kopija	591
Upotreba alatke mysqldump	591
Vežba 15.01 – Pravljenje rezervnih kopija pomoću mysqldumpa	593
Upotreba mysqlpumpa	595
Vežba 15.02 – Pravljenje rezervne kopije pomoću mysqlpumpa	596
Planiranje pravljenja rezervnih kopija	596
Potpuno vraćanje	599
Delimično vraćanje	599
Vežba 15.03 – Vraćanje jedne rezervne kopije šeme	600
Upotreba point-in-time oporavka pomoću binlog datoteka	601

GTID format.....	603
Upotreba mysqlbinloga za proveru sadržaja binloga	605
Aktivnost 15.01 – Pravljenje rezervne kopije i vraćanje jedne šeme	606
Rezime	608

DODATAK

Rešenje za aktivnost 1.1	609
Rešenje za aktivnost 2.1	612
Rešenje za aktivnost 3.1	615
Rešenje za aktivnost 3.2	619
Rešenje za aktivnost 4.1	621
Rešenje za aktivnost 5.1	624
Rešenje za aktivnost 5.2	628
Rešenje za aktivnost 6.1	631
Rešenje za aktivnost 7.1	632
Rešenje za aktivnost 8.1	637
Rešenje za aktivnost 8.2	642
Rešenje za aktivnost 9.1	648
Rešenje za aktivnost 10.1	649
Rešenje za aktivnost 10.2.....	652
Rešenje za aktivnost 10.3.....	654
Rešenje za aktivnost 10.4.....	657
Rešenje za aktivnost 10.5.....	659
Rešenje za aktivnost 11.1	662
Rešenje za aktivnost 12.1	667
Rešenje za aktivnost 13.1	674
Rešenje za aktivnost 14.1	678
Rešenje za aktivnost 15.1	679
Rešenje za aktivnost 15.2.....	681

INDEKS	685
---------------------	------------

Predgovor

Da li želite da naučite kako da efikasno kreirate i održavate baze podataka? Da li tražite jednostavne odgovore na osnovna MySQL pitanja i jednostavne primere koje možete koristiti na vašem poslu? Ako su vaši odgovori potvrdni, ova radionica je pravi izbor za vas.

U ovoj knjizi, koja je osmišljena tako da „izgradi“ vaše samopouzdanje, pomoću korisne prakse se koristi jednostavan pristup koji se fokusira na praktičnost, tako da možete odmah da pređete na rad, bez potrebe da čitate stranice i stranice dosadne, suvoparne teorije.

Zahvaljujući kratkim vežbama i aktivnostima, naučićete kako da koristite različite MySQL alatke za kreiranje baze podataka i za upravljanje podacima u njoj. Videćete kako se prenose podaci između MySQL baze podataka i drugih izvora i kako se koriste skupovi podataka iz realnog sveta da biste stekli dragoceno iskustvo u manipulaciji podacima i dobili informacije iz podataka. Dok budete čitali ovu knjigu, otkrićete kako da zaštitite svoju bazu podataka upravljanjem korisničkim dozvolama, izvršavanjem logičkih rezervnih kopija i vraćanjem u prethodno stanje.

Ako ste već pokušali da naučite SQL, ali niste mogli da pređete sa razumevanja jednostavnih upita na kreiranje projekata „uživo“ pomoću stvarnog sistema upravljanja bazom podataka, „MySQL radionica“ će vas odvesti na pravi put.

Kada pročitate ovu knjigu u celosti, imaćete znanje, veštine i samopouzdanje da unapredite svoju karijeru i da se uhvatite u koštac sa svojim ambicioznim projektima pomoću MySQL-a.

Za koga je ova knjiga?

Ova knjiga je za svakoga ko želi da nauči kako da koristi MySQL na produktivan i efikasan način. Ako ste potpuno novi u korišćenju MySQL-a, ova knjiga će

vam pomoći da počnete rad ili, ako ste ranije koristili MySQL, popuniće sve „praznine“, konsolidovati ključne koncepte i obezbediti dragocenu korisnu praksu. Prethodno poznavanje jednostavnog SQL-a ili osnovnih tehnika programiranja pomoći će vam da brzo shvatite koncepte koji se razmatraju u knjizi, ali ono nije neophodno.

Šta obuhvata ova knjiga?

U Poglavlju 1, „Osnovni koncepti“, predstavljeni su koncepti baza podataka, sistema za upravljanje bazama podataka, relacionih baza podataka i opšte strukture MySQL-a.

U Poglavlju 2, „Kreiranje baze podataka“, razmatramo kako se kreira baza podataka u MySQL-u. Naučićete kako da kreirate bazu podataka i tabelu, kako da postavite indekse i ključeve i kako da modelujete sisteme baza podataka, koristeći ER i EER dijagrame.

U Poglavlju 3, „Korišćenje SQL-a za rad u bazi podataka“, prikazano je kako se SQL može koristiti za rad u MySQL bazama podataka. Videćete načine za pravljenje rezervnih kopija i vraćanje baza podataka. Takođe ćemo razmotriti načine za kreiranje baza podataka i tabela, ali i za umetanje podataka, ažuriranje podataka, menjanje struktura tabela, skraćivanje tabela, brisanje podataka i ispuštanje tabela.

U Poglavlju 4, „Izbor, agregacija i primena funkcija“, razmotreni su metodi odabira i analize podataka iz baza podataka. Pogledaćete izbor i filtriranje podataka, ali i metode za primenu funkcija i agregiranja na podatke.

U Poglavlju 5, „Korelacija podataka u tabelama“, predstavljeni su metodi spajanja tabela. Takođe ćete upoznati podupite i uobičajene izraze tabela.

U Poglavlju 6, „Uskladištene procedure i drugi objekti“, razmotreni su različiti tipovi objekata baze podataka koji postoje u MySQL-u, uključujući prikaze, funkcije, procedure skladištenja, okidače i transakcije.

U Poglavlju 7, „Kreiranje klijenata baze podataka na platformi Node.js“, upoznaćete metode korišćenja platforme Node.js u MySQL bazi podataka. Pogledaćete podešavanje razvojnih MySQL servera, osnove platforme Node.js i metode povezivanja sa MySQL-om za kreiranje baza podataka i tabela.

Poglavlje 8, „Upotreba podataka pomoću platforme Node.js“, proširiće vaše znanje o korišćenju platforme Node.js za interfejs zajedno sa MySQL-om. Videćete kako možete da umetnete, da ažurirate i da prikazete podatke pomoću platforme Node.js. Takođe ćete naučiti kako da podesite i koristite ODBC veze.

U Poglavlju 9, „MS Access, deo 1“, prikazano je kako možemo da se povežemo sa MySQL-om pomoću MS Accessa. Pogledaćete metode za konfigurisanje MS Accessa, podešavanje svojstava polja i migraciju podataka za povezivanje sa MySQL-om.

U Poglavlju 10, „MS Access, deo 2“, obrađene su napredne teme interakcije MS Accessa sa MySQL-om, uključujući upotrebu prolaznih upita, pozivanje MySQL objekata i upotrebu MS Access obrazaca.

U Poglavlju 11, „MS Excel VBA i MySQL, deo 1“, koristimo MS Excel pomoću VBA za povezivanje sa MySQL bazama podataka radi preuzimanja i izmene podataka.

Poglavlje 12, „MS Excel VBA i MySQL, deo 2“, proširiće vaše znanje o MS Excelu kako biste mogli da razmotrite metode čitanja, umetanja, ažuriranja i prosleđivanja podataka iz Excela u MySQL.

U Poglavlju 13, „Dodatne aplikacije MySQL-a“, razmotrene su različite aplikacije koje možemo koristiti da bismo unapredili svoje MySQL veštine i sposobnosti. Naučićete kako da koristite X DevAPI i ispitaćete koncepte, kao što su umetanje dokumenata, učitavanje podataka iz CSV-ova i izvoz/uvoz različitih formata datoteka.

U Poglavlju 14, „Korisničke dozvole“, prikazano je kako se korisničke dozvole koriste za omogućavanje bezbednog pristupa MySQL bazama podataka. Videćete kako se kreiraju korisnički nalozi, kako se dodeljuju dozvole i kako se korisnički nalozi mogu koristiti u MySQL bazi podataka.

U Poglavlju 15, „Logičke rezervne kopije“, prikazano je kako se kreiraju logičke rezervne kopije u MySQL-u. Učićete o različitim tipovima vraćanja rezervnih kopija i metodima za zakazivanje kreiranja rezervnih kopija na MySQL serveru.

Izvucite maksimum iz ove knjige

SOFTVER/HARDVER KOJI JE RAZMATRAN U KNJIZI

MySQLCommunity Server 8.0.28

MySQL Workbench 8.0.28

Node.js 16.14.2

MS Office 2016

ZAHTEVI OPERATIVNOG SISTEMA

Windows, macOS ili Linux

Preuzimanje datoteka primera koda

Možete da preuzmete datoteke sa primerima koda za ovu knjigu sa adrese GitHuba

<https://github.com/PacktWorkshops/The-MySQL-Workshop/>

Kod, ako postoji njegovo ažuriranje, biće ažuriran u GitHub spremištu.

Preuzmite slike u boji za ovu knjigu

Takođe smo vam obezbedili PDF datoteku koja sadrži slike u boji ekrana/ dijagrama upotrebljenih u ovoj knjizi. Tu datoteku možete da preuzmete sa adrese

https://static.packt-cdn.com/downloads/9781839214905_ColorImages.pdf

Korišćene konvencije

U ovoj knjizi se koristi niz konvencija.

Kod u tekstu - Označava kodne reči u tekstu, nazive tabela baza podataka, nazive direktorijuma, nazive datoteka, ekstenzije datoteka, nazive putanja, lažne URL adrese, korisnički unos i Twitter postove. Evo primera: „Koristimo `WHERE` da bismo filtrirali redove koji nas zanimaju.“

Blok koda je postavljen na sledeći način:

```
SQL = "SELECT Count(capacityindicatorstats.ID) AS RecCount
FROM capacityindicatorstats;"
Call CreatePassThrough(SQL, "CISCount", True, False)
Set RS = CurrentDb.OpenRecordset("CISCount", dbOpenDynaset)
```

Kada treba da privučemo vašu pažnju na određeni deo bloka koda, relevantne linije ili stavke su ispisane podebljanim slovima:

```
RS.MoveFirst
Me.cntSeries = RS.Fields("SeriesCount")
RS.Close
```

Svi unosi ili ispisi komandne linije napisani su na sledeći način:

```
$ mkdir css
$ cd css
```

Podobljana slova - Novi termini, važne reči ili reči koje vidite na ekranu - na primer, u menijima ili okvirima za dijalog, prikazani su u tekstu **podobljanim slovima**. Na primer: „Ako nije, kliknite desnim tasterom miša na njega u panelu **Navigate** i izaberite **Design View**.“



Saveti ili važne napomene se prikazuju ovako.

Stupite u kontakt

Povratne informacije naših čitalaca su uvek dobrodošle.

Štamparske greške -Iako smo preduzeli sve mere da bismo obezbedili tačnost sadržaja, greške su moguće. Ako pronađete neku grešku u ovoj knjizi, bili bismo zahvalni ako biste nam to javili. Otvorite veb stranicu knjige:

<https://knjige.kombib.rs/mysql-radionica-praktican-vodic-za-rad-sa-podacima-i-bazama-podataka>

i ostavite komentar.

Piraterija - Ako na Internetu pronađete ilegalne kopije naših knjiga, u bilo kojoj formi, molimo vas da nas o tome obavestite i da nam pošaljete adresu lokacije ili naziv veb sajta. Pošaljite nam poruku na adresu copyright@packt.com ili kombib@gmail.com i pošaljite nam link ka sumnjivom materijalu..

Ako ste zainteresovani da postanete autor - Ako postoji tema za koju ste stručni i zainteresovani ste za pisanje ili doprinos knjizi, posetite stranicu authors.packt-pub.com.

Podelite svoje mišljenje

Želimo da saznamo vaše mišljenje o knjizi „MySQL radionica“. Posetite veb stranicu ove knjige na našem sajtu

<https://knjige.kombib.rs/mysql-radionica-praktican-vodic-za-rad-sa-podacima-i-bazama-podataka>

i napišite komentar.

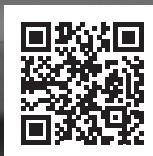
Vaša recenzija je važna i nama i tehničkoj zajednici i pomoći će nam da isporučujemo sadržaje visokog kvaliteta.



Postanite član Kompjuter biblioteke

Kupovinom jedne naše knjige stekli ste pravo da postanete član Kompjuter biblioteke. Kao član možete da kupujete knjige u pretplati sa 40% popustai učestvujete u akcijama kada ostvarujete popuste na sva naša izdanja. Potrebno je samo da se prijavite preko formulara na našem sajtu. Link za prijavu: <http://bit.ly/2TxeK5a>

Skenirajte QR kod
registrujte knjigu
i osvojite nagradu



Deo 1

Kreiranje baze podataka

Ovaj deo knjige obuhvata osnove MySQL-a, relacionih baza podataka i sistema za upravljanje bazama podataka. Biće u njemu reči o načinima na koje možete kreirati baze podataka i umetati, menjati, ispitivati i brisati podatke koji se nalaze u njima.

Ovaj deo se sastoji od sledećih poglavlja:

Poglavlje 1, „Osnovni koncepti“

Poglavlje 2, „Kreiranje baze podataka“

Poglavlje 3, „Korišćenje SQL-a za rad u bazi podataka“

Poglavlje 4, „Izbor, agregacija i primena funkcija“



1

Osnovni koncepti

U ovom poglavlju ćete učiti o osnovnim tipovima baza podataka i načinima na koji ih ljudi obično koriste. Naučićete kako MySQL implementira specifične koncepte, kao što su strukture baze podataka, slojevi i organizacija, i kako izgleda njegova arhitektura. Istražićete šta je **sistem za upravljanje relacionim bazama podataka** MySQL i po čemu se on razlikuje od standardnog **sistema za upravljanje bazama podataka**. Takođe ćete učiti o normalizaciji podataka i modelovanju podataka.

Kada pročitate ovo poglavlje u celosti, imaćete dobar pregled šta je baza podataka i koje su njene različite komponente. Takođe ćete naučiti šta MySQL čini posebnim i kako se uklapa u eko-sistemu baza podataka i njenih komponenata.

Ovo poglavlje obuhvata sledeće teme:

- Uvod u bazu podataka
- Istraživanje MySQL-a
- Vežba 1.01 - Organizovanje podataka u relacionom formatu
- Istraživanje MySQL arhitekture
- Mašine za skladištenje (InnoDB i MyRocks)
- Modelovanje podataka
- Normalizacija
- Aktivnost 1.01 - Kreiranje optimizovane tabele za projekat zaposlenih

Uvod u bazu podataka

Postoji mnoštvo informacija, tj. sve veća gomila malih delova podataka koji pokreću svaki aspekt vašeg života; što je veća ta gomila, postaje vrednija. Na primer, razmislite o situaciji u kojoj morate da pretražujete Internet u potrazi za određenim informacijama – na primer, kako da kreirate MySQL bazu podataka. Da biste pronašli određene informacije, šaljete upit pretraživaču, koji, zatim, analizira velike skupove podataka da bi pronašao relevantne rezultate. Ručno stavljanje svih tih podataka u neki oblik korisnog konteksta, kao što je unošenje u softver za tabelarno izračunavanje, oduzima mnogo vremena.

Korišćenjem baza podataka lakše je automatizovati unos i obradu podataka. Sada možete da skladištite sve te podatke u baze podataka koje se stalno povećavaju i da te baze podataka „gurate“, „izvlačite“, „stiskate“ i „razvlačite“ da biste iz njih, za tren oka, dobili informacije koje nikada ranije niste mogli ni zamisliti. Baza podataka je organizovana kolekcija strukturiranih podataka. Podaci postaju informacije nakon obrade. Na primer, imate bazu podataka za skladištenje servera i njihovih informacija, kao što su broj procesora, memorija, skladište i lokacija. Sami ovi podaci nisu odmah upotrebljivi za poslovne odluke i analize. Međutim, detaljni izveštaji o korišćenju servera na određenim lokacijama sadrže informacije koje možete preuzeti iz baze podataka.

Da bi se obezbedio brz i tačan pristup i da bi se zaštitili svi dragoceni podaci, baza podataka je obično smeštena u spoljnu aplikaciju posebno dizajniranu za efikasno skladištenje i upravljanje velikim količinama podataka. MySQL je jedna takva aplikacija. U skoro svim slučajevima, sistem za upravljanje bazom podataka ili server baze podataka je instaliran na namenskom računaru. Na ovaj način, mnogi korisnici mogu istovremeno da se povežu sa centralizovanim serverom baze podataka. Bez obzira na broj korisnika, i podaci i baza podataka su važni – pošto se u njima čuvaju poverljivi podaci i korisni uvidi – i moraju biti na odgovarajući način zaštićeni i efikasno korišćeni. Na primer, baza podataka se može koristiti za čuvanje informacija iz evidencija ili informacija o prihodima kompanije.

U ovoj knjizi ćete naučiti kako da upravljate bazom podataka. Takođe ćete naučiti kako da primenite bazu podataka, kako da upravljate njom i kako da joj postavljate upite.

U sledećem odeljku su detaljnije opisane baze podataka.

Arhitektura baze podataka

Baza podataka je kolekcija povezanih podataka koji su prikupljeni i uskladišteni, radi lakšeg pristupa i upravljanja. Svaka diskretna stavka podataka u bazi podataka nije, sama po sebi, ni korisna, ni vredna, ali kolekcija podataka u celini (uz lakoću korišćenja i brz pristup) obezbeđuje izuzetno moćnu alatku za poslovnu i ličnu upotrebu. Na primer, ako imate skup podataka koji pokazuju koliko vremena korisnik provede na određenoj stranici, možete pratiti korisničko iskustvo u svojoj aplikaciji. Kako obim podataka raste i njihov istorijski sadržaj se proteže dalje u prošlost, tako oni postaju korisniji u identifikaciji prošlih i predviđanju budućih trendova, a vrednost podataka za njihovog vlasnika se povećava. Baze podataka omogućavaju korisnicima da logički razdvoje podatke i da ih uskladište u dobro strukturiranom formatu koji im omogućava da kreiraju izveštaje i identifikuju trendove.

Da biste razumeli prednosti baza podataka, razmotrite telefonski imenik koji se koristi za čuvanje imena, adresa i telefonskih brojeva ljudi. Telefonski imenik je dobar primer ručnog skladištenja podataka, u kome su podaci organizovani po abecednom redu kako bismo informacije lako pronašli (doduše, ručno). Zahvaljujući telefonskom imeniku, skladištenje velikih skupova podataka kreira glomazan fizički objekat, koji se mora ručno pretraživati da bismo pronašli podatke koje želimo. Proces pretraživanja podataka je dugotrajan, a podatke možemo pretraživati samo po imenima zbog načina na koji su organizovani.

Da biste poboljšali ovaj proces, možete koristiti računarske informacione sisteme za skladištenje podataka u tabelama ili u ravnim datotekama. Ravne datoteke čuvaju podatke u formatu običnog teksta. Datoteke sa ekstenzijama `.csv` ili `.txt` su, obično, ravne datoteke.

```
test,password1
test2,password2
test3,password3
```

Slika 1.1 Primer ravne datoteke

Tabele čuvaju podatke u redovima i kolonama, omogućavajući da logički odvojite podatke i da ih uskladištite.

username	password
test	password1
test2	password2
test3	password3

Slika 1.2 Primer tabele

Koristite baze podataka u skoro svemu što radite u svom životu. Kad god se povežete na veb sajt, izgled ekrana i informacije prikazane na ekranu preuzimaju se iz baze podataka. Mobilni telefon koji koristite u svakodnevnom životu skladišti telefonske brojeve u bazi podataka. Kada gledate emisiju na striming servisu, vaši podaci za prijavljivanje, informacije o emisiji i sama emisija se skladište u bazi podataka.

Postoji mnogo različitih tipova sistema baza podataka. Većina tipova je prilično slična, ali se i prilično razlikuje. Jedni tipovi su usmereni ka određenoj vrsti aktivnosti, a drugi su opšti u svojoj primeni. U narednim odeljcima videćete dva uobičajena sistema za upravljanje bazama podataka DBMS i RDBMS, koje danas koristi većina preduzeća.

Centralizovana baza podataka je baza podataka koja se nalazi, čuva i održava na jednom mestu. Najjednostavniji primer centralizovane baze podataka je MS Access datoteka uskladištena na SharePointu koju koristi više ljudi. Distribuirana baza podataka je složenija, jer se podaci ne skladište na jednom mestu, već na više lokacija. Distribuirana baza podataka pomaže korisnicima da brzo preuzmu informacije, jer se podaci skladište bliže krajnjim korisnicima.

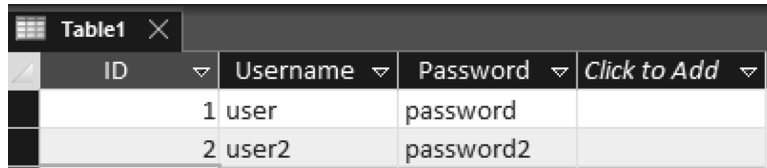
Na primer, ako imate bazu podataka koja se distribuira širom Amerike, Evrope i Azije, američki korisnici će pristupiti bazi podataka uskladištenoj u Americi, evropski korisnici će pristupiti bazi podataka uskladištenoj u Evropi itd. Međutim, to ne znači da Amerikanci ne mogu da pristupe podacima iz Evrope ili Azije. Samo je pristup podacima koji su im bliži brži.

Relacione baze podataka i baze podataka zasnovane na objektima predstavljaju ideje o načinu na koji se podaci čuvaju „iza scene“. Relacione baze podataka uključuju baze podataka, kao što su MySQL i MSSQL, dok objektne baze podataka uključuju baze podataka, kao što je PostgreSQL. Relacione baze podataka koriste koncept koji je objašnjen u ovom poglavlju, dok baze podataka zasnovane na objektima koriste koncept inteligentnih objekata i objektno-orijentisanog programiranja i u njima elementi „znaju“ za šta se koriste.

U sledećem odeljku ćete videti nekoliko primera uobičajenih rešenja za upravljanje bazama podataka koje koriste programeri.

MS Access kao baza podataka

MS Access je Microsoft aplikacija baze podataka. To je jedan od najjednostavnijih primera baze podataka. Omogućava korisnicima da manipulišu podacima pomoću makrofunkcija, upita i izveštaja kako bi mogli da ih dele korišćenjem različitih tehnika vizuelizacije, kao što su grafikoni i Venovi dijagrami. To je alatka za smanjenje brojeva i odlična je za analizu brojeva, predviđanje budućih vrednosti i izvođenje scenarija „šta ako“.



ID	Username	Password	Click to Add
1	user	password	
2	user2	password2	

Slika 1.3 MS Access datoteka

Međutim, MS Access nije najbolja dostupna baza podataka, zbog određenih ograničenja u funkcionalnostima. Na primer, ako su kancelarije vaše kompanije rasute na više lokacija, moguće je deliti Access bazu podataka. Međutim, postoji ograničenje broja korisnika koji se mogu povezati u jednom trenutku. Osim toga, postoje ograničenja u veličini Access datoteka baze podataka, koja omogućavaju skladištenje samo ograničenih skupova podataka. Pristup najbolje funkcioniše u situacijama kada su grupe koje pristupaju bazi podataka male, a i skup podataka mali, u opsegu od 1 milion zapisa ili manje.

Uzmimo kao primer situaciju u kojoj osiguravajuće društvo kreira bazu podataka za korisničku podršku da bi pristupilo korisničkim podacima o polisama osiguranja. Ako je tim na početku mali, sa tri agenta za korisničku podršku i 300 zapisa, MS Access dobro funkcioniše, pošto je opseg upotrebe ograničen. Međutim, dok osiguravajuće društvo raste, može se uključiti više agenata za korisničku podršku i kreirati više zapisa. Kako baza podataka raste, tako MS Access postaje manje praktičan i na kraju Access više neće funkcionisati.

Zbog ovih ograničenja koriste se alternativni sistemi upravljanja bazama podataka.

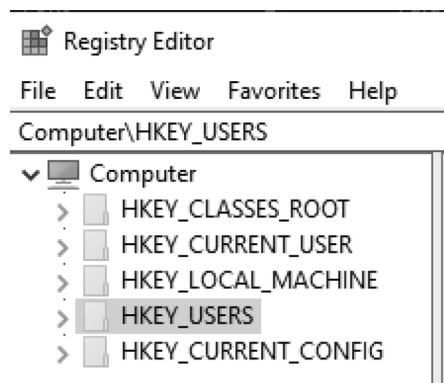
Sistem za upravljanje bazama podataka

Sistemi za upravljanje bazama podataka (DBMS - Database Management Systems) treba da krajnjim korisnicima obezbede fino podešen pristup podacima zasnovanim u kontrolisanom okruženju. Ovi sistemi omogućavaju da definišete podatke i upravljate njima na strukturirani način. Postoji mnogo različitih tipova DBMS-ova koji se koriste u aplikacijama, a svaki od njih ima različite prednosti i mane. Prilikom izbora DBMS-a važno je utvrditi koji je najbolji za određeni problem.

Razmotrimo prethodni primer osiguravajućeg društva koje je kreiralo bazu podataka za agente korisničke podrške. Ako su programeri želeli da ne koriste MS Access, mogli bi da skladište podatke unutar generičkog DBMS-a. Ovi sistemi mogu pomoći da se podaci organizuju slično kao podaci u Access bazi podataka, uz uklanjanje ograničenja veličine i povezivanja koje je kreirao Access. Ovo rešava problem ograničenosti sistema baze podataka. Međutim, i dalje postoje ograničenja u strukturi podataka zasnovanoj na generičkom DBMS rešenju. Neka DBMS rešenja će jednostavno organizovati podatke u tabelarnim formatima, bez ikakvih strukturalnih prednosti. Ove situacije nisu baš idealne za velike skupove podataka.

Ovi problemi se mogu eliminisati pomoću **istema za upravljanje relacionim bazama podataka (Relational Database Management Systems - RDBMS)**.

Primeri DBMS-a uključuju sistem datoteka vašeg računara, FoxPro i Windows Registry.



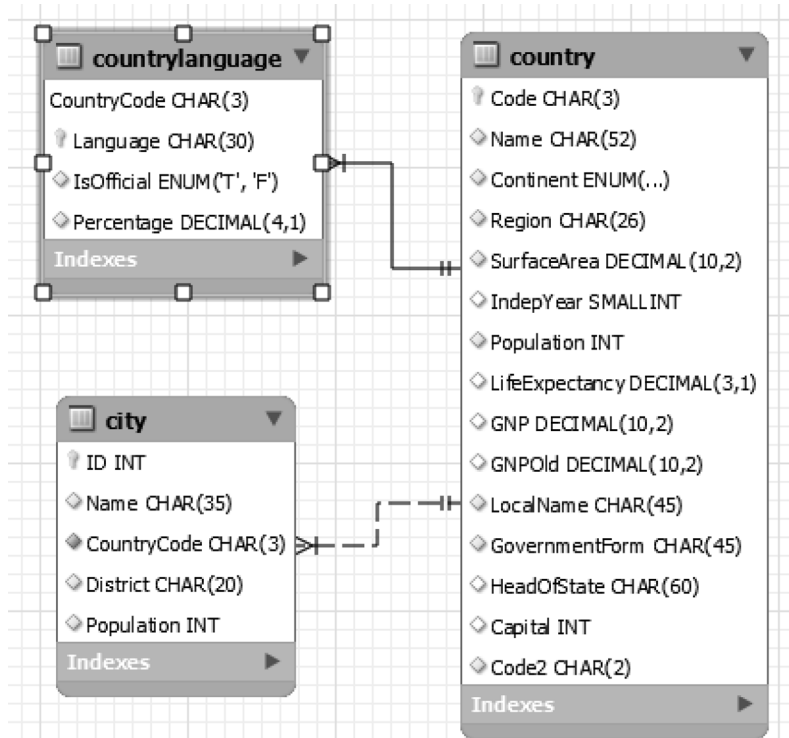
Slika 1.4 Windows Registry je primer osnovnog DBMS-a

RDBMS

Relaciona baza podataka skladišti podatke u dobro strukturiranom formatu redova, kolona i tabela. Red sadrži skup podataka koji se odnosi na jedan entitet. Kolona sadrži podatke o jednom polju ili deskriptoru tačke podataka. Uzmimo kao primer tabelu koja sadrži podatke o korisnicima. Svaki red će sadržati podatke o jednom korisniku. Svaka kolona opisuje korisnike i skladišti tačke podataka, kao što su njihovo korisničko ime, lozinka i slične informacije. Različiti tipovi relacija se mogu definisati između tabela, a određena pravila se primenjuju na kolone. Ovo je poboljšana verzija DBMS koncepta, koja je predstavljena 1970. godine. Dizajnirana je da podrži hijerarhiju klijent-server, istovremeni pristup više korisnika ili aplikacija, bezbednosne funkcije, transakcije i druge objekte koji čine upotrebu podataka iz ovih sistema ne samo bezbednom, već i efikasnom.

RDBMS je robusniji od opštih DBMS ili MS Access baza podataka. Uz primer baze podataka osiguravajućeg društva sada možete da kreirate strukturu oko podataka koji se čuvaju za predstavnike korisničke podrške. Ova struktura predstavlja odnose između različitih skupova podataka, pri čemu se olakšava izvlačenje zaključaka iz povezanih podataka. Osim toga, i dalje dobijate sve prednosti DBMS-a, pa imate najbolji sistem koji odgovara vašim potrebama.

Na sledećoj slici je primer baze podataka u MySQL-u. Kao što vidite, baza podataka ima više tabela (countrylanguage, country i city), koje su međusobno povezane. Naučićete kako da povežete različite tabele u Poglavlju 10, „MS Access, deo 2“.



Slika 1.5 Dijagram RDBMS relacija entiteta

Neki od popularnih RDBMS sistema su MySQL, Microsoft SQL Server i MariaDB. Učitate o MySQL-u u sledećem odeljku.

Istraživanje MySQL-a

MySQL je RDBMS otvorenog koda koji za komunikaciju sa bazom podataka koristi intuitivne ključne reči, kao što su `SELECT`, `INSERT INTO` i `DELETE`. Ove ključne reči se koriste u upitima koji ukazuju serveru kako treba da upravlja podacima, kako da čita i upisuje podatke ili kako da obavlja operacije na objektima baze podataka ili na serveru, kao što su kreiranje ili modifikovanje tabela, uskladištenih procedura, funkcija i prikaza. Objekti baze podataka se definišu i njima se manipuliše korišćenjem SQL komandi, a sva komunikacija i instrukcije koje klijentske aplikacije izdaju bazi podataka izvršavaju se pomoću SQL koda.

MySQL se u velikoj meri primenjuje u poslovanju, uključujući skladištenje podataka, upravljanje inventarom, evidentiranje korisničkih sesija na veb stranicama i skladištenje zapisa o zaposlenima.

MySQL je zasnovan na modelu klijent-server. Taj model omogućava MySQL-u da upravlja istovremenim vezama više korisnika i da hostuje veliki broj baza podataka sa tabelama i fino podešenim bezbednosnim dozvolama kako bi se osiguralo da podacima pristupaju samo odgovarajući korisnici.

U sledećem odeljku ćete upoznati neke od tipova podataka koji se koriste u MySQL-u za skladištenje podataka.

Tipovi podataka

Svaka kolona u tabeli baze podataka zahteva tip podataka za identifikaciju tipa podataka koji će biti uskladišteni u njoj. MySQL koristi dodeljeni tip podataka da utvrdi kako će koristiti podatke.

U MySQL verziji 8.0 postoje tri osnovna tipa podataka. Oni su poznati kao string, numeric i date i time. Detaljnije su opisani u sledećoj tabeli.

- `string` - Znakovni nizovi (strings) su tekstualni prikazi podataka. Postoje različiti tipovi znakovnih nizova podataka, uključujući `CHAR`, `VARCHAR`, `BINARY`, `VARBINARY`, `BLOB`, `TEXT`, `ENUM` i `SET`. Oni mogu predstavljati podatke od pojedinačnih tekstualnih znakova u tipovima `CHAR`, do punih znakovnih nizova teksta u tipovima `VARCHAR`. Veličina znakovnog niza promenljivih može da varira od 1 bajta do 4 GB, u zavisnosti od tipa i veličine podataka koji se skladište. Da biste saznali više o ovim tipovima podataka, posetite stranicu <https://dev.mysql.com/doc/refman/8.0/en/string-types.html>.
- `numeric` - Numerički tipovi podataka skladište samo numeričke vrednosti. Postoje različite tipovi numeričkih podataka, uključujući `INTEGER`, `INT`, `SMALLINT`, `TINYINT`, `MEDIUMINT`, `BIGINT`, `DECIMAL`, `NUMERIC`, `FLOAT`, `DOUBLE` i `BIT`. Ovi tipovi podataka mogu predstavljati brojeve različitih formata. Tipovi `DECIMAL` i `FLOAT` predstavljaju decimalne vrednosti, dok tipovi `INTEGER` mogu predstavljati samo celobrojne vrednosti. Opseg uskladištene veličine zavisi od numeričkog tipa podataka koji je dodeljen polju i može da ima od jedan do osam bajtova, u zavisnosti od toga da li su podaci potpisani i da li tip podržava decimalne vrednosti. Da biste saznali više o ovim tipovima podataka, posetite stranicu <https://dev.mysql.com/doc/refman/8.0/en/numeric-types.html>.
- `date` i `time` - Postoji pet tipova podataka datuma i vremena: `Date`, `Time`, `Year`, `DateTime` i `TimeStamp`. `Date`, `Time` i `Year` skladište različite komponente datuma u odvojenim kolonama, `DateTime` beleži i datum i vreme, a `TimeStamp` pokazuje koliko sekundi je prošlo od fiksne tačke u vremenu. Tipovi podataka zasnovani na datumu obično imaju

osam bajtova, u zavisnosti od toga da li skladište i vreme i datum. Da biste saznali više detalja, posetite link <https://dev.mysql.com/doc/refman/8.0/en/date-and-time-types.html>.

Kao programer, vi treba da izaberete odgovarajući tip i veličinu podataka za informacije koje ćete sačuvati u koloni. Ako znate da će polje koristiti samo pet znakova, definišite njegovu veličinu kao 5.

U sledećoj vežbi naučićete kako da organizujete skup podataka u relacionom formatu pomoću odgovarajućih tipova podataka za svako polje.

Vežba 1.01 – Organizovanje podataka u relacionom formatu

Pretpostavimo da radite za kompaniju „ABC Corp“. Vaš menadžer bi želeo da razvije bazu podataka u kojoj se skladište informacije o klijentima, ali i porudžbine koje je klijent kreirao. Od vas je zatraženo da odredite kako ćete organizovati podatke u relacionom formatu. Osim toga, kompanija zahteva da definišete tipove podataka koji su odgovarajući za svako polje. Ovo je lista svojstava koja treba da se uskladište u relacionom modelu:

Podaci o klijentu:

- ID klijenta
 - ime klijenta
 - adresa klijenta
 - broj telefona klijenta
- Podaci o porudžbini:
 - ID klijenta
 - ID porudžbine
 - cena porudžbine

Da biste kreirali relacionu strukturu baze podataka, izvršite sledeće korake:

1. Prvo utvrdite tipove podataka koji su odgovarajući za podatke. ID polja treba da budu tip podataka `int`, pošto su ID-ovi obično numerički. Za polja koja sadrže imena, adrese i telefonske brojeve tip podataka `varchar` je odgovarajući, jer može da uskladišti opšti tekst. Na kraju, cena se može definisati kao tip podataka `double`, pošto treba da uskladišti decimalne vrednosti.
2. Utvrdite koliko tabela treba da imate. U ovom slučaju imate dva skupa podataka, što znači da bi trebalo da imate dve tabelle – `CustomerData` i `OrderData`.

3. Razmotrite kako su tabele međusobno povezane. Pošto klijent može imati porudžbinu u podacima o porudžbini, možete zaključiti da su kupci i porudžbine međusobno povezani.
4. Zatim, pogledajte koje su kolone iste u dva skupa podataka. U ovom primeru obe tabele sadrže kolonu `CustomerID`.

Na kraju, kombinujte sve informacije. Imate dve tabele `CustomerData` i `OrderData`. Možete ih povezati pomoću iste kolone koju imaju, a to je kolona `CustomerID`. Relacioni model bi izgledao kao na sledećoj slici.

CustomerData		OrderData	
CustomerID	int	CustomerID	int
CustomerName	varchar	OrderID	int
CustomerAddress	varchar	OrderPrice	double
CustomerPhoneNumber	varchar		

Slika 1.6 Podaci o klijentima i porudžbinama organizovani u relacionom formatu

Sada imate potpuno definisanu relacionu strukturu za vaše podatke. Ova struktura zajedno sa tipovima podataka može se koristiti za konstruisanje odgovarajuće relacione baze podataka.

„Zaronićete“ u arhitekturu MySQL-a u sledećem odeljku.

Istraživanje MySQL arhitekture

Kada malo bolje pogledate, videćete da se svi računarski sistemi sastoje od nekoliko slojeva. Svaki sloj ima specifičnu ulogu u celokupnom dizajnu sistema. Sloj je odgovoran za jedan ili više zadataka. Zadaci su podeljeni na manje module namenjene jednom aspektu uloge sloja. Svaka operacija treba da prođe kroz sve slojeve da bi bila uspešna. Ako nije uspešna u jednom sloju, ne može da pređe na sledeći sloj i pojavljuje se greška.

MySQL server takođe ima nekoliko slojeva. Fizički sloj je odgovoran za skladištenje stvarnih podataka u optimizovanom formatu. Njemu se onda pristupa pomoću logičkog sloja. Logički sloj je odgovoran za strukturiranje podataka u odgovarajućem formatu, uz primenu svih potrebnih dozvola i struktura. Najviši je aplikacioni sloj, koji obezbeđuje interfejs za veb aplikacije, skriptove ili bilo koju vrstu aplikacija koje imaju API za interakciju sa bazom podataka.

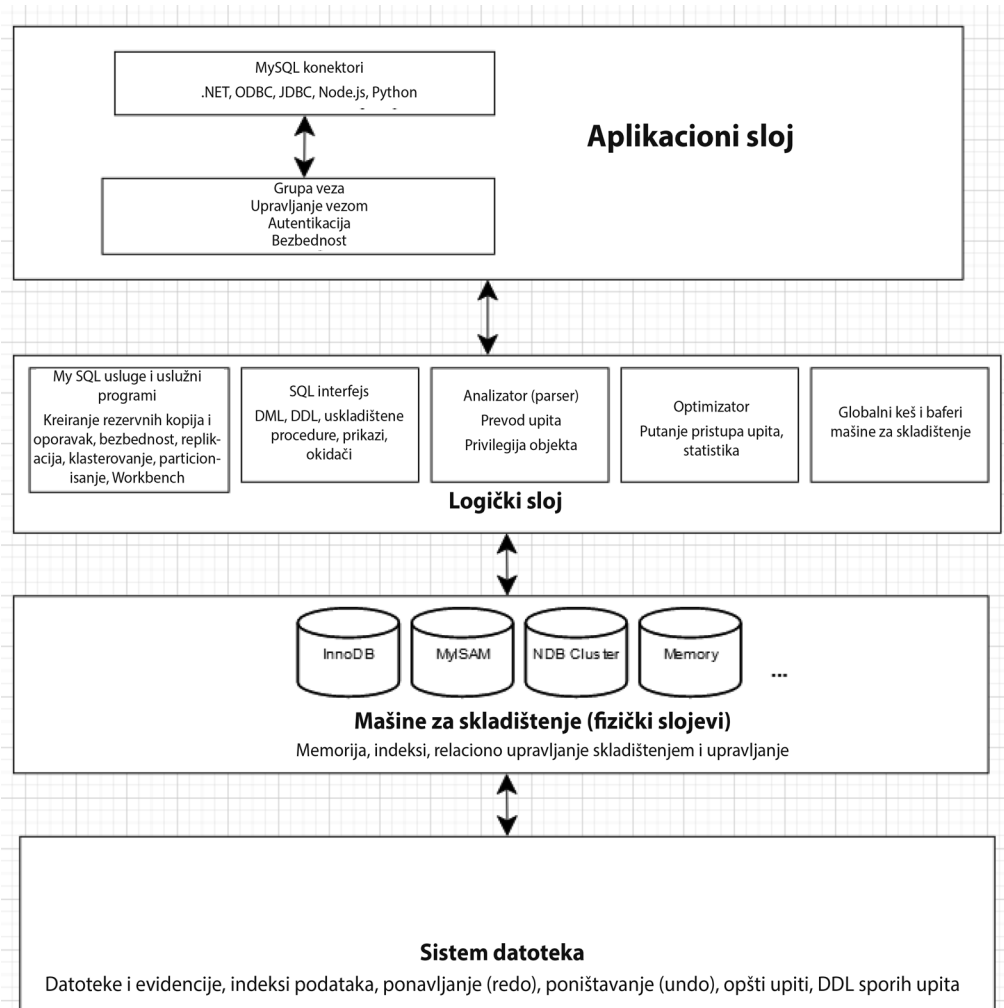
Kao što je ranije rečeno, RDBMS sistem obično ima arhitekturu klijent-server. Vi i vaša aplikacija ste klijent, a MySQL je server.

MySQL slojevi

Postoje tri sloja na MySQL serveru:

- aplikacioni sloj
- sloj za skladištenje
- fizički sloj

Ovi slojevi su od suštinskog značaja za razumevanje koji deo je odgovoran za način na koji se vaši podaci tretiraju. Sledi grafički prikaz osnovne arhitekture MySQL servera. Prikazano je kako su različite komponente unutar MySQL sistema međusobno povezane.



Slika 1.7 MySQL arhitektura

Aplikacioni sloj – veza sa klijentom

Aplikacioni sloj prihvata vezu, koristeći bilo koju od klijentskih tehnologija (JDBC, ODBC, .NET, PHP). Ima grupu veza koje predstavljaju API za aplikacioni sloj koji upravlja komunikacijom sa različitim „potrošačima“ podataka, uključujući aplikacije i veb servere. Aplikacioni sloj obavlja sledeće zadatke:

- **Upravljanje vezom** – Klijentu se dodeljuje programska nit tokom kreiranja veze; zamislite tu nit kao „cevovod“ u serveru. Klijent svaku radnju izvršava pomoću nje. Nit se kešira, tako da klijent ne mora da se prijavljuje uvek kada pošalje zahtev. Ona se uništava kada klijent prekine vezu. Svi klijenti imaju svoje niti. Kada klijent želi da se poveže sa bazom podataka, on će prvo poslati zahtev serveru baze podataka, koristeći svoje akreditivne. Obično će zahtevi uključivati i detalje o bazi podataka sa kojom klijent želi da se poveže na serveru. Server će, zatim, potvrditi njegov zahtev, uspostaviti sesiju sa serverom i vratiti vezu korisniku.
- **Autentikacija** – Kada se veza uspostavi, server će izvršiti autentikaciju klijenta, koristeći podatke o korisničkom imenu i lozinci koji su poslani uz zahtev. Ako su podaci za prijavljivanje netačni, klijentu neće biti dozvoljeno da nastavi dalje. Ako su podaci za prijavljivanje tačni, klijent će preći na bezbednosne provere.
- **Provera bezbednosti** – Kada se klijent uspešno poveže sa serverom, MySQL će proveriti šta je korisničkom nalogu dozvoljeno da radi u njemu. Proveriće klijentov status čitanja/pisanja/ažuriranja/brisanja, a nivo bezbednosti za nit će biti postavljen za sve zahteve izvršene na ovoj vezi i niti.

Kada se klijent poveže sa serverom, nekoliko usluga se aktivira u grupi veza serverskog sloja.

MySQL serverski sloj (logički sloj)

Ovaj sloj ima svu logiku i funkcionalnost MySQL RDBMS-a. Njegov prvi sloj je grupa veza, koja prihvata klijentske veze i izvršava autentikaciju klijentskih veza. Ako se klijent uspešno poveže, ostali slojevi MySQL servera će mu biti dostupni u okviru ograničenja. Serverski sloj ima sledeće komponente:

- **MySQL usluge i uslužni programi** – Ovaj sloj obezbeđuje usluge i uslužne programe za administriranje i održavanje MySQL sistema. Još nekoliko usluga i uslužnih programa se može dodati po potrebi. To je jedan od glavnih razloga zašto je MySQL veoma popularan. Neke usluge i neki uslužni programi uključuju rezervne kopije i oporavak, bezbednost, replikaciju, klasterovanje, partitionisanje i MySQL Workbench.

- **SQL interfejs** – SQL je alatka za obezbeđivanje interakcije između MySQL klijenta i MySQL servera. SQL alatke koje obezbeđuje sloj SQL interfejsa između ostalih uključuju **jezik za manipulaciju podacima (Data Manipulation Language - DML)**, **jezik za definiciju podataka (Data Definition Language - DDL)**, uskladištene procedure, prikaze i okidače. Ti koncepti će biti detaljno proučavani u ovoj knjizi.
- **Analizator (parser)** – MySQL ima svoj interni jezik za obradu zahteva podataka. Kada se SQL iskaz prosledi MySQL serveru, on će prvo proveriti keš memoriju. Ako MySQL otkrije da je identičan iskaz prethodno pokrenuo bilo koji klijent, jednostavno će vratiti keširane rezultate. Ako ne pronađe upit koji je prethodno pokrenut, MySQL analizira iskaz i kompajlira ga u MySQL interni jezik.

Analizator ima tri osnovne operacije koje će primeniti na SQL iskaz:

- leksičku analizu koja prihvata tok znakova (SQL iskaz) i kreira listu reči koja čini iskaz
- sintaksičku analizu koja prihvata reči i kreira strukturiranu reprezentaciju sintakse, proveravajući da li je sintaksa ispravno definisana
- generisanje koda koje konvertuje sintaksu generisanu u *koraku 2* u interni jezik MySQL-a, tj. prevodi sintaksički ispravne upite na interni jezik MySQL-a
- **Optimizer** – Interni kod iz analizatora se, zatim, prosleđuje u sloj optimizatora za koji će se ispostaviti da je najbolji i najefikasniji za izvršavanje koda. Optimizator može da prepíše upit, da utvrdi redosled skeniranja tabela i da izabere ispravne indekse koje treba da koristite.
- **Keš** – MySQL će keširati kompletan skup rezultata za `SELECT` iskaze. Keširani rezultati se skladište u slučaju da bilo koji klijent, uključujući i vas, pokrene isti upit. Ako je pokrenut isti upit, analiziranje se preskače, a keširani rezultati se vraćaju. Primetićete ovo na delu ako dva puta pokrenete upit. Prvog puta će proći malo više vremena dok se rezultati ne vrate, a naredna pokretanja upita će biti brža.

Sloj mašine za skladištenje (fizički sloj)

Sloj mašine za skladištenje upravlja svim operacijama umetanja, čitanja i ažuriranja podataka. MySQL koristi tehnologiju priključnih mašina za skladištenje. To znači da možete dodati mašine za skladištenje koje su prikladne za vaše potrebe. One su često optimizovane za određene zadatke ili tipove skladištenja i funkcionisaće bolje od drugih u svojoj „specijalnosti“.

Videćete različite tipove mašina za skladištenje podataka u sledećem odeljku.

Mašine za skladištenje (InnoDB i MyRocks)

MySQL mašine za skladištenje su softverski moduli koje MySQL server koristi za pisanje, čitanje i ažuriranje podataka u bazi podataka. Postoje dva tipa mašina za skladištenje – **sa transakcijom** i **bez transakcije**:

- **Mašine za skladištenje sa transakcijom** dozvoljavaju da se operacije pisanja ponište ako ne budu uspešne, tako da izvorni podaci ostaju nepromenjeni. Transakcija može da obuhvata nekoliko operacija pisanja. Zamislite prenos sredstava sa jednog računa na drugi u računovodstvenom sistemu preduzeća; zaduživanje sredstava sa jednog računa i njihovo odobrenje na drugom je jedna transakcija. Ako se neuspeh dogodi pri kraju transakcije, sve prethodne operacije će biti poništene i ništa u transakciji neće biti izvršeno. Ako su svi zadaci pisanja bili uspešni, transakcija će biti izvršena, a sve promene će biti trajne. Većina mašina za skladištenje je transakciona, kao što je InnoDB.
- **Mašine za skladištenje podataka bez transakcije** skladište podatke odmah nakon izvršenja. Ako operacija pisanja ne bude uspešna pred kraj niza operacija pisanja, prethodne operacije moraju da budu poništene ručno pomoću koda. Zato će korisnik verovatno morati da zabeleži stare vrednosti negde drugde da bi ih uvek imao pri ruci. U primeru računovodstva zamislite da su sredstva zadužena sa prvog računa, ali da nisu bila odobrena na drugom, a početno zaduženje nije stornirano. U ovom slučaju sredstva bi jednostavno nestala. Primer ovog tipa mašine je MyISAM.

Pri izboru mašine za skladištenje treba uzeti u obzir da li je usklađena sa ACID-om.

ACID usklađenost

ACID usklađenost osigurava integritet podataka u slučaju povremenih „kvarova“ na različitim slojevima, kao što su prekinuta veza, greška prilikom skladištenja i otkazivanje procesa servera:

- **Atomičnost** osigurava da se sve različite operacije unutar transakcije tretiraju kao jedna jedinica, što znači da, ako jedna ne bude uspešna, sve će biti neuspešne. Ovo osigurava da ni jedna transakcija ne ostane delimično izvršena. Ako je transakcija uspešna, izmene se prenose na sloj za skladištenje i zagarantovana je tačnost podataka.

- **Konzistentnost** osigurava da transakcija ne može dovesti bazu podataka u nevažeće stanje. Svi napisani podaci moraju biti u skladu sa svim definisanim pravilima u bazi podataka, uključujući ograničenja, kaskade, okidače i referentni integritet primarnog i spoljnog ključa. Ovo će sprečiti oštećenje podataka izazvano nelegalnom transakcijom.
- **Izolacija** osigurava da ni jedan deo transakcije nije vidljiv drugim korisnicima ili procesima dok se cela transakcija ne završi.
- **Trajnost** osigurava da transakcija ostane izvršena, čak i u slučaju kvara sistema ili nestanka struje. Transakcija se snima u memoriji prijavljivanja (logon store), koja je trajna.

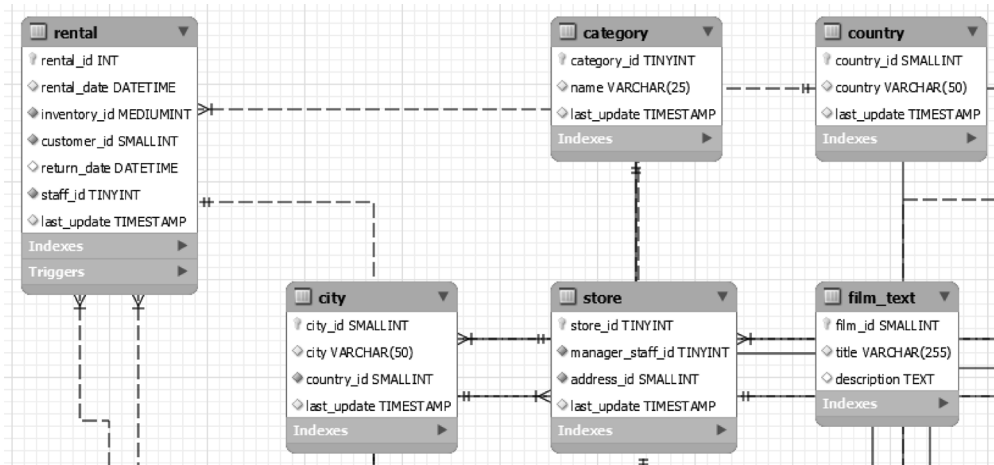
Podrazumevana mašina za skladištenje MySQL-a je InnoDB, koja je usklađena sa ACID-om. Postoje i drugi tipovi mašina za skladištenje koje drugačije skladište podatke i manipulišu njima. Ako ste zainteresovani da saznate više detalja o tipovima mašina za skladištenje koji su dostupni za MySQL, pogledajte link <https://dev.mysql.com/doc/refman/8.0/en/storage-engines.html>.

U sledećem odeljku ćete videti kako različite aplikacije mogu da se povežu sa vašom bazom podataka pomoću aplikacionog sloja.

Modelovanje podataka

Modelovanje podataka je konceptualni i logički prikaz predložene fizičke baze podataka u vizuelnom formatu koji koristi dijagrame **entiteta i relacija (Entity Relationship - ER)**. ER dijagram predstavlja sve entitete baze podataka, tako da definiše njihove relacije i svojstva. Cilj za ER dijagram je da postavi strukturu entiteta, tako da budu laki za razumevanje i da se kasnije implementiraju u sistemu baze podataka.

Da biste razumeli modelovanje podataka, morate da poznajete dva ključna koncepta. Prvi je primarni ključ. On se koristi za jedinstvenu identifikaciju određenog zapisa ili reda u bazi podataka. Za sada, trebalo bi da znate da primarni ključevi „nameću“ tabeli da ne bude duplih redova sa istim ključem. Drugi koncept je spoljni ključ. On omogućava da povežete tabele sa jednim poljem ili kolekcijom polja koja se odnose na primarni ključ druge tabele.



Slika 1.8 Model podataka baze podataka sakila

Na prethodnom snimku ekrana prikazani su delovi modela podataka za bazu podataka sakila. Prikazano je kako su različite tabele povezane i kakve su njihove relacije. Možete čitati relacije u poljima koja se dele između povezanih tabela. Na primer, tabela `rental` i tabela `category` su povezane poljem `last_update`. Tabela `category` je zatim povezana sa tabelom `country` takođe pomoću polja `last_update`. Ovo pokazuje opštu strukturu relacija tabele.

Model podataka osigurava da su svi potrebni objekti podataka (uključujući tabele, primarne ključeve, spoljne ključeve i uskladištene procedure) predstavljeni i da su relacije između njih pravilno definisane. Model podataka takođe pomaže da se identifikuju podaci koji nedostaju ili koji su suvišni.

MySQL obezbeđuje **poboljšani dijagram entiteta i relacija** za modelovanje podataka sa kojim možete direktno da komunicirate da biste dodali, modifikovali i uklonili objekte baze podataka i postavili relacije i indekse. Ovome se može pristupiti pomoću Workbench-a (to je detaljno objašnjeno u sledećem poglavlju). Kada je model završen, može se koristiti za kreiranje fizičke baze podataka ako ona ne postoji ili za ažuriranje postojeće fizičke baze podataka.

U sledećim koracima predstavljen je proces pomoću kojeg nastaje baza podataka:

1. Neko dobije ideju da kreira bazu podataka i aplikaciju.
2. Analitičar baze podataka ili programer su angažovani da kreiraju bazu podataka.
3. Izvodi se analiza kako bi se utvrdilo koji podaci moraju da budu sačuvani. Ove izvorne informacije mogu da pristignu iz drugog sistema, dokumenata ili verbalnih zahteva.

4. Analitičar normalizuje podatke da bi definisao tabele.
5. Baza podataka je modelovana korišćenjem normalizovanih tabela.
6. Baza podataka je kreirana.
7. Razvijaju se aplikacije koje koriste bazu podataka za izveštavanje, obradu i izračunavanje.
8. Baza podataka je aktivirana.

Na primer, pretpostavimo da izrađujete sistem koji skladišti video-snimke za korisnike. Prvo, morate da utvrdite kako će baza podataka biti strukturirana, tj. da utvrdite koje podatke treba uskladištiti, koja polja su relevantna i koje tipove podataka polja treba da imaju i relacije između podataka. Za primer vaše video-baze podataka možda ćete želeći da memorišete lokaciju videa na serveru, naziv videa i opis videa koje možete da povežete sa tabelom baze podataka koja sadrži ocene i komentare za video. Nakon toga, možete da kreirate bazu podataka koja odgovara predloženoj strukturi. Na kraju, možete postaviti bazu podataka na server, tako da bude aktivirana i dostupna korisnicima.

U sledećem odeljku ćete učiti o normalizaciji baze podataka, tj. o činu kreiranja optimizovane šeme baze podataka sa što manje redundancije pomoću ograničenja i uklanjanja funkcionalne zavisnosti podelom baze podataka na manje tabele.

Normalizacija

Normalizacija je jedna od najvažnijih veština za svakoga ko planira da dizajnira i održava baze podataka. To je tehnika dizajna koja pomaže u eliminisanju nepoželjnih karakteristika, kao što su anomalije pri umetanju, ažuriranju i brisanju podataka, i smanjuje redundantnost podataka. Anomalije umetanja mogu biti posledica nedostatka primarnih ključeva ili prisustva funkcionalne zavisnosti. Jednostavno rečeno, pojaviće se dupli zapisi kada ne bi trebalo da ih bude.

Ako imate veliku tabelu sa milionima zapisa, operacije traženja, ažuriranja i brisanja oduzimaju vam mnogo vremena. Prvo možete da dodelite više resursa serveru, ali to nije dobro. Sledeće što treba da uradite je normalizacija tabele, tj. treba da podelite veliku tabelu koju imate na manje tabele i da povežete te manje tabele pomoću relacija korišćenjem primarnog i spoljnog ključa.

Ovu tehniku je izmislio Edgar Codd. Ona ima sedam različitih formi koje se nazivaju **normalne forme**. Lista sadrži **prvu normalnu formu (First Normal Form - 1NF)**, sve do **šeste normalne forme (Sixth Normal Form - 6NF)** i još jednu dodatnu formu, tj. **Bojs Kodovu normalnu formu (Boyce-Codd Normal Form - BCNF)**.

Na osnovu normalne forme, svaka ćelija treba da sadrži jednu vrednost i svaki zapis treba da bude jedinstven. Na primer, pretpostavimo da imate bazu podataka koja skladišti informacije o zaposlenima. Prva normalna forma podrazumeva da svaka kolona u vašoj tabeli sadrži jednu informaciju.

EmployeeName	Designation	EmployeeLocation
Jeff	Database Administrator	Canada
Sarah	Programmer	United States
Bob	Accounting	Europe
Jane	Operations	Canada

Slika 1.9 Primer tabele za 1NF

Druga normalna forma označava da je baza podataka u prvoj normalnoj formi i da mora imati primarni ključ u jednoj koloni. U prethodnom primeru trenutno nemate jednu jedinstvenu kolonu, pošto bi ime zaposlenog, naslov i lokacija mogli da se dupliraju. Da biste pretvorili bazu podataka u drugu normalnu formu, možete dodati ID kao jedinstveni identifikator.

EmployeeID	EmployeeName	Designation	EmployeeLocation
1	Jeff	Database Administrator	Canada
2	Sarah	Programmer	United States
3	Bob	Accounting	Europe
4	Jane	Operations	Canada

Slika 1.10 Primer tabele za 2NF

Treća normalna forma zahteva da baza podataka bude u drugoj normalnoj formi i zabranjeno je imati tranzitivne funkcionalne zavisnosti. Tranzitivna funkcionalna zavisnost postoji kada kolona u jednoj tabeli zavisi od druge kolone koja nije primarni ključ. To znači da je svaki odnos u bazi podataka samo između primarnih ključeva. Baza podataka se smatra normalizovanom ako dostigne treću normalnu formu. Tabela je ovde u trećoj normalnoj formi, pošto ima primarni ključ koji se može koristiti za povezivanje sa bilo kojom drugom tabelom, bez upotrebe polja bez ključa.

EmployeeID	EmployeeName	Designation	EmployeeLocation
1	Jeff	Database Administrator	Canada
2	Sarah	Programmer	United States
3	Bob	Accounting	Europe
4	Jane	Operations	Canada

Slika 1.11 Primer tabele za 3NF

Da biste saznali više detalja, posetite sajt <https://docs.microsoft.com/en-us/office/troubleshoot/access/database-normalization-description>.

Nakon što ste naučili sve o upotrebi skupova podataka, izvršićete jednu aktivnost da biste rezimirali sve što ste do sada naučili u ovom poglavlju.

Aktivnost 1.01 - Kreiranje optimizovane tabele za projekat zaposlenih

Vaš menadžer vas je zamolio da kreirate bazu podataka koja sadrži informacije o mrežnim uređajima na lokaciji vaše korporativne mreže. Možda imate više uređaja sa istim nazivom na istoj lokaciji. Od vas se zahteva da uskladite tabele sa 3NF formom kako bi bile što efikasnije. Osim toga, potrebno je da utvrdite odgovarajuće tipove podataka za svaku kolonu u tabeli. Na kraju, od vas se traži da utvrdite koje kolone treba da budu primarni ključevi, tako da 3NF bude zadovoljen. Odlučili ste da izvršite sledeće korake.

1. Analizirajte sledeću tabelu.

Hostname	Location	OperatingSystem	Layerlevel
PINKY	Ground Floor A	IOS	L2
PINKY	Ground Floor A	NXOS	L2
HERETIC	First Floor A	JUNOS	L3
HERETIC	First Floor A	NXOS	L3
HERETIC	Ground Floor B	IOS	L3

Slika 1.12 Tabela uređaja na mreži

2. Identifikujte obrasce da biste odredili tipove podataka i moguće primarne ključeve. Možda ćete morati da dodate kolonu u tabelu ako odgovarajući primarni ključ još uvek ne postoji. Zatim, dovedite tabelu u 1NF formu.

3. Dovedite tabelu u 2NF formu, podelite tabelu i pokušajte da je dovedete u 2NF formu u skladu sa pravilom.
4. Dovedite tabelu u 3NF formu, još više je podelite i dovedite je u 3NF formu, tako da bude u 2NF formi sa odgovarajućim ograničenjima.



Rešenje za ovu aktivnost nalazi se u „Dodatku“ ove knjige.

Sada ste podesili optimizovanu tabelu i moći ćete da koristite ovu tehniku da biste efikasno optimizovali svoju bazu podataka pre nego što počnete da je popunjavate podacima i primenjujete u proizvodnom okruženju.

Rezime

U ovom poglavlju naučili ste šta je relaciona baza podataka i koje su razlike između DBMS baze podataka i RDBMS baze podataka. Učili ste o modelu klijent-server koji koristi MySQL i pročitali ste kratak uvod u MySQL arhitekturu da biste videli kako MySQL funkcioniše.

Zatim ste istražili koji slojevi čine MySQL i kako možete da definišete različite modele podataka i dodali ste tabele tim modelima podataka. Takođe ste naučili osnovne koncepte ACID-a i kako se inicijalizuje baza podataka.

U sledećem poglavlju ćete dopuniti svoje znanje o modelovanju podataka, o relacijama entiteta i o načinu upotrebe MySQL Workbench za podešavanje/konfigurisanje baza podataka.