

LYNDA - MYSQL ESSENTIAL

①

- MySQL je database management system
- MySQL se sastoji od:
 - database servera
 - client librarija - interfejsi za programerske jezike
 - command line client - na ovom se mogu raditi ne stvari

DBMS - Database Management System

Storage engine - dio DBMS-a koji sprema i dohvaća podatke

SQL - structured query language, jezik koji komunicira sa DBMS-om

Database - skladište podataka. Obično na disku, može biti u memoriji ili negdje drugo. Obično se sastoji od sistema fajlova, nekad je i samo 1 fajl.

Table - jedna klasa podataka

② XAMPP skripte

/Applications/XAMPP/xamppfiles/bin

/mysql

→ mysql komanda

(alias xmysql)

- ③ show databases; - daje prikaz svih baza
exit - izlazi iz mysql konzole
→ sve komande završavaju sa ;

PROMJENA ROOT PASSWORDA ZA MYSQL

- MYSQL ima 2 root usera

\$ mysql -u root - polazice shell kao root

> UPDATE mysql.user SET Password = PASSWORD('5894944')
updatea tablicu user dupac password
↓
FJA ENKRIPTIRAN
PASSWORD
— promjena passworda za root here (njih 2 ili 3)
PASSWORD ODABRANI

↳ po defaultu je mysql root password blank

WHERE User = 'root';
ovo se u 1. liniji

> FLUSH PRIVILEGES; - ponovno učitava tablice sa passwordima i tako ih applicira
> exit

\$ mysql -u root -p → treba staviti -p kad root ima setene passworde

MYSQL USERI

User creation

→ mysql ima dobro razrađena pravila/procedure i mogućnosti za user

• STVARANJE NOVOG USERA (loginan kao root)

KODICI
DODAVANJE ZA
USERA

1. > CREATE USER admin@localhost;

- stvara usera admin

→ znači da se user može loginati na ovom stroju. Ne može SSH-ati na komp i onda se ulogirati

2. > GRANT ALL ON *.* TO admin@localhost WITH GRANT OPTION;

SVA PRAVA

NA SVIM BAZAMA
i NA SVIM
TABLICAMA

MOGUĆNOST DA
DRUGIM USERIMA DAJE
PRAVA

> CREATE USER web@localhost;

COMA SPACE RAZMACI

> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, FILE, INDEX, ALTER, CREATE TEMPORARY TABLES, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, EXECUTE ON *.* TO web@localhost;

DATA SVA PRAVA NA
TABLICAMA i BAZAMA

NA SVIM BAZAMA
i NA SVIM TABLICAMA

3. > FLUSH PRIVILEGES;

- ponovo pročitaj privilegije

• DODAVANJE PASSWORDA NOVIM USERIMA

(isto kao za root)

1. > UPDATE mysql.user SET Password = PASSWORD('SP44944') WHERE User = 'web';
→ 'web';
→ 'admin';

2. > FLUSH PRIVILEGES;

↳ pokretanje skripte koje odradi komande u MYSQL shellu

```
$ mysql -u admin -p < test-mysql.sql
```

→ odradi me komande iz fila kao da su napisane u mysql shellu

> use test; - za komande naredje upotrebljavam test bazu

> show tables; - pokazuje tablice od baze definirane u koraku prije

↓
customer
item
sale

> select * from item; → pokazuje me stupce i retke u tablici item unutar odabrane baze

> select count(*) from item; → vraća broj redaka iz tablice

MYSQL EXPORT

NETA RAZPRA

```
$ mysqldump -u username -ppassword database-name > dump.sql
```

OPCIJE

- ↳ --no-data - backup samo strukturu bez podataka
- ↳ --no-create-info - samo podaci, bez info o kreiranju baze, tablice...
- ↳ --databases ime1 ime2 - dumpanje više baza
- ↳ --all-databases - dumpa me baze

MYSQL RESTORE & IMPORT

```
$ mysql -u username -ppassword database-name < dump.sql
```

ALTERNATIVNI EXPORT (ako ovaj prvi ne radi)

```
$ mysql dump --user=root --password=pas --opt baza > db-dump.sql
```

5

Import / export baze

Kreiranje i dropanje baza i tablica

KREIRANJE BAZE

- > CREATE DATABASE sales; - kreira bazu sales
- > SHOW DATABASES; - pokazuje dostupne baze
- > DROP DATABASE sales; - BRISJE BAZU (i ne može više restorovati!)
- ne pita confirmation

KREIRANJE TABLICE

> use test; → (odabrat neku bazu)

```
> CREATE TABLE test_customer (  
    id INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255),  
    address VARCHAR(255),  
    city VARCHAR(255),  
    state CHAR(2),  
    zip CHAR(10)  
);
```

VARIJABILNA DULJINA
ZNAKOVA

FIXNA DULJINA
ZNAKOVA

OVO SU
STUPCI U
BAZI

- > DESCRIBE test_customer; - pokaže shemu tablice
- > DROP TABLE test_customer; - briše tablicu

⑥

MYSQL

DATA

TYPES

Tipovi podataka

• date tipovi :

- numerički - za brojeve
- string - za text i riječi
- large storage - za dokumente i datoteke
- date & time
- bit values
- enumeration - za mnemonic values

• NUMERIC TYPES

- INT - integers - cijeli brojevi
- FLOAT - floating point - nije fiksni broj znamenki, NIJE 100% PRECIZNO
- DECIMAL - fixed point - decimalni brojevi sa fiksnom preciznošću
- može TOČAN rezultat

• STRING TYPES

- fixed - uvijek iste dužine
- variable - mogu biti varijabilne dužine
- CHAR - za tekstualne podatke
 - ↳ CHAR - fiksna dužina
 - ↳ VARCHAR - varijabilna dužina
- BINARY - za ne-tekstualne podatke
 - ↳ BINARY - fiksna dužina
 - ↳ VARBINARY - varijabilna dužina

• LARGE STORAGE

- **BLOB** - binary large object, za ne-textualne podatke, slike, ...
 - ↳ **TINYBLOB** - do 256 b
 - ↳ **BLOB** - do 64 kb
 - ↳ **MEDIUMBLOB** - do 16 Mb
 - ↳ **LONGBLOB** - do 4Gb

- **TEXT** - za textualne podatke. Mogu se pretvoriti u tekstualne u MySQL-u
 - ↳ **TINYTEXT** - do 256 b
 - ↳ **TEXT** - do 64 kb
 - ↳ **MEDIUMTEXT** - do 16 Mb
 - ↳ **LONGTEXT** - do 4Gb

♥ DATE AND TIME

- **DATE** - za datume u rasponu od 1000-ite godine do godine 9999
- **TIME** - za vremena u rasponu ± 840 sati (npr. od trenutnog vremena i sl.)
 - ↳ ovaj je 839:59:59
 h min sec
- **DATETIME** - za kombinacije date/time (2011-05-04 15:32:29)
- **TIMESTAMP** - za logiranje događaja, automatski se - sa umetanjem retka

• BIT - tip

↳ bit(8) }
 ↳ bit(10) } 8 bitova
 10 bitova

- insert into bittest (b1, b2) values (b'11110000', b'01001')

↳ pomoću b označimo da su podaci binarni

- čitanje binarnih funkcija

SELECT BIN(b1), BIN(b2) FROM bittest

\ /
 FUNKCIJA ZA
 BINARNI PRIKAZ

- za čitanje binarnih podataka

select b1 + 0, b2 + 0 FROM bittest

↳ zbrajanje $+ 0 \rightarrow$ bitovi postoje brojevi

b1	b2
240	9

FUNKCIJE ZA PRIKAZ BINARNIH PODATAKA

```
> SELECT BIN(b1), OCT(b1), HEX(b1) FROM bittest
```

└─ 11110000 └─ 360 └─ F0

• BOOLEAN VALUES

→ u MySQL-u nema boolean tipa, MySQL isprema kao integer

→ BOOL je alias za TINYINT

TRUE je 1
FALSE je 0

PRIMERI:

```
> SELECT 5 = 5;
```

└─ 1 (true)

```
> SELECT 5 = 7;
```

└─ 0 (false)

• ENUMERATION TIP

```
> create table enumtest (
    color enum ('red', 'blue', 'green')
);
```

```
> insert into enumtest (color) values ('red');
```

→ vrijednosti zapravo
sprema kao
integere, a ne
kao text

↓
- ako ovdje umetnemo
re-definiranu vrijednost,
SQL će tiho failat i upisat
prazan red

ENUMERATION TIP
TAKODER
↓

• SET TIP

→ kao enum, samo što može biti odabrano više vrijednosti u
jednoj koloni. ~~Spec~~ Ovaj tip se sprema kao bit

```
> create table settest (
    attrib SET ('bold', 'underline', 'italic')
);
```

→ vrijednost može biti
kombinacija 1, 2 ili
3 vrijednosti od
pomenutih

```
> insert into settest (attrib) values ('bold');
```

```
> insert into settest (attrib) values ('bold,italic');
```

↓
BEZ RAZMAKA OVOJE

Mysql funkcije

7

MYSQL FUNKCIJE

Funkcije se upotrebljavaju da bi dale vrijednosti uzete od drugih podataka

> use world;

> SELECT COUNT(*) FROM Country;

↑
NAZIV
FUNKCIJE

↑
ARGUMENT, JEDAN ILI
VIŠE - IDU U ZAGRADE

STRING FUNKCIJE

- SELECT 'Hello, World';

↳ literal string

↑
FUNKCIJA

↑
ARGUMENTI

> SELECT CONCAT('Hello', 'World');

↳ HelloWorld

CONCAT WITH SEPARATOR
/

> SELECT CONCAT_WS(':', 'Hello', 'World');

↑
1 ARGUMENT JE
SEPARATOR

↳ Hello:World

LPAD, RPAD

- vraća string određene dužine, truncatiran ili popunjen
odabranim znakom

> use world;

> SELECT LPAD(title, 30, ' ') FROM album;

RPAD

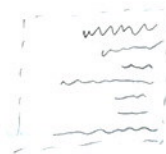
↑
STUPAC U
TABLICI

↑
ZNAK KOJIM
'PUNI' AKO
NEMA DOVOLJNO

REZULTATNI
BROJ
ZNAKOVA

TABLICA
ALBUM

↳



- rezultatni text je right aligned, (padding se s lijeve strane).

- sve linije imaju točno 30 znakova

- ako je neka linija duža od označenog broja znakova - "kida" se

NUMERIC FUNCTIONS

> SELECT 320;

↳ 320 (mali običan broj)

> SELECT 320 / 5;

↳ 64 (mali rezultat operacije)

→ DAJE MATEMATIČKE FJE (+, -, /, *)

> SELECT 320 DIV 60;

↳ 5 (rezultat dijenja bez ostatka)

> SELECT 320 MOD 60;

↳ 20

CONV

> SELECT CONV (125, 10, 16);

↓
BROJ 125
PO BAZI 10

PRETVARA U BROJ
PO BAZI 16 (HEXADECIMALNO)

→ 7D (je rezultat)

> SELECT CONV (1111101, 2, 10); → 125

↓
BINARNI BROJ (BAZA 2)
U DEKADSKI (BAZA 10)

HEX

> SELECT HEX (125);

↳ 7D

BIN

> SELECT BIN (125);

↳ 1111101

OCT

↓
ISTO KAO
OVE PRETHODNE
DVIJE

CRC32

> SELECT CRC32 ('Hello, World');

↳ 643532486

→ DAJE CRC32 HASH OD
STRINGA

> SELECT HEX (CRC32 ('Hello, World'));

↳ 265B86C6

→ HEXANI CRC 32 (ČESTO SE
UPOTREBLJAVA)

MATH FJE

> SELECT DEGREES (PI());
↳ 180°

PI(π) u stupnjevima

> SELECT RADIANS (180);
↳ 3.14...

FORMAT → dává broj decimala i stavljá ^{zareze} ~~točke~~ (za tsnice) u broj

> SELECT FORMAT (10000000, 2);
↳ 10,000,000.00 ↳ BROJ DECIMALA

POWER

→ 16 ON THE POWER OF 2 = 16²
> SELECT POW (16, 2);
↳ 256

RANDOM

> SELECT RAND (); → vraća random broj manji od 1.
↳ 0.31485278 → vrlo korisno

> SELECT Name, Region FROM Country ORDER BY RAND () LIMIT 10;
↳ RANDOMIZIRAN POREDAK

DATE TIME FJE

NOW, CURDATE, CURTIME

> SELECT NOW();

- vraća trenutno vrijeme (datum + vrijeme)

> SELECT CURDATE();

- vraća samo datum

> SELECT CURTIME();

- vraća samo time

≠

UTC_TIMESTAMP

- vraća "Londonsko" vrijeme

> SELECT UTC_TIMESTAMP();

DATE TIME MATH

> SELECT NOW(), UTC_TIMESTAMP(), TIME(NOW() - UTC_TIMESTAMP());

↳ 2:00:00

FJA LJEPD
PRIKAŽE
VRIJEME

DATA REZULTAT U
MEKOM INTERNOM FORMATU

DATE DIFF

- vraća razliku u danima

> SELECT DATEDIFF(NOW(), '2009-05-14');

↳ 903 → broj dana

DATE_FORMAT

- formira vrijeme kako želimo

> SELECT DATE_FORMAT(NOW(), '%W, %D %M %Y, %T');

↑
DATUM

DAY OF
MONTH

4 DIGIT
YEAR

↑
DAY OF
WEEK

↑
MONTH

↑
TIME

TOKENI

↳ Thursday, 21st May 2011, 11:15:36

• TIME ZONES

- informaciju o vremenskoj zoni MySQL ima u sistemskoj varijabli `time_zone`

> SELECT @@time_zone;

↳ "SYSTEM"

→ upotrebljena vremenska zona od sistema

> SET time_zone = 'US/Eastern';

→ postavlja vremensku zonu

→ ako radi error, time zone

> SET time_zone = 'SYSTEM';

tablice nisu instalirane/postavljene

- u bazi se timestamp stupac AUTOMATSKI postavlja (sa unosom retka). Timestamp se sprema u UTC formatu, a prikazuje u ovisnosti o postavljenoj vremenskoj zoni (`SELECT @@time_zone;`)

• INSTALIRANJE TIME ZONE TABLICA

- za podržavanje promjene vrem-zona

> SET time_zone = 'US/Eastern';

→ error

> USE mysql;

> SELECT COUNT(*) FROM time_zone_name;

→ 0

→ Baza i tablica postoje, ali su prazne

- skini package sa dev.mysql.com/downloads/timzones.html (paket bez leap seconds)

- kopiraj dobivene file u: `Applications/XAMPP/xamppfiles/var/mysql/mysql`

- ne u ovaj dir ↑

- `$ chmod 775 time*`

→ promijeniti permissions time fileama

- uć u mysql shell

> use mysql;

> select count(*) from time_zone_name;

→ 536... ✓

• AGGREGATE FUNKCIJE

COUNT

COUNT (*) → STANDARDNO VRAĆA SVE

> SELECT COUNT (DISTINCT Region) FROM Country;

↳ VRAĆA UNIQUE VRIJEDNOSTI

> SELECT Region, COUNT (Region) FROM Country GROUP BY Region;

↳

Region	count (Region)
...	5
...	8
...	12
...	7

→ Broj osoblja za region

> SELECT Region, GROUP_CONCAT (Name) FROM Country GROUP BY Region;

↳ Daje popis regiona i concatenirana imena koja pripadaju tom regionu

> SELECT Region, GROUP_CONCAT (Name ORDER BY Name SEPARATOR ' / ')
FROM Country GROUP BY Region;

↳ ispiše kao i prije, poreda imena alfabetski i koristi drugi separator

FULLTEXT SEARCH

Full text search

```
> CREATE TABLE articles(  
  id INT AUTO INCREMENT NOT NULL PRIMARY KEY;  
  title VARCHAR(255),  
  body TEXT,  
  FULLTEXT (title, body)  
);
```

- stvara fulltext index na title i body stupcima

UNOS 2 VRIJEDNOSTI ODJEDNOM

```
> INSERT INTO articles (title, body) VALUES ('Mysql vs YourSql', 'In this database...');
```

QUERY ZA FULLTEXT SEARCH

```
> SELECT * FROM articles WHERE MATCH (title, body) AGAINST ('database');
```

SPECIFICIRAN KOLONA
ZA FULLTEXT SEARCH

SPECIFICIRAN
SEARCH TERM

VRATI RETKE SA RIJEČI DATABASE (CASE INSENSITIVE)

NATURAL LANGUAGE SEARCH

↳ NE vraća rezultate za search terms koji postoje u više od 50% redaka
(da se ne bi vraćalo previše rezultata)

```
> SELECT * FROM articles WHERE MATCH (title, body) AGAINST ('database tutorial');
```

2 odvojena search termina.
Vraća retke koji imaju ili prvu ili drugu ili obje riječi.

BOOLEAN MODE SEARCH

> SELECT * FROM articles WHERE MATCH (title, body) AGAINST ('+database
+management' IN BOOLEAN MODE);

↓
STAVIT + KRAJ RIJEČI KOJE SE OBAVEZNO
MORAJU POJAVIT

ISKLJUČIVANJE REZULTATA

> SELECT * FROM articles WHERE MATCH (title, body) AGAINST ('+database -management'
IN BOOLEAN MODE);

↓
SA - SE ISKLJUČUJU ONI RETCI
KOJI SADRŽE POREĐENU RIJEČ

WITH QUERY EXPANSION SEARCH

1. PRVI PUT U TRAŽENJU KROZ BAZU PRONALAZI SPECIFICIRANE REZULTATE
2. U DRUGOM PROLAZKU - NAĐE IZ REZULTATA RIJEČI KOJE SE ČESTO PONAVLJAJU I TRAŽI I NJIH

> SELECT * FROM articles WHERE MATCH (title, body) AGAINST ('database' WITH QUERY
EXPANSION);