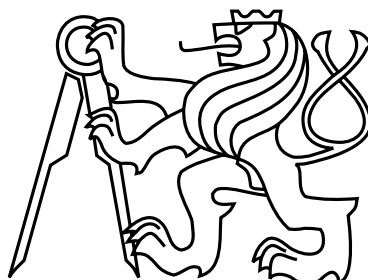


České vysoké učení technické v Praze  
Fakulta elektrotechnická



Diplomová práce

## **Klient pro správu databází MySQL**

*Zbyněk Munzar*

Vedoucí práce: Ing. Michal Valenta, Ph.D.

Studijní program: Elektrotechnika a informatika strukturovaný magisterský

Obor: Informatika a výpočetní technika

leden 2008



## Poděkování

Děkuji svým rodičům, kteří mi umožnili studia a své ženě, která mi pomohla a pomáhá na cestě k dokončení studií. Dále děkuji kočičce Micce za jméno pro aplikaci a svému ročnímu synovi, že spí přes den alespoň dvě hodiny, kdy se mohu věnovat školní práci.



## **Prohlášení**

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona c. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 18.1.2008

.....



## Abstrakt

Cílem diplomové práce je vyvinout rychlý a efektivní nástroj pro správu MySQL databáze. Hlavním účelem aplikace je zaplnit trhlinu v dostupnosti kvalitního freewarového databázového klienta pro práci s daty.

Při vytváření programu je kladen důraz na intuitivní a praktické uživatelské rozhraní, rychlost operací a podporu různých znakových sad. Dalším důležitým kritériem je snadná rozšiřitelnost o nové funkce a možnosti související s neustálým vývojem MySQL.

Tato práce zachycuje koncepci programu a jeho okamžitý stav, který je však potřeba nadále a neustále vyvíjet.

## Abstract

The aim of the dissertation is to develop fast and effective tool for the MySQL database administration. The main object of the application is to fill a gap in the field of the availability of quality freeware client for processing the data.

The program was created with the stress on intuitive and practical user interface, the rate of operations and the support of different character sets. Another important criterion is the possibility of easy enhancement by new functions and options connected with continuous development of MySQL.

The work captures the philosophy of the program and its present state, but there is a necessity of its further and continuous development.



# Obsah

<b>Seznam obrázku</b>	<b>xi</b>
<b>Seznam tabulek</b>	<b>xiii</b>
<b>1 Úvod</b>	<b>1</b>
1.1 Cíle diplomové práce . . . . .	1
1.2 Struktura práce . . . . .	1
<b>2 MySQL</b>	<b>2</b>
2.1 Specifikace . . . . .	2
2.2 Vlastnosti . . . . .	2
2.3 Architektura ukládání dat . . . . .	3
2.4 Získání . . . . .	4
2.5 Instalace . . . . .	5
2.6 Jiné databázové systémy . . . . .	6
<b>3 Správa databáze</b>	<b>7</b>
3.1 Administrátor . . . . .	7
3.2 Datový specialista . . . . .	8
3.3 Vývojář . . . . .	9
3.4 Nástroje . . . . .	9
<b>4 Přehled dostupných nástrojů</b>	<b>10</b>
4.1 Webové . . . . .	10
4.1.1 phpMyAdmin . . . . .	10
4.1.2 phpMinAdmin . . . . .	11
4.2 Zdarma . . . . .	11
4.2.1 HeidiSQL . . . . .	12
4.2.2 MySQL GUI Tools by MySQL AB . . . . .	12
4.3 Komerční . . . . .	13
4.3.1 SQL-Front . . . . .	13
4.3.2 EMS SQL Manager for MySQL . . . . .	14
4.3.3 Navicat . . . . .	15
4.4 Micka . . . . .	15
4.5 Shrnutí použitelnosti nástrojů . . . . .	17
<b>5 Implementace</b>	<b>19</b>
5.1 Motivace . . . . .	19
5.2 Vývojové prostředí a programovací jazyk . . . . .	19
5.3 Popis implementace . . . . .	20
5.3.1 Architektura . . . . .	20
5.3.2 Uživatelské rozhraní . . . . .	21
5.4 Omezení . . . . .	22
5.5 Rozšiřování . . . . .	22
5.5.1 Další databázové systémy . . . . .	22
5.5.2 Nové funkce . . . . .	24
<b>6 Testování</b>	<b>25</b>
6.1 Nasazení v praxi . . . . .	25

6.2	SourceForge.net . . . . .	26
<b>7</b>	<b>Závěr</b>	<b>27</b>
7.1	Vyhodnocení použitelnosti navržené aplikace . . . . .	27
7.2	Náměty pro další práci . . . . .	27
7.3	Možnosti rozvoje prostředí . . . . .	27
<b>8</b>	<b>Seznam literatury</b>	<b>29</b>
<b>A</b>	<b>Instalační a uživatelská příručka</b>	<b>31</b>
A.1	Vlastnosti Micky . . . . .	31
A.2	Distribuce . . . . .	31
A.3	Instalace . . . . .	31
A.3.1	Systémové požadavky . . . . .	31
A.3.2	Postup instalace . . . . .	32
A.4	Připojení k MySQL serveru . . . . .	32
A.4.1	Připojení a odpojení . . . . .	32
A.4.2	Nastavení připojení . . . . .	32
A.4.3	Nastavení znakové sady . . . . .	33
A.4.4	Zobrazení objektů databáze . . . . .	33
A.4.5	SQL log a stavový řádek . . . . .	34
A.4.6	Paměť poslední relace . . . . .	34
A.5	Práce s databází . . . . .	35
A.6	Práce s tabulkou . . . . .	35
A.6.1	Struktura a vlastnosti tabulky . . . . .	35
A.6.1.1	Vytvoření nové tabulky . . . . .	36
A.6.1.2	Úprava struktury tabulky . . . . .	36
A.6.1.3	Úprava tabulky . . . . .	38
A.6.2	Data . . . . .	38
A.6.2.1	Editace hodnot . . . . .	38
A.6.2.2	Označování a kopírování hodnot . . . . .	38
A.6.2.3	Operace s řádky . . . . .	39
A.6.2.4	Zobrazení . . . . .	39
A.6.2.5	Navigace . . . . .	40
A.7	Export a import . . . . .	40
A.7.0.6	Export . . . . .	40
A.7.0.7	Import . . . . .	42
A.8	SQL okno . . . . .	42
A.9	Okno procesů . . . . .	43
A.10	Nastavení . . . . .	43
A.11	Aktualizace . . . . .	43
A.12	Seznam klávesových zkratk . . . . .	43
<b>B</b>	<b>Deklarace objektu TSenMySQL</b>	<b>45</b>
<b>C</b>	<b>Obsah přiloženého CD</b>	<b>47</b>

## Seznam obrázků

2.1	Architektura MySQL[11]	4
3.1	Datový model	8
3.2	Databáze	9
4.1	phpMyAdmin náhled	11
4.2	phpMinAdmin náhled	12
4.3	HeidiSQL náhled	13
4.4	MySQL GUI Tools náhled	14
4.5	SQL-Front náhled	15
4.6	EMS SQL Manager for MySQL náhled	16
4.7	Navicat náhled	16
4.8	Micka náhled	17
5.1	Delphi	20
5.2	Vrstvy komunikace	21
5.3	GUI - Hlavní okno	22
5.4	GUI - SQL okno	23
5.5	Rozšíření na další systémy komunikace	23
6.1	Webový formulář pro hlášení chyb	25
6.2	Micka na SourceForge.net	26
A.1	Obsah distribuce	31
A.2	Volba připojení	32
A.3	Dialog připojení	32
A.4	Micka po připojení a označení databáze	33
A.5	SQL log a stavový řádek	34
A.6	Dialog pro vytvoření nebo úpravu databáze	35
A.7	Kontextové menu stromu objektů	36
A.8	Dialog pro vytvoření nové tabulky	36
A.9	Struktura tabulky	37
A.10	Úprava tabulky	38
A.11	Nastavení limitů a filtr	39
A.12	Menu sloupečku	40
A.13	Dialog pro export	41
A.14	SQL okno	42
A.15	Okno procesů	43



## Seznam tabulek

2.1	Standardně podporované operační systémy a platformy pro MySQL 5.1[13]	. . .	5
4.1	Srovnání nástrojů, hodnocení 1 až 5 (1 - výborně, 5 - nedostatečně)	. . . . .	17



# 1 Úvod

Počítače dnes slouží k mnoha účelům, ale jeden z nejdůležitějších a také nejčastějších je uchovávání a zpracování dat. Díky počítačovým sítím a zejména Internetu se mohou informace, tedy data, téměř okamžitě šířit po celém světě.

Čím více dat, tím větší potřeba mít je někde přehledně a bezpečně uložená a kdykoli se k nim moci dostat. Lidé chtějí mít přístup k datům odkudkoli na světě, chtějí rychle vyhledávat, chtějí mít opravdu velké množství dat a přitom se o ně nemusí příliš starat. Na základě těchto požadavků vzniká databázový systém.

Tato práce se týká databázového systému MySQL, ale není primárně o MySQL. Předmětem zájmu je klient, který se k databázovému systému připojuje. Jeho prostřednictvím se dá spravovat celá databáze včetně dat.

## 1.1 Cíle diplomové práce

Cílem této diplomové práce je navrhnout a implementovat kvalitního a funkčního klienta pro správu databází MySQL. Na trhu není široký výběr z takovýchto nástrojů, které by byly dostupné zdarma. Nástroj bude obsahovat všechny běžné funkce pro správu relační databáze<sup>1</sup>. Vznikne desktopová aplikace jako forma tlustého klienta. Proběhne také diskuse nad výběrem implementační platformy. Aplikace se zaměří především na uživatelskou přívětivost a na efektivní práci s daty a na rychlou navigaci ve větších databázích i tabulkách. Samozřejmě s podporou dostupných znakových sad pro texty.

Aplikaci autor pojmenuje **Micka** na počest jeho kočky.

## 1.2 Struktura práce

**Databázový systém** Práce se bude věnovat nejdříve stručnému popisu databázového systému MySQL z pohledu vývojáře databázové aplikace, to znamená cílové skupiny Micky, kapitola 2.

**Správa databáze** Dále se podíváme na správu databáze, kdo a proč pracuje s databází, kapitola 3.

**Nástroje** Následuje přehled nástrojů pro práci s MySQL, jejich popis, shrnutí funkčnosti a srovnání. Na závěr kapitoly 4 je doporučení, který nástroj je pro koho vhodný.

**Implementace** Těžištěm práce je implementace. Kapitola 5 popisuje právě implementaci, dále vývojové prostředí, architekturu a uživatelské rozhraní. Součástí je úvaha nad omezeními a dalším směřováním.

**Testování** Testování realizované aplikace najdete v kapitole 6.

**Závěr** Shrnutí výsledků diplomové práce je v závěru, kapitola 7.

**Přílohy** Na konci jsou přílohy. Nejzajímavější je příloha A, instalační a uživatelská příručka, která poskytuje vyčerpávající popis celé Micky z pohledu uživatele.

---

<sup>1</sup>Termín databáze je v tomto textu používán ve dvou smyslech. Za prvé jako zkrácený výraz pro databázový systém a za druhé pro označení konkrétní databáze, jako objektu s tabulkami. Konkrétní význam je zřejmý z kontextu.

## 2 MySQL

MySQL je databázový systém vyvíjený švédskou firmou MySQL AB. Začal vznikat v osmdesátých letech minulého století a na jeho počátku stáli tři tvůrci David Axmark, Allan Larsson a Michael "Monty" Widenius. Tak jako u mnoha podobných projektů byla hlavní motivací vzniku potřeba kvalitního a použitelného systému, který nebyl k dispozici, nebo byl k dispozici jen jako komerční za nepřijatelnou cenu. MySQL je zdokonalením původního systému mSQL, za použití původního API<sup>1</sup>. Jméno MySQL je po dceři jednoho ze zakladatelů, My.

Zajímavý je systém dvojího licencování. Je k dispozici buď zdarma včetně zdrojových kódů pod licenci GPL[3] nebo pod komerční licenci, která má v ceně nástroje pro monitorování, optimalizaci a samozřejmě plnou technickou podporu včetně aktualizací.

MySQL je multiplatformní databáze, kterou lze nainstalovat na mnoha různých operačních systémech (Linux, MS Windows, BSD, ...). Díky tomu a také díky volnému šíření má dnes tento databázový systém značný podíl na trhu. Kombinace MySQL, PHP<sup>2</sup> a Apache<sup>3</sup> je velmi oblíbená při nasazování na webové servery.

### 2.1 Specifikace

MySQL je založeno na relačním modelu databáze. Data jsou uspořádána do jednotlivých tabulek (relací) a nad nimi jsou definované přípustné operace. Tabulky mají jednotlivé sloupce (pole, vlastnosti) a řádky. Každý řádek představuje jeden záznam, například v tabulce zákazníků je jeden řádek jeden zákazník. Záznamy jsou identifikovány zpravidla takzvaným klíčem. Může to být speciální sloupec id nebo jakýkoli jiný sloupec nebo kombinace několika sloupců. Většinou je žádoucí, aby každý záznam byl jedinečně identifikovatelný.

Abychom mohli záznamy přidávat, upravovat, prohlížet a mazat, přidávat tabulky a dělat všechny ostatní operace nad databází, potřebujeme komunikovat. MySQL pro komunikaci mezi klientem a databázovým serverem používá, jak sám název napovídá, jazyk SQL (Structured Query Language). SQL je standardizovaný jazyk pro přístup do databáze a je definován ANSI/ISO SQL normou. SQL standard byl vyvinut v roce 1986 a od té doby vyšlo několik verzí: SQL-92, SQL:1999 a aktuální verze SQL:2003. MySQL v dnešní době používá upravenou verzi SQL:2003.

### 2.2 Vlastnosti

Od začátku je MySQL optimalizováno především na rychlost. Umí rychle a spolehlivě pracovat i s rozsáhlými objemy dat. Až donedávna mu však chyběly některé poměrně podstatné vlastnosti, například vnořené dotazy, pohledy, triggerů nebo uložené procedury. Ty byly dodány až v posledních letech především pod tlakem tvůrců webových aplikací, kdy s narůstající složitostí a rozsahem narůstají i požadavky na použité nástroje. Ještě dnes se dá poměrně často narazit na MySQL 3.2 nebo 4.0, které umí v podstatě jen úplně nejzákladnější věci typu select a create table a jeho zásadním nedostatkem je naprosto nevyřešená práce s různými znakovými sadami. Migrace na verzi alespoň 4.1, je tedy poměrně náročná a pro textová data s háčky a čárkami i nebezpečná.

Nejdůležitější podporované vlastnosti:

---

<sup>1</sup>API (Application Programming Interface), rozhraní pro programování aplikací. Je to sbírka funkcí nebo tříd, které může používat programátor při použití dané knihovny.

<sup>2</sup>Definice PHP k kapitole 4.1.1 na straně 10

<sup>3</sup>Apache HTTP Server je softwarový webový server s otevřeným kódem pro Linux, BSD, MS Windows a další platformy. V současné době dodává prohlížečům na celém světě většinu internetových stránek.[17]

- cizí klíče (od verze 3.23 podporovány v tabulkách typu InnoDB (druh uložště, viz dále))
- transakce (od verze 3.23 podporovány v tabulkách typu InnoDB)
- podpora různých znakových sad a časových pásem v datech (od verze 4.1)
- vnořené dotazy (od verze 4.1)
- uložené procedury (od verze 5.0)
- triggery (od verze 5.0)
- pohledy (od verze 5.0)

### 2.3 Architektura ukládání dat

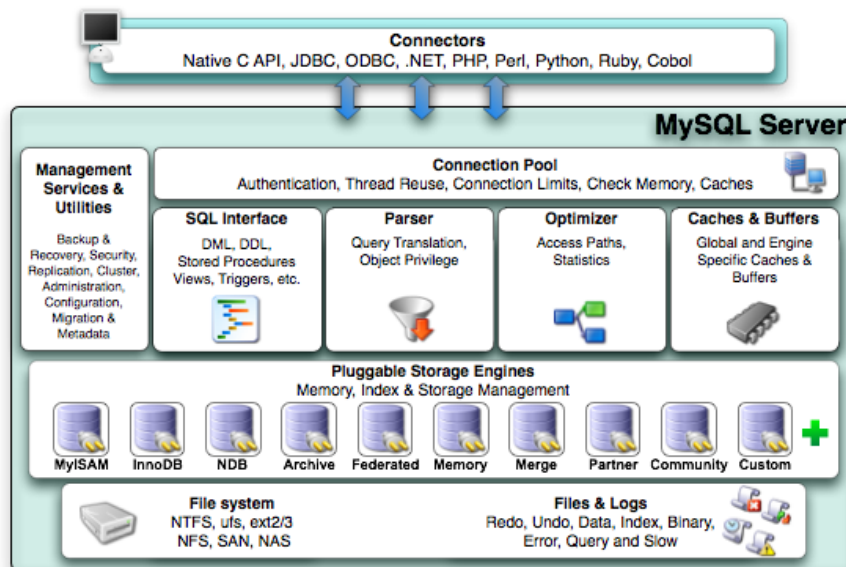
MySQL nabízí několik způsobů ukládání databázových tabulek (storage engine, uložště). Při vytváření tabulky se dá způsob volitelně zadat. Pokud se nezadá, použije se implicitně uložště MyISAM.

```
CREATE TABLE test(i INT, c CHAR(10)) ENGINE = BLACKHOLE;
```

Různá uložště se liší účelem použití, tím co podporují a umí, a způsobem samotného ukládání tabulek.

Přehled uložšť:

- MyISAM - nejpoužívanější, implicitní, bez podpory transakcí. Každá tabulka se ukládá na disk do tří souborů (formát tabulky, samotná data, index).
- InnoDB - podporuje transakce, všechny tabulky se ukládají do jednoho souboru. Umí efektivně pracovat s velkými objemy dat, ale z hlediska výkonnosti se nedoporučuje mít na stejném disku jiný databázový stroj.
- Falcon - podporuje transakce, navržen speciálně pro 32 nebo 64 bitové systémy s velkou pamětí a pro vícejádrové procesory nebo vícevláknové prostředí. (od verze 6.0)
- MERGE - jedná se v podstatě o kolekce identických MyISAM tabulek. Identické znamená, že musí mít stejné sloupečky a indexy. Sloučené tabulky nemusí být ve stejné databázi. MERGE tabulky podporují operace SELECT, DELETE, UPDATE, a INSERT. Pro takovou tabulku se ukládá soubor s definicí a soubor se seznamem sloučených tabulek.
- MEMORY (HEAP) - data se ukládají pouze do operační paměti, jen definice tabulek se ukládá na disk. Při ukončení serveru se data ztratí. Je vhodný pro extrémně rychlé dočasné tabulky také díky hashovaným indexům.
- FEDERATED - umožňuje lokálně zpřístupnit data ze vzdálené MySQL databáze bez replikace. Dotazy na lokálním serveru budou automaticky spouštěny na vzdálených tabulkách. Žádná data nejsou ukládána lokálně.
- ARCHIVE - komprimované tabulky, bez podpory indexů
- CSV - ukládání dat v prostých textových souborech
- NDBCLUSTER - uložště pro clusterované databáze. (od verze 5.0)



Obrázek 2.1: Architektura MySQL[11]

- BLACKHOLE - uložiště, kde se dá vytvořit tabulka, dá se do ní vkládat, ale nic se neuloží. Dotaz vždy vrátí prázdný výsledek.

Podrobné specifikace jsou uvedeny přímo v manuálu k MySQL[11]. Podrobný seznam uložišť včetně krátkého popisu a podpory v dané distribuci se získá dotazem

```
SHOW ENGINES;
```

## 2.4 Získání

Kompletní instalaci MySQL serveru lze stáhnout přímo z webových stránek věnovaných MySQL na adrese <http://dev.mysql.com/downloads/>. Open source varianta pod licencí GPL[3] se jmenuje MySQL Community Server. K dispozici je vždy několik posledních verzí, v době psaní této práce jsou ke stažení verze 4.1, 5.0, 5.1, 6.0. Při volbě verze je třeba vždy myslet na to, kterým klientem a kterou aplikací se budeme k serveru připojovat. Pokud máme například webovou aplikaci a používáme v ní pohledy, potřebujeme verzi 5.0 a novější, protože starší verze je nepodporují. Většina syntaxe je zpětně kompatibilní, ale přesto se v nové verzi občas nějaký dotaz přestane používat a i na to je potřeba při instalaci nové verze myslet. Naštěstí má MySQL velice precizně napsanou dokumentaci ke každé nové verzi a vždy je zde uveden seznam změn a zpětně nepodporovaných konstrukcí včetně případných alternativ. Dokumentace je dostupná v několika formátech ke stažení i online verze na <http://dev.mysql.com/doc/>. U každé verze jsou dostupné varianty pro různé operační systémy a platformy. Tabulka 2.1 ukazuje přehled standardně podporovaných operačních systémů a platform pro MySQL verzi 5.1, která je dnes nejpoužívanější pro nově instalovaný server.

Dostupný je také samotný zdrojový kód. MySQL je napsáno v jazyce C a C++, takže jej lze teoreticky zkompilovat i pro jakoukoli jinou platformu, která má vhodný překladač. Případně si i každý může upravit kód podle vlastních potřeb a zkompilovat právě pro sebe. Například firma Google používá pro své aplikace upravenou MySQL databázi.

Windows  
 Windows x64  
 Linux (non RPM packages)  
 Linux (non RPM, Intel C/C++ compiled, glibc-2.3)  
 Red Hat Enterprise Linux 3 RPM (x86)  
 Red Hat Enterprise Linux 3 RPM (Intel IA64)  
 Red Hat Enterprise Linux 4 RPM (x86)  
 Red Hat Enterprise Linux 4 RPM (AMD64 / Intel EM64T)  
 Red Hat Enterprise Linux 4 RPM (Intel IA64)  
 Red Hat Enterprise Linux 5 RPM (x86)  
 Red Hat Enterprise Linux 5 RPM (AMD64 / Intel EM64T)  
 SuSE Linux Enterprise Server 9 RPM (AMD64 / Intel EM64T)  
 SuSE Linux Enterprise Server 9 RPM (Intel IA64)  
 SuSE Linux Enterprise Server 10 RPM (x86)  
 SuSE Linux Enterprise Server 10 RPM (AMD64 / Intel EM64T)  
 Linux x86 generic RPM (dynamically linked)  
 Linux AMD64 / Intel EM64T generic RPM  
 Linux Intel IA64 generic RPM  
 Solaris (pkgadd packages)  
 Solaris (TAR packages)  
 FreeBSD (TAR packages)  
 Mac OS X (TAR packages)  
 HP-UX (depot packages)  
 HP-UX (TAR packages)  
 IBM AIX  
 Novell NetWare

Tabulka 2.1: Standardně podporované operační systémy a platformy pro MySQL 5.1[13]

## 2.5 Instalace

Protože se celá tato práce zabývá implementací databázového klienta v systému MS Windows, bude zde jen stručně popsána instalace právě v tomto operačním systému. Zvláště pro výukové účely a pro testování. Na reálném serveru běžícím na jakémkoli operačním systému by se instalaci měl věnovat zkušený administrátor.

### Popis instalace [6]

Stáhněte si patřičnou verzi instalátoru (viz diskuse výše) a spusťte ji. Zvolte uživatelskou instalaci. Po instalaci se spustí MySQL konfigurátor. Zvolte si zde konfiguraci pro vývojářskou stanici, výchozí znakovou sadu UTF8 a nastavte nějaké heslo pro uživatele root.

Pokud vám konfigurátor MySQL instance nahlásí chybu v právech, ukončete jej a spusťte ručně znovu MySQLInstanceConfig.exe. Napodruhé to vždy funguje, napoprvé to někdy hlásí chyby.

Pokud již máte starší MySQL server nainstalovaný, nejprve jej zastavte, odinstalujte starou verzi, následně smažte soubory z `mysql\data\` (jen soubory v tomto adresáři, ne podadresáře, to jsou vlastní databáze) a proveďte instalaci dle popisu výše.

Potřebujete-li MySQL databázi pro provozování vlastního lokálního webového serveru pro vývoj webových stránek v PHP, což je nejčastější důvod instalace na vlastní osobní počítač, lze si ušetřit spoustu práce s instalací a konfigurací celé kombinace Apache, PHP

a MySQL použitím kompletního instalačního balíčku. Existuje například projekt VertrigoServ <<http://vertrigo.sourceforge.net/>>, který obsahuje kompletní instalaci Apache, PHP, MySQL, SQLite, PhpMyAdmin, ZendOptimizer a SQLiteManager.

## 2.6 Jiné databázové systémy

Existuje i mnoho jiných databázových systémů než MySQL. Jak komerčních tak dostupných zdarma.

Z komerčních jsou nejznámější Oracle Database <<http://www.oracle.com>> a MS SQL <<http://www.microsoft.com/sql/>> od Microsoftu. Obě velké společnosti nabízejí volně ke stažení takzvanou express verzi, kterou je možno nainstalovat a provozovat zcela zdarma, ovšem ne ke komerčním účelům. Je to taková ochutnávka toho nejlepšího, co ve světě databází existuje.

Z open source databázových systémů je nejznámější a nejpoužívanější právě MySQL. Další, podle mnoha názorů vyspělejší, je PostgreSQL <<http://www.postgresql.org>>, který se pomalu ale jistě v oblíbenosti dotahuje na MySQL.

Velice zajímavý je open source databázový systém SQLite[17] <<http://www.sqlite.org>>, Na rozdíl od databází založených na principu klient-server (všechny ostatní zde zmíněné), kde je databázový server spuštěn jako samostatný proces, je SQLite pouze malá knihovna, která se přilinkuje k aplikaci a pomocí jednoduchého rozhraní ji lze využívat.

Podrobnější popis nebo rozsáhlejší přehled by byl jistě zajímavý, ale přesahuje rámec tohoto dokumentu. Pro získání více informací je možné navštívit uvedené internetové stránky projektů nebo články zabývající se databázemi na Wikipedii <<http://cs.wikipedia.org/wiki/Databze>>.

## 3 Správa databáze

Tato kapitola pojednává, kdo a proč potřebuje spravovat databázi. Je pojatá z pohledu uživatele a i když pro přehlednost zjednodušená, měla by pravdivě odrážet situaci, která dnes panuje zejména ve firemní sféře.

### 3.1 Administrátor

Administrátor přistupuje k databázi, respektive databázovému systému na té nejnižší úrovni. Zabývá se především péčí o programové vybavení, o samotný MySQL server. Stará se o přístup k serveru ze strany jeho uživatelů.

Základní povinnosti administrátora by šly shrnout do několika bodů[1]:

**Spouštět, vypínat a restartovat server** Administrátor musí mít přístup k souborovému systému, musí vědět jak server spustit, jak zařídit aby startoval spolu s operačním systémem. Když server nenaběhne nebo hlásí chyby, jak odhalit závadu, opravit jí a dokončit spuštění.

**Správa uživatelských účtů** Jak účty upravovat a přidávat. Administrátor by měl mít přehled a starat se, kdo, do které databáze a odkud se připojuje. Hlídat bezpečnost databáze, aby každý viděl jen tam, kam vidět má. Měl by také umět poradit novým uživatelům, jak správně nastavit klienta pro bezchybné připojení do správné databáze. Musí umět obnovit zapomenuté heslo. Odstraňovat takzvané mrtvé účty, které už nikdo nepoužívá.

**Údržba protokolů (logů)** Je dobré vědět, které informace jsou užitečné a které jen zabírají místo. Jak dlouho je udržovat, jaký mít cyklus rotace logů.

**Zálohování a kopírování databází** Zálohování je životně důležitá operace. V případě havárie serveru nebo chybné manipulace s daty ze strany uživatele, je třeba mít možnost obnovit data s co možná nejmenšími ztrátami. Zpravidla se dělají denní zálohy a pak speciální zálohy před potenciálně nebezpečnými operacemi na serveru, jako je například přehodnocení komponent nebo celého serveru. Ze zálohy by měl být administrátor schopen obnovit jak celou databázi tak i jen část jedné tabulky.

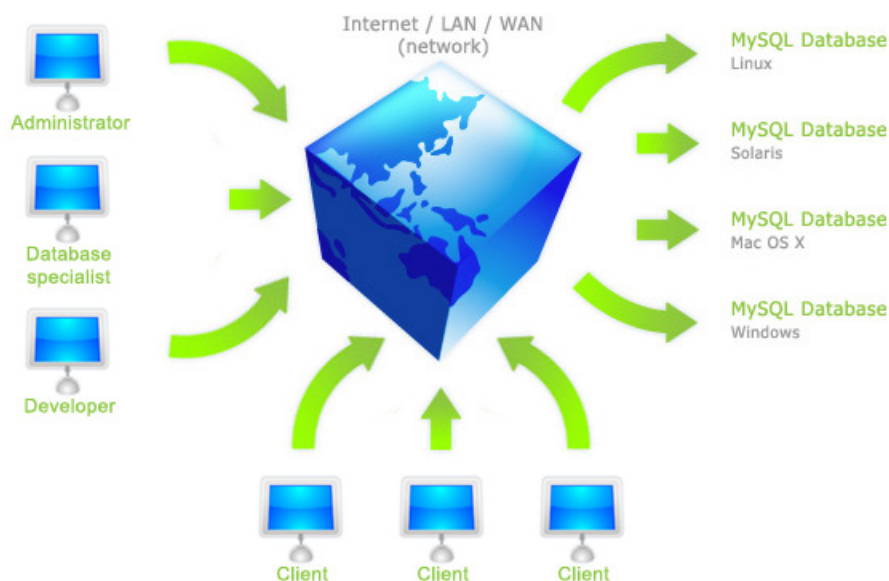
Podobná operace je kopírování databáze. Tu by měl administrátor zvládnout například při spouštění nového výkonnějšího serveru.

**Databázové repliky** Záloha nebo kopie databáze je její obraz v daný okamžik. Další možností jsou repliky. Znamená to připravit dva servery, které spolupracují tak, že změny v databázích, které spravuje jeden server, se kontinuálně promítají do odpovídajících databází, které spravuje druhý server.

**Konfigurace a ladění serveru** Uživatelé mají určité požadavky, jak by se měl server chovat. Nemusí to být jen výkon, ale třeba podporované znakové sady nebo nastavení některých proměnných prostředí. Administrátor musí umět konfigurovat jakékoli nastavení a vyladit potřeby uživatelů, výkon a bezpečnost.

**Práce s několika servery** Může nastat potřeba provozovat několik serverů současně s různou verzí MySQL nebo různým nastavením zabezpečení, i to by měl administrátor zvládnout.





Obrázek 3.2: Databáze

Musí umět data i přijímat v různých formách.

### 3.3 Vývojář

Vývojářem je zde myšlen programátor, který při vývoji produktů přijde na různé úrovni do styku s databází. Může to být tvůrce nebo správce internetových stránek, který se dnes již bez znalosti databází neobejde. Nebo vývojář firemní aplikace, která na intranetu komunikuje přes databázi. V dnešní době mají databáze a data klíčový význam nejen ve světě počítačů. Vývojář je většinou tak trochu administrátor i tak trochu datař a ještě něco navíc. Je to univerzální uživatel databáze.

Měl by si umět nainstalovat a alespoň na základní úrovni spravovat MySQL server u sebe na stanici, aby mohl testovat, na čem právě pracuje.

Musí si prohlížet nebo přímo vytvářet strukturu tabulek a databáze. Musí umět pracovat s daty v tabulkách. Prohlížet, vkládat, exportovat, vybírat, importovat, zálohovat.

Navíc často pracuje se samotným API MySQL. Používá různá rozhraní různých jazyků pro komunikaci se serverem. Občas je nucen si sám napsat třídu pro zpracování výstupu nebo vstupu z MySQL. Zkrátka potřebuje komplexní pohled i znalosti databázových systémů.

Bohužel dost často, obzvlášť ve webových aplikacích, vznikají díla, která spíš zatěžují server, síť a zkušenějšího kolegu, než že by něco efektivního prováděla. A je to zpravidla důsledkem neznalosti základních principů databázových systémů a absence komplexního pohledu na věc.

### 3.4 Nástroje

Nabídka dostupných nástrojů na správu databáze je uvedena v následující kapitole 4. Úvaha nad vhodností jednotlivých nástrojů a doporučení použitelnosti pro uvedené skupiny uživatelů je v kapitole 4.5 na straně 17.

## 4 Přehled dostupných nástrojů

Standardní instalace MySQL serveru obsahuje textovou konzoli, kam lze zadávat dotazy a do konzole se textově vypíše výsledek dotazu. Největší výhodou tohoto nástroje je jeho přímá dostupnost v každé instalaci serveru. Není třeba jej shánět, stahovat a samostatně instalovat. Ale možnosti konzole jako správce databáze jsou velmi omezené. Pro zadávání jednoduchých dotazů, pokud uživatel dobře zná syntaxi jazyka, nebo pro případ největší nouze je použitelná. Ovšem pro správu většího množství tabulek, složitější struktury nebo prohlížení a editaci dat je naprosto nevhodná. Pro tyto účely existuje mnoho vhodnějších produktů.

Tato práce se zabývá implementací desktopové aplikace v prostředí 32 bitového Microsoft Windows, takže srovnání dostupných konkurentů se až na několik výjimek zaměří taktéž na desktopové řešení. MS Windows je drtivě nejpoužívanější operační systém a tomu odpovídá i dostupnost nástrojů, která je nepoměrně lepší než například pro Linux. V dnešní době se rozvíjí trend multiplatformnosti, nicméně jediný zástupce napsaný v Javě, který by měl být teoreticky spustitelný na libovolné platformě s nainstalovaným JRE<sup>1</sup>, se nepodařilo zprovoznit ani na několika různých počítačích. Úplně jinou skupinu tvoří webové aplikace, které jsou nezávislé na klientském operačním systému. Zdánlivě do této diplomové práce nepatří, ale jejich využitelnost je natolik vysoká, že dva nejzajímavější zástupci zde budou zmíněni.

### 4.1 Webové

První zajímavá skupina jsou webové aplikace. Běží na serveru, často na stejném jako samotná databáze, a ovládají se odkudkoli ze světa pomocí okna webového prohlížeče. Tento přístup má své velké výhody, ale také některé nevýhody. Zmíněny zde budou dva rozdílné, velmi zajímavé projekty, oba dostupné zdarma.

#### 4.1.1 phpMyAdmin

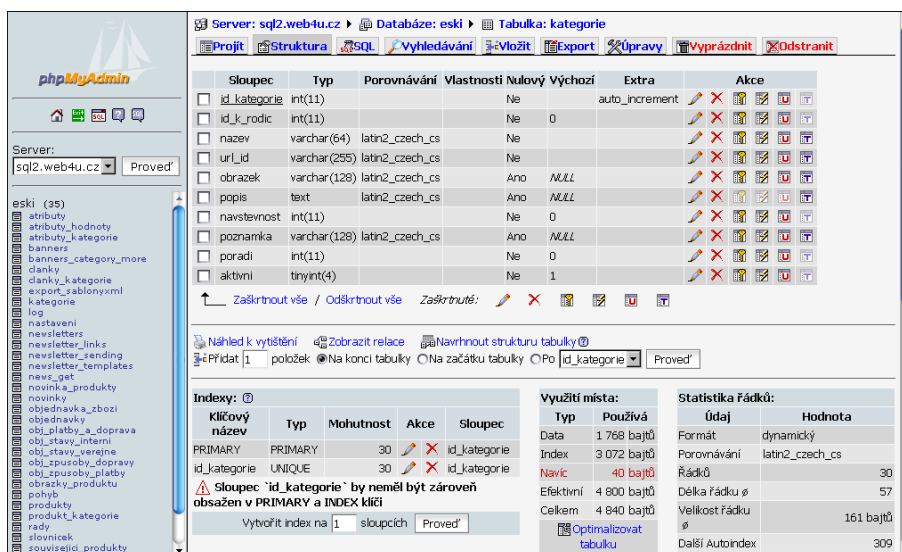
phpMyAdmin ( <<http://www.phpmyadmin.net>>, verze 2.10.1) je robustní webová aplikace napsaná v jazyce PHP<sup>2</sup> a využívá všech moderních dostupných technologií. Jak typ napovídá, ovládá se pomocí okna internetového prohlížeče. Na vývoji tohoto nástroje se podílí poměrně velká skupina lidí z celého světa a používá ji většina uživatelů MySQL. A právě proto pružně zachycuje nové vlastnosti každé vydané verze MySQL a přizpůsobuje se požadavkům uživatelů. Dá se dokonce říci, že je to takový etalon funkčnosti MySQL klienta. Výborně se hodí ke správě struktury databáze. Téměř každý webhosting nabízející MySQL databázi jej dává k dispozici ke vzdálené správě. Jediné zásadní nevýhody pramení z jeho povahy webové aplikace. Pro práci s obsahy tabulek, vyhledávání, filtrování, sestavování složitějších dotazů je z principu výrazně méně efektivní a pohodlný než desktopová aplikace. Na druhou stranu těží některé dobré vlastnosti z toho, že zpravidla běží na stejném stroji nebo alespoň na stejné lokální síti jako samotný databázový stroj. Díky tomu nemá konkurenci například v rychlosti importu dat, kdy konkurenty zdržuje posílání mnoha dotazů za sebou ke vzdálenému počítači.

- + Kvalitní nástroj zadarmo
- + Pravidelný vývoj

---

<sup>1</sup>Java Runtime Environment, také JRE, nebo J2RE je software vytvořený firmou Sun Microsystems, který umožní operačnímu systému spouštět Java aplikace, více viz [16].

<sup>2</sup>PHP (rekurzivní zkratka PHP: Hypertext Preprocessor, původně Personal Home Page) je skriptovací programovací jazyk, určený především pro programování dynamických internetových stránek. Nejčastěji se začleňuje přímo do struktury jazyka HTML, XHTML či WML, což je velmi výhodné pro tvorbu webových aplikací. PHP lze ovšem také použít i k tvorbě konzolových a desktopových aplikací.[17]



Obrázek 4.1: phpMyAdmin náhled

### 4.1.2 phpMinAdmin

phpMinAdmin (<http://phpminadmin.sourceforge.net>, verze 1.4.0) je minimalistická webová aplikace českých autorů tvořená právě jedním php skriptem. Není tak propracovaná jako phpMyAdmin, ale její kouzlo spočívá v tom, že pokud máme FTP<sup>3</sup> přístup na vzdálený server, vzdálenou databázi, která má zakázaný přístup z venku a chceme s ní cokoli udělat, stačí nakopírovat jediný soubor a můžeme svou databázi spravovat. Je to extrémně jednoduché a elegantní řešení v situaci, do které se každý webmaster<sup>4</sup> čas od času dostane. Zajímavý je způsob exportu, kdy ukládá několik řádků do jednoho dotazu insert, tím se minimalizuje jednak velikost exportovaného textového souboru, ale především počet dotazů při následujícím importu.

Tato aplikace trochu překvapivě umí velmi příjemně pracovat i s uloženými procedurami, triggerry a pohledy. Přes svou nenápadnost je velice použitelná. Slabou stránkou je úprava a prohlížení dat, ale to logicky vyplývá z podstaty webové aplikace.

- + Celá aplikace v jednom souboru
- – Méně funkcí

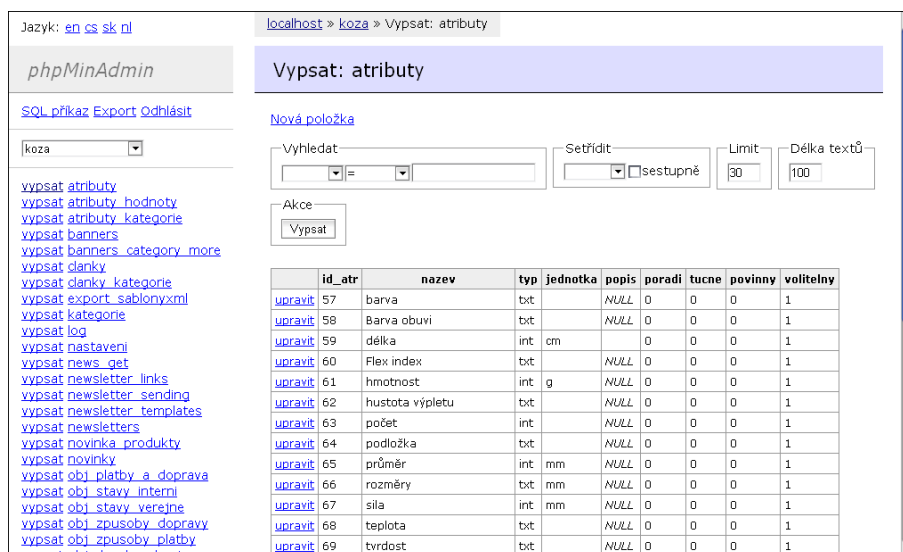
## 4.2 Zdarma

Další skupinou jsou desktopové nástroje pro správu databáze MySQL. V této kapitole budou zmíněny ty, které jsou dostupné zdarma. Každý si je může stáhnout z Internetu a bez jakéhokoli omezení používat. Často jsou k nim dostupné i zdrojové kódy.

Grafické rozvržení je u všech následujících programů, včetně komerčních, podobné. Po levé straně je strom, kde uzly tvoří databáze a jejich potomky objekty databází, především tabulky, u vyspělejších nástrojů také pohledy, triggerry nebo uložené procedury. V hlavním

<sup>3</sup>FTP (File Transfer Protocol) je protokol určený pro přenos souborů mezi počítači na TCP/IP síti, na kterých mohou běžet libovolné rozdílné operační systémy.

<sup>4</sup>V této práci rozumíme webmasterem správce internetových stránek (webu), zpravidla programátor ovládající nějaký skriptovací jazyk, značkovací jazyk formátování webových stránek a alespoň povrchně databáze. Dnes poměrně častá a žádaná profese.



Obrázek 4.2: phpMinAdmin náhled

pravém okně se pak zobrazují podrobná data související s objektem ve stromě. V případě tabulky například její struktura a v další záložce data. Každý klient dále obsahuje textové okno, kam lze zkopírovat nebo přímo napsat libovolný SQL dotaz a rovnou vidíme výsledek. Mezi standardní funkce patří editace struktury databáze a tabulky, práce s daty a export a import dat. Různé nástroje pak mají další množství funkcí viz dále. Výhodou je SQL log, kam se vypisují všechny dotazy, jimiž aplikace komunikuje s databázovým strojem.

#### 4.2.1 HeidiSQL

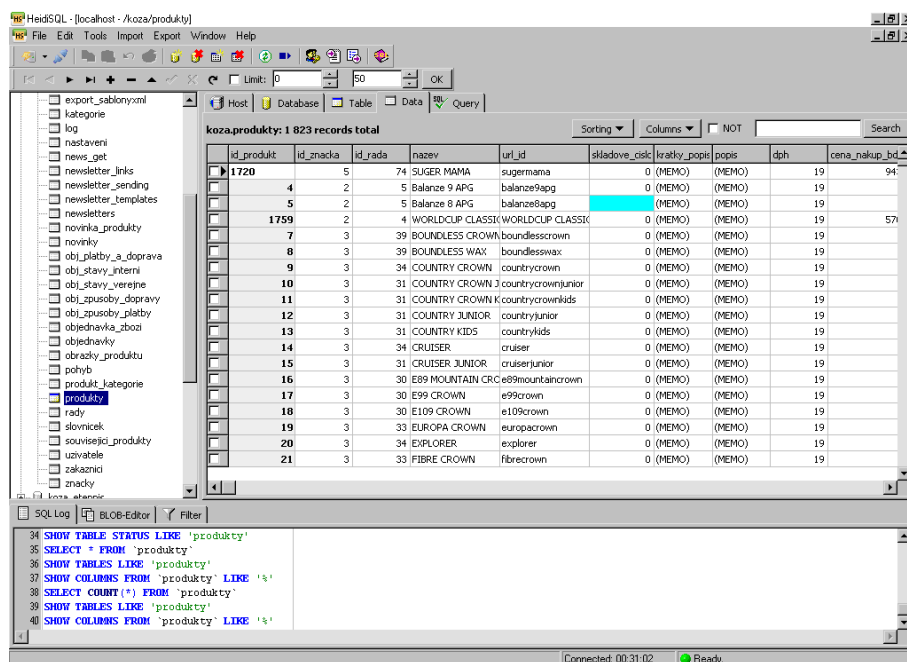
HeidiSQL (<<http://www.heidisql.com>>, verze 3.2) je desktopový klient. Jedná se o projekt, který vyvíjí několik nadšenců a dává jej volně k dispozici včetně zdrojových kódů. Obsahuje všechny zmíněné standardní funkce. Za zmínku určitě stojí funkce synchronizace tabulek mezi databázemi. Nabízí mnoho možností, pokud potřebujeme nějaký exotičtější export do SQL. Lze volit verzi výstupního jazyka a podrobně formu exportu dat, jako insert, update nebo replace. Na druhou stranu je výsledný export zbytečně robustní a nepodporuje více řádků tabulky do jednoho dotazu. Přestože se jedná již o třetí stabilní verzi, některé funkce nejsou úplně dotažené tak, aby šlo program každodenně pohodlně používat k práci, zejména pokud jde o prohlížení a editaci dat v tabulkách. Pro občasnou úpravu v databázi nebo pravidelné zálohování dat se vzhledem ke své jednoduchosti a dostupnosti hodí výborně.

- + Jednoduchý, ale použitelný program zdarma
- – Některé nedotažené detaily

#### 4.2.2 MySQL GUI Tools by MySQL AB

Sada nástrojů MySQL GUI Tools (<<http://dev.mysql.com/downloads/gui-tools>>) pro MySQL 5.0 vydaná přímo tvůrcem databáze MySQL AB. Slouží primárně ke správě, zálohování a monitorování uživatelů, databází, tabulek, pohledů nebo uložených procedur, nikoli však k práci se samotnými daty. Je to precizní nástroj nezbytný pro systémového správce, ovšem nepoužitelný například pro datové oddělení. Je dodáván vždy k dané verzi MySQL, to znamená buď pod GPL[3] nebo komerční licenci.

- + Profesionální program pro systémovou správu celé databáze



Obrázek 4.3: HeidiSQL náhled

- – Nevhodný pro práci s daty

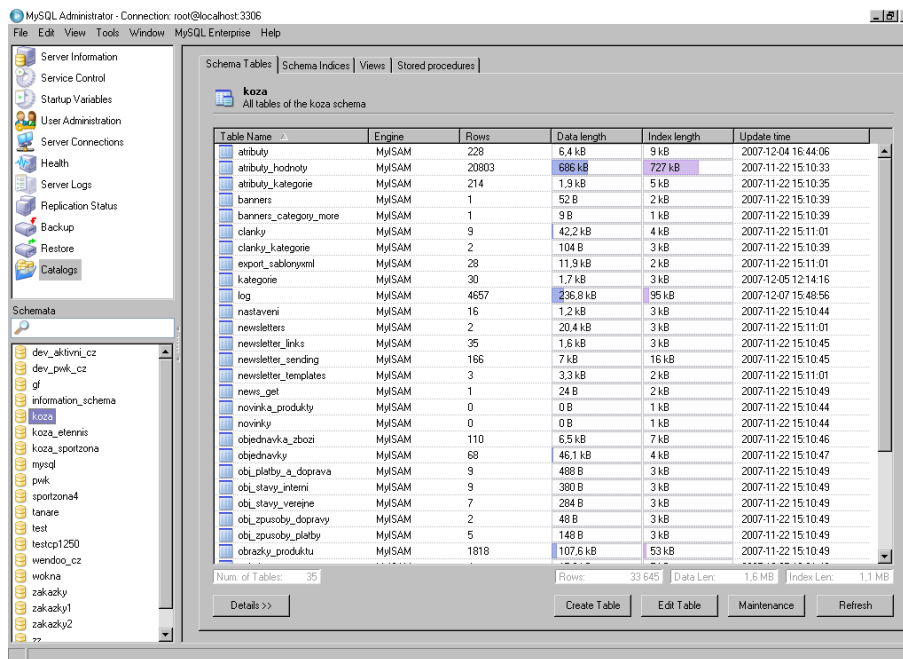
Existuje samozřejmě mnoho dalších nástrojů dostupných zdarma, ale ty jsou zpravidla prakticky nepoužitelné. Nereflektují vývoj MySQL, nelze je nainstalovat nebo obsahují příliš mnoho chyb.

### 4.3 Komerční

Nabídek komerčních aplikací pro práci s MySQL databázemi pro systém Windows je mnoho. Zde jsou vybrané tři, které jsou velmi používané, kvalitní a jsou dostupné k vyzkoušení ve své plné funkčnosti.

#### 4.3.1 SQL-Front

SQL-Front (<http://www.sql-front.com>, verze 4.0) lze považovat do značné míry za vzor freewarových kolegů. Aplikace je poměrně jednoduchá s velice intuitivním ovládáním, obsahuje všechny potřebné funkce pro kvalitní správu struktury i dat. Nemá žádné zvláštní funkce navíc, ale ty pro běžnou práci ani nejsou třeba. Cena je příznivých 35 €, to znamená asi 1000 Kč za jednu licenci včetně všech budoucích aktualizací. Pro více licencí se cena snižuje. Ideální volba pro firmu, která potřebuje pracovat s databázemi na střední úrovni. Velice nedávno vyšla čtvrtá verze, která přejímá všechny příjemné vlastnosti předchozí vyladěné verze a navíc umí pracovat s triggerly, uloženými procedurami a pohledy, i když zatím ne příliš pohodlně. Jako nevýhoda se může v určitých situacích jevit načítání celé struktury databáze a tabulek při připojování k serveru. Pokud máme vzdálené připojení s mnoha databázemi a s mnoha tabulkami, doba čekání je nepříjemně dlouhá. Ve většině případů to však není problém. Velice efektivní vlastnost je možnost přímo z kontextového menu databáze nebo tabulky otevřít nové okno nebo záložku. Zvláště systém záložek, na který jsou uživatelé zvyklí například z moderních webových prohlížečů, zlepšuje práci s více tabulkami najednou. Obvykle je to řešeno méně přehlednými dalšími okny nebo podokny. Za zmínku ještě stojí SQL log, kam se vypisují



Obrázek 4.4: MySQL GUI Tools náhled

všechny dotazy, kterými aplikace komunikuje. Uživatel tak má přehled, co se přesně děje. Jen pro zajímavost je také v češtině.

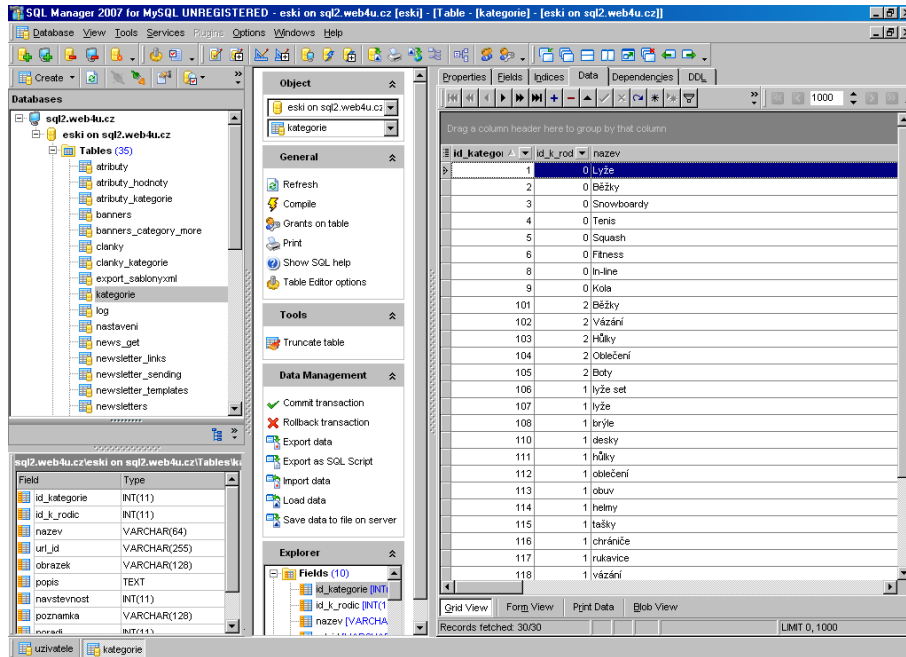
- + Jednoduché a intuitivní ovládání
- + Nízká cena
- – Žádné speciální funkce, pro běžnou práci ale naprosto dostačující

### 4.3.2 EMS SQL Manager for MySQL

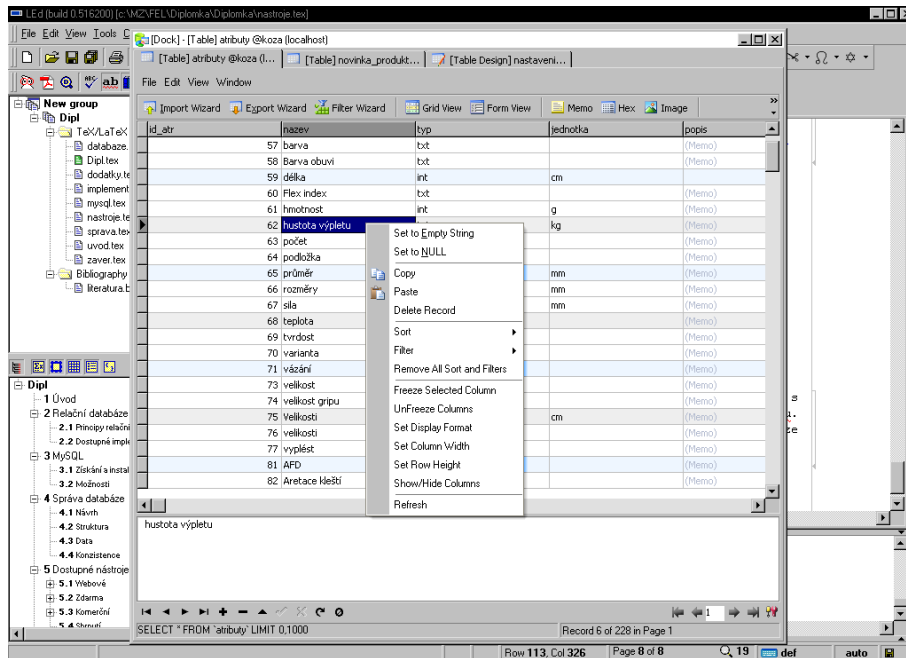
EMS SQL Manager for MySQL (<http://www.sqlmanager.net/en/products/mysql/manager>), verze 2007) je opravdu špičkový nástroj pro práci s MySQL databázemi, umí toho jednoznačně nejvíce ze všech zmínovaných nástrojů. Reflektuje všechny nové možnosti každé nové verze MySQL, takže má propracovanou podporu uložených procedur, triggerů, pohledů. Nabízí nepřehledné možnosti zobrazení a editace struktury i dat, exporty, průvodce pro filtrování, synchronizaci, vyspělý SQL editor a mnoho dalšího, podrobná specifikace je na webu produktu. Nemá SQL log, což je určitě škoda. Umí toho tolik, že se stává pro neškoleného uživatele velice nepřehledným. Kdo ho chce používat, musí určitě věnovat mnoho času zaučení a experimentování. Nicméně kdo to myslí opravdu vážně s MySQL databází, je to ideální nástroj. Bohužel rozsahu odpovídá i cena. Jedna licence s podporou na rok za nejlevnější a nejjednodušší business variantu stojí 175 \$, to je přes 3000 Kč a kompletní studio s podporou na 3 roky už stojí 655 \$.

Zvláštní funkce, která je poměrně jedinečná a stojí za podrobnější zmínku, je takzvaný tunneling. Všechny uvedené desktopové aplikace přistupují k serveru přímo. Některé servery však mají přístup z venku kvůli bezpečnosti zakázaný, a tedy i tyto nástroje jsou pro vzdálenou administraci nepoužitelné. EMS SQL Manager obsahuje variantu připojení tunnelingem, kdy nepoužívá přímé připojení, ale administrátor nakopíruje speciální php nebo asp skript na server, a program pak komunikuje přes port 80 a http protokol, které jsou téměř vždy povolené.

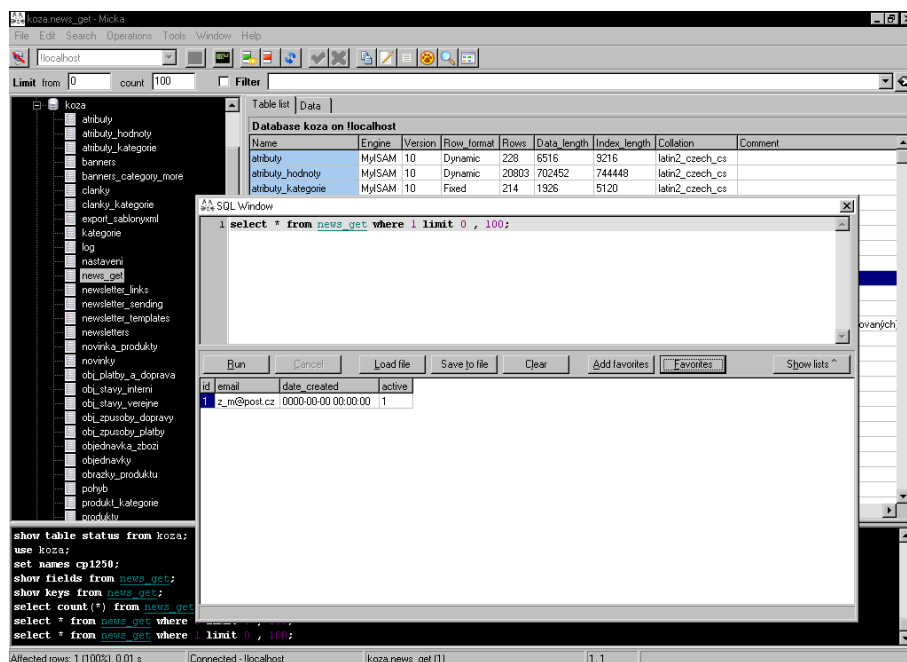




Obrázek 4.6: EMS SQL Manager for MySQL náhled



Obrázek 4.7: Navicat náhled



Obrázek 4.8: Micka náhled

- + Rychlost
- + Propracované ovládání
- – Absence vyspělejších funkcí

#### 4.5 Shrnutí použitelnosti nástrojů

Následující tabulka subjektivně hodnotí jednotlivé nástroje podle různých kritérií. Každé kritérium je hodnoceno známkou 1 až 5 jako ve škole, to znamená, 1 je nejlepší, 5 nejhorší. Dostupnost vyjadřuje v zásadě cenu produktu. Všechny produkty se dají v určité verzi stáhnout z Internetu, některé zcela zdarma (známka 1), některé chtějí po 30 dnech nemalý obnos peněz (známka 3). Platformy vystihují, pro jaké různé operační systémy je daný nástroj dostupný. Jedničku dostaly jen webové aplikace, které mohou běžet všude tam, kde je webový prohlížeč, dvojka je za několik určitých operačních systémů a trojka za jediný podporovaný OS.

Nástroj	Dostupnost	Funkčnost	Přehlednost	Platformy	Tlustý klient
phpMyAdmin	1	2	1	1	NE
phpMinAdmin	1	3	1	1	NE
HeidiSQL	1	3	1	3	ANO
GUI Tools	1	2	1	1	ANO
SQL-Front	2	2	1	3	ANO
EMS	3	1	3	3	ANO
Navicat	3	1	1	2	ANO
Micka	1	3	1	3	ANO

Tabulka 4.1: Srovnání nástrojů, hodnocení 1 až 5 (1 - výborně, 5 - nedostatečně)

Při výběru aplikace, kterou bude používat možná několik následujících let, by si měl potenciální uživatel především uvědomit, k čemu ji bude potřebovat a kolik peněz je za ni ochoten

utratit nebo kolik peněz je ochotna dát jeho firma za kolik licencí.

V kapitole 3 na straně 7 jsou definovány tři základní typy uživatelů. Teď se na ně podíváme z pohledu nástrojů, které by nejlépe využili.

**Administrátor** potřebuje přímý a neomezený přístup k databázovému serveru, potřebuje spravovat uživatele a jejich práva, potřebuje zálohovat, případně obnovovat ze zálohy. Musí mít přístup k logu, ke konfiguraci MySQL k procesům, které se systémem komunikují. Musí mít práva k restartování serveru. Občas může potřebovat vytvořit, nebo naopak odstranit jednotlivou databázi. Zpravidla nepotřebuje pracovat se samotnými daty nebo strukturou tabulek. Maximálně si nějakou tabulku prohlédnout. K takovým účelům se asi nejlépe hodí přímo program **MySQL GUI Tools** od tvůrce databázového systému firmy MySQL AB. Všechny výše zmíněné operace zvládne bezpečně a přehledně. Dokonce lze říci, že je přímo pro administrátory vyvinut. V placené verzi umí i vyspělé monitorování a optimalizaci databáze. Další nástroje, které by mohl administrátor použít, jsou EMS a Navicat. Ty toho sice umí zbytečně moc, což může být i potenciálně nebezpečné, ale splňují všechny požadavky. Zde je však velice aktuální hledisko financování.

**Datař** také uvítá spíše desktopovou aplikaci než webovou. Moc ho nezajímá databázový server jako takový, ale především struktura databází, tabulek a samotná data. Potřebuje dělat náročné a složité operace nad daty. Výběry z více tabulek, filtrování, exporty, importy. Často ručně píše sekvence dlouhých dotazů. Potřebuje přehledný, efektivní nástroj, který umí perfektně pracovat s daty a má vyspělou textovou konzoli<sup>5</sup>. Pro tyto účely se jistě hodí EMS a Navicat, už není třeba se rozepisovat o tom, že jsou výborné, ale drahé. Použitelný je jistě **SQL-Front a Micka**, která je i zadarmo. HeidiSQL nemá příliš vyladěnou konzoli, ani práci s daty, takže spíš s přimhouřením oka.

**Vývojář** je nejuniverzálnější uživatel, ten potřebuje občas všechno. Ten nejvíce ocení a nejvíce se mu vyplatí investovat do takových programů, jako je EMS nebo Navicat. Pro určitý styl práce zajisté ocení i vymoženosti při práci s Mickou. Pokud není moc náročný a s databází nepracuje denně, rozhodně stačí i HeidiSQL. Vývojář často rád využije i možnosti webových aplikací, které jsou v některých případech přímo nenahraditelné. Shrnout by to šlo tak, že pro vývojáře jsou vhodné všechny výše zmíněné nástroje, snad kromě MySQL GUI Tools, který je příliš zaměřen na administraci samotného systému.

Ještě malý pohled z hlediska investic a bezpečnosti. Pro vyspělou a bezpečnou administraci databáze na úrovni databázového specialisty se vyplatí koupit kvalitní a precizně otestovaný nástroj. Nicméně uživatel, který používá databázi jako jeden z mnoha nástrojů v projektech, což je dnes například každý webmaster, se pohodlně spokojí i s kvalitní aplikací zdarma.

---

<sup>5</sup>Máme na mysli textové okno pro přímé zadávání dotazů a se zpracováním výsledků

## 5 Implementace

Program Micka začal vznikat na začátku roku 2003, kdy byla zásadní absence kvalitní databázové desktopové aplikace dostupné zdarma, kterou by šlo především rychle a efektivně prohlížet, upravovat a přidávat data. Existoval phpMyAdmin, který je zmiňován v předchozí kapitole, ideální nástroj pro vytvoření tabulek, správu struktury celé databáze. V té době ještě MySQL neumělo triggerů nebo uložené procedury, a stačilo tedy řešit pouze sloupcečky v tabulkách a klíče. Jakmile chtěl ale uživatel začít například zadávat data stylem jako do Excelu nebo vybírat podle různých kritérií z větší tabulky, narazil na zásadní omezení webové aplikace, zvláště když nebylo běžné rychlejší připojení k Internetu. Byl zapotřebí nějaký nástroj, který dokáže s minimální režií přehledně a rychle zobrazovat libovolná data a umožní je i bezpečně upravovat a přidávat. Nezbyvalo, než si takový nástroj naprogramovat.

### 5.1 Motivace

Hlavním požadavkem na program Micka byla maximální uživatelská přívětivost a co nejpříjemnější a nejefektivnější práce se samotnými textovými daty. První verze neuměly pracovat se strukturou tabulek, neuměly vytvářet nebo rušit celé databáze, ale přesto byly velice použitelné. Na samotnou správu databáze existuje mnoho jiných nástrojů a je zbytečné duplikovat všechny funkce.

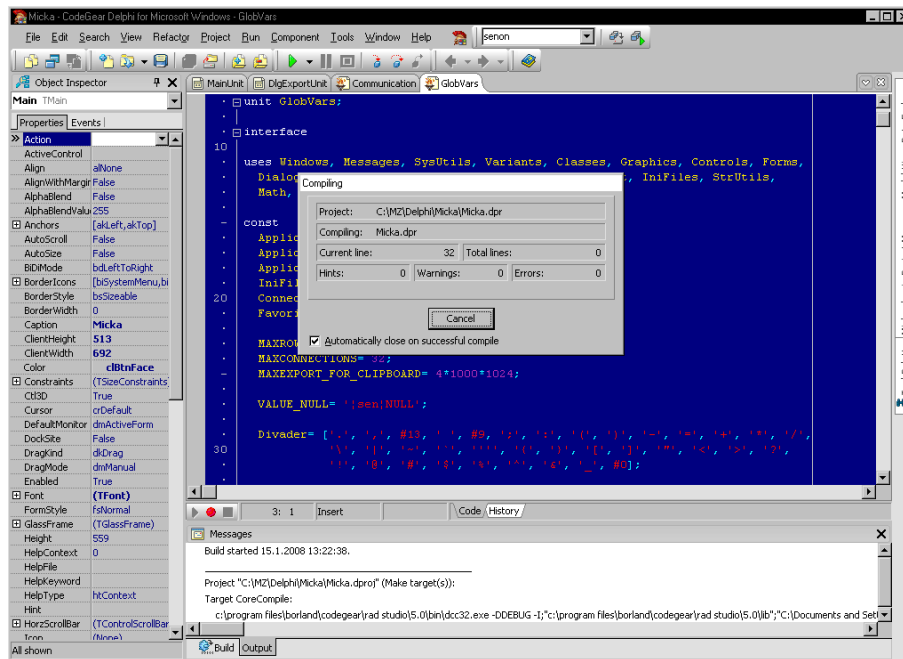
Od počátku byla Micka zaměřená na práci s daty. Po připojení k serveru si program zjistí jen minimum nezbytných informací, jako je znaková sada připojení a seznam databází. Některé jiné aplikace zjišťují seznamy a strukturu všech tabulek, hodnoty všech možných proměnných, což samotné připojení nesmírně zpomaluje. Je možné namítnout, že v dnešní době vysokorychlostního internetu na tom příliš nezáleží. Na druhou stranu je čím dál běžnější pracovat nikoli z firemní kanceláře na páteční síti, ale třeba z vlaku jedoucího do Ostravy, horské chalupy v Lužických horách nebo apartmánu v Chorvatsku. Zkrátka všude tam, kde ještě není pokrytí mobilními sítěmi třetí generace, a přesto se potřebujeme podívat, co se děje v našem internetovém obchodě nebo vyřešit problém chybného zobrazení názvu produktu váženého klienta.

Micka se snaží co nejvíce uživateli zpříjemnit práci, snaží se vychytat desítky detailů, které často právě freewareové projekty zanedbávají. Pamatuje si cesty ke všem souborům, které se ukládají nebo načítají, rozvržení, pozice a stav dialogových oken, abychom například při každém exportu nemuseli dokola zaklikávat nebo odklikávat pět různých políček. Je zde možnost zkopírovat jakýkoli název tabulky, sloupcečku, typu porovnání nebo obsahu buňky do schránky a následně vložit do skriptu, dokumentu nebo kamkoli bez nepříjemného přepisování. Jako jediný ze všech podobných programů, které byly testovány, umí exportovat cokoli pouze do schránky. Chcete-li zkopírovat strukturu tabulky nebo obsah menší tabulky do jiné databáze nebo na jiný server, netřeba vyplňovat název textového souboru pro export, pak soubor najít, otevřít v textovém editoru, zkopírovat jeho část a opět vložit do jiného SQL editoru. Stačí zaškrtnout příslušnou volbu v dialogu. Máte tak plnou kontrolu nad operacemi v databázi. Díky zpětné vazbě od mnoha každodenních uživatelů se tak Micka chová opravdu přívětivě.

### 5.2 Vývojové prostředí a programovací jazyk

Na počátku vývoje jakéhokoli software stojí vždy poptávka. Již od počátku Micka nevznikala jen pro potřebu samotného tvůrce, ale především jako přívětivý nástroj pro několik firemních klientů a spolupracovníků na mnoha projektech, především webových stránkách, elektronických obchodech a databázových systémech. Drtivá většina jednotlivců i firem používá operační systém (dále jen OS) MS Windows. Stejně jako všichni, pro které původně projekt vznikl.

Vzhledem k požadavku rychlosti a efektivitě bylo ideální začít s nějakým vývojovým



Obrázek 5.1: Delphi

prostředím, které je nativní pro tento OS a existuje pro něj rozhraní pro přímou komunikaci s databází bez dalších knihoven a rozhraní třetích stran. Takové požadavky výborně splňuje vývojové prostředí Borland Delphi s jazykem Object Pascal v kombinaci s komponentou Direct SQL[14] pro přímý přístup k databázovému serveru. Navíc s ním má autor nejvíce zkušeností. Delphi je kvalitní a moderní prostředí, v němž mohou vznikat špičkové aplikace. Projekt Micka tedy začal vznikat v jazyce Object Pascal v prostředí Borland Delphi 6[8].

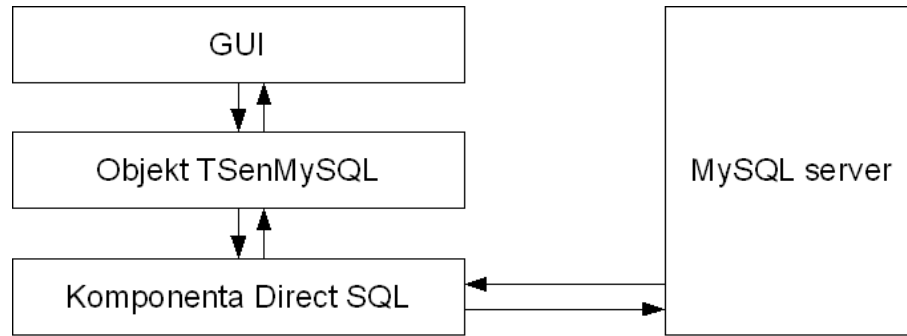
## 5.3 Popis implementace

### 5.3.1 Architektura

Implementace je postavená na jednotlivých od sebe oddělených vrstvách. Tato architektura umožňuje především snadnou rozšiřitelnost, i když má i mnoho jiných dobrých vlastností. Flexibilitu při aktualizaci nebo změny komponenty pro samotné připojení a komunikaci se serverem. Globální ošetření výjimek a chyb. Možnost použít celou vrstvu v jiné speciální aplikaci.

Nejnižší je komponenta Direct SQL[14] publikovaná pod GPL[3] licencí na <http://sourceforge.net/projects/directsql/>, která komunikuje přímo se samotným MySQL serverem. Komponenta obsahuje několik tříd a objektů. Nejdůležitější je třída `TMysqlClient`, jejíž metody se starají o samotnou komunikaci. Metody jako `connect()` nebo `query()` nepotřebují vysvětlovat. Další důležitou třídou je `TMysqlResult`, kterou vrací metoda `query()` i některé další a jejíž pomocí se pracuje s výsledkem dotazu. Výsledek obsahuje, kromě vlastností sloupečků a různých pomocných hodnot, ukazatel na aktuální řádek výběru a pomocí metod `prev` a `next` jej posunujeme. Celá komponenta je napsaná elegantně a pracuje s celým připojením rychle a efektivně. Jen je třeba pečlivě uvolňovat paměť po každém dotazu, protože Delphi neobsahuje garbage collector<sup>1</sup> a po delším běhu programu by nastaly problémy

<sup>1</sup>Garbage collector je obvykle část běhového prostředí (programovacího) jazyka nebo přídavná knihovna, podporovaná kompilátorem, hardware, operačním systémem nebo jakoukoli kombinací těchto tří. Má za úkol automaticky určit, která část paměti programu je už nepoužívaná, a připravit ji pro další znovupoužití.[17]



Obrázek 5.2: Vrstvy komunikace

s neúměrným nárůstem obsazené paměti.

Tuto komponentu "vidí" v celé aplikaci pouze jediný objekt `TSenMySQL` se kterým pak pracuje zbytek programu. `TSenMySQL` odstíní samotný přístup k MySQL, takže pokud by se například zastavil vývoj komponenty Direct SQL, jiná komponenta se ukázala výhodnější, nebo by bylo potřeba celé prostředí přizpůsobit pro komunikaci s třeba PostgreSQL, stačí v zásadě přepsat několik metod v `TSenMySQL` a zbytek bude fungovat korektně. Deklarace objektu je v dodatku B na straně 45.

Poslední, vlastně uživatelskou vrstvou, jsou samotné formuláře. Micka je klasická formulářová aplikace řízená událostmi. Například uživatel klikne na tlačítko *Připojit*, událost zavolá proceduru, která podle zaškrtnuté položky v menu zjistí parametry připojení a zavolá metodu `connect()` objektu `TSenMySQL` a ukončí se. Úspěšné připojení vyvolá další událost, která zavolá jinou proceduru obsahující kód pro ošetření menu a inicializuje výpis databázi. A tak dále.

### 5.3.2 Uživatelské rozhraní

Uživatelské rozhraní jinak také GUI<sup>2</sup> je standardní SDI<sup>3</sup> aplikační okno systému Windows. V záhlaví se zobrazuje aktuální databáze a tabulka. Obsah okna je rozvržen podobně jako u ostatních aplikací pro správu databáze, včetně webových. Toto rozvržení je přehledné a uživatelé jsou na něj zvyklí.

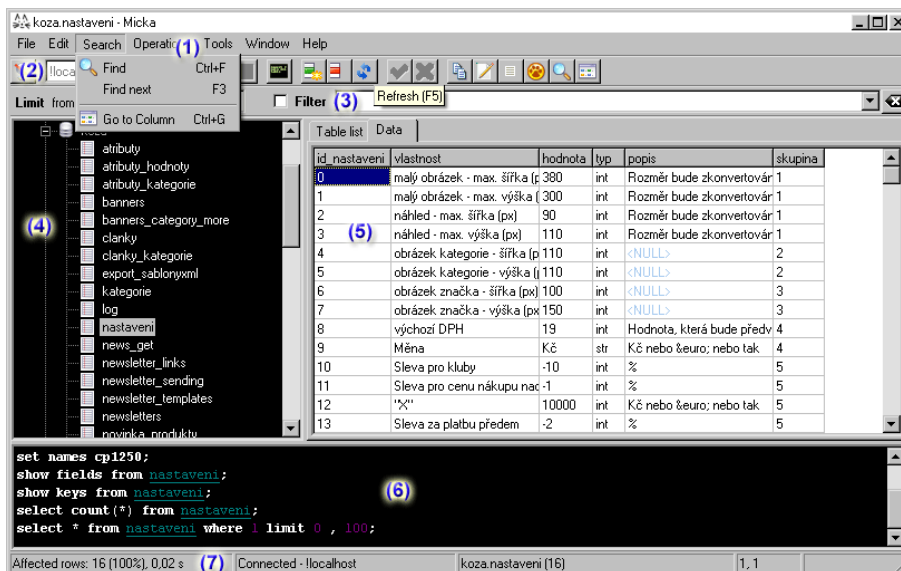
Uživatelské rozhraní je vidět na obrázku 5.3. Nahoře standardní rozbalovací menu (1). Dále tlačítková lišta (2), která přehledně zobrazuje seznam připojení a tlačítka pro nejčastější operace. Po najetí kurzorem myši se jako kontextová nápověda (hint) zobrazí klávesová zkratka. Další lišta slouží pro nastavení zobrazení výsledku a filtraci (3). Následuje samotná pracovní plocha. Po levé straně strom objektů (4) (nikoli objekty ve smyslu objektového modelu, ale jednotlivé entity databáze). Vpravo jsou záložky (5) pro seznam tabulek, data a podobně. Pod tím vším je aktivní SQL log (6), kam se vypisuje veškerá komunikace. Na konci, úplně dole, je stavový řádek (7) se stavovými hodnotami. Podrobnosti o zobrazení jsou v uživatelské příručce v dodatku A na straně 31.

Další důležitou součástí je takzvané **SQL okno**. Je to textové pole, kam lze zapsat libovolný dotaz nebo sérii dotazů a ty pak poslat serveru. V dolní části okna se zobrazí výsledek posledního dotazu. Do textového pole se dá načíst textový soubor, uložit jednotlivé dotazy do oblíbených nebo uložit obsah pole do textového souboru. Samotný editor umí highlighting<sup>4</sup>,

<sup>2</sup>GUI (Graphical User Interface), grafické uživatelské rozhraní

<sup>3</sup>SDI (Single Document Interface), organizace GUI, kde v uvnitř hlavního okna je otevřen vždy pouze jeden dokument. Druhá varianta je MDI (Multiple Document Interface), kdy hlavní okno je rodič a může mít otevřeno více dokumentů jako potomky.

<sup>4</sup>Zvýraznění syntaxe



Obrázek 5.3: GUI - Hlavní okno

doplňování a zvýraznění názvů tabulek a sloupečků, zalamování řádků a ostatní funkce běžné pro textové pole ve Windows, jako je kopírování, vkládání atd.. Pro tento účel je použita komponenta SynEdit[5] dostupná na <http://synedit.sourceforge.net>.

## 5.4 Omezení

Nejzásadnější omezení pramení ze zvoleného vývojového prostředí Borland Delphi. Aplikaci Micku nelze portovat na jiný operační systém než, je 32 bitová Microsoft Windows. Zdůvodnění volby takového vývojového prostředí je uvedeno v kapitole 5.2 na straně 19. Pokud Borland (CodeGear), jako tvůrce Delphi, uvolní verzi překladače pro 64 bitové Windows nebo pro Linux, nebude problém Micku portovat.

Komponenta pro připojení k MySQL je napsaná tak, aby se uměla připojit odkudkoli i z Linuxu. Dokonce byla testována na Kylixu 3 od Borlandu, což je obdoba Delphi 6 pro Linux, dnes již bohužel poněkud zastaralá. To je jediná potenciální překážka při portování celého projektu. Zbytek je důsledně napsán jen se standardními komponentami.

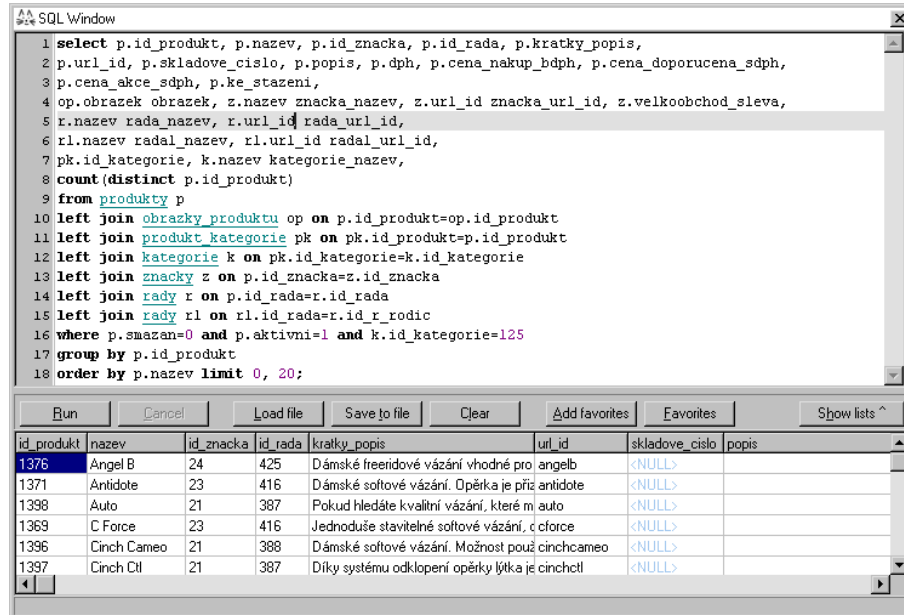
Další omezení vyplývá z vývojového týmu, který tvoří právě jeden programátor pracující v podstatě zadarmo. Další vývoj je sice výrazně motivován mnoha různými potřebami, ale jeden člověk je vždy zdrojem potenciální nestability. Pokud by se našel větší tým programátorů, další vývoj by byl jistě a spolehlivě zajištěn.

Určitou překážkou rozšiřování vývojového týmu může být komerční povaha vývojového nástroje. To předpokládá, aby se potenciální spolupracovník sám významně zabýval prací s Delphi a měl tedy licenci. I když cena licence, zejména pod tlakem konkurenčních prostředí a jazyků, často nabízených i zdarma, se postupně snižuje.

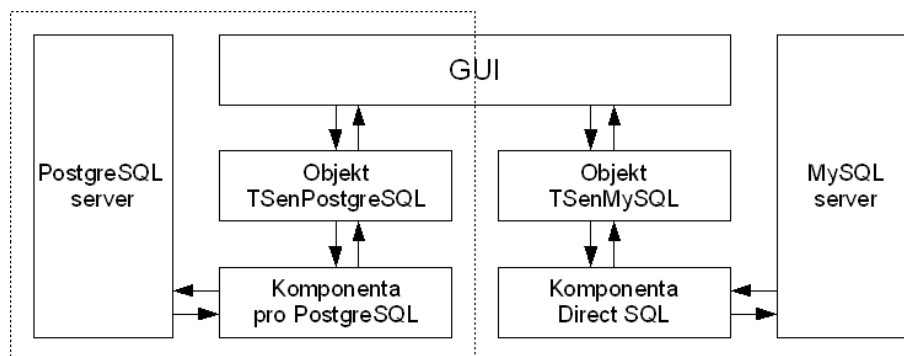
## 5.5 Rozšiřování

### 5.5.1 Další databázové systémy

Celá aplikace je navržena tak, aby bylo uživatelské rozhraní odděleno od samotné komunikace s daným databázovým strojem. Převod celého programu pro jinou databázi než MySQL nebo možnost použít GUI pro více rozdílných databází spočívá v zásadě v napsání jednoho nového objektu pro přístup k dané databázi a rozvětvení na uživatelské úrovni.



Obrázek 5.4: GUI - SQL okno



Obrázek 5.5: Rozšíření na další systémy komunikace

### 5.5.2 Nové funkce

Je třeba také myslet na poměrně rychlý vývoj MySQL. Klient by měl reflektovat všechny nové vlastnosti a možnosti databázového stroje. Použitelnost se do značné míry odvíjí právě od neustálého vývoje a vydávání nových verzí. Tato práce zachycuje koncepci programu a jeho okamžitý stav, který je však potřeba nadále a neustále vyvíjet.

Další funkce a schopnosti programu Micka se do značné míry odvíjejí od času, který bude vývojář ochoten obětovat nad rámec svých pracovních i jiných povinností.

Vývojové prostředí Delphi je dostatečně silné na zaimplementování téměř jakýchkoli funkcí.

**Přenos dat** Určitě by byl velice užitečný silný nástroj pro přenos dat mezi databázemi. Tím by se jedním průvodcem, jedním oknem navíc, vyřešilo hned několik funkcí. Jako přesun nebo kopírování tabulek a celých databází. Navíc i mezi různými MySQL servery. Zatím to Mice chybí.

**Tisk** Občas se hodí vytisknout obsah tabulky nebo i strukturu. Všechny tabulkové výpisy, včetně seznamu tabulek nebo seznamu sloupečků, jsou realizovány komponentu `TStringGrid`. Stačilo by napsat proceduru nebo možná spíš třídu i formulářem pro nastavení tisku, které by se jako parametr předal ukazatel na jakýkoli `StringGrid` a jednoduše tak máme prostředek, jak tisknout téměř cokoli z celé aplikace.

**Hybridní MDI rozhraní** K efektivnějšímu procházení více tabulek zároveň by pomohlo přidání hybridního MDI rozhraní se záložkami, jaké známe například z moderních internetových prohlížečů.

**Pohledy, trigger, uložené procedury** V Programu prozatím chybí podpora pro pohledy, trigger a uložené procedury, které MySQL podporuje od verze 5. To bude náročná úprava, která vyžaduje několik změn v celém programu. Kromě přidání záložek pro jejich zobrazení, jak je zmíněno v předchozím odstavci, je třeba přebudovat strom objektů, kam přibude hned několik nových druhů. A v neposlední řadě bude nutné přepsat parser<sup>5</sup> SQL dotazů, který umí rozpoznat jednotlivé jednoduché dotazy, ale neumí poznat složitější konstrukce například pro deklaraci procedur nebo triggerů.

Jak je vidět, práce je spousta. Ale je položen kvalitní základ s dobrým uživatelským rozhraním, na kterém je možno a také nutno do budoucna stavět. Podobných projektů není mnoho a bylo by škoda zahodit potenciál a spoustu práce, která je již vykonána.

Také by asi stálo za to, nepublikovat spustitelnou verzi jen jako zip archiv, ale i ve formě instalátoru. Některé uživatele to může odradit, na druhou stranu má jistotu, že se mu nežádoucím způsobem nezapíše do systému, jak to s oblibou dělají některé programy. Některé knihovny do systémového adresáře, některé do uživatelského adresáře a spousta záznamů do registrů systému Windows. A při odinstalaci tam toho půlka zbude. To u Micky rozhodně nehrozí.

---

<sup>5</sup>Parser je algoritmus, který analyzuje předložená data (obvykle textový soubor) a vyhledává v něm prvky odpovídající definici určitého jazyka. Obvykle provádí syntaktický rozbor zdrojových kódů napsaných v daném programovacím jazyce[17].

## 6 Testování

Aplikace Micka je testována v první řadě v průběhu vývoje. Každá nově naprogramovaná část se důkladně testuje na funkčnost. Pokud má napojení na jiné části, musí se testovat i ty, zda nedošlo k negativnímu ovlivnění. A pak je samozřejmě ještě třeba zkoušet, jak pracuj celek.

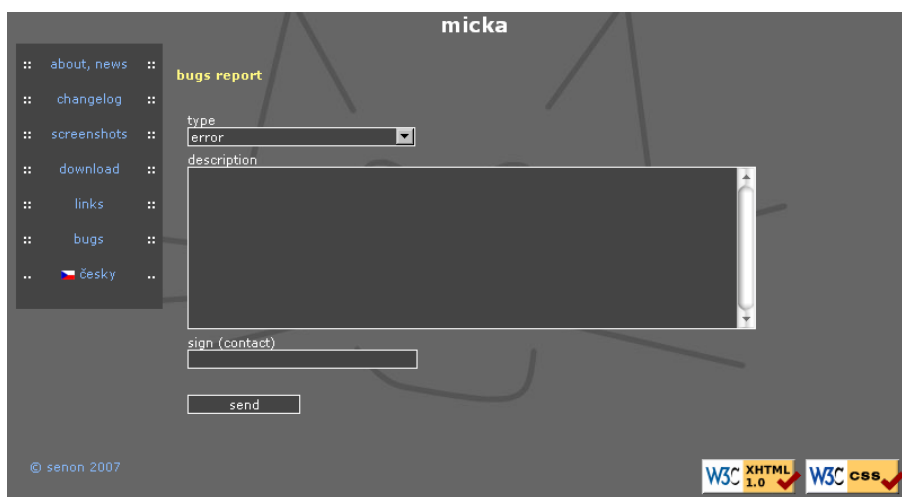
Toto testování při vývoji, navíc stejnými lidmi, kteří aplikaci tvoří, nestačí. Odhalí zásadní chyby, ale těžko může dokonale odhalit například záludnosti uživatelského rozhraní, které nakonec vždy zjistí až nezávislý uživatel. Nebo chování programu v nestandardních podmínkách, se vstupy, které by vývojáře nikdy nenapadly. V případě Micky i připojení na nějaký exotický MySQL server ve smyslu verze a konfigurace.

Protože Micka je určena v první řadě do praxe, nejvíce přínosné je právě praktické testování.

### 6.1 Nasazení v praxi

Testování použitelnosti uživatelského rozhraní i funkčnosti probíhá již bezmála dva roky téměř denně v reálném pracovním prostředí. Nástroj je nejvíce používán v Direct marketingové agentuře Wunderman v interaktivním oddělení, které se zabývá vývojem software, zejména webových aplikací. Odtud pramení nejvíce podnětů a upozornění na chyby. Dále se program používá pro správu dat administrátorem několika elektronických obchodů<sup>[9]</sup> a v neposlední řadě pro správu databáze dalších dvou firem, odkud je také výrazná zpětná vazba. Díky dlouhodobému nasazení v pracovním prostředí je program Micka poměrně dobře odladěný a vyladěný pro uživatelskou přívětivost.

Na samotném webu produktu `<http://www.senon.cz/micka/>` je formulář pro hlášení chyb. Na formulář vede přímý odkaz z aplikace *Help - Micka bugs report*, který otevře okno prohlížeče s požadovanou stránkou.



Obrázek 6.1: Webový formulář pro hlášení chyb

Ve formuláři lze vybrat, zda je to chyba nebo podnět. Příspěvky se ukládají do speciálního systému pro správu ToDo seznamu<sup>1</sup>. Po příchodu se zpráva zařadí do skupiny a určí se jí priorita. Postupně se pak zapracovávají a označují za hotové, případně komentují.

<sup>1</sup>ToDo - z anglického to do, k udělání. Je to zpravidla seznam položek, většinou organizovaných do skupin, které čekají na vyřízení nebo implementování. Do seznamu se dají obvykle zadávat komentáře a priorita položky.

The screenshot shows the SourceForge.net project page for 'micka'. The page layout includes a top navigation bar with 'Home', 'Browse Software', 'Marketplace', 'Community', and 'Create Project'. Below this is a search bar with 'Software' selected and 'Search' and 'Advanced' buttons. The main content area features a project summary for 'micka', a 'Download micka' button, and a 'Latest News' section with three entries: 'Micka 1.0.0 for win32 released' (2007-12-19), 'Micka 0.9.8 for win32 released' (2007-11-23), and 'Micka 0.9.2 for win32 released' (2007-07-24). There are also advertisements for 'IntelliJ IDEA' and 'HP'.

Obrázek 6.2: Micka na SourceForge.net

## 6.2 SourceForge.net

Od 15.12.2006 je Micka publikována na portálu SourceForge.net <<http://sourceforge.net/projects/micka/>>, který sdružuje open source projekty, obsahuje systém pro hlášení chyb, fóra a mnoho dalších nástrojů pro usnadnění vývoje. Odtud byla Micka stažena v době psaní této práce asi třístokrát. Poprvé verze 0.0.10, která již měla plnou funkcionalitu pro práci s daty, ale to bylo také všechno co uměla. Z tohoto zdroje nepřišlo žádné upozornění na chyby ani nové podněty. Pravděpodobně proto, že kdo je spokojený nepíše a kdo není, program smaže a nainstaluje jiný. To by musel být vysloveně nadšenec, který by se chtěl aktivně podílet na dalším vývoji.

## 7 Závěr

### 7.1 Vyhodnocení použitelnosti navržené aplikace

Podářilo se vyvinout aplikaci, databázového klienta pro MySQL, který nejen, že by měl být teoreticky v praxi velice dobře použitelný, ale je i skutečně využívaný pro každodenní práci s databázemi.

V mnoha parametrech předčí komerční produkty nebo se jim přinejmenším vyrovná. Jeho výhodou je zejména rychlost připojení a efektivní navigace v databázových objektech. Snaží se minimalizovat komunikaci se serverem. Proto je i řádově rychlejší při pomalém připojení a dlouhých odezvách sítě, například přes GPRS<sup>1</sup>.

Vyniká přehledným GUI a pohodlným ovládním plným drobností, které při mnoha běžných operacích zpřijemňují a zefektivňují práci. Zejména se to týká navigace ve velkých databázích s mnoha tabulkami a v rozsáhlých tabulkách, jak řádkově, tak sloupcově.

Program je zaměřen na práci s daty. Chybí mu sice vyspělé nástroje na administraci serveru nebo propracovaní průvodci synchronizací, zato se ale výborně hodí na prohlížení, vyhledávání, filtrování a úpravy dat. Zvládne všechny serverem podporované znakové sady, dlouhé texty v buňkách i větší tabulky. Podporuje export i import dat. Umí všechny základní operace s databází, které může vývojář potřebovat. Přehled všech aktuálních funkcí a popis práce s nimi je v instalační a uživatelské příručce na straně 31.

### 7.2 Náměty pro další práci

Databázový systém MySQL se stále poměrně rychle vyvíjí. Tato práce zachycuje stav programu Micka a jeho vlastnosti ve vztahu k MySQL právě v tuto chvíli. Aby byl program použitelný i v budoucnu, musí se stále vyvíjet spolu s databázovým systémem. Měl by reflektovat nové funkce, nové možnosti a případné zásadní změny a měl by přitom zůstat pokud možno zpětně kompatibilní.

Tato diplomová práce je začátkem dlouhé cesty tohoto klienta se serverem ruku v ruce.

### 7.3 Možnosti rozvoje prostředí

Hlavním přínosem je samotná koncepce tohoto nástroje a jeho propracované GUI. Pod souhrnem formulářů, tabulek a klávesových zkratk běží poměrně jednoduchá a přímá třída, která komunikuje právě s MySQL serverem. Pokud bychom chtěli toto prostředí použít pro jiný databázový systém, bylo by to možné relativně s minimem práce. Architektura je podrobněji popsána v kapitole 5.3.1 na straně 20.

Cíle diplomové práce tedy byly splněny. Přesto bude vývoj databázového klienta dále pokračovat a bude se stále dál snažit zefektivnit a zpřijemnit co nejvíce uživatelům práci.

Aktuální informace o vývoji Micky jsou dostupné na internetové adrese <http://www.senon.cz/micka/> a na <http://sourceforge.net/projects/micka/>.

---

<sup>1</sup>General Packet Radio Service (GPRS) je mobilní datová služba přístupná pro uživatele GSM mobilních telefonů[17].



## 8 Seznam literatury

- [1] DUBOIS, P.: *MySQL profesionálně*. Mobil Media, a.s., 2003, ISBN 80-86593-41-X.
- [2] FISCHER, O.: *Syntaxí řízený editor jako webová aplikace*. Diplomová práce, FEL ČVUT, 2007.
- [3] GNU: GNU General Public License, version 2. Online, 2007.  
URL <<http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>>
- [4] HANDZLIK, D.: VertrigoServ. Online, 2007.  
URL <<http://vertrigo.sourceforge.net>>
- [5] HARMON, D.; kolektiv: Delphi komponenta SynEdit. Online, 2007.  
URL <<http://synedit.sourceforge.net>>
- [6] HULÁN, R.: Instalace Apache, MySQL a PHP na Windows. Online, 2008.  
URL <<http://myego.cz/item/instalace-apache-mysql-a-php-na-windows>>
- [7] MARTÍNEK, D.: Český styl pro BibTeX. Online, 2007.  
URL <<http://www.fit.vutbr.cz/~martinek/latex/czechiso.html.cs.iso-8859-2>>
- [8] MILLER, T.; POWELL, D.; kolektiv: *Mistrovství v Delphi 3*. Computer Press, 1998, ISBN 80-7226-110-X.
- [9] MUNZAR, Z.: Projekt internetového obchodu Koza. online, 2007.  
URL <<http://www.etennis.cz>>
- [10] MySQL\_AB: *MySQL 3.23, 4.0, 4.1 Reference Manual*. 2006.
- [11] MySQL\_AB: *MySQL 5.1 Reference Manual*. 7918 vydání, 2007.
- [12] MySQL\_AB: *MySQL 6.0 Reference Manual*. 9490 vydání, 2008.
- [13] MySQL\_AB: Oficiální portál MySQL. online, 2008.  
URL <<http://www.mysql.com/>>
- [14] NICOLA, C.; kolektiv: Delphi komponenta Direct SQL. Online, 2007.  
URL <<http://sourceforge.net/projects/directsql>>
- [15] POKORNÝ, J.; HALAŠKA, I.: *Databázové systémy*. Skriptum FEL ČVUT, 2003.
- [16] WIKIPEDIA: Java Runtime Environment. Online, 2007.  
URL <[http://en.wikipedia.org/wiki/Java\\_Runtime\\_Environment](http://en.wikipedia.org/wiki/Java_Runtime_Environment)>
- [17] WIKIPEDIA: Wikipedia česky. Online, 2008.  
URL <<http://cs.wikipedia.org>>



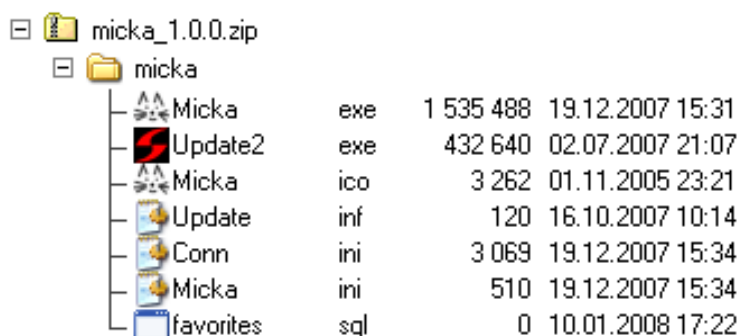
## A Instalační a uživatelská příručka

### A.1 Vlastnosti Micky

- Klíčové vlastnosti
  - Plná podpora MySQL od verze 4.1 a částečná podpora od verze 3.23
  - Extrémně rychlá komunikace s MySQL serverem
  - Podpora všech znakových sad
  - Profily připojení
  - Správa databází
  - Operace s tabulkami, správa struktury tabulek
  - Výborná práce s daty
  - Propracovaný SQL editor
  - Export do SQL a CSV
  - Monitorování procesů
- Výhody
  - Paměť poslední relace
  - Filtrování dat, fulltextové vyhledávání
  - Rychlá navigace v rozsáhlých databázích a velkých tabulkách
  - SQL log
  - Mnoho drobných detailů, které usnadňují práci

### A.2 Distribuce

Program Micka je distribuován jako komprimovaný zip archiv `micka_1.0.0.zip`. Stáhnout jej můžete buď přímo z internetových stránek projektu Micka <<http://www.senon.cz/micka/>>, nebo z portálu SourceForge <<http://sourceforge.net/projects/micka/>>.



File Name	Extension	Size	Date	Time
Micka	exe	1 535 488	19.12.2007	15:31
Update2	exe	432 640	02.07.2007	21:07
Micka	ico	3 262	01.11.2005	23:21
Update	inf	120	16.10.2007	10:14
Conn	ini	3 069	19.12.2007	15:34
Micka	ini	510	19.12.2007	15:34
favorites	sql	0	10.01.2008	17:22

Obrázek A.1: Obsah distribuce

### A.3 Instalace

#### A.3.1 Systémové požadavky

- Počítač typu PC s operačním systémem Microsoft Windows 98/XP/Vista
- Funkční připojení k databázovému serveru MySQL (localhost, LAN, Internet, ...)

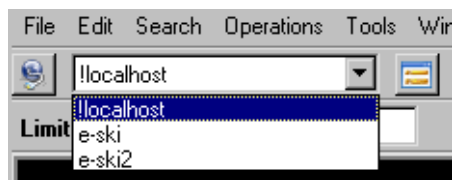
### A.3.2 Postup instalace

1. Rozbalte komprimovaný zip archiv do adresáře, ze kterého budete chtít program používat, například `c:\Program Files\`. Archiv přímo obsahuje adresář Micka a v něm všechny potřebné soubory, jak je vidět na obrázku A.1. Nové verze Windows umí nativně pracovat s formátem zip. Nebo k rozbalení můžete použít nějaký další nástroj, například ExtractNow ( <http://www.extractnow.com> ).
2. Samotný program Micka se spouští souborem `Micka.exe`, ostatní jsou pomocné. Vhodné je vytvořit zástupce na ploše nebo v panelu snadné spuštění právě pro tento exe soubor.
3. Instalace je hotova a můžete spustit `Micka.exe` nebo jeho zástupce.

## A.4 Připojení k MySQL serveru

### A.4.1 Připojení a odpojení

Jedna spuštěná instance Micky se umí připojit právě k jednomu MySQL serveru. Umožňuje však uložit až 16 různých profilů připojení, z nichž si uživatel vybere vždy jeden a použije jej. Profil vyberete z roletové nabídky v nástrojové liště a připojíte se stisknutím tlačítka *Connect* vedle vlevo, nebo výběrem z hlavního menu *File - Connect*, klávesová zkratka je `<F11>`. Odpojuje se stejným tlačítkem.



Obrázek A.2: Volba připojení

### A.4.2 Nastavení připojení

Pokud neexistuje žádný profil připojení, po stisknutí tlačítka *Connect* se objeví dialog *Connection setting*. Ten lze vyvolat i z hlavního menu *Tools - Connection setting* nebo tlačítkem na nástrojové liště vlevo od výběru profilu `<Shift+Ctrl+F12>`. V dialogovém okně je seznam připojení a vpravo tlačítka pro operace:

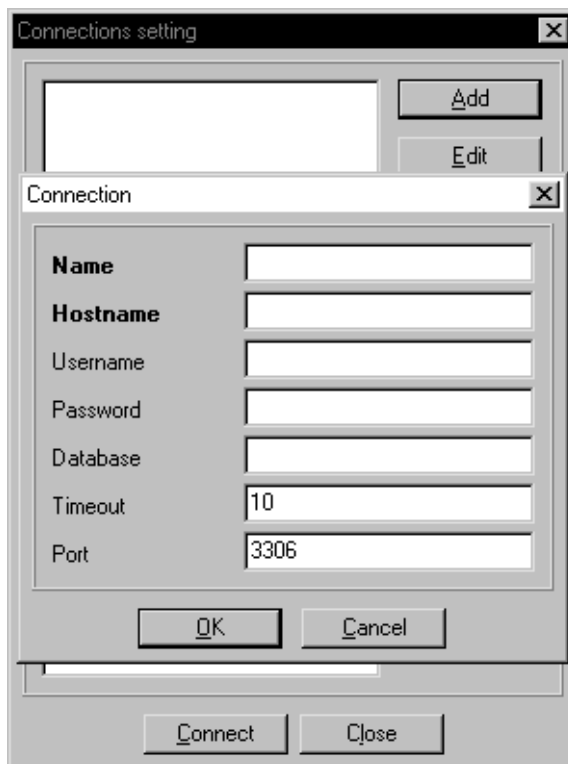
**Add** - přidání nového připojení

**Edit** - úprava připojení

**Duplicate** - zkopírování připojení

**Remove** - odstranění připojení

Po kliknutí na *Add*, *Edit* nebo *Duplicate* se otevře formulář pro konkrétní připojení. Vyplnění polí v případě zkopírování připojení se stejně jako editace, jen není vyplněn název připojení. Textové pole *Name* je pro zadání názvu připojení, je to povinný parametr a můžete tam zadat v podstatě jakékoli jméno. Název musí být jedinečný. Nepovinný parametr *Database* říká, která jedna databáze se má zobrazit, pokud má uživatel<sup>1</sup> přístupová práva



Obrázek A.3: Dialog připojení

<sup>1</sup>Uživatel ve smyslu přihlášení k databázovému serveru.

k více databázím. Když pole zůstane prázdné, zobrazí se kompletní seznam. *Timeout* je čas v sekundách, který klient čeká na odezvu serveru. Význam ostatních políček je zřejmý.

Po potřebných úpravách v seznamu připojení můžete dialog zavřít a připojit se dle postupu uvedeném v A.4.1 nebo se rovnou připojit k označenému připojení tlačítkem *Connect*.

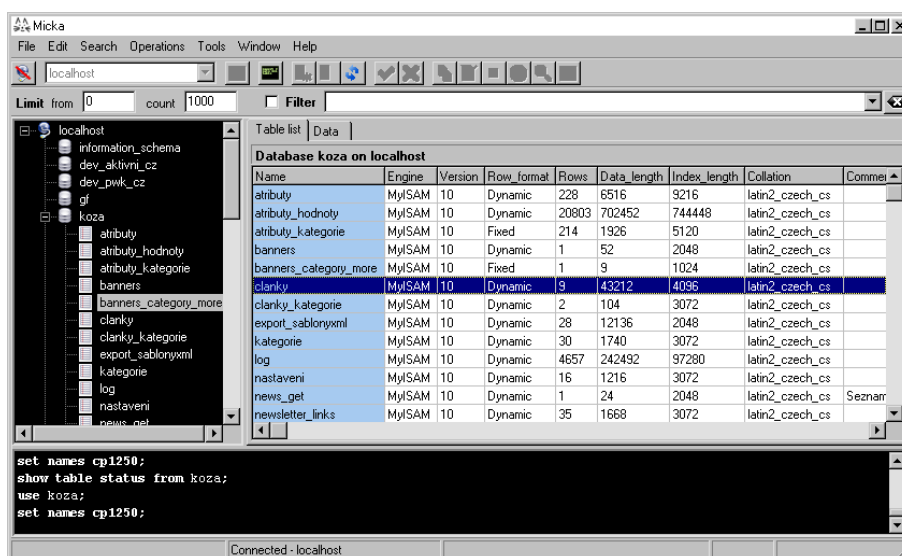
### A.4.3 Nastavení znakové sady

Znaková sada, kterou bude server používat pro komunikaci s klientem, se nastavuje při otevření databáze. Místa pro zobrazování výsledků dotazů používá standardní komponenty Windows, proto je pro korektní zobrazení diakritiky potřeba používat také standardní kódování dané verze Windows. Pro střední Evropu je to cp1250. Výstupní znakovou sadu serveru můžete změnit v dialogu nastavení, *Tools - Setting - MySQL*. Více o nastavení v kapitole A.10 na straně 43.

### A.4.4 Zobrazení objektů databáze

Po úspěšném připojení k serveru v levé části okna s černým pozadím zobrazí strom objektů k danému databázovému systému. Kořenem stromu je název připojení. Potomci jsou databáze, ke kterým má přihlášený uživatel právo. Jinými slovy, na druhé úrovni stromu je seznam dostupných databází. Pokud poklepete na název databáze, rozbalí se další větve stromu, seznam tabulek zvolené databáze. Tímto způsobem můžete rozkliknout několik databází. Ze serveru se vždy načítají jen nezbytně nutné informace o databázi, se kterou právě pracujete.

Po rozvinutí dané databáze se seznam tabulek i s podrobnostmi zobrazí také v hlavním okně aplikace. Názvy tabulek jsou podsvíceny modře. V tomto seznamu lze rychle vyhledávat a filtrovat pomocí editačního okénka. Když máte kurzor v modrém seznamu tabulek a začnete psát písmena nebo čísla, objeví se editační okénko a kurzor se přesouvá jen po těch tabulkách, které odpovídají napsanému vzoru. Po stisku <Enter> nebo poklepnutí na jméno tabulky se v hlavním okně zobrazí záložka data, ve které je výpis zvolené tabulky. Výpis se řídí druhou nástrojovou lištou, kam se dají zadat limity a filtr. Filtr si aplikace pamatuje pro každou tabulku zvlášť, stejně jako seřazení, ale o tom více v kapitole o práci s tabulkou A.6 na straně 35.



Obrázek A.4: Miska po připojení a označení databáze

Celý strom objektů lze aktualizovat ze serveru. Můžete zvolit z hlavního menu *Edit - Refresh*

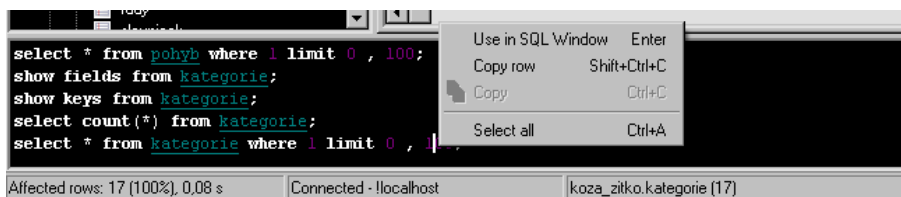
nebo stisknout klávesu <F5> v okně stromu, nebo zvolením položky *Refresh* v kontextovém menu<sup>2</sup> stromu.

Ze stromu objektů můžete jakýkoli název zkopírovat do schránky pro použití například v PHP skriptu nebo v dokumentech. Příkaz *Copy name* <Ctrl+C> zvolte z kontextového menu stromu, jak ukazuje obrázek A.7 na straně 36.

Za zmínku ještě stojí položka kontextového menu stromu *Properties*. Tato položka zobrazí okno vlastností toho objektu, na němž máte právě kurzor. Vlastnosti připojení k serveru znamenají výpis Proměnných MySQL a aktuálního statusu serveru. Vlastnosti databáze zobrazí seznam tabulek s podrobnostmi a opět status serveru. Vlastnosti tabulky otevřou dialog pro správu sloupečků a klíčů tabulky, podrobnosti opět v kapitole A.6.

#### A.4.5 SQL log a stavový řádek

Další užitečnou součástí je SQL log v dolní části okna. Sem se vypisují veškeré dotazy, kterými Micka komunikuje s MySQL serverem. Dotazy mají zvýrazněnou syntaxi a zvýrazněné názvy tabulek. SQL log má své kontextové menu, kde jsou navíc dvě netradiční položky. *Copy row* zkopíruje přímo celý řádek, na němž je kurzor do schránky, ušetří nám tak označování textu. Položka *Use in SQL Window* vezme aktuální řádek a vloží jej do SQL okna, které se zobrazí a dostane focus<sup>3</sup>. Tak můžete snadno a rychle použít generované dotazy k přípravě svých vlastních.



Obrázek A.5: SQL log a stavový řádek

Ještě pod SQL logem je stavový řádek. Ten obsahuje několik políček. V tuto chvíli jsou pro nás zajímavé první tři. V prvním políčku se zobrazuje počet ovlivněných řádků posledním dotazem, který vidíte v logu. V závorce hned za ním je v procentech počet ovlivněných řádků, vztažených relativně k počtu řádků aktuálně otevřené tabulky. Hodnota, která je čistě informativní, ale v mnoha případech zajímavá. Například, filtrujete-li v tabulce objednávek ještě nezaplacené, zde se okamžitě dozvíte, jaký je jejich podíl, a nemusíte nic přepočítávat. Poslední hodnota v tomto políčku je orientační doba, kterou trvalo vykonání posledního dotazu. Do této doby je započítána i režie na komunikaci, takže u rychlých dotazů je nepřesná. U náročných dotazů je však doba samotné komunikace zanedbatelná. V dalším políčku se zobrazí stav a jméno aktuálního připojení. Ve třetím políčku je název právě otevřené tabulky včetně databáze a v závorce počet záznamů v tabulce. Stavový řádek má ještě jedno políčko, kam se vypisuje pozice kurzoru v tabulce, to bude zmíněno v kapitole A.6.

#### A.4.6 Paměť poslední relace

Když pracujete Mickou na nějaké databázi, je poměrně pravděpodobné, že při dalším spuštění budete chtít pracovat na té samé. Proto si Program umí zapamatovat, kam jste byli při vypnutí programu připojení, a tuto relaci po dalším spuštění obnoví. Pamatuje si server a konkrétní

<sup>2</sup>Kontextové menu každého objektu ve Windows se vyvolá kliknutím pravého tlačítka na daný objekt.

<sup>3</sup>Komponenta (vizuální prvek Windows, tlačítko, seznam, strom, editační okno, ...) má focus, pokud je označena, například tlačítko, nebo v případě textového pole v ní bliká kurzor. Okno s focusem je zpravidla vpředu a má barevně odlišen pruh záhlaví. Focus může mít vždy jen jedno okno a na něm jedna komponenta.

databázi. Tato funkce jde vypnout v nastavení programu, *Tools - Setting - Environment*. Více o nastavení v kapitole A.10 na straně 43.

## A.5 Práce s databází

Aplikace umí kompletně pracovat s jednotlivými databázemi na MySQL serveru. S databázemi se pracuje, stejně jako s ostatními objekty v celém programu, primárně pomocí kontextového menu. Když kliknutím pravého tlačítka myši na objekt ve stromě vyvoláte kontextové menu, můžete si povšimnout, že ke každé úrovni objektů přísluší jiné položky v tomto menu. Jednotlivé varianty vidíte na obrázku A.7 na straně 36.



Obrázek A.6: Dialog pro vytvoření nebo úpravu databáze

Databázi můžete vytvořit, upravit, odstranit a exportovat. Exportu se podrobně věnuje kapitola A.7 na straně 40. K odstranění databáze není co dodávat. Snad jen upozornění, že Micka, stejně jako jiné aplikace pro správu databází, nemá žádné undo<sup>4</sup>. To znamená, že ukvapené odstranění databáze nelze vrátit. Jedině obnovit ze zálohy. Před odstraněním se samozřejmě zobrazí potvrzující dotaz "Opravdu chcete ... ?".

**Vytváření i úprava databáze** se provádí pomocí dialogu, jež vidíte na obrázku A.6. Dá se zde zvolit výchozí znaková sada a porovnání pro celou databázi. Nově vytvořená tabulka pak bude implicitně používat toto nastavení.

Micka zatím neumí databázi přejmenovat, respektive jednoduchý SQL dotaz pro přejmenování se plánuje až do dalších verzí MySQL. To znamená, že přejmenování by šlo realizovat vytvořením nové databáze se stejnými vlastnostmi, ale novým jménem, zkopírováním všech tabulek a zrušením původní databáze. Toho se chce tento program vyvarovat.

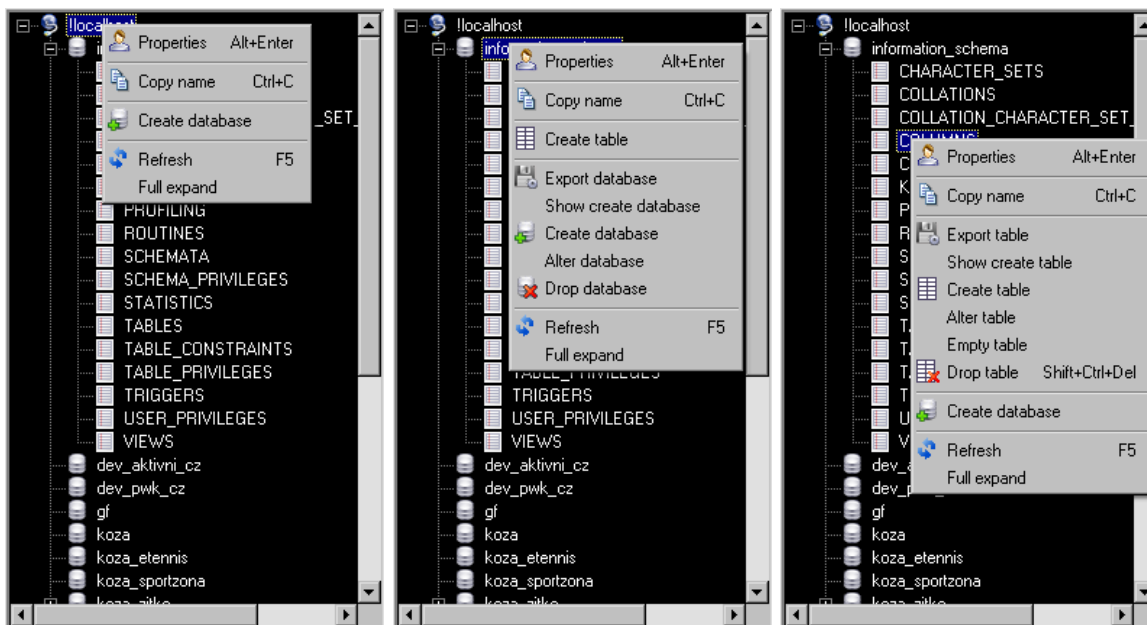
## A.6 Práce s tabulkou

Celá Micka je primárně navržena pro práci s tabulkami. Proto zde nabízí nejvíce možností a funkce zobrazování, úprav i navigace jsou nejvíce propracované.

### A.6.1 Struktura a vlastnosti tabulky

Přístup k funkcím pro práci se strukturou a vlastnostmi tabulky je opět z kontextového menu stromu objektů. To vidíte na obrázku A.7 na straně 36 v části Tabulka. Tabulku můžete vytvořit, upravit, vyprázdnit, odstranit, upravit strukturu a exportovat. Export je popsán v kapitole A.7 na straně 40.

<sup>4</sup>Undo = zpět, funkce, která vrátí aplikaci, dokument nebo jiný objekt do stavu před poslední operací.



(a) Server

(b) Databáze

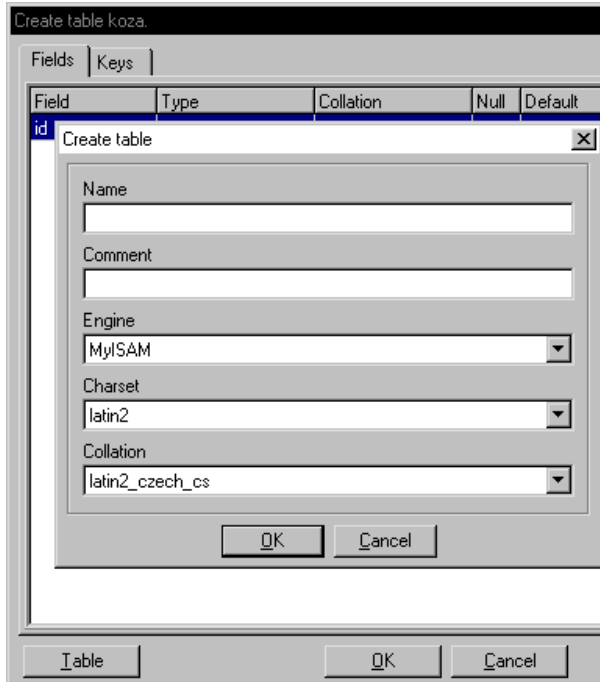
(c) Tabulka

Obrázek A.7: Kontextové menu stromu objektů

#### A.6.1.1 Vytvoření nové tabulky

Tabulku můžete **vytvořit** zvolením položky *Create table*. Objeví se dva dialogy, ty vidíte na obrázku A.8. Na vrchu je dialog vlastností tabulky. Zde vyplníte název tabulky, komentář. Výběr uložště (Engine) může být pro nezkušeného uživatele nejistý. Zde doporučujeme nahlédnout do manuálu MySQL k příslušné verzi serveru, ten je dostupný na <http://dev.mysql.com/doc/>. Pro většinu běžných případů je vhodné zvolit výchozí MyISAM. Zbývá jen nastavit znakovou sadu a porovnání pro tabulku. Ty budou implicitně používat její textové sloupečky, i když i u nich se dá explicitně nastavit něco jiného.

Po odsouhlasení tohoto dialogu se dostanete do dalšího dialogu pro vytvoření struktury, to znamená sloupečků a klíčů. Dialog vlastností můžete opětovně vyvolat stisknutím tlačítka *Table* vlevo dole. Podrobný popis práce se strukturou je v následující kapitole A.6.1.2.



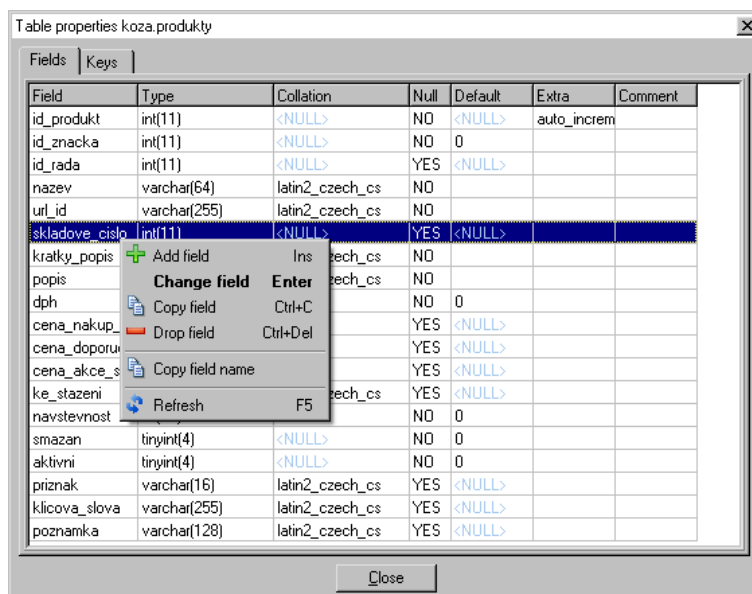
Obrázek A.8: Dialog pro vytvoření nové tabulky

#### A.6.1.2 Úprava struktury tabulky

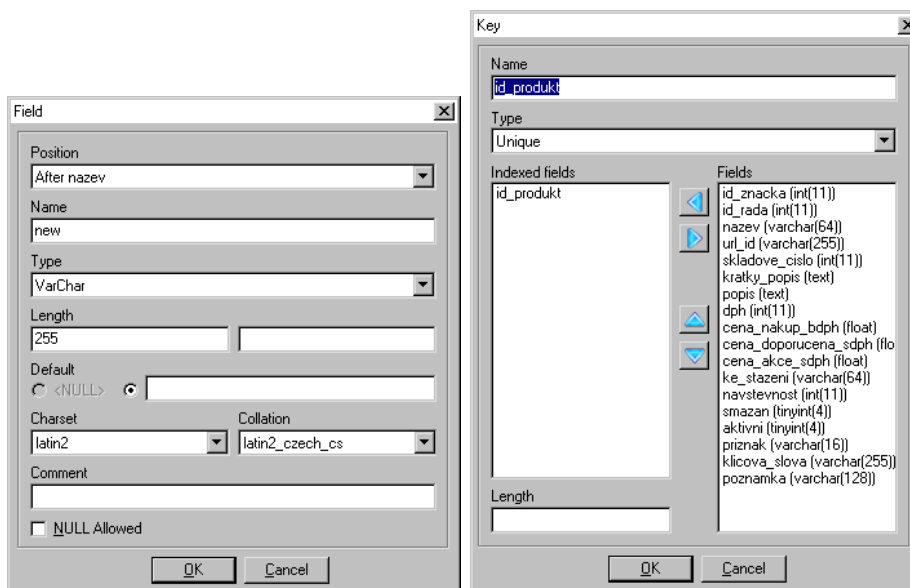
Úpravou struktury tabulky máme na mysli **správu sloupečků a klíčů** (indexů). Sem se dostanete zvolením položky *Properties* z kontextového menu tabulky. Zde dochází k malé kolizi české a anglické terminologie. V samotném

programu má položka *Properties* význam právě struktury tabulky, zatímco úprava vlastností tabulky je pod heslem *Alter table*.

Zobrazí se dialog *Table properties jméno tabulky*. Ten má dvě záložky, sloupečky a klíče.



(a) Sloupečky



(b) Sloupeček

(c) Klíč

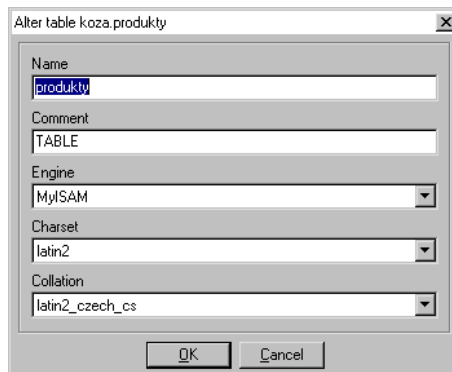
Obrázek A.9: Struktura tabulky

**Sloupečky** V okně je seznam všech sloupečků se zobrazenými důležitými vlastnostmi (A.9(a)). Jednotlivé operace provádí pomocí kontextového menu ke každé položce. Operace obnovit (*Refresh*) znovu načte celý seznam z databáze. *Copy field* má stejný význam jako *Add field*, jen do formuláře pro sloupeček (A.9(c)) předvyplní hodnoty podle zvolené položky. Tento formulář se mění podle zvoleného datového typu. Nabídka datových typů a jejich možnosti jsou vzaty z verze MySQL 5.1, proto se může stát, že u editace nižší verze server zahlásí chybu. Aplikace je postavená tak, aby taková chyba nevedla ke ztrátě informací.

**Klíče** Pro správu klíčů platí stejná pravidla jako pro sloupce.

### A.6.1.3 Úprava tabulky

Úpravou tabulky máme na mysli změny **vlastností tabulky** jako objektu. V menu je to položka *Alter table*. Zobrazí se nám podobný dialog jako u vytváření nové tabulky. Můžete zde měnit jméno tabulky, komentář, uložení a znakovou sadu s porovnáním. Odsouhlasení dialogu může vést až ke třem různým dotazům serveru podle toho, co jste přesně upravili. Tento formulář, podobně jako formulář pro úpravu databáze, se mění podle verze MySQL. Zde uvedené ukázky odpovídají verzi 4.1 a vyšší.



Obrázek A.10: Úprava tabulky

### A.6.2 Data

Správa dat a navigace je navržena tak, aby byla pokud možno co nejpohodlnější a zároveň efektivní. Po poklepání na jméno tabulky ve stromě objektů nebo v seznamu tabulek se obsah tabulky zobrazí v hlavní části okna.

Samotná tabulka dat spojuje pohodlí přímé editace hodnot a sílu tabulkového výběru.

#### A.6.2.1 Editace hodnot

Každá buňka tabulky je editovatelná. Dovnitř se dostanete poklepáním myši, nebo s kurzorem na dané buňce a stiskem klávesy <Enter> nebo <F2>. Každá z těchto metod vkročí do buňky trochu jiným způsobem. Pokud buňka obsahuje delší text, můžete jej editovat ve zvláštním textovém okně. Stiskem <F4> nebo zvolením *Edit cell* z kontextového menu zobrazíte velké textové okno. Textové pole umí pro větší přehlednost zalamovat řádky textu.

Bez komunikace se serverem můžete vždy editovat jeden celý řádek a změny se pošlou databázi automaticky opuštěním daného řádku, tedy například přesunem kurzoru na jiný řádek nebo přepnutím do jiného okna. Ručně můžete potvrdit změny stiskem <F9> nebo zvolením *Operations - Use changes* z hlavního menu, případně tlačítkem na nástrojové liště. Změny můžete i zrušit, ale jen dokud se neposlaly serveru. Zruší se stiskem <Ctrl+Alt+Shift+F9> nebo opět z menu nebo nástrojové lišty. Zde platí také pravidlo, že neexistuje žádná funkce zpět. Jakkmile se změna dat odešle databázi, jediná cesta zpět je ze zálohy.

Občas potřebujete vložit NULL hodnotu. To provedete stiskem <Shift+Space> na zvolené buňce nebo výběrem *Insert null* z kontextového menu, hlavního menu *Edit* nebo z tlačítkové lišty.

Obnova všech dat je <F5> nebo volba z menu. Micka nemá vestavěnou kontrolu datových typů ani žádnou jinou kontrolu vstupu. Kvůli rychlosti se ani po každé úpravě hodnot znovu nenačítá skutečný řádek. Může se tedy stát, že v programu uvidíte něco jiného, než leží na serveru. Nestává se to často, ale pro kontrolu je vhodné obnovit data. Při obnově dat si tabulka pamatuje stav zobrazení a pozici kurzoru, proto není třeba mít obavy ze ztráty pečlivě nalezené polohy v tabulce.

#### A.6.2.2 Označování a kopírování hodnot

Z tabulky se dá kopírovat jedna nebo více označených buněk do schránky <Ctrl+C>. Pokud je to výběr více buněk, v textové schránce jsou sloupce odděleny znakem tabulátor (#9, \t) a řádky znakem odřádkování (#13, \n).

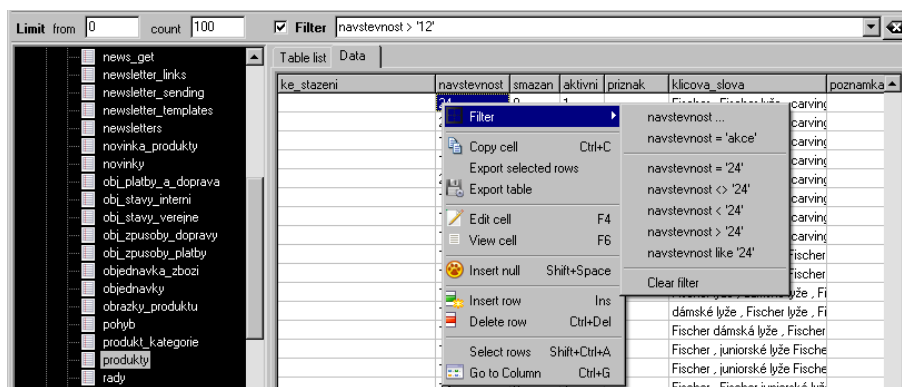
Obdélníkový výběr se provádí tažením myši přes tabulku. Pokud chcete označit celé řádky, ale museli byste skrolovat daleko doprava, stačí označit jen jeden sloupec přes požadované řádky a z kontextového menu zvolit *Select rows* <Shift+Ctrl+A>. Označit celé řádky lze také tažením myši v prázdné oblasti úplně vpravo mezi koncem tabulky a scrollbarem.

### A.6.2.3 Operace s řádky

Řádky, jinými slovy záznamy v tabulce můžete přidávat <Ins> a odstraňovat <Ctrl+Del>. Tyto příkazy jsou dostupné také z kontextového menu, hlavního menu *Edit* nebo nástrojové lišty. Opět platí, že operace s řádky jsou nevratné.

### A.6.2.4 Zobrazení

Zobrazení dat v tabulce ovlivňuje několik faktorů.



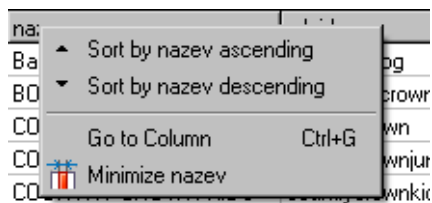
Obrázek A.11: Nastavení limitů a filtr

**Limity** Limity jsou dva a týkají se řádků tabulky. Limit Od říká, od kolikátého řádku se budou data zobrazovat. Řádky se počítají vždy od 0. Druhý limit říká, kolik řádků se zobrazí. Zpravidla je zbytečné zobrazovat celou tabulku, to bývá zdlouhavé a náročné na paměť. Skutečný celkový počet řádků se zobrazuje ve stavovém řádku. Chceme-li tedy zobrazit prvních 100 řádků, limity budou 0, 100. Dalších 100 řádků budou limity 100, 100. Limity se zadávají ve druhém řádku nástrojové lišty. Po změně limitu se musí obnovit zobrazení *Refresh* <F5>.

**Filtr** Filtr je množina podmínek dotazu. Podmínky se dají zadávat ručně do textového pole nad tabulkou. Celý vepsaný řádek se zařadí do dotazu serveru, musíte tedy sami kontrolovat správnou syntaxi. V případě chyby program vypíše chybovou hlášku zaslanou MySQL. Pro snazší zadávání podmínek má Micka speciální podmenu v kontextovém menu tabulky. Když na nějaké buňce vyvoláte tuto nabídku, automaticky se předvyplní položky tak, že je nabídka filtru vztažená ke konkrétnímu sloupečku a buňce. Stačí si pak vybrat jednu z možností a ta se připsá do pole filtru spolu s logickou spojkou and. Menu je vidět na obrázku A.11. Druhá položka je jméno sloupečku vztažené k obsahu schránky Windows. Vpravo od textového pole je tlačítko, které zruší filtr a obnoví zobrazení.

Aplikace si pamatuje historii všech filtrů od spuštění, jsou uvedeny v roletce. Také si pamatuje aktivní filtr pro každou tabulku. Pokud tedy přepínáte mezi více tabulkami nebo databázemi, můžete se bez úhony vrátit zpět k filtrované tabulce.

**Seřazení** Při prvním načtení tabulky od spuštění programu jsou data seřazena, tak jak je poslal server. Dají se pak seřadit vzestupně nebo sestupně podle kteréhokoli sloupečku a toto nastavení si program v rámci jednoho spuštění pamatuje pro každou tabulku. Neřadí se hned po jednom kliku na název sloupečku, jak je to běžné u jiných aplikací. Pokud totiž máte velmi dlouhou tabulku a chcete ji seřadit sestupně, musíte kliknout dvakrát a tedy řadit dvakrát a to trvá dlouho, zvláště pokud na daném sloupečku není index. Zde se po kliku levým tlačítkem myši na název



Obrázek A.12: Menu sloupečku

ze kterého si vyberete, jak chcete tabulku seřadit.

**Minimalizace sloupečků** Pro vyšší přehlednost v tabulce s mnoha sloupečky můžete některé, které vás nezajímají, minimalizovat. Příkaz *Minimize nabez sloupečku* najdete v menu sloupečku, viz obrázek A.12. Po přenačtení tabulky *Refresh* <F5> se opět všechny sloupečky obnoví do automatické velikosti.

#### A.6.2.5 Navigace

**Vyhledávání** V zobrazené tabulce lze vyhledávat fulltextově. Buď v aktuálním sloupci, řádku, nebo v celé tabulce po řádcích. Vyhledává se na částečnou shodu kdekoli v textu. Příkaz vyhledat najdete v hlavním menu *Search - Find* nebo standardní klávesová zkratka <Ctrl+F>. Samozřejmě nechybí funkce Najít další *Search - Find next* <F3>. Ale pozor, vyhledává se jen v aktuálním zobrazení bez komunikace se serverem.

**Jít na sloupeček** Další pomůcka pro rychlou navigaci je funkce Jít na sloupeček *Search - Go to column* <Ctrl+G>. Tato funkce je dostupná i z tlačítkové lišty a menu sloupečku z obrázku A.12. V levé části tabulky se zobrazí seznam sloupečků, kde můžete vyhledávat psaním prvních písmen názvu.

## A.7 Export a import

### A.7.0.6 Export

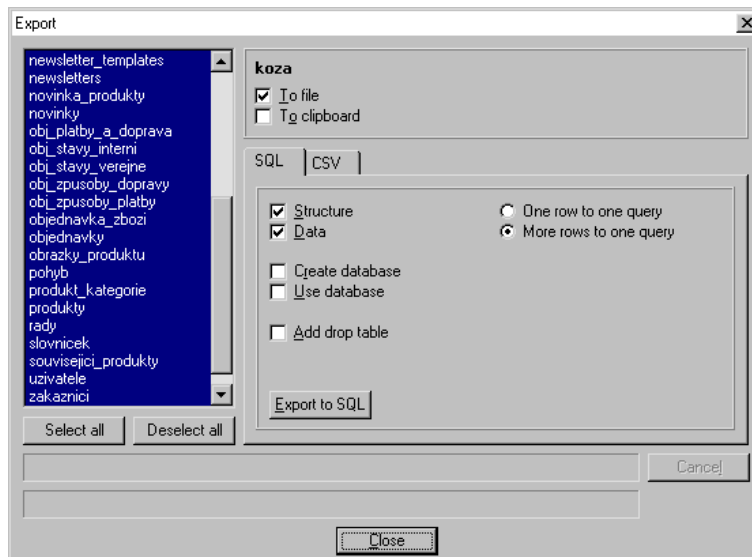
Program Micka má nástroj pro export do SQL (pro MySQL) a do CSV<sup>5</sup>. Exportovat lze několik vybraných tabulek nebo výběr z jedné tabulky. Příkaz pro export je v kontextovém menu databázových objektů a zobrazené tabulky. Exportovací dialog je okno se záložkami s nastaveními pro každý formát.

Výstup lze nasměrovat do souboru nebo do schránky. Export do schránky je výjimečná funkce, která se hodí například při rychlém přenesení výběru dat nebo jen struktury více tabulek zároveň do jiné databáze nebo jiné aplikace. Další uplatnění je při kopírování celé tabulky do tabulkového procesoru, například Excelu nebo Calcu. Nemusíte ukládat soubor a pak jej opět otevírat.

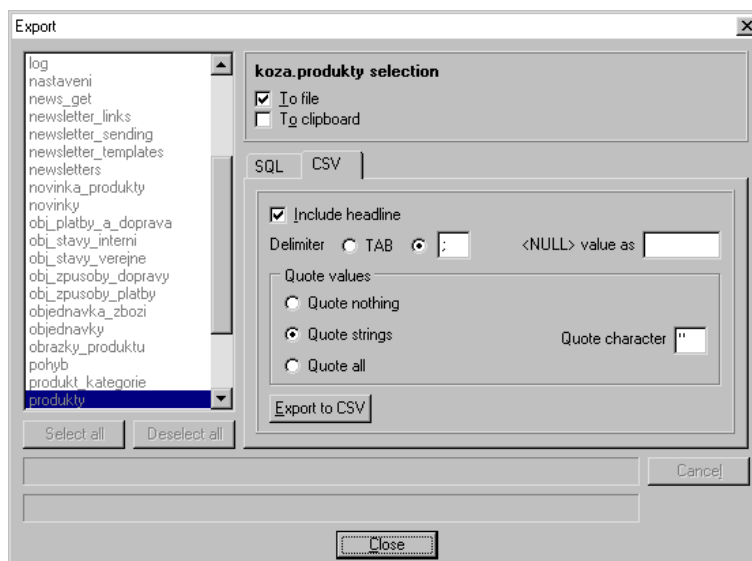
Při exportu do SQL si můžete zvolit, zda generovat pro každý záznam v tabulce samostatný insert<sup>6</sup> nebo zařadit více záznamů do jednoho insertu. Druhá varianta je úspornější při ukládání výsledného textového souboru a hlavně při následujícím importu minimalizuje počet dotazů a tím jej výrazně urychlí. Určitou malou nevýhodou může být složitější výběr jen některých řádků z exportu.

<sup>5</sup>Formát prostého textu, kdy jsou data oddělena libovolným znakem.

<sup>6</sup>Dotaz insert... slouží v SQL k přidávání do tabulky.



(a) SQL



(b) CSV

Obrázek A.13: Dialog pro export

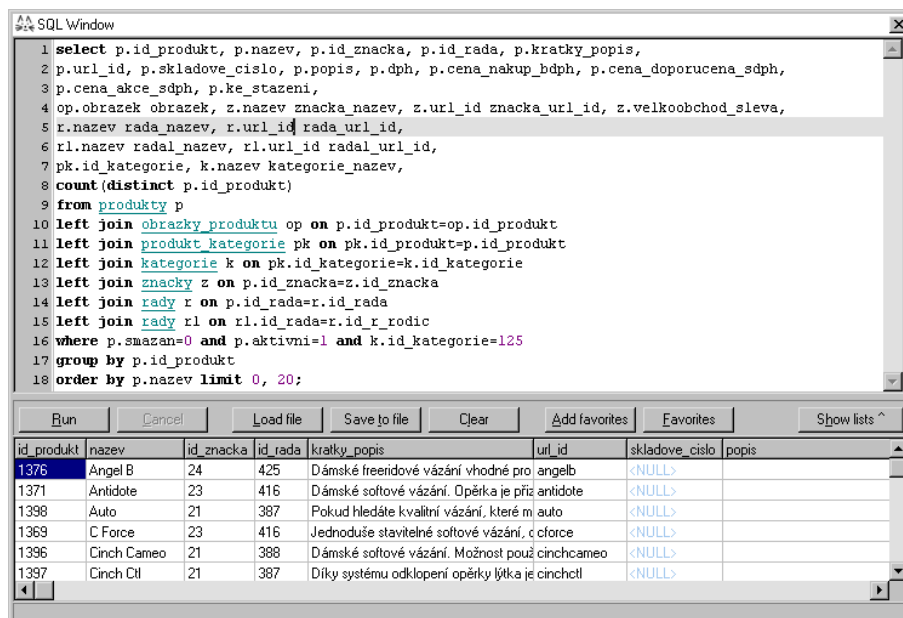
### A.7.0.7 Import

Micka nemá speciální nástroj pro import dat. Soubor ve formátu SQL lze jednoduše načíst do SQL okna a spustit. Více v následující kapitole A.8.

## A.8 SQL okno

*SQL window* je samostatně plovoucí okno, částečně napojené na hlavní okno. Není to samostatná aplikace, která by byla vidět na liště systému Windows, ale ani to není modální okno<sup>7</sup>. Je to zkratka samostatné okno. Micka si stejně jako u hlavního okna pamatuje jeho pozici a rozvržení. Dá se vyvolat z menu *Window - SQL window <Ctrl+K>*. Přepínat mezi hlavním a SQL oknem se dá klávesou *<F12>*.

Okno je rozděleno na dvě základní části. V horní části je textové pole a v dolní prostor pro zobrazování výsledků. Do horní části lze zobrazit seznam tabulek z aktuální databáze a seznam sloupečků z aktuální tabulky pro snadné vkládání do textového pole. Seznamy se zobrazí tlačítkem *Show lists*.



Obrázek A.14: SQL okno

Do textové oblasti lze psát ručně jeden nebo více dotazů. Můžete sem načíst textový soubor. Celou oblast můžete uložit do souboru. Jednotlivé dotazy nebo sérii dotazů lze uložit do oblíbených a odtud je kdykoli v budoucnu používat.

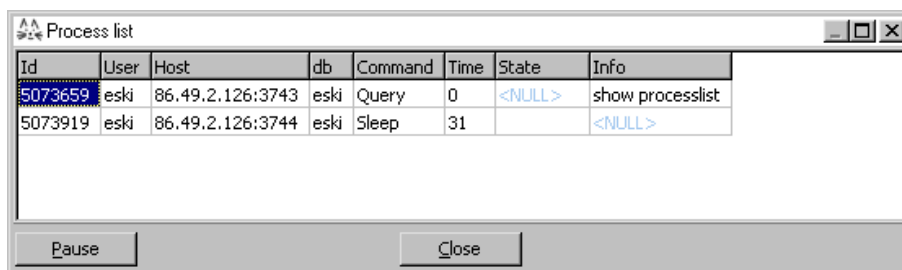
Po kliknutí na tlačítko *Run <F9>* se spustí celá série dotazů, které jsou zde napsané. Dole se zobrazí vždy výsledek posledního dotazu. Všechny vykonávané dotazy se zapisují do SQL logu v hlavním okně a informace o dotazu se také zobrazuje ve stavovém řádku hlavního okna, viz A.4.5 na straně 34.

Editor zvýrazňuje syntaxi SQL a názvy tabulek z aktuální databáze. Má funkci dokončování, takže buď automaticky nebo po stisku *<Ctrl+Space>* vám nabídne seznam tabulek a sloupečků.

<sup>7</sup>Když aplikace zobrazí modální okno, uživatel má přístup jen do tohoto okna a ne do celé aplikace. Jsou to všechny dialogy i třeba chybová hláška, kterou dokud neodklepnete, visí nad zastavenou aplikací

## A.9 Okno procesů

Tento nástroj zobrazuje jednoduchý seznam procesů, který se v pravidelných intervalech obnovuje. Můžete tak sledovat provoz na serveru. Funkce *Pause* umožňuje zastavit monitorování ve chvíli, která je zajímavá. Vyvolá se *Window - Process list*.



Obrázek A.15: Okno procesů

## A.10 Nastavení

Zvolením položky *Tools - Setting* z hlavního menu zobrazíte dialog nastavení. Ten má dvě záložky, MySQL a Prostředí (*Environment*).

Na záložce MySQL se nastavuje, zda v dotazech používat zpětné apostrofy. MySQL umožňuje tímto znakem uvozovat názvy databází, tabulek a sloupečků. Musí se používat u názvů, které neodpovídají SQL normě, například tabulka s názvem 2002. Obvykle není potřeba. Další nastavení se týká výstupní znakové sady. Před nastavením této položky je třeba se připojit k MySQL serveru, aby program mohl zjistit seznam dostupných znakových sad.

Záložku Prostředí není potřeba vysvětlovat.

## A.11 Aktualizace

Standardní distribuce Micky obsahuje i nástroj pro automatickou aktualizaci. Zvolíte-li v menu *Tools - Update Micka*, aplikace Micka se končí a spustí aktualizací program. Pokud ten zjistí, že je serveru dostupná novější verze, stáhne jí, nainstaluje a spustí.

## A.12 Seznam klávesových zkratk

### Globální

<F11> Připojit/Odpojit

<F12> Hlavní/SQL okno

<Alt+X> Zavřít

<Shift+Ctrl+F11> Nastavení

<Shift+Ctrl+F12> Nastavení připojení

### Hlavní okno

<F5> Obnovit objekty databáze/data

<Ctrl+K> SQL okno

- <Alt+Enter> Vlastnosti serveru/databáze/tabulky
- <Shift+Ctrl+Del> Odstranit tabulku
- <Ctrl+C> Kopírovat název objektu/výběr tabulky
- <F9> Potvrdit změny
- <Ctrl+Alt+Shift+F9> Zrušit změny
- <Ctrl+F> Najít
- <F3> Najít další
- <Ctrl+G> Jít na sloupeček
- <Ins> Vložit řádek
- <Ctrl+Del> Odstranit řádek
- <F4> Upravit buňku v textovém okně
- <F6> Zobrazit obsah buňky
- <Shift+Space> Vložit null hodnotu
- <Shift+Ctrl+A> Označit řádky

#### **SQL okno**

- <F9> Spustit dotazy
- <Ctrl+S> Uložit do souboru
- <Ctrl+Space> Nabídka automatického dokončení (okno editoru)

#### **Ve všech textových oknech fungují standardní klávesové zkratky Windows**

- <Ctrl+C>, <Ctrl+Ins> Kopírovat
- <Ctrl+V>, <Shift+Ins> Vložit
- <Ctrl+X> Vyjmout
- <Ctrl+A> Označit vše

## B Deklarace objektu TSenMySQL

```

TSenMySQL= object
private
  MySQLClient: TMySQLClient;
public
  ActualDatabase: string;
  ActualTable: TTableInfo;          // table, which is open
  ServerVersion: integer; // 40002 (4.0.2), 41, 50, 50102 (5.1.2), ...

  procedure Create; // constructor
  function ShowError(Error: string; EnableCancelBtn: boolean): integer;
  function Connected: boolean;
  function Connect(Hostname: string; Username: string; Password: string;
    Database: string; Timeout, Port: integer): boolean;
  procedure Disconnect;
  function SendQuery(Query1: AnsiString; var sg: TStringGrid;
    EnableCancelBtn: boolean; GetData: boolean= TRUE;
    OnlyRows: TRowsSet= []): integer; // if data then 0, mrResult from error dialog
  function SendEasyQuery(Query1: AnsiString): TArrayS; // return first col
  function SendEasyQueryOne(Query1: AnsiString): string; // return first cell
  procedure LoadDatabaseTree(DatabaseName, Table: string; var Tree: TTreeView;
    NoExpand: boolean= FALSE);
  function LoadTableData(DatabaseName, Table: string; var sg: TStringGrid;
    OrderBy: string; Descending: integer; Filter: string;
    LimitFrom, LimitCount: integer): integer; // if data then 0, else 1
  procedure UseDatabase(DatabaseName: string);
  function ConnectionProperties(var sg1, sg2: TStringGrid): integer; // like sendquery
  function DatabaseProperties(DatabaseName: string;
    var sg1, sg2: TStringGrid): integer; // like sendquery
  function TableProperties(DatabaseName: string; TableName: string;
    var sg1, sg2: TStringGrid): integer; // like sendquery
  procedure UpdateCommand(DatabaseName, TableName, SetS, WhereS: string; Limit: integer);
  procedure DeleteCommand(DatabaseName, TableName, WhereS: string; Limit: integer);
  procedure InsertCommand(DatabaseName, TableName, SetS: string);
  function ShowCreateTable(DatabaseName, TableName: string): string;
  function ShowCreateDatabase(DatabaseName: string): string;
  procedure DropTable(DatabaseName1, TableName1: string);
  procedure EmptyTable(DatabaseName1, TableName1: string);
  procedure RenameTable(DatabaseName1, TableName1, NewName1: string);
  procedure CloseActualTable;
  function GetCollationList: TStrings;
  function GetCharsetList: TStrings;
  function GetEngines: TStrings;
  procedure CreateDatabase(Name, Charset, Collation: string);
  procedure AlterDatabase(Name, NewName, Charset, Collation: string);
  procedure DropDatabase(DatabaseName1: string);
  function LastError: string;

  function ExportToMySQL(Query1: AnsiString; TableName: string;

```

```
OneRowOneQuery: boolean; var Progressbar: TProgressbar): TStrings;  
function ExportToCSV(Query1: AnsiString; TableName: string;  
var Progressbar: TProgressbar;  
Headline: boolean; Delimiter: char; NullString: string;  
QuoteType: integer; QuoteChar: string): TStrings;  
end;
```

## C Obsah příloženého CD

Součástí diplomové práce je přiložené CD se samotnou aplikací, s jejími zdrojovými kódy a elektronickou verzí této zprávy. Obsah CD má následující strukturu:

**Micka\** - Adresář se spustitelnou verzí

**Micka\_src\** - Adresář s kompletními zdrojovými kódy, včetně použitých komponent

**dp\_micka.pdf** - Elektronická verze tohoto dokumentu