

# Laboration 1, Databashantering med MySQL

Av: Marcus Rejås <[marcus@rejas.se](mailto:marcus@rejas.se)>

I denna laboration skall du se till att du kommer åt databasmotorn och att det fungerar. Du kommer också att skapa en tabell i din databas och mata in några poster i den.

Denna laboration skapades för elever vid Roslagens Högskola men kan användas av vem som helst. Namnen på serverna måste i så fall ändras.

Kapitel 7 i Björk<sup>1</sup> är en bra hjälp.

## Installera PuTTY

Ifall du inte sedan förr har PuTTY installerat så installera denna och gör dig hemmastadd i den nu. Hur man gör står i ett separat blad som du fick samtidigt som detta.

## Logga in på `ssh.rhs.se`

Hur man gör detta står också i dokumentet om PuTTY, när du loggat in skall du på sista raden se en **kommandoprompt** som ser ut ungefär så här:

```
rejas@ssh:~$
```

Det är vid denna prompt som du ger kommandon till servern. Du kan prova kommandot **"ls"** som betyder "lista filer", varje kommando körs när du trycker <Enter> (Du skriver det som är fet-stil nedan, resten skriver systemet).

```
rejas@ssh:~$ ls
Mail bin data htaccess www wwwdata
rejas@ssh:~$
```

Vill du se aktuell tid ger du kommandot **"date"**:

```
rejas@ssh:~$ date
Wed Feb 11 19:03:53 CET 2004
rejas@ssh:~$
```

För att skriva ett kommando du nyss skrivit kan du använda historikfunktionen, den fungerar så att du kan bläddra mellan tidigare skrivna kommandon med hjälp av upp- och ner-piltangenterna. Prova att trycka pil upp och tryck <Enter> när kommandot **"ls"** visas igen.

## Starta databasklienten

För att komma åt databasen måste man använda ett program som kallas för en databasklient. Det finns flera olika sådana och vi kommer längre fram i kursen att ansluta till databasen med PHP men tills vidare skall vi använda den klient som följer med MySQL. Denna databasklient heter **"mysql"** och kan startas med en mängd olika argument. De argument vi skall använda ser ut så här: **"mysql -h obelix.rhs.se -p rejas"**.

**"-h obelix.rhs.se"** betyder att vi skall använda databasservern **"obelix.rhs.se"**, (**-h** står för host som betyder värd, eller dator).

**"-p"** betyder att vi skall ange lösenordet när vi loggar in

**"rejas"** är den databas som vi skall använda, det **skall vara ditt inloggningsnamn**.

Så här kan det se ut:

---

1 <http://www.jonashjork.net/databas/>

```
rejas@ssh:~$ mysql -h obelix.rhs.se -p rejas
Enter password:

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 102 to server version: 4.0.16-log

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Som du ser så står du nu vid en ny kommandoprompt som ser ut så här:

```
mysql>
```

Vid denna skriver du **kommandon till databasen**. Jämför med prompten innan som var för att ge kommandon till systemet. Även vid denna prompt har du samma historikfunktion som du hade vid systemprompten, men här bläddrar du mellan kommandona till databasen. Vi testar lite kommandon så att vi ser att det fungerar. Kommandot "SHOW TABLES" visar alla tabeller i databasen. Alla kommandon till databasen avslutas med semikolon ";" (det behöver de inte göra vid systemprompten).

```
mysql> SHOW TABLES;
Empty set (0.00 sec)

mysql>
```

I detta fall fanns det inga tabeller att visa eftersom vi inte har skapat några ännu. Men vad hade hänt om vi glömt att skriva ett semikolon på slutet?

```
mysql> SHOW TABLES
->
-> ;
Empty set (0.00 sec)

mysql>
```

Nu när jag tryckte <Enter> efter "SHOW TABLES" visades en ny prompt. Denna prompt visar att jag är på en ny rad men fortfarande på samma kommando. Jag tryckte <Enter> två gånger, sedan skrev jag ett semikolon för att avsluta kommandot. Som du såg så gick det bra, det går att dela upp kommandon på flera rader om man vill bara avsluta dem med semikolon.

För att lämna databasklienten så ger du kommandot "\q". De kommandon som börjar med bakstreck "\ " är specialkommandon till databasklienten och dessa skall inte avslutas med semikolon.

```
mysql> \q
Bye
rejas@ssh:~$
```

Starta nu databasklienten igen och upprepa det vi gjort nu **minst en gång**. Detta skall du kunna göra i sömnen. Anteckna gärna och titta i den lilla **lathunden till MySQL** som du har fått.

## Skapa en tabell

Skapar en tabell gör man med kommandot CREATE TABLE vi börjar med att skapa en liten testtabell med ett fält som vi kallar för namn och som innehåller text (max 10 tecken). Syntaxen för kommandot CREATE TABLE ser i sin enklaste form ut så här:

```
mysql> CREATE TABLE test ( namn char(10) );
Query OK, 0 rows affected (0.01 sec)

mysql>
```

Vi skapar en tabell som heter "test" och som innehåller ett fält av typen char(10), för mer information om typer se **appendix a** i kursboken (Björk).

Vi kollar att tabellen verkligen skapades:

```
mysql> SHOW TABLES;
+-----+
| Tables_in_rejas |
+-----+
| test            |
+-----+
1 row in set (0.00 sec)

mysql>
```

Jodå, den verkar ju ha skapats. Vi kan titta närmare på fälten med kommandot "SHOW FIELDS FROM ..."

```
mysql> SHOW FIELDS FROM test;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| namn  | char(10)  | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.04 sec)

mysql>
```

## Mata in data i tabellen

Nu när vi har vår lilla tabell, vill vi stoppa in lite värden i den. De gör vi med kommandot "INSERT INTO"

```
mysql> INSERT INTO test (namn) VALUES ('kalle');
Query OK, 1 row affected (0.00 sec)

mysql>
```

Stoppa nu in några fler namn.

## Ställa frågor till databasen, välja ur en tabell

Att ställa frågor till en databas innebär att man gör vissa operationer på poster. Den enklaste frågan är att välja poster ur en tabell. Det gör man med kommandot "SELECT ... FROM ...:

```
mysql> SELECT * FROM test;
+-----+
| namn  |
+-----+
| kalle |
| kajsa |
| knatte |
| tjatte |
| fnatte |
+-----+
5 rows in set (0.00 sec)

mysql>
```

Tecknet \* betyder ungefär "allt", man kan läsa ut kommandot ovan som "VÄLJ ALLT FRÅN test"

## Radera poster från en tabell

Man kan radera poster från en tabell med kommandot **"DELETE FROM ..."**, observera att detta är ett destruktivt kommando som raderar saker. Det finns inget **"undo"**. Gör man en **"DELETE FROM tabell"** så kommer allt i tabellen att tas bort. Vi måste begränsa vad som skall tas bort med hjälp av **"WHERE"**. Se nedanstående exempel:

```
mysql> DELETE FROM test WHERE namn='fnatte';
Query OK, 1 row affected (0.80 sec)

mysql>
```

I exemplet ovan raderades raden där fältet namn innehåller **"fnatte"**. En WHERE-sats kan man använda tillsammans med de flesta andra satser. Till exempel **"SELECT"**. Som du ser så raderas posten utan minsta förvarning ovan. Ofta kan det vara bra att testa med ett oförstörande kommando som till exempel **"SELECT"** först.

```
mysql> SELECT * FROM test WHERE namn='tjatte';
+-----+
| namn   |
+-----+
| tjatte |
+-----+
1 row in set (0.00 sec)

mysql>
```

Som vi ser så verkar en post matcha **"WHERE namn='tjatte'"** så är det den vi vill radera är det bara att sätta igång:

```
mysql> DELETE FROM test WHERE namn='tjatte';
Query OK, 1 row affected (0.03 sec)

mysql>
```

Nu skall "tjatte" vara borta, vi kollar med **"SELECT"** igen:

```
mysql> SELECT * FROM test WHERE namn='tjatte';
Empty set (0.00 sec)

mysql>
```

Nu vet vi att "tjatte" är borta!

## Radera tabeller

Ibland vill man ta bort en hel tabell. Det kan man göra med **"DROP TABLE"**. Kommandon som börjar med DROP skall man vara extremt försiktiga med, nu snackar vi riktigt destruktiva saker, och som vanligt inget **"undo"**.

```
mysql> DROP TABLE test;
Query OK, 0 rows affected (0.34 sec)

mysql>
```

Så, nu är tabellen **"test"** ett minne blott.

## Skapa tabellen bilar

Nu skall du skapa tabellen bilar. Den skall innehålla fyra fält. Ett med **reg**, **marke**, **modell** och **arsmodell** (observera att vi undviker svenska tecken).

Den skall se ut enligt skärmdumpen nedan och du skall fylla den med data som nedan.

Var noga med att det blir exakt som nedan. Använda de kommandon du lärt dig för att skapa tabellen. Se till att den är rätt innan du börjar mata in bilarna. Resultatet skall bli exakt som detta:

```
mysql> EXPLAIN bilar;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null  | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| reg        | char(10)  | YES   |      | NULL    |      |
| marke      | char(50)  | YES   |      | NULL    |      |
| modell     | char(50)  | YES   |      | NULL    |      |
| arsmodell  | int(11)   | YES   |      | NULL    |      |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.03 sec)

mysql> SELECT * FROM bilar;
+-----+-----+-----+-----+
| reg      | marke      | modell  | arsmodell |
+-----+-----+-----+-----+
| ABC123   | Saab       | 9-5     | 2003      |
| DEF123   | Volvo      | S80     | 2002      |
| GHI123   | Mazda      | 626     | 2001      |
| JKL123   | Audi       | A8      | 2001      |
| MNO123   | BMW        | 323     | 1998      |
| PQR123   | Ford       | Mondeo  | 2001      |
| STU123   | Volvo      | 740     | 1987      |
| VYX123   | Volkswagen | Golf    | 1988      |
| ABC456   | Volkswagen | Polo    | 2003      |
| DEF456   | Toyota     | Carina II | 1998      |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>
```

Lycka till!