

Timothy Boronczyk

 sitepoint



**OKAMŽITĚ**

computer  
press

**OVLÁDNĚTE MySQL ZA VÍKEND**

# MySQL Okamžitě

Vyšlo také v tištěné verzi

Objednat můžete na  
[www.computerpress.cz](http://www.computerpress.cz)  
[www.albatrosmedia.cz](http://www.albatrosmedia.cz)



**Timothy Boronczyk**  
**MySQL Okamžitě – e-kniha**  
Copyright © Albatros Media a. s., 2016

Všechna práva vyhrazena.  
Žádná část této publikace nesmí být rozšiřována  
bez písemného souhlasu majitelů práv.

  
**ALBATROS** MEDIA a.s.

# Obsah

<b>O autorovi</b>	<b>5</b>
<b>Co je databáze?</b>	<b>7</b>
<b>Od Edgara Codda k MySQL: stručná historie</b>	<b>9</b>
<b>Alternativy a budoucnost MySQL</b>	<b>11</b>
<b>Komu je tato kniha určena</b>	<b>12</b>
<b>Použité konvence</b>	<b>12</b>
<b>Zpětná vazba od čtenářů</b>	<b>14</b>
<b>Errata</b>	<b>14</b>
 KAPITOLA 1	
<b>Začínáme pracovat s MySQL</b>	<b>15</b>
<b>Instalace MySQL na Linuxu</b>	<b>16</b>
<b>Instalace MySQL ve Windows</b>	<b>21</b>
<b>Komunikace se serverem</b>	<b>23</b>
<b>Účty a zabezpečení MySQL</b>	<b>25</b>
<b>Shrnutí</b>	<b>26</b>
 KAPITOLA 2	
<b>Ukládání dat</b>	<b>29</b>
<b>Vytváření tabulek</b>	<b>30</b>
<b>Přidávání dat</b>	<b>43</b>
<b>Shrnutí</b>	<b>47</b>
 KAPITOLA 3	
<b>Načítání a aktualizace dat</b>	<b>49</b>
<b>Instalace databáze Sakila</b>	<b>49</b>
<b>Načítání dat</b>	<b>51</b>
<b>Udržování dat v aktuálním stavu</b>	<b>61</b>
<b>Shrnutí</b>	<b>64</b>

## KAPITOLA 4

<b>Práce s několika tabulkami</b>	<b>67</b>
<b>Spojování tabulek</b>	<b>68</b>
<b>Abstrakce pomocí pohledů</b>	<b>75</b>
<b>Normální formy</b>	<b>77</b>
<b>Změny tabulek</b>	<b>82</b>
<b>Shrnutí</b>	<b>84</b>

## KAPITOLA 5

<b>Připojení k databázi pomocí kódu</b>	<b>85</b>
<b>Připojení k MySQL z jazyka Python pomocí knihovny Connector/Python</b>	<b>85</b>
<b>Připojení k MySQL z jazyka PHP pomocí rozšíření PDO</b>	<b>91</b>
<b>Připojení k MySQL z jazyka R pomocí knihovny RMySQL</b>	<b>97</b>
<b>Shrnutí</b>	<b>101</b>

## KAPITOLA 6

<b>Programování databáze</b>	<b>103</b>
<b>Základy programování databáze</b>	<b>104</b>
<b>Funkce</b>	<b>107</b>
<b>Uložené procedury</b>	<b>110</b>
<b>Triggery</b>	<b>112</b>
<b>Události</b>	<b>116</b>
<b>Uživatelsky definované funkce</b>	<b>118</b>
<b>Shrnutí</b>	<b>124</b>

## KAPITOLA 7

<b>Zálohování a replikace</b>	<b>127</b>
<b>Logické zálohy</b>	<b>127</b>
<b>Fyzické zálohy</b>	<b>130</b>
<b>Replikace</b>	<b>132</b>
<b>Plánujte dopředu</b>	<b>137</b>
<b>Shrnutí</b>	<b>138</b>
<b>Rejstřík</b>	<b>139</b>

# O autorovi

Timothy Baranczyk žije ve městě Syracuse ve státě New York a pracuje jako vývojář ve společnosti ShoreGroup, Inc. Webovým technologiím se věnuje již od roku 1998. Svoje vysokoškolské vzdělání završil získáním titulu v oboru programování. Kromě toho je také držitelem certifikátu Zend Certified Engineer. Svůj volný čas, kterého ale nemá příliš mnoho, tráví rád s přáteli. S oblibou také hovoří esperantem a miluje spaní s nohama vyčnívajícíma z postele. Poměrně snadno jej rozptýlí jakékoliv blýskavé předměty.



# Úvod

Lze říci, že s daty se v současné době potkáte všude: počínaje ručně naškrábanými nákupními seznamy a konče rozsáhlými sadami dat (zvanými „big data“) v datových centrech nadnárodních korporací. Korporace shromažďují tolik dat, kolik jim umožňují jejich technologické možnosti, a následně je využívají k formulování nových obchodních a podnikových strategií. Vědci studují data proto, aby s jejich pomocí našli odpovědi na otázky, které mohou zachránit další lidské životy, zlepšit naše prostředí a vysvětlit naše místo v celém vesmíru. Dokonce i průměrný člověk shromažďuje během svého života poměrně značné množství dat, k nimž patří jak údaje v domácím účetnictví, popisující finanční chování každého z nás, tak i telefonní čísla v seznamu, uloženém v chytrém telefonu. Ukládání a celková organizace dat se natolik usnadnilo, že mnohé z databázových konceptů a algoritmů, díky nimž se všechny tyto aktivity staly vůbec možnými, dnes již považujeme za samozřejmé.

Tato kniha je úvodem do základních konceptů práce s relačními systémy řízení báze dat (RDBMS – Relational Database Management System), především tedy oblíbeným a volně šiřitelným systémem MySQL. Podobně jako další knihy z řady *Okamžitě* se i takto kniha snaží o to, aby vám poskytla určitý náskok v porozumění zvolené technologii. Velice rychle – a doufejme, že i bez velkých bolestí hlavy – se seznámíte se základy, díky čemuž budete moci dále pokračovat ve svém dalším studiu.

Jsem velmi vděčný za to, že mi byla nabídnuta možnost napsat tuto knihu. Co ji od ostatních knih v této řadě odlišuje, je, že se věnuje technologii, která se využívá jak při vývoji webových aplikací, tak i mimo něj. Tím samozřejmě nechceme říci, že MySQL není oblíbená u vývojářů zabývajících se tvorbou webových aplikací – právě naopak! Databáze se však používají i v mnoha dalších oblastech, což jsem se snažil postihnout svým výběrem témat.

## Co je databáze?

Ačkoliv máme obecnou tendenci spojovat význam slova **databáze** s digitálním světem počítačů, tento pojem ve skutečnosti označuje jakoukoliv uspořádanou kolekci dat. To tedy znamená, že databáze může být jak digitální (elektronická), tak také fyzická. Typickou fyzickou databází je například i skříň stojící ve vaší domácí pracovně a obsahující složky, plné faktur a dalších účetních dokladů. Anebo jí může být i sbírka kuchařek, uložených na polici, majících stránky s osahanými rohy a přetékajících dodatečnými recepty, které jste si vystříhli z nějakých časopisů a založili do jednotlivých knih.

V digitálním světě jsou databáze členěny podle způsobu, jímž jsou v nich ukládána a uspořádávána data. Mezi běžné typy digitálních databází patří:

- **Prosté databázové soubory** – v nich jsou data ukládána sekvenčně, velice často pak ve formátu prostého textu. Tyto soubory lze sice poměrně snadno vytvářet a přidávat do

nich další data, avšak jejich používání je provázeno i několika nevýhodami. Vyhledávání dat v prostých databázových souborech je poměrně pomalé, tyto soubory mohou obsahovat i redundantní data, a navíc je možné je vcelku snadno poškodit. Příkladem takového typu databáze může být textový soubor vytvořený při hře Solitaire a obsahující informace o výsledcích jednotlivých hráčů.

- **Hierarchické databáze** – v nich jsou data ukládána ve stromové struktuře (pomocí vztahů rodič – dítě). Jedná se o vysoce uspořádané databáze, nabízející velice efektivní vyhledávání. Jejich nevýhodou však je poměrně obtížné procházení v případě, že neznáte použité vztahy. V průběhu času se může stát obtížnou i údržba vztahů mezi daty. Příkladem hierarchické databáze může být registr systému Windows.
- **Dokumentově orientované databáze** – v nich jsou data ukládána ve volné podobě, přičemž jsou indexována pomocí klíčových či hashovacích hodnot. Tyto databáze na jedné straně nabízejí poměrně dobrou škálovatelnost v rozsáhlých sítích, na straně druhé je však jejich používání spojeno i s mnoha problémy typickými pro prosté databázové soubory. Tyto databáze totiž mohou obsahovat redundantní data, nejsou v nich udržovány žádné vztahy mezi daty a vyhledávání dat může být pomalé. Redis a CouchDB představují poměrně oblíbené „NoSQL“ databázové systémy, sloužící k řízení těchto typů databází.
- **Relační databáze** – v nich jsou data ukládána do tabulek tvořených řádky a sloupci. Do podobných tabulek mohou být uspořádány například i vytištěné ceníky či jízdni řády autobusů. Relační databáze podporují indexování velkých objemů dat, umožňující jejich rychlé načítání, nicméně vztahy (vazby) mezi jednotlivými tabulkami mohou být velice složité.

Nad většinou moderních digitálních databází se nachází **databázový server**, což je aplikace, která je speciálně navržena k řízení databází a je odpovědná za umožnění a řízení přístupu k uloženým datům. V těchto systémech totiž platí, že nikdy nepracujete přímo s daty. Namísto toho na server odesíláte požadavky na přidání nějakých dat, jejich změnu, výmaz či načtení. Následně server za vás provede požadované operace a výsledky vám předá. Kniha, kterou právě čtete, je zaměřena na MySQL, což je databázový server určený k řízení relačních databází.

Standardním jazykem používaným pro komunikaci s relačními systémy řízení báze dat je zhruba od poloviny 80. let minulého století jazyk **SQL** (Structured Query Language – strukturovaný dotazovací jazyk). Jazyk SQL sestává z příkazů pro vkládání, načítání a řízení dat, příkazů pro vytváření a údržbu tabulek, a dokonce i z příkazů pro správu databází. Jednotlivé příkazy mohou být rozděleny do kategorií či „podjazyků“ daných jejich určením: příkazy sloužící pro práci se samotnými daty vytvářejí DML (Data Manipulation Language – jazyk pro manipulaci s daty); příkazy sloužící ke správě tabulek a databází vytvářejí DDL (Data Definition Language – jazyk pro definici dat); a nakonec příkazy sloužící k řízení přístupu k databázi vytvářejí DCL (Data Control Language – jazyk pro řízení dat).



Je dobré, abyste o tomto členění příkazů jazyka SQL věděli, neboť by se mohlo stát předmětem diskuze na nějaké party databázových administrátorů, které se třeba zúčastníte; já však v této knize takto podrobné členění používat nebudu. Jinými slovy řečeno, všechny příkazy jazyků DML, DDL a DCL budu souhrnně označovat jako SQL.

## Od Edgara Codda k MySQL: stručná historie

V původních databázích byla data uspořádávána do stromových struktur. Přístup k takovým datům pak vyžadoval to, aby programátor napsal kód přímo procházející tyto struktury. Lze říci, že tento způsob práce s daty byl poměrně křehký a jakékoliv operace související s přidáváním dalších dat, aktualizacemi dat stávajících či změnami uspořádání dat byly dosti riskantní. Edgar Codd zkritizoval tento přístup ve své práci, nazvané *A Relational Model of Data for Large Shared Data Banks*, kterou vydal roku 1970. Tvrdil, že daleko lepší by bylo data uspořádat do tabulek a pracovat s nimi nezávisle na informacích o vzájemných vazbách, řazení a indexování. V té době se jednalo o velice zajímavý a inovativní koncept, kterého se chopila společnost IBM. Její inženýři ve vývojových laboratořích v San Jose začali pracovat na projektu s označením System R, který měl ověřit platnost teorií Edgara Codda.

Výsledkem projektu System R byla nejen první implementace jazyka SQL, ale také jasné důkazy potvrzující platnost relačních konceptů prosazovaných Edgarem Coddem. Když se Larry Ellison doslechl o výzkumu vedoucím k prototypu System R, byl těmito informacemi tak zaujat, že myšlenky Edgara Codda a jazyk SQL začlenil do svého vlastního databázového serveru nazvaného Oracle. Mimochodem, při uvádění produktu na trh Larry Ellison porazil společnost IBM, a v roce 1979 se tak Oracle stal prvním komerčně dostupným relačním systémem řízení báze dat.

Mezitím se však o práci Edgara Codda začali zajímat i profesori počítačových věd pracující na univerzitě státu Kalifornie se sídlem v Berkeley. Univerzitě se podařilo získat dostatek finančních zdrojů z National Science Foundation a z výzkumných divizí vojenského letectva a pozemních vojsk armády USA. Díky tomu byla univerzita schopna vytvořit rotující tým studentů vedený Michaellem Stonebrakerem a pracující na projektu INGRES. Tento projekt podrobně prozkoumal mnohé z relačních myšlenek publikovaných Edgarem Coddem, avšak kromě nich implementoval i vlastní dotazovací jazyk, nazvaný QUEL. Současně s tím, jak jednotliví studenti dokončovali svoje studia a odcházeli pracovat do jiných softwarových společností, se na trhu začaly objevovat databázové systémy inspirované projektem INGRES. K těm nejzajímavějším patřil Sybase, jehož licenci si později zakoupila společnost Microsoft. Ta jej pak přejmenovala na Microsoft SQL Server. I samotný INGRES byl nakonec uveden na trh v komerční podobě a poměrně rychle zaujal na trhu vůdčí pozici.

Ta se však začala otrásat zhruba okolo roku 1985, kdy počítačová veřejnost začala upřednostňovat jazyk SQL před QUEL. A navíc roku 1987 byl jazyk SQL přijat jako standard, a to jak americkou organizací ANSI (American National Standards Institute), tak i mezinárodní

organizací ISO (International Organization for Standardization). Na konci této dekády pak trhu jasně vévodil databázový systém Oracle spolu s jazykem SQL.

V roce 1993 David Hughes pracoval na aplikaci pro monitorování sítě, která si ukládala data do databáze řízené systémem Postgres (což byl jeden z nástupců systému INGRES). Z důvodu přenositelnosti však chtěl zajistit přístup k těmto datům i prostřednictvím jazyka SQL, a proto napsal překladač z jazyka QUEL do jazyka SQL, který nazval miniSQL. Během dalšího vývoje této monitorovací aplikace však byl David Hughes čím dál více zklamán hardwarovými nároky databázového systému Postgres, a proto se rozhodl, že na základě miniSQL vyvine svůj vlastní hardwarově nenáročný systém řízení báze dat. Řešení miniSQL upřednostňovalo malé hardwarové nároky před striktním a úplným dodržováním standardů SQL, a proto v něm byly implementovány pouze ty nejdůležitější části tohoto standardu. David Hughes byl schopen distribuovat svůj systém za zlomek nákladů ostatních komerčně dostupných databázových systémů. Lze tedy říci, že miniSQL se stalo prvním nízkonákladovým relačním systémem řízení báze dat založeným na standardu SQL. Vše bylo připraveno pro příchod MySQL.

V téže době Monty Widenius vyvíjel webovou aplikaci pro – v té době – bouřlivě se rozvíjející internet. Základem jeho aplikace byl databázový server UNIREG, který si sám vytvořil. Monty Widenius zjistil, že přístup k systému UNIREG při generování dynamických webových stránek vyžaduje příliš mnoho prostředků, a proto se začal poohlížet po alternativním řešení. Jeho pozornost zaujal systém miniSQL, který se stal velmi populárním především díky své cenové strategii, zejména pak mezi poskytovateli webhostingových služeb. Systém miniSQL však neimplementoval některé z funkcí, které Monty Widenius potřeboval. Z tohoto důvodu se Monty Widenius rozhodl přepsat svůj systém UNIREG tak, aby zajišťoval vyšší výkon. Současně však využil tuto příležitost i k tomu, aby reimplementoval API systému UNIREG tak, aby nabízelo kompatibilitu s API systému miniSQL. Díky tomu by pak mohl využít mnoho nástrojů třetích stran nabízených pro systém miniSQL. Monty Widenius přejmenoval takto upravený systém na MySQL a jeden z jeho přátel jej přesvědčil, aby jej zveřejnil.

Systém MySQL byl zpřístupněn pod licencí GNU General Public License a roku 1995 Monty Widenius spolu s Davidem Axmarkem a Allanem Larssonem založil společnost MySQL AB, jejímž cílem bylo vést další vývoj MySQL a nabízet alternativní licencování a podporu pro komerční zákazníky. A zatímco systém miniSQL byl cenově dostupný, systém MySQL byl pro většinu zákazníků prakticky zdarma.

Jelikož licenční podmínky MySQL umožňovaly začlenění tohoto systému do většiny distribucí Linuxu a jelikož API systému MySQL bylo kompatibilní s API miniSQL, přičemž ale nabízelo více možností, systém MySQL velice rychle zaujal tržní pozici miniSQL. Lze říci, že v současné době je systém MySQL druhým nejrozšířenějším relačním systémem řízení báze dat založeným na jazyku SQL (přičemž první místo zaujímá systém SQLite, a to především díky využití v chytrých telefonech a vestavěných systémech).

# Alternativy a budoucnost MySQL

Roku 2008 společnost Sun Microsystems zakoupila společnost MySQL AB za cenu 1 miliardy USD a o 2 roky později Oracle Corporation získala společnost Sun Microsystems včetně veškerých jejích aktiv (tedy včetně MySQL) za cenu 7,4 miliardy USD. Tatáž společnost, která na konci 80. let minulého století porazila IBM a INGRES, se tak stala vlastníkem autorských práv k MySQL. Přitom společnost Oracle již měla svůj vlastní databázový systém, díky čemuž se veškeré obavy o budoucnost MySQL, které začala mít komunita ve chvíli, kdy se vlastníkem stala společnost Sun, pouze zesílily.

Avšak díky licenci GPL platí, že každý smí MySQL vzít a dále ji vylepšovat. Jedinou podmínkou je, že veškeré provedené změny musí mít odpovídající licenci. To znamená, že ostatní mohou databázový systém MySQL nejen rozšiřovat, ale dokonce mohou vyvíjet i svoje vlastní varianty (ve světě otevřeného softwaru nazývané forky) a veřejně je zpřístupňovat. A forků MySQL dnes existuje již celá řada:

- **Dorsal Source** – první fork MySQL, připravený společností Proven Scaling jako reakce na pomalý proces uvolňování dalších verzí ze strany Sun Microsystems. Kromě toho tato společnost také spravovala opravy chyb a rozšíření připravená komunitou. V současné době již tento projekt není dále aktivní.
- **Drizzle**<sup>1</sup> – fork MySQL připravený Brianem Akerem, jehož cílem je nabídnout rychlejší variantu MySQL obsahující pouze nejdůležitější a nezákladnější funkce. Tento fork je určený především pro podporu webových aplikací. Základní funkcionalita je zajišťována jádrem, zatímco doplňkové funkce jsou zajišťovány zásuvnými moduly. Projekt sice ještě nezanikl, nicméně se zdá, že vývoj se na nějakou dobu pozastavil.
- **Percona Server**<sup>2</sup> – fork MySQL podporovaný konzultační společností Percona LLC. Cílem tohoto forku je vytvořit rychlou náhradu MySQL nabízející zvýšený výkon a některé další funkce, které nejsou součástí komunitní edice zveřejněné společností Oracle.
- **MariaDB**<sup>3</sup> – fork MySQL, který vytvořil sám Monty Widenius jako reakci na to, že společnost MySQL AB byla nejprve prodána společnosti Sun Microsystems a posléze se s i s ní stala součástí Oracle. Tento fork chce být především komunitní náhradou zajišťující rovnocenné funkce ve většině případů.

## Další informace

Jednotlivé forky MySQL jsou podrobněji rozebrány v přednášce „Different MySQL Forks for Different Folks“<sup>4</sup>, kterou přednesla Sheeri Cabral na konferenci Confoo v roce 2013.

1 <http://www.drizzle.org/>

2 <https://www.percona.com/software/mysql-database/percona-server>

3 <https://mariadb.org/>

4 <https://www.youtube.com/watch?v=dcWoHusSAsE>

I přes určité napětí v komunitě je dlouhodobý výhled pro „značku“ MySQL dobrý. Na rozdíl od očekávání mnoha lidí společnost Oracle další vývoj databázového systému MySQL neukončila. Dokonce lze říci, že kvalita jednotlivých vydání se pod vedením společnosti Oracle ve skutečnosti zvýšila. Jednotlivé forký tak zajišťují konkurenci, což může – doufejme – systému MySQL jedině prospět. I když se na celý vývoj podíváme tím nejpesimističtějším náhledem, zjistíme, že v poslední dekádě došlo v podnicích k nárůstu využívání otevřeného softwaru, díky čemuž nelze očekávat, že by se systém MySQL měl v nejbližší době stát pouhou historií.

## Komu je tato kniha určena

Tato kniha je určena především pro ty z vás, kteří se zajímají o práci s daty a chtějí se naučit používat MySQL. A aby pro vás některé části této knihy byly skutečně přínosné, měli byste mít alespoň nějaké předcházející zkušenosti z oboru programování, byť znalost nějakého konkrétního jazyka není nutná.

## Použité konvence

V této knize narazíte na řadu typografických konvencí a různých stylů rozvržení, které budou zvýrazňovat různé typy informací. Prohlédněte si proto následující konvence.

## Ukázky zdrojového kódu

Zdrojový kód bude napsán neproporcionálním písmem, například takto:

```
<h1>Skvělý letní den</h1>
<p>Byl krásný letní den, který přímo vybízel na procházku parkem. Ptáčci zpívali a všechny děti byly zpátky ve škole.</p>
```

Když budeme k současnému příkladu přidávat další zdrojový kód, takový kód se bude zobrazovat tučně:

```
function animuj() {
    nova_promenna = "Ahoj";
}
```

Místo již napsaného zdrojového kódu bude výpis obvykle obsahovat tři tečky:

```
function animuj() {
    ...
    return nova_promenna;
}
```

## Tipy, poznámky a upozornění



**Tip: Haló, vy tam!**

Tipy vám poskytují malé cenné rady.



**Poznámka: Ehm, promiňte...**

Poznámky jsou doplňující informace, které se týkají tématu, ale nejsou kriticky důležité.



**Upozornění: Nezapomeňte vždy...**

... věnovat pozornost těmto sdělením.

## Zpětná vazba od čtenářů

Nakladatelství a vydavatelství Computer Press, které pro vás tuto knihu přeložilo, stojí o zpětnou vazbu a bude na vaše podněty a dotazy reagovat. Můžete se obrátit na následující adresy:

*Computer Press*

*Albatros Media a.s., pobočka Brno*

*IBC*

*Příkop 4*

*602 00 Brno*

nebo

*sefredaktor.pc@albatrosmedia.cz*

**Computer Press neposkytuje rady ani jakýkoli servis pro aplikace třetích stran. Pokud budete mít dotaz k programu, obraťte se prosím na jeho tvůrce.**

## Errata

Přestože jsme udělali maximum pro to, abychom zajistili přesnost a správnost obsahu, chybám se úplně vyhnout nelze. Pokud v některé z našich knih nějakou najdete, ať už v textu nebo v kódu, budeme rádi, pokud nám ji oznámíte.

Veškerá existující errata zobrazíte na adrese <http://knihy.cpress.cz/K2274> po klepnutí na odkaz Soubory ke stažení. (Nejsou-li žádná errata zatím k dispozici, není odkaz Soubory ke stažení dostupný.)

# Začínáme pracovat s MySQL

## V této kapitole:

- Instalace MySQL na Linuxu
- Instalace MySQL ve Windows
- Komunikace se serverem
- Účty a zabezpečení MySQL
- Shrnutí

V této kapitole se seznámíte s prvními kroky, které budete muset provést, budete-li chtít začít používat MySQL. Ukážeme si, jak se MySQL instaluje jak na platformě Linuxu, tak i na platformě Windows. Proto při čtení textu dávejte pozor na to, abyste postupovali podle informací vztahujících se k vám zvolené platformě. Poté uvidíte, jak se pracuje s klientem příkazového řádku systému MySQL, neboť právě tohoto klienta využijeme pro připojení k databázovému serveru a vytvoření naší první databáze.

Prvním krokem při instalování mnoha aplikací velice často bývá určení vhodné verze. Z tohoto důvodu je nyní vhodné připomenout, že systém MySQL je dostupný v několika edicích či variantách. Společnost Oracle vám nabízí jak volně dostupnou edici Community Edition, tak také placené edice s názvy Standard, Enterprise a Cluster Carrier Grade. Rozdíly mezi Community Edition a placenými edicemi spočívají především v licencování a podpoře možnosti využívání některých dodatečných serverových zásuvných modulů a nástrojů pro zálohování a monitorování.

Jak jsme však již řekli, MySQL je volně šiřitelný software, publikovaný pod licencí GNU General Public License, a proto by vás nemělo překvapit to, že existují i alternativní forky. Dvěma oblíbenými forky jsou MariaDB, což je komunitou udržovaná „rozšířená náhrada“ MySQL, a Percona Server, podporovaný konzultační společností Percona LLC. Běžný uživatel si však rozdílů mezi MySQL, MariaDB a Percona Server většinou ani nevšimne.

Samozřejmě můžete použít kteroukoliv ze zmíněných verzí či edicí MySQL, nicméně v rámci snahy o udržení pozornosti a konzistence budu já osobně pracovat s edicí MySQL Community Edition verze 5.7.11 (stabilní verze, aktuální v době překladu této knihy). Současně budu veškeré dále popisované postupy omezovat na operační systémy Debian/Ubuntu, RedHat/CentOS a Windows Server 2012. Domnívám se totiž, že toto jsou ty

operační systémy, na nichž budete nasazovat MySQL v produktivním prostředí s největší pravděpodobností.



#### **Poznámka: Lokální vývojové prostředí**

Těm z vás, kteří si chtějí připravit instalaci MySQL pro účely lokálního vývoje a testování, doporučuji vytvoření virtuálního stroje pomocí nástroje VirtualBox<sup>1</sup> společnosti Oracle. V tomto virtuálním stroji si můžete nainstalovat jeden z výše uvedených operačních systémů a následně – podle instrukcí, které najdete níže – si nainstalovat i MySQL. Toto řešení vám dává možnost nejenom pracovat ve vývojovém prostředí, jehož konfigurace se bude v maximální možné míře podobat konfiguraci produktivního prostředí, aniž by bylo vázáno na určitý server či určitou síť, ale navíc i váš lokální systém bude ušetřen instalace dalších služeb a aplikací, a to bez ohledu na to, zda tím lokálním systémem je Linux, Windows či Mac OS X.

## Instalace MySQL na Linuxu

Operační systém Linux rozhodně nepředstavuje homogenní platformu. Mimo jiné i proto, že jednotlivé distribuce se liší i preferovaným způsobem instalace softwaru. V této části si ukážeme, jak se MySQL instaluje na systémech Debian/Ubuntu a Red Hat/CentOS pomocí správce balíčků a jak se dá MySQL zkompilovat a následně i nainstalovat ze zdrojových kódů. Získáte tak dovednosti nezbytné k provedení instalace MySQL na systém Linux téměř ve všech scénářích, se kterými se můžete setkat.

### Instalace pomocí správce balíčků

Většina moderních systémů řady Linux využívá ke značnému zjednodušení instalací dalších softwarů nějakého správce balíčků. A jelikož je databázový systém MySQL natolik rozšířený, je velice pravděpodobné, že buď přímo MySQL, nebo některý z jeho forků již existuje v repozitáři balíčků vámi zvolené distribuce. Pokud je nám známo, například Debian/Ubuntu nabízí ve svých repozitářích edici MySQL Community Edition, takže uživatelé mohou MySQL nainstalovat zadáním jednoduchého příkazu `sudo apt-get install mysql-server`. Naopak v repozitářích Red Hat/CentOS byl systém MySQL nedávno nahrazen forkem MariaDB; v tomto případě je tedy nutné k instalaci použít příkaz `su -c 'yum install mariadb-server'`.

Instalace softwaru z repozitáře udržovaného tvůrci distribuce je pro většinu uživatelů optimální, nicméně je nutné si uvědomit, že budete-li spoléhat na tyto repozitáře, může se stát, že výsledkem instalace nebude nejnovější verze. Naštěstí se ale nemusíme vzdát pohodlí, které nám práce s balíčky nabízí. Společnost Oracle totiž udržuje aktuální balíčky ve formátech RPM i DEB, které si můžete nainstalovat pomocí příkazů `rpm` a `dpkg`. Kromě toho ale

---

<sup>1</sup> <https://www.virtualbox.org/>



společnost Oracle udržuje i repozitáře APT a Yum a nabízí speciální balíčky automaticky přidávající tyto repozitáře do seznamu známých repozitářů vašeho systému.

Pomocí následujících kroků můžete ve svém systému zaregistrovat jeden z repozitářů společnosti Oracle a následně z něj nainstalovat MySQL Community Edition. Neběží-li na vašem serveru grafické uživatelské rozhraní a nemůžete-li použít nějaký textový webový prohlížeč, například Lynx, budete muset provést první čtyři kroky na nějakém jiném systému a nakopírovat takto získaný soubor na váš server.

1. Spustíte si webový prohlížeč a přejděte na stránku repozitářů MySQL, mající adresu <http://dev.mysql.com/downloads/repo>.
2. V závislosti na správci balíčků vámi použité distribuce Linuxu klepněte na odkaz **DOWNLOAD** nacházející se v části **MySQL Yum Repository** anebo **MySQL APT Repository**. Budete přeměrováni na stránku obsahující seznam různých konfiguračních balíčků.
3. Dále klepněte na tlačítko **Download** nacházející se vpravo od balíčku vhodného pro váš systém. Například uživatelé systému Red Hat/CentOS 7 by si měli stáhnout balíček s názvem **Red Hat Enterprise Linux 7 / Oracle Linux 7 (Architecture Independent), RPM Package**. Naopak uživatelé systému Ubuntu 14.04 Trusty Tahr by si měli stáhnout balíček s názvem **Ubuntu / Debian (Architecture Independent), DEB**.
4. V dalším kroku se vás společnost Oracle bude snažit přesvědčit k tomu, abyste si u ní vytvořili nějaký účet. Tento krok však není povinný, a proto se přesuňte do spodní části stránky, kde klepněte na odkaz **No thanks, just start my download**. Poté by se již mělo zahájit stahování vámi zvoleného balíčku.
5. Otevřete si okno terminálu, přejděte do složky, do níž jste stáhli (či nakopírovali) zvolený balíček, a spuštěním odpovídajícího příkazu tento balíček nainstalujte:
  - Uživatelé systému Red Hat/CentOS by k instalaci měli použít příkaz:
 

```
rpm -i mysql57-community-release-e17-7.noarch.rpm
```
  - Uživatelé systému Debian/Ubuntu by k instalaci měli použít příkaz:
 

```
dpkg -i mysql-apt-config_0.6.0-1_all.deb
```
6. Repozitář je nyní zaregistrován a vy již můžete nainstalovat MySQL Community Edition pomocí správce balíčků své distribuce:
  - Uživatelé systému Red Hat/CentOS by k instalaci měli použít příkaz:
 

```
su -c 'yum install mysql-community-server'
```
  - Uživatelé systému Debian/Ubuntu by k instalaci měli použít příkaz:
 

```
sudo apt-get install mysql-server-5.7
```

Uživatelé systému Ubuntu budou v průběhu instalace navíc vyzváni k zadání hesla uživatele root systému MySQL (uživatelé systémů Debian a Red Hat/CentOS budou moci zadat toto heslo až v následujícím kroku, po spuštění jednoho poinstallačního příkazu). V této souvislosti je nutné zdůraznit, že MySQL si udržuje svůj vlastní seznam uživatelských účtů,

kteře jsou nezávislé na uživatelských účtech operačního systému – jinými slovy řečeno, byť uživatelské jméno může být shodné, uživatel root databázového systému MySQL není shodný s uživatelem root Linuxu.

Uživatelé systému Red Hat/CentOS by poté měli spustit tyto poinstallační příkazy, pomocí nichž nastaví heslo uživatele root MySQL, zaregistrují MySQL jako systémovou službu a spustí její instanci (systémy Debian/Ubuntu automaticky registrují a spouštějí MySQL):

1. K nastavení hesla uživatele root databázového systému MySQL použijte příkaz:  
`mysqladmin -u root heslo`
2. Spouštění MySQL během startu operačního systému zajistíte příkazem:  
`su -c 'chkconfig --level 2345 mysqld on'`
3. A nakonec spusťte server MySQL příkazem:  
`su -c 'systemctl start mysql'`

Tímto je dokončena instalace MySQL Community Edition na váš systém. Pro vaši budoucí práci níže uvádíme příkazy sloužící ke spuštění serveru MySQL, jeho zastavení a kontrole jeho stavu:

- Spuštění MySQL
  - Ubuntu – `sudo service mysql start`
  - Debian – `sudo systemctl start mysqld`
  - Red Hat/CentOS – `su -c 'systemctl start mysql'`
- Zastavení MySQL
  - Ubuntu – `sudo service mysql stop`
  - Debian – `sudo systemctl stop mysqld`
  - Red Hat/CentOS – `su -c 'systemctl stop mysql'`
- Zjištění aktuálního stavu MySQL
  - Ubuntu – `service mysql status`
  - Debian – `sudo systemctl status mysqld`
  - Red Hat/CentOS – `su -c 'systemctl status mysql'`



#### **Poznámka: Jednodušší budoucnost**

Ke spuštění, zastavování a kontrole stavu MySQL se používají rozdílné příkazy proto, že Ubuntu 14.04 používá Upstart, zatímco ostatní popisované distribuce používají systemd. Vývojáři Ubuntu předpokládají, že přechod k init systému systemd zahájí od verze 15.04. V době, kdy bude dostupná verze 16.04 s dlouhodobou podporou, by již příkazy pro provádění těchto úkonů měly být zcela shodné s těmi, které se používají v distribuci Debian.

## Instalace ze zdrojů

Systémoví administrátoři kompilují software z jeho zdrojových kódů stále méně často, nicméně taková kompilace vám nabídne možnost úplné kontroly funkcí aplikace, její optimalizace a kontroly jejich konfiguračních nastavení. Jak však asi tušíte, jedná se o nejnáročnější metodu instalace.

V následujícím postupu jsou popsány kroky, které musíte provést, chcete-li si stáhnout zdrojový kód serveru MySQL Community Edition, zkompilovat jej a nakonec nainstalovat. Opět platí, že nemáte-li přístup ke grafickému uživatelskému rozhraní či textovému webovému prohlížeči přímo na serveru, budete muset provést prvních několik kroků na jiném počítači a stažené soubory přenést na požadovaný počítač.

1. Spusťte si webový prohlížeč a přejděte na stránku MySQL Community Downloads, mající adresu <http://dev.mysql.com/downloads>.
2. Klepnutím na odkaz **MySQL Community Server** přejděte na stránku Download MySQL Community Server. Pro výběr požadované cílové platformy můžete využít rozbalovací seznam **Select Platform**.
3. V tomto seznamu si zvolte **Source Code**, poté se na stránce přesuňte dolů k položce **Generic Linux (Architecture Independent)**, **Compressed TAR Archive** a vpravo vedle ní klepněte na odkaz **Download**.
4. Vytvoření účtu není pro pokračování ve stahování nezbytné. Proto přejděte do spodní části stránky a zde klepněte na odkaz **No thanks, just start my download**.
5. Otevřete si okno terminálu a pomocí následujících příkazů vytvořte uživatelský účet, určený pouze ke spouštění serveru MySQL:

```
sudo groupadd mysql
sudo useradd -r -g mysql mysql
```

6. Poté přejděte do složky, do níž jste stáhli zkomprimovaný zdrojový kód. Tento archiv rozbalte a poté přejděte do složky s kódem:

```
cd /tmp
gzip -cd mysql-5.7.11.tar.gz | tar xvf -
cd mysql-5.7.11
```

7. Spuštěním příkazu `cmake` vytvořte skripty pro sestavení. V následující ukázce tohoto příkazu neuvádím žádné volby, nicméně úplný přehled konfiguračních voleb najdete v online dokumentaci<sup>2</sup>.

```
cmake
```

8. Poté spusťte příkaz `make`, jímž server MySQL zkompilujete. Následuje spuštění příkazu `make install`, vyžadující nejvyšší oprávnění. Díky tomu budou výsledné

<sup>2</sup> <http://dev.mysql.com/doc/refman/5.7/en/source-configuration-options.html>

binární soubory, pomocné nástroje, knihovny a soubory dokumentace přepokopírovány do nové domovské složky nacházející se ve vašem systému:

```
make
sudo make install
```

9. Níže vidíte příkazy, jimiž můžete zajistit to, že nainstalované soubory budou mít přiřazeného správného vlastníka a budou k nim nastavena odpovídající přístupová oprávnění:

```
sudo chown -R mysql /usr/local/mysql
sudo chgrp -R mysql /usr/local/mysql
```

10. Datová složka serveru MySQL a jeho systémové tabulky musí být inicializovány skriptem `mysql_install_db`, který najdete v podsložce `scripts` instalační složky. Skript pracuje s cestami zadávanými relativně vůči instalační složce, a proto tento skript raději spouštějte z instalační složky, a nikoliv z podsložky `scripts` či odkudkoliv jinud:

```
cd /usr/local/mysql
sudo scripts/mysql_install_db --user=mysql
```

11. Spusťte server MySQL a nastavte heslo jeho uživatele `root`:

```
sudo mysql_secure_installation &
mysqladmin -u root heslo
```

Tímto je celá instalace serveru MySQL dokončena. Existují však ještě další úkony, související s konfigurací systému, které byste měli zvážit. Doporučuji vám, abyste přidali podsložku `bin` instalační složky do proměnné prostředí `PATH`. Díky tomu budete moci spouštět pomocné nástroje serveru MySQL, aniž byste vždy museli zadávat plnou cestu. Za předpokladu, že používáte Bash, můžete do konfiguračního souboru `/etc/profile` přidat následující řádky:

```
PATH=/usr/local/mysql/bin:$PATH
export PATH
```



#### Poznámka: Práce s proměnnou prostředí `PATH`

Nastavením hodnoty proměnné prostředí `PATH` v souboru `/etc/profile` zjednodušíte přístup k pomocným nástrojům MySQL všem uživatelům systému. Pokud ovšem chcete, abyste takto jednoduchý přístup ke zmiňovaným nástrojům měli pouze vy, přidejte výše uvedené řádky do souboru `~/ .bash_profile` či `~/ .bashrc`.

Je také velice pravděpodobné, že budete chtít zajistit automatické spuštění MySQL v době náběhu systému. Níže uvedené kroky vycházejí z předpokladu, že používáte Linux založený na init systému System V.

1. Přejděte do podsložky `support-files` zdrojového kódu. V ní byste měli najít soubor `mysql.server`, který je nutné zkopírovat do složky `init.d` vašeho systému a poté jej změnit na spustitelný:

```
sudo cp /tmp/mysql-5.7.11/support-files/mysql.server /etc/init.d/mysql
sudo chmod 755 /etc/init.d/mysql
```

2. V jednotlivých požadovaných úrovních běhu vytvořte symbolické odkazy ukazující na tento skript:

```
ln -s /etc/init.d/mysql /etc/rc3.d/S99mysql
ln -s /etc/init.d/mysql /etc/rc0.d/K01mysql
```

A nyní již můžete spustit server MySQL příkazem `sudo /etc/init.d/mysql start`, zatímco k jeho zastavení můžete použít příkaz `sudo /etc/init.d/mysql stop`.

## Instalace MySQL ve Windows

I přesto, že společnost Microsoft vždy aktivně udržuje a podporuje několik verzí operačního systému Windows najednou, lze říci, že ve srovnání se systémy Linux jsou operační systémy řady Windows poměrně homogenní. Níže uvedené kroky se týkají systému Windows Server 2012, nicméně lze předpokládat, že ve větší či menší míře budou použitelné i pro desktopový systém Windows 8 či Windows 8.1.

1. Spusťte si webový prohlížeč a přejděte na stránku MySQL Community Downloads, mající adresu <http://dev.mysql.com/downloads>.
2. Klepnutím na odkaz **MySQL on Windows (Installer & Tools)** přejděte na stránku MySQL on Windows. Na ní klepněte na první odkaz, jímž je **MySQL Installer**.
3. Dostanete se tak na stránku Download MySQL Installer. Přesuňte se dolů k položce **Windows (x86, 32-bit), MSI Installer** a vpravo vedle ní klepněte na odkaz **Download**.
4. V dalším kroku se vás společnost Oracle bude snažit přesvědčit k tomu, abyste si u ní vytvořili nějaký účet. Tento krok však není povinný, a proto se přesuňte do spodní části stránky, kde klepněte na odkaz **No thanks, just start my download**. Poté by se již mělo zahájit stahování vámi zvoleného balíčku.
5. Přejděte do složky, do níž jste si uložili stažený soubor ve formátu MSI, a poklepaním na něj spusťte průvodce instalací.
6. Na obrazovce License Agreement zaškrtněte volbu, jíž potvrdíte souhlas s licenčními podmínkami, a poté klepněte na tlačítko **Next**.
7. Dostanete se tak na obrazovku Choose Setup Type, na níž zvolte **Server Only**. Následně klepnutím na tlačítko **Execute** zahajete samotnou instalaci. V závislosti na nastavených politikách zabezpečení se může stát, že se vám zobrazí dialog User Account Control.
8. Po dokončení instalace klepněte na tlačítko **Next**, čímž přejdete ke konfiguraci MySQL.

9. Prvním krokem je nastavení typu konfigurace. Doporučujeme vám vybrat volbu **Development Machine**, neboť díky ní by MySQL server měl využívat pouze minimální množství systémových prostředků.
10. Pokud se týká ostatních voleb tohoto dialogu (použití portu 3306 protokolu TCP/IP), ponechte je ve výchozím nastavení a pokračujte klepnutím na tlačítko **Next**.
11. V dalším kroku budete požádáni o zadání hesla uživatele root. Zadejte tedy heslo a ve spodní části dialogu opět klepněte na tlačítko **Next**.
12. Volby, které se vám zobrazí nyní, se týkají spouštění serveru MySQL jako služby systému Windows. Ve většině případů asi budete chtít, aby server MySQL byl nakonfigurován jako služba systému Windows a aby se tato služba spouštěla spolu s operačním systémem. Ve spodní části dialogu pak můžete nastavit účet, pod nímž bude tato služba spouštěna. Opět lze říci, že ve většině případů bude vyhovovat výchozí nastavení, tj. spouštění služby pod identitou uživatele SYSTEM. Pokračujte klepnutím na tlačítko **Next**.
13. Dostáváte se k poslednímu kroku, jímž je provedení vlastní konfigurace. Nejprve se vám zobrazí seznam všech potřebných úkonů. Celý proces spusťte klepnutím na tlačítko **Execute**.
14. Jsou-li všechny úkony dokončeny úspěšně, u každého z nich se zobrazí zelená ikona zaškrtnutí a ve spodní části dialogu se zpřístupní tlačítko **Finish**. Klepnutím na něj se vrátíte do průvodce instalací, v němž rovnou klepněte na tlačítko **Next**.
15. Dostanete se tak na poslední obrazovku průvodce instalací, která vám nabídne pouze možnost zkopírování protokolu celé instalace do schránky. Celý proces ukončete klepnutím na tlačítko **Finish**.

Nyní podle následujícího postupu přidejte podsložku `bin` té složky, do níž byl instalován server MySQL, do systémové proměnné `PATH`.

1. Otevřete si okno **System Properties**.
  - a. Stisknutím klávesové kombinace **Win+C** si zobrazte uživatelské rozhraní Edge.
  - b. Klepněte na charm **Search**, zadejte Control Panel, a jakmile se ve výsledcích objeví ikona Control Panel, klepněte na ni.
  - c. Zobrazí-li se vám ovládací panely pomocí pohledu **Category**, klepněte na položku **System and Security** a následně na položku **System**. Jestliže se vám ovládací panely zobrazí v pohledu **Icon**, klepněte na ikonu **System**.
2. V levé části dialogu klepněte na odkaz **Advanced system settings**, jímž si otevřete okno **System Properties**.
3. Pokud se vám toto okno neotevřelo rovnou na záložce **Advanced**, přejděte na ni a v její spodní části pak klepněte na tlačítko **Environment Variables**.
4. V sekci **System variables** si vyberte proměnnou **Path** a klepněte na tlačítko **Edit**.
5. Na konec stávající hodnoty přidejte cestu ke složce `bin` (`C:\Program Files\MySQL\MySQL Server 5.7\bin`), přičemž tuto novou položku oddělte od hodnoty stávající středníkem.

## 6. Klepnutím na tlačítko **OK** postupně všechny dialogy uzavřete.

V tuto chvíli jste tedy schopni spouštět jednotlivé pomocné nástroje z okna příkazového řádku, aniž byste přitom museli zadávat úplnou cestu: složka `bin` je totiž součástí toho seznamu cest, v němž systém Windows vyhledává spustitelné soubory. A jelikož byl server MySQL již během instalace zaregistrován jako služba systému Windows, bude se spouštět automaticky během náběhu systému a budete jej moci řídit pomocí správce služeb systému Windows. V případě potřeby si však můžete otevřít okno příkazového řádku – pochopitelně s oprávněními administrátora – a server MySQL spustit, příp. zastavit, pomocí následujících příkazů:

- Spuštění serveru MySQL – `net start MySQL57`
- Zastavení serveru MySQL – `net stop MySQL57`

# Komunikace se serverem

Na první pohled se vám může zdát, že server MySQL vlastně jen nečinně běží a čeká na to, až obdrží nějaké dotazy. Jakmile server nějaký dotaz přijme, provede jej za nás a odešle zpět výsledek. Přitom existuje několik způsobů, jimiž můžeme se serverem MySQL komunikovat: například s ním můžeme komunikovat programově z prostředí aplikace, kterou jsme si sami napsali, anebo s ním můžeme komunikovat interaktivně pomocí vyhrazeného klientského programu. Ve zbývajících částech této knihy budeme ke spojení a komunikaci používat převážně klienta příkazového řádku, který je součástí standardní instalace serveru MySQL; výjimkou bude kapitola 5, v níž se budeme bavit o odesílání příkazů jazyka SQL z prostředí nějakého programu.

Otevřete si okno příkazového řádku (či okno terminálu) a zadejte do něj příkaz `mysql -u root -p`. Volba `-u` určuje uživatelský účet serveru MySQL, který bude použit pro připojení, zatímco díky volbě `-p` se zobrazí výzva pro zadání hesla tohoto uživatele. Jakmile vás o to program požádá, zadejte heslo uživatele `root`, které jste zadali v předcházejících částech této kapitoly.



### Poznámka: Mnoho voleb

Volby `-u` a `-p` jsou pouze dvěma z mnoha voleb, s nimiž je klientský program schopen pracovat. Níže najdete přehled některých dalších voleb, které možná začnete využívat poměrně často (chtěte-li si zobrazit úplný přehled všech voleb, spusťte klienta s volbou `-?`):

- `-A` – vypnutí automatického dokončování názvů
- `-B` – běh v dávkovém režimu (při zobrazování výsledků se jako oddělovač využívá znak tabulátoru)
- `-e` *příkaz* – spuštění zadaného příkazu jazyka SQL
- `-h` *název\_hostitele* – zadání hostitelského názvu vzdáleného databázového serveru
- `-N` – potlačení zobrazování názvů sloupců ve výstupu

- `-p` – zobrazení výzvy pro zadání hesla toho uživatelského účtu, který má být použit k připojení
- `-u jméno_uživatele` – umožňuje zadání jména toho uživatele, který má být použit k připojení
- `-?` – zobrazení seznamu všech podporovaných voleb

Jakmile se úspěšně připojíte k serveru MySQL, zobrazí se v okně příkazového řádku výzva `mysql>`. A právě z této výzvy budeme zadávat všechny příkazy jazyka SQL. Po zadání nějakého příkazu klient zobrazí nejen samotnou odpověď serveru, ale i informace o době, kterou ke zpracování dotazu server potřeboval, a o veškerých chybách, na něž během provádění dotazu narazil.

Server MySQL je samozřejmě schopen udržovat najednou více databází. Chcete-li zjistit, které databáze daný server spravuje, zadejte na řádku výzvy příkaz `SHOW DATABASES;`. Výsledkem takového dotazu bude seznam všech databází aktuálně spravovaných daným serverem. Připojíte-li se k nově nainstalované instanci, zobrazí se vám pouze čtyři databáze využívané samotným serverem MySQL: `information_schema`, `mysql`, `performance_schema` a `sys`. V některých případech se můžete setkat i s databází `test`, vznikající po spuštění nástroje `mysql_install_db` a určené pro testování.

Chcete-li vytvořit novou databázi, použijte příkaz `CREATE DATABASE`. Například je-li vaším cílem vytvoření databáze s názvem „`mysql_okamzite`“, zadejte na řádku výzvy příkaz `CREATE DATABASE mysql_okamzite;`. Poté znovu proveďte příkaz `SHOW DATABASES;` a ve výstupu již uvidíte svoji první databázi.

Potřebujete-li serveru říci, že chcete pracovat s určitou databází, použijte příkaz `USE`. Zadáte-li na řádku výzvy příkaz `USE mysql_okamzite;`, budou všechny následně zadané příkazy prováděny v databázi `mysql_okamzite`. Cílovou databázi je možné zadat i ve chvíli, kdy se k serveru připojujete pomocí klienta příkazového řádku. Typickou ukázkou by mohl být příkaz `mysql -u root -p mysql_okamzite`.

Příkaz `SHOW TABLES` serveru MySQL říká, že nám má zobrazit seznam všech tabulek právě aktivní databáze (tj. té, k níž jsme se připojili pomocí příkazu `USE`). Je zřejmé, že jelikož jsme v databázi `mysql_okamzite` nevytvořili žádné tabulky, bude odpověď na příkaz `SHOW TABLES` poměrně jednoduchá: „Empty set“. Vytvoření správné tabulky však vyžaduje poněkud delší plánování a přípravu a navíc jsme již probrali poměrně dost nových témat. Z těchto důvodů se tematice vytváření nových tabulek budeme věnovat až v následující kapitole.

Chcete-li ukončit připojení k serveru MySQL pomocí klienta příkazového řádku, jednoduše buď zadejte příkaz `exit`, anebo stiskněte klávesovou kombinaci **Ctrl+D**.



# Účty a zabezpečení MySQL

Posledním tématem, jemuž bychom se měli v této kapitole věnovat, jsou uživatelské účty serveru MySQL. Byť uživatel root serveru MySQL není totožný se stejnojmenným uživatelem operačního systému vašeho počítače, rozhodně není určen k běžnému užívání. Uživatel root serveru MySQL by měl být používán pouze k účelům správy serveru, například při vytváření dalších uživatelských účtů, při nastavování oprávnění či při čištění mezipaměti přístupu. Pro každodenní práci by rozhodně měly být využívány účty s nižšími oprávněními.

Chcete-li vytvořit nový uživatelský účet, pomocí klienta příkazového řádku se připojte k serveru MySQL, a to jako uživatel root. Poté proveďte následující příkaz CREATE USER:

```
CREATE USER 'cpress'@'localhost' IDENTIFIED BY 'tajne';
```

Tento příkaz vytvoří nový uživatelský účet s názvem „cpress“ a heslem „tajne“, umožňující uživateli ověření z tétohož systému, na němž běží server MySQL. Namísto localhost můžete použít názvy jiných počítačů či jejich IP adresy, čímž umožníte připojení z rozdílných systémů a sítí. Mějte však na mysli, že server MySQL považuje každou kombinaci jméno uživatele / název hostitelského počítače za samostatný účet. To ovšem znamená, že cpress@localhost a cpress@192.168.1.100 jsou ve skutečnosti považovány za dva nezávislé a samostatné účty, z nichž každý má přiřazenou svoji vlastní sadu oprávnění.



## Poznámka: Zástupné znaky

Znaky \_ a % jsou zástupnými znaky, které můžete použít v části výše uvedeného příkazu pro identifikaci hostitelského počítače, a to za účelem zajištění částečných shod. Například tedy můžete použít definice typu „192.168.1.10\_“ anebo „%.example.com“. Přitom znak \_ nahrazuje jeden libovolný znak, zatímco znak % nahrazuje libovolný počet znaků. Z toho vyplývá, že následující příkaz můžete využít k vytvoření účtu, který je schopen se ověřit odkudkoliv, z kteréhokoliv hostitelského systému – což je sice velmi pohodlná, avšak na druhou stranu velice nebezpečná praxe:

```
CREATE USER 'cpress'@'%' IDENTIFIED BY 'tajne';
```

To, zda server MySQL umožní určitému uživateli provedení nějakého úkonu, závisí na oprávněních přiřazených jeho uživatelskému účtu. Nové účty jsou vytvářeny bez jakýchkoliv oprávnění, což znamená, že jim musíme explicitně přiřadit všechna oprávnění, která budou potřebovat. Uživatel „cpress“ bude potřebovat několik oprávnění, neboť jej budete využívat i v dalších částech této knihy. V tuto chvíli stačí, když tomuto účtu přiřadíme základní sadu oprávnění (další oprávnění pak můžete přiřadit ve chvíli, kdy začnou být nezbytná). Zadejte následující příkaz:

```
GRANT CREATE, DROP, ALTER, INSERT, UPDATE, SELECT, DELETE, INDEX, REFERENCES ON mysql_okamzite.* TO 'cpress'@'localhost';
```

Syntaxe příkazu GRANT serveru MySQL je natolik flexibilní, že rozsah oprávnění můžeme zúžit pouze na vybrané sloupce určité tabulky či na vybrané tabulky databáze. V našem

případě jsme však serveru MySQL řekli, aby uživateli „cpress“ přihlašujícímu se z počítače „localhost“ přiřadil veškerá oprávnění ke všem tabulkám databáze `mysql_okamzite` (což je dáno znakem \*). Přiřazená oprávnění jsou:

- CREATE – umožňuje uživateli vytváření databází a tabulek
- DROP – umožňuje uživateli výmaz celých tabulek a databází
- ALTER – umožňuje uživateli změnu definice existující tabulky
- INSERT – umožňuje uživateli přidávání nových záznamů do již existující tabulky
- UPDATE – umožňuje uživateli aktualizaci záznamů v tabulce
- SELECT – umožňuje uživateli načítání již existujících záznamů z tabulky
- DELETE – umožňuje uživateli výmaz existujících záznamů z tabulky
- INDEX – umožňuje uživateli vytváření či výmaz indexů

Přehled všech možných oprávnění a toho, co vlastně uživatelskému účtu umožňují, najdete v dokumentaci<sup>3</sup>. Jak jsme si řekli již výše, pokud v budoucnu zjistíte, že určitý účet vyžaduje nějaká nová oprávnění, stačí provést další příkaz GRANT. Obdobně platí, že oprávnění, která nejsou nadále potřebná, lze uživatelskému účtu odebrat; k těmto účelům je však nutné použít jiný příkaz, a to příkaz REVOKE, jehož syntaxe je velice podobná syntaxi příkazu GRANT:

```
REVOKE CREATE, DROP, ALTER, INDEX ON mysql_okamzite.* TO 'cpress'@'localhost';
```

Kdykoliv však provádíte nějakou změnu týkající se uživatelského účtu či jeho oprávnění, musíte vzápětí spustit příkaz FLUSH PRIVILEGES, jímž serveru MySQL řeknete, aby si znovu načel mezipaměť informací o účtech, kterou si udržuje. Jedině tak zajistíte to, že změny vstoupí v platnost okamžitě. Neucíníte-li tak, provedené změny v podstatě nebudou mít žádný vliv, a to až do doby následného restartu serveru MySQL:

```
FLUSH PRIVILEGES;
```

Po provedení tohoto příkazu ukončete připojení k serveru MySQL z klienta příkazového řádku a poté se znovu připojte pomocí účtu „cpress“. Pokud jste zadali všechny výše uvedené příkazy správně a pokud po zobrazení výzvy zadáte i správné heslo, znovu se vám zobrazí výzva `mysql>`.

## Shrnutí

V této kapitole jsme probrali spoustu témat. Naučili jste se instalovat MySQL na různé platformy, dozvěděli jste se, jak se můžete připojit k serveru MySQL pomocí klienta příkazového řádku, seznámili jste se s postupem vytváření nových databází a nakonec jste alespoň ve stručnosti viděli, jak se v prostředí serveru MySQL spravují uživatelé.

<sup>3</sup> <http://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html>

---

Ačkoliv nyní možná chcete ihned pokračovat studiem další kapitoly, já osobně vám doporučuji, abyste si nejdříve alespoň stručně pročetli online manuál serveru MySQL – přinejmenším ty části, v nichž jsou popsána dosud probraná témata. Seznamte se s podrobnostmi týkajícími se příkazů `CREATE USER` a `GRANT`. Zjistěte si, jak se dá změnit heslo nějakého existujícího uživatelského účtu a jak se dá vymazat účet, který není nadále potřebný. Zamyslete se nad tím, jaká oprávnění byste přiřadili účtu používanému pro ukládání a načítání dat v rámci nějaké webové aplikace.

V kapitole 2 se budeme věnovat tématům souvisejícím s ukládáním dat v databázi. Uvidíte, jak se vytváří nová tabulka a jak se do ní vkládají nové řádky. Dozvíte se také, jaké typy dat lze do tabulek ukládat, co se míní pojmem databázové úložiště (storage engine) a jak naše volba databázového úložiště ovlivňuje způsob, jímž server MySQL spravuje naše data.



# Ukládání dat

## V této kapitole:

- Vytváření tabulek
- Přidávání dat
- Shrnutí

Data uložená v relační databázi jsou uspořádána do **tabulek**. Podíváme-li se pak na takovou tabulku, uvidíme, že data jsou v ní uložena do jakési mřížky, v níž každý **řádek** tvoří jeden záznam a každý **sloupec** identifikuje určitou hodnotu záznamu. Pro představu jsme níže připojili tabulku ukazující počet medailí, které získala každá z pěti medailově nejúspěšnějších zemí účastnících se zimních olympijských her v roce 2014. V každém řádku vidíte název země a dále počet zlatých, stříbrných a bronzových medailí, které vyhráli sportovci reprezentující danou zemi. V posledním sloupci pak vidíte celkový počet medailí.

Země	Zlato	Stříbro	Bronz	Celkem
Rusko	13	11	9	33
USA	9	7	12	28
Norsko	11	5	10	26
Kanada	10	10	5	25
Holandsko	8	7	9	24

Tabulka, kterou jsme si ukázali, je „fyzická“, a to proto, že ji můžete vidět vytištěnou v knize či nakreslenou na tabuli. Taková tabulka je omezená pouze dostupným fyzickým prostorem.

Na druhou stranu databázová tabulka je nehmotnou strukturou uloženou kdesi na pevném disku či v paměti počítače. Můžeme si ji tedy pouze představit anebo si ji zkusit nakreslit. Databázová tabulka je interpretována počítačovým procesem (například serverem MySQL), přičemž omezení tohoto procesu definují i omezení týkající se tabulky. Počet sloupců, počet řádků, a dokonce i typy hodnot, které lze ukládat do jednotlivých sloupců to vše závisí na tom, co je schopen zpracovat nejenom použitý databázový server, ale i daný počítač. Avšak bez ohledu na tato omezení platí, že databázová tabulka je ve skutečnosti velice flexibilní. Můžeme vytvářet relace (tj. vazby) mezi jednotlivými tabulkami, můžeme kombinovat více tabulek dohromady, můžeme třídit záznamy a zobrazovat si pouze vybrané záznamy, můžeme odstraňovat záznamy a v neposlední řadě můžeme provádět i nejrůznější výpočty s daty uloženými v jednotlivých záznamech.

Toto je pouze náhled elektronické knihy. Zakoupení její plné verze je možné v elektronickém obchodě společnosti eReading.