

MySQL – Lathund

Av: Marcus Rejås <marcus@rejas.se>

Detta är en liten lathund till MySQL. Den beskriver de vanligaste kommandona i deras enklaste form. Dessa bör man lära sig utantill, vilket som regel sker automatiskt när man använt dem ett tag.

Detta dokument är framtaget för elever vid Roslagens Högskola. Det kan med små modifieringar användas av alla.

Anslut till databasen

Logga in på `ssh.rhs.se` med **PuTTY**, se separat lathund.

Anslut till databasen med följande kommando:

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

Byt ut "rejas" mot ditt användarnamn, ange ditt lösenord när det efterfrågas. Lösenordet syns inte när du skriver in det, men skriv ändå.

Skapa en tabell (CREATE TABLE)

```
mysql> CREATE TABLE test (id INT, namn CHAR
(50));
Query OK, 0 rows affected (0.05 sec)
```

Ovanstående skapar tabellen **test** med två fält, id och namn. Namn är ett fält som innehåller text (upp till 50 bokstäver) och id är en siffra. Denna siffra räknar upp sig själv så man behöver inte ange detta fält.

Mata in data (INSERT INTO)

```
mysql> INSERT INTO test (namn) VALUES ('Kalle
Anka');
Query OK, 1 row affected (0.14 sec)
```

Ovanstående matar in "Kalle Anka", som **namn** i tabellen **test**. Vill jag mata in värden i flera fält åtskiljer jag dem med komma (,), likadant gör jag då med värdena. Till exempel

```
mysql> INSERT INTO test (id, namn) VALUES (10,
'Kajsa Anka');
Query OK, 1 row affected (0.07 sec)
```

Ställa frågor (SELECT ... FROM)

```
mysql> SELECT * FROM test;
+-----+-----+
| id  | namn      |
+-----+-----+
| NULL | Kalle Anka |
| 10  | Kajsa Anka |
+-----+-----+
2 rows in set (0.10 sec)
```

Ovanstående väljer allt (*) från tabellen **test**.

Ändra data (UPDATE)

WARNING!! Detta kommando förändrar data, det finns inget "undo", var försiktig!

```
mysql> UPDATE test SET id=9 WHERE id=10;
Query OK, 1 row affected (0.07 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
```

Ovanstående uppdaterar fältet id till 9 för alla poster där id är 10. Observera att **ALLA** poster med id=10 kommer att få id 9, vilket kanske inte är vad man vill alla gånger. **Se till att det finns ett fält som är unikt för alla poster.**

Radera poster (DELETE FROM)

WARNING!! Detta kommando tar bort data, det finns inget "undo", var försiktig!

```
mysql> DELETE FROM test WHERE id=9;
Query OK, 1 row affected (0.06 sec)

mysql>
```

Ovanstående tar bortt **ALLA** poster med id=9.

Tips och trix!

- Du kan bläddra mellan tidigare kommandon du skrivit med tangenterna **pil upp** och **pil ned**.
- Dokumentera dina kommandon genom att kopiera dem och klistra in dem i ett textredigeringsprogram, de kan vara bra att ha senare.
- Betydligt mer information finns i Björks bok, "Databashantering". Den finns på <http://www.jonashjork.net/databas/>