

SQL

relační databázový systém



Úvodní kurz jazyka SQL.

Tento studijní materiál nenahrazuje studentům originální manuál databázového systému.

Pro úspěšné absolvování kurzu a porozumění problematiky je nutné se seznámit s referenčními zdroji databáze.

Neproporcionální text v rámečcích obsahuje autentickou odezvu z konzole systému.

Použité logo a obchodní název aplikace © 1995-2007 MySQL AB, 2008 Sun Microsystems, Inc.

<http://www.mysql.com/>

základní předpoklady:

- na jednom stroji může běžet v jednom okamžiku několik instancí db serverů MySQL
- každá instance může obsahovat několik databází
- každá databáze může obsahovat několik tabulek, obsahující data

terminálový (konzolový) klient MySQL je aplikace **mysql.exe**
umístěná v adresáři **c:\mysql\bin**

připojení k databázi:

mysql -u root -h 127.0.0.1 -P 3306

příklad připojení k lokálně nainstalovanému serveru (127.0.0.1)
na jeho defaultní port (3306) jako správce databáze (root) a to vše
bez hesla ! ... bezpečnostní faux-pas, ale pro testování optimální s

příkazy v prostředí terminálového klienta MySQL

příkazy lze psát i na několik řádků dokud není ukončeno znakem ";"
sekvence \c na další řádce zruší vstup rozepsaného příkazu

vypiš databáze v systému

show databases;

přepni se do databáze ve které se nyní bude pracovat, zvol jí

use reko;

vypiš tabulky ve zvolené (aktuální) databázi

show tables;

vytvoření databáze "reko" (nová a prázdná)

create database reko;

likvidace databáze

drop database reko;

likvidace tabulky

drop table matrix;

vytvořit novou tabulku

```
create table osoby (  
    pk int auto_increment primary key,  
    jmeno char(25) not null,  
    prijmeni char(40) not null,  
    bydliste int  
    ) ENGINE=MyISAM DEFAULT CHARSET=cp1250;
```

tabulka "osoby" bude obsahovat sloupce (pk, jmeno, prijmeni, bydliste)

"pk" a "bydliste" bude číselný typ

"jmeno" a "prijmeni" bude řetězec znaků dlouhý 25 a 40 znaků

auto_increment - pro vložené řádky bez vyplnění této hodnoty bude hodnota automaticky vyplněna - automatické číslování, vzestupné

primary key - hodnoty musí být v rámci tabulky unikátní, neopakovatelné

ENGINE=MyISAM DEFAULT CHARSET=cp1250 - informace pro db o typu tabulky a použité znakové sadě pro písmena národních abeced

vložit záznam (datovou větu) do tabulky

```
insert into osoby (jmeno,prijmeni,bydliste) values  
("karel","capek",1);
```

do tabulky osoby vložit do zmíněných položek data, položka "pk" byla při vytváření tabulky nastavena jako **auto_increment** a tak se bude vyplňovat sama

vymazání datové věty (záznamu) z tabulky

```
delete from matrix;
```

vymaže celý obsah tabulky, ne však tabulku jako takovou !

```
delete from matrix where model=1;
```

mazání datových vět s podmínkou - **where**

- smaže záznam, kde datová položka model obsahuje hodnotu "1"

```
delete from matrix where ((model=1) and (prodejce > 5));
```

složená podmínka - výmaz se provede za předpokladu platnosti obou podmínek, závorky slouží pouze k zpřehlednění - nejsou povinné, ale nanejvýš vhodné je to takhle prostě více cool !

nejpoužívanější logické operátory: AND, OR, XOR

(nejen pro příkaz **delete**, používají se při **selectech** a pod.)

SELECT data mining - dolování dat z databáze

příkazy v následujících příkladech jsou funkční na následujících vzorových datech:

- příklad databáze aut v autobazaru

tabulka matrix

pk	spz	model	prodejce	kupující	cena_n	cena_p	datum_n	datum_p
1	cbj 44-45	1	1	NULL	10000	NULL	2007-02-14	NULL
2	cbp 74-15	2	2	NULL	15000	NULL	2007-02-14	NULL
3	cbr 47-45	1	3	NULL	20000	NULL	2007-03-18	NULL
4	cbh 54-35	4	3	NULL	32000	NULL	2007-08-18	NULL
5	cbk 45-25	7	3	NULL	13000	NULL	2007-05-05	NULL
6	cbe 49-75	9	3	NULL	5000	NULL	2007-04-07	NULL
7	cbj 22-11	4	3	NULL	8000	NULL	2007-04-04	NULL
8	cbm 41-27	3	4	NULL	50000	NULL	2007-07-22	NULL
9	cbj 57-25	4	1	NULL	68000	NULL	2007-04-14	NULL

tabulka modely

pk	vyrobce	model
1	skoda	105 s
2	skoda	120 s
3	skoda	125 l
4	skoda	136 l
5	skoda	110 r

tabulka zakaznik

pk	jmeno	prijmeni	bydliste
1	jan	novacek	a.michny 7 Budejovice
2	petr	blaha	a.michny 7 Budejovice
3	karel	malik	a.michny 7 Budejovice
4	ivan	kollar	a.michny 7 Budejovice
5	david	matulka	a.michny 7 Budejovice

```
select 1+1;
```

spočti 1+1 a vypiš výsledek

```
mysql> select 1+1;
+-----+
| 1+1 |
+-----+
|    2 |
+-----+
1 row in set (0.00 sec)
```

```
select password("Hello world");
```

```
select md5("Hello world");
```

zahesluj a zobraz požadovaný řetězec "Hello world"

```
mysql> select password("Hello world");
+-----+
| password("Hello world") |
+-----+
| *10682B6D54CE72511B299AE5A0C1EAAADEA71349 |
+-----+
1 row in set (0.00 sec)

mysql> select md5("Hello world");
+-----+
| md5("Hello world") |
+-----+
| 3e25960a79dbc69b674cd4ec67a72c62 |
+-----+
1 row in set (0.00 sec)
```

```
select * from matrix;
```

vypiš celou tabulku "matrix"

```
mysql> select * from matrix;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| pk | spz      | model | prodejce | kupující | cena_n | cena_p | datum_n | datum_p |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | cbj 44-45 | 1 | 1 | NULL | 10000 | NULL | 2007-02-14 | NULL |
| 2 | cbp 74-15 | 2 | 2 | NULL | 15000 | NULL | 2007-02-14 | NULL |
| 3 | cbr 47-45 | 1 | 3 | NULL | 20000 | NULL | 2007-03-18 | NULL |
| 4 | cbh 54-35 | 4 | 3 | NULL | 32000 | NULL | 2007-08-18 | NULL |
| 5 | cbk 45-25 | 7 | 3 | NULL | 13000 | NULL | 2007-05-05 | NULL |
| 6 | cbe 49-75 | 9 | 3 | NULL | 5000  | NULL | 2007-04-07 | NULL |
| 7 | cbj 22-11 | 4 | 3 | NULL | 8000  | NULL | 2007-04-04 | NULL |
| 8 | cbm 41-27 | 3 | 4 | NULL | 50000 | NULL | 2007-07-22 | NULL |
| 9 | cbj 57-25 | 4 | 1 | NULL | 68000 | NULL | 2007-04-14 | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
9 rows in set (0.00 sec)
```

```
select count(*) from matrix;
```

vypiš počet datových vět (záznamů) v tabulce

```
mysql> select count(*) from matrix;
+-----+
| count(*) |
+-----+
|          9 |
+-----+
1 row in set (0.00 sec)
```

```
select pk, prodejce from matrix;
```

vypiš jen požadované sloupce (**pk**, **prodejce**) z tabulky "matrix"

```
mysql> select pk,prodejce from matrix ;
```

```
+-----+
| pk | prodejce |
+-----+
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 3 |
| 5 | 3 |
| 6 | 3 |
| 7 | 3 |
| 8 | 4 |
| 9 | 1 |
+-----+
```

```
9 rows in set (0.00 sec)
```

```
select * from modely where model like "%20_s";
```

výběr dat, které splňují podmínku **where**

použití wildcards (zástupné znaky) - v podmínce musí být **like**

- znak "%" nahrazuje řetězec znaků

- znak "_" nahrazuje jediný znak

pokud potřebujeme vyhledávat přímo tyto znaky (" _ %) použijeme před nimi backslash "\" **like "%20"** platí pro %20

```
mysql> select * from modely where model like "%20_s";
```

```
+-----+
| pk | vyrobce | model |
+-----+
| 2 | skoda | 120 s |
+-----+
```

```
1 row in set (0.00 sec)
```

operátory v podmínce WHERE - pro porovnání číselných položek

= , > , < , >= , <= , <> , is null

```
select distinct model from matrix;
```

výstupem jsou pouze rozdílné hodnoty ... za každou hodnotu zobrazí pouze jednoho zástupce

```
mysql> select distinct model from matrix;
```

```
+-----+
| model |
+-----+
| 1 |
| 2 |
| 4 |
| 7 |
| 9 |
| 3 |
+-----+
```

```
6 rows in set (0.00 sec)
```

```
select matrix.pk, matrix.spz, zakaznik.prijmeni "prodejce"
from matrix, zakaznik
where matrix.prodejce=zakaznik.pk;
```

příklad joinu - vypiš požadované sloupcečky z požadovaných tabulek

použit tzv.tečkovaný tvar - tabulka.záznam

nadpis pro sloupeček **zakaznik.prijmeni** bude nadepsán "prodejce"

sloupec **matrix.prodejce** obsahuje čísla, které mají být v tomto selectu nahrazeny záznamy z tabulky "zakaznik" ... **zakaznik.prijmeni**
- je vytvořeno logické propojení **matrix.prodejce=zakaznik.pk**
- propojení je platné jen pro okamžik výpisu, pro další select může být omylem napojen **matrix.prodejce** na typy aut ... je to sice logická chyba, ale jen logická, bude to tedy fungovat a provede se výpis - pro databázi zcela korektní záležitost, db lidské blbosti nerozumí !

```
mysql> select matrix.pk, matrix.spz, zakaznik.prijmeni "prodejce"
from matrix, zakaznik
where matrix.prodejce=zakaznik.pk;
```

pk	spz	prodejce
1	cbj 44-45	novacek
9	cbj 57-25	novacek
2	cbp 74-15	blaha
3	cbr 47-45	malik
4	cbh 54-35	malik
5	cbk 45-25	malik
6	cbe 49-75	malik
7	cbj 22-11	malik
8	cbm 41-27	kollar

9 rows in set (0.00 sec)

```
select matrix.pk, matrix.spz, modely.model "model"
from matrix, modely
where matrix.model=modely.pk;
```

ještě jednou podobný select

```
mysql> select matrix.pk, matrix.spz, modely.model "model"
from matrix, modely
where matrix.model=modely.pk;
```

pk	spz	model
1	cbj 44-45	105 s
3	cbr 47-45	105 s
2	cbp 74-15	120 s
8	cbm 41-27	125 l
4	cbh 54-35	136 l
7	cbj 22-11	136 l
9	cbj 57-25	136 l

7 rows in set (0.00 sec)

..... kam se ale poděl záznam matrix.pk 5 a 6 ???

- není vypsán, protože **matrix.model** který nemá odpovídající protějšek mezi záznamy **modely.pk** se nepropojí a nevypíše


```
select matrix.pk, matrix.spz, modely.model "model"
from matrix left join (modely)
on matrix.model=modely.pk;
```

provedení joinu s **přesahem** - vypíše se i záznamy, které nemají v druhé tabulce "svě protějšky" - tyto záznamy mají v propojovaných datech prázdný datový záznam - hodnota NULL (datové vakuum)

```
mysql> select matrix.pk, matrix.spz, modely.model "model"
from matrix
left join (modely)
on matrix.model=modely.pk;
```

pk	spz	model
1	cbj 44-45	105 s
2	cbp 74-15	120 s
3	cbr 47-45	105 s
4	cbh 54-35	136 l
5	cbk 45-25	NULL
6	cbe 49-75	NULL
7	cbj 22-11	136 l
8	cbm 41-27	125 l
9	cbj 57-25	136 l

9 rows in set (0.00 sec)

```
select matrix.pk, matrix.spz, modely.model "model"
from matrix left join (modely)
on matrix.model=modely.pk
where modely.model is null;
```

vypíše se jen záznamy bez svých protějšků - kde při spojení vzniklo vakuum, podmínka **is null** - "**= null**" nebude fungovat

```
mysql> select matrix.pk, matrix.spz, modely.model "model"
from matrix
left join (modely)
on matrix.model=modely.pk
where modely.model is null;
```

pk	spz	model
5	cbk 45-25	NULL
6	cbe 49-75	NULL

2 rows in set (0.00 sec)

```
select matrix.pk, matrix.spz
  from matrix
 where matrix.model in (select modely.pk from modely)
```

hledání záznamů, které nemají své protějšky v jiné tabulce
tentokrát verze s **vnořeným selectem**
- pokud není záznam v seznamu **in** (vygenerovaným vnořeným selectem)
nebude vypsan

je to možná přehlednější zápis než s "joinem s přesahem", ale pro db
mnohem pomalejší ... pokud zpracováváme 1000 a více záznamů

```
mysql> select matrix.pk, matrix.spz from matrix where matrix.model in (select modely.pk from modely);
+-----+-----+
| pk | spz      |
+-----+-----+
| 1 | cbj 44-45 |
| 2 | cbp 74-15 |
| 3 | cbr 47-45 |
| 4 | cbh 54-35 |
| 7 | cbj 22-11 |
| 8 | cbm 41-27 |
| 9 | cbj 57-25 |
+-----+-----+
7 rows in set (0.01 sec)
```

```
select matrix.pk, matrix.spz
  from matrix
 where matrix.model not in (select modely.pk from modely);
```

negace předchozí logiky

```
mysql> select matrix.pk, matrix.spz from matrix where matrix.model not in (select modely.pk from modely);
+-----+-----+
| pk | spz      |
+-----+-----+
| 5 | cbk 45-25 |
| 6 | cbe 49-75 |
+-----+-----+
2 rows in set (0.00 sec)
```

```
select pk, spz from matrix where pk in (1,3,5);
```

další příklad na IN - pracuj jen se záznamy uvedenými **v seznamu**

```
mysql> select pk, spz from matrix where pk in (1,3,5);
+-----+-----+
| pk | spz      |
+-----+-----+
| 1 | cbj 44-45 |
| 3 | cbr 47-45 |
| 5 | cbk 45-25 |
+-----+-----+
3 rows in set (0.00 sec)
```

```
select pk, spz from matrix where pk between 2 and 5;
```

vymezení podmínky rozsahem hodnot - datové věty musí mít záznam v datové položce **pk** v rozsahu mezi 2 - 5

```
mysql> select pk, spz from matrix where pk between 2 and 5;
```

pk	spz
2	cbp 74-15
3	cbr 47-45
4	cbh 54-35
5	cbk 45-25

```
4 rows in set (0.03 sec)
```

```
select prodejce, count(*) from matrix group by prodejce;
```

vypiš různé prodejce a počet záznamů, kde se vyskytují

```
mysql> select prodejce, count(*) from matrix group by prodejce;
```

prodejce	count(*)
1	2
2	1
3	5
4	1

```
4 rows in set (0.00 sec)
```

```
select prodejce, count(*)  
  from matrix  
  group by prodejce  
  having count(*) >= 2;
```

vypiš různé prodejce a počet záznamů, kde se vyskytují, ale jen ty prodejce kteří se vyskytují 2 a vícekrát

```
mysql> select prodejce, count(*) from matrix group by prodejce having count(*) >= 2;
```

prodejce	count(*)
1	2
3	5

```
2 rows in set (0.00 sec)
```

obsluha databáze

naplnění tabulky daty z externího souboru

soubor ve formátu CSV - comma separated values, data oddělené čárkou

```
***** file: data.csv *****
10;porsche;l15 z;
11;porsche;l2 b;
12;mercedes;a 2200;
13;mercedes;a 2100;
14;mercedes;c 6255;
15;mercedes;d 6565;
```

```
*****
```

```
load data local
infile "e:/data.csv"
into table modely fields
terminated by ";;";
```

```
mysql> load data local
      infile "e:/data.csv"
      into table modely
      fields terminated by ";;";

Query OK, 6 rows affected, 6 warnings (0.00 sec)
Records: 6  Deleted: 0  Skipped: 0  Warnings: 6
```

parametr **"local"** - data jsou na disku klienta, jinak se soubor hledá na serveru

data v souboru musí na každém řádku končit příslušným znakem, např. středníkem ";", při exportu dat z excelu tento znak většinou chybí a musí se ručně do textového souboru doplnit

poslední záznam v datovém souboru by měl být (prázdná řádka), při více prázdných řádkách jsou vkládány do db nulové záznamy

naplnění tabulky daty ze selectu

novou tabulku **tab2** lze vytvořit jako stín stávající tabulky **matrix**
create table tab2 like matrix;

naplnění daty pomocí selectu nad jinou tabulkou
insert into mx select * from matrix where pk is not null;

```
mysql> insert into mx select * from matrix where pk is not null;
Query OK, 9 rows affected (0.09 sec)
Records: 9  Duplicates: 0  Warnings: 0
```

transakce

transakce lze použít u tabulek, které mají obsluhu innodb enginem

show table status;

vypíše vlastnosti tabulek z aktuální databáze

alter table matrix engine = innodb;

alter table matrix engine = myisam;

pro tabulku "matrix" přepne engine (mechanismus obsluhy) db stroje

jako výchozí je nastaven při startu nové session autocommit, ten je potřeba pro transakční zpracování vypnout a nastartovat transakční mechanismus

set autocommit=0;

start transaction;

nyní jsou innodb tabulky schopny transakčního zpracování (ne všechny operace podléhají transakčnímu přístupu, např. drop table nelze vrátit)

výsledek příkazu insert nebo delete lze vrátit příkazem

rollback;

nebo potvrdit příkazem

commit;

pro jemnější ošetření lze vkládat v průběhu zpracování transakce záložky, ke kterým se lze vrátit (pokud se vrátíme k nějaké záložce, mladší záložky se zruší)

vlož záložku, vrať se k záložce, smaž záložku

savepoint stav_a;

rollback to savepoint stav_a;

release savepoint stav_a;

triggery

hlídací pes v databázi, číhající na definovaný povel

```
delimiter //
create trigger tr_01 before insert on modely for each row
begin
    if new.pk < 4 then
        set new.pk = new.pk + 10;
    elseif new.pk > 4 then
        set new.pk = new.pk + 20;
    end if;
end; //
delimiter ;
```

delimiter // - povel bloku, středníky nejsou brány (až do odvolání)
za ukončení příkazu

// blok ukončen

delimiter; - zrušení sekvence // jako delimiteru

tr_01 - název triggeru

before / **after** - doba zapůsobení triggeru

insert / **update** / **delete** - hlídaná operace, povel pro hlídacího psa

new.položka / **old.položka** - položka po / před hlídanou operací

drop trigger reko.tr_01;
smazání triggeru tr_01 v databázi reko

show triggers;
zobrazení aktivních triggerů v databázi

```
mysql> show triggers;
+-----+-----+-----+-----+-----+-----+-----+-----+
| Trigger | Event | Table | Statement | Timing | Created | sql_mode | Definer |
+-----+-----+-----+-----+-----+-----+-----+-----+
| tr_01   | INSERT | modely | begin
  if new.pk < 4 then
    set new.pk = new.pk + 10;
  elseif new.pk > 4 then
    set new.pk = new.pk + 20;
  end if;
end | BEFORE | NULL | | root@localhost |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.02 sec)
```

bezpečnostní politika systému

založení nového uživatele - **jan** s heslem **hello**
create user jan identified by 'hello';

heslo je hashováno funkcí `password()`

smazání uživatele
drop user jan;

pro okamžitou akceptaci nových práv musí být proveden příkaz
flush privileges;

přidělení práv pro uživatele **jan**
grant all on reko.* to jan;
přidělí se všechna práva - **all**
pro všechny objekty databáze reko (**reko.***)

přidělení práv jen pro jednu konkrétní tabulku
grant all on reko.matrix to jan;

odebrání práv pro **mazání** datových vět pro uživatele **jan**
revoke delete on reko.matrix from jan;

správa databáze pomocí utility MySQL Admin

výpis procesů na serveru
mysqladmin -u root processlist
(trochu detailnější je parametr **proc stat**)

```
C:\MySQL\bin>mysqladmin -u root processlist
```

Id	User	Host	db	Command	Time	State	Info
1	root	localhost:1087	mysql	Sleep	114		
4	root	localhost:1098	reko	Sleep	45		
19	jan	localhost:1121	reko	Sleep	553		
32	root	localhost:1134		Query	0		show processlist

ukončení procesu prostřednictvím jeho identifikátoru ID
mysqladmin -u root kill 19

vzorová data - databáze "autobazar"

SQL soubor s příkazy **reko.sql** je spuštěn jako dávka příkazem DOSu
c:\mysql\bin\mysql -h 127.0.0.1 -u root -t -vvv< reko.sql >reko.out.txt
Dosový příkaz je efektivní uložit do do dávkového souboru BAT a
spouštět jím dávku SQL příkazů. Výstup SQL dávky je přesměrován z
obrazovky do souboru **reko.out.txt**

```
***** file: reko.sql *****
# database creation

create database if not exists reko;
drop database reko;
create database reko;
use reko;

# table constructor
create table matrix (
  pk int auto_increment primary key,
  spz char(10),
  model int not null,
  prodejce int not null,
  kupujici int,
  cena_n int,
  cena_p int,
  datum_n date,
  datum_p date
) ENGINE=MyISAM DEFAULT CHARSET=cp1250;

create table zakaznik (
  pk int primary key,
  jmeno char(25) not null,
  prijmeni char(40) not null,
  bydliste char(40)
) ENGINE=MyISAM DEFAULT CHARSET=cp1250;

create table modely (
  pk int primary key,
  vyrobce char(25) not null,
  model char(40) not null
) ENGINE=MyISAM DEFAULT CHARSET=cp1250;

# test-data import

insert into matrix (spz,model,prodejce,cena_n,datum_n) values ("cbj 44-45",1,1,10000,"2007-02-14");
insert into matrix (spz,model,prodejce,cena_n,datum_n) values ("cbp 74-15",2,2,15000,"2007-02-14");
insert into matrix (spz,model,prodejce,cena_n,datum_n) values ("cbr 47-45",1,3,20000,"2007-03-18");
insert into matrix (spz,model,prodejce,cena_n,datum_n) values ("cbh 54-35",4,3,32000,"2007-08-18");
insert into matrix (spz,model,prodejce,cena_n,datum_n) values ("cbk 45-25",7,3,13000,"2007-5-5");
insert into matrix (spz,model,prodejce,cena_n,datum_n) values ("cbe 49-75",9,3,5000,"2007-4-7");
insert into matrix (spz,model,prodejce,cena_n,datum_n) values ("cbj 22-11",4,3,8000,"2007-4-4");
insert into matrix (spz,model,prodejce,cena_n,datum_n) values ("cbm 41-27",3,4,50000,"2007-7-22");
insert into matrix (spz,model,prodejce,cena_n,datum_n) values ("cbj 57-25",4,1,68000,"2007-4-14");

insert into zakaznik (pk,jmeno,prijmeni,bydliste) values (1,"jan","novacek","a.michny 7 Budejovice");
insert into zakaznik (pk,jmeno,prijmeni,bydliste) values (2,"petr","blaha","a.michny 7 Budejovice");
insert into zakaznik (pk,jmeno,prijmeni,bydliste) values (3,"karel","malik","a.michny 7 Budejovice");
insert into zakaznik (pk,jmeno,prijmeni,bydliste) values (4,"ivan","kollar","a.michny 7 Budejovice");
insert into zakaznik (pk,jmeno,prijmeni,bydliste) values (5,"david","matulka","a.michny 7 Budejovice");

insert into modely (pk,vyrobce, model) values (1,"skoda","105 s");
insert into modely (pk,vyrobce, model) values (2,"skoda","120 s");
insert into modely (pk,vyrobce, model) values (3,"skoda","125 l");
insert into modely (pk,vyrobce, model) values (4,"skoda","136 l");
insert into modely (pk,vyrobce, model) values (5,"skoda","110 r");
*****
```


ovládání serveru z příkazové řádky

console start

C:\MySQL\bin\mysqld-nt --defaults-file=C:\MySQL\my-opts1.cnf

console stop

C:\mysql\bin\mysqladmin --port=3306 -u root shutdown

service - install

**C:\MySQL\bin\mysqld-nt --install-manual MySQL3306
--defaults-file=c:\mysql\my-opts1.cnf**

service - uninstall

C:\MySQL\bin\mysqld-nt --remove MySQL3306

service start / stop

net start mysql3306

net stop mysql3306