

MYSQL LABORATION 1

Ni kommer se två punkter där ni skall göra något. Punkt ett är Windows, punkt två är för MacOS. Vi arbetar med Steg 1 till och med 7 tillsammans under lektionspasset. Steg 8 genomför du individuellt på lektion eller valfri plats, deadline för steg 8 finner du på Progress.

Steg 1:

För att kunna börja arbeta med våra databaser måste vi starta vår databasmodul på vår webserver.

Det gör vi genom att;

- Starta XAMPP Control Panel och klickar på start till höger om modulen MySQL.
- Starta manager-osx markera MySQL Database och klicka start.
 - Kan behöva startas via terminalen, skriv:

```
Sudo /Applications/XAMPP/xamppfiles/bin/mysql.server start .
```

När vi har startat vår server kan vi öppna kommandoprompten för just MySQL. Det gör vi genom att;

- Klicka på knappen Shell i menyn till höger på XAMPP Control Panel.
 - Nu är vi inne i kommandoprompten för servern men för att kunna komma in till vår databasmodul måste vi logga in i den med kommandot:

```
# mysql -u root .
```

- Öppnar terminalen och starta kom igång med mysql.

```
Sudo /Applications/XAMPP/xamppfiles/bin/mysql -u root .
```

Massa välkomsttext visas då förhoppningsvis och om du gjort rätt så ska din kommandoprompt visa följande rad:

```
MariaDB [(none)]>
```

Steg 2:

När vi är i kommandoprompten så måste vi avsluta varje förfrågan med ett ; som skrivs med Shift+komma.

Testa skriv:

SHOW DATABASES *(utan semikolon)*

Prompten visar då:

```
MariaDB [(none)]> SHOW DATABASES
```

```
->
```

Men om du nu skriver ett ; så kommer frågan att skickas iväg och vi får se vilka databaser som just nu finns på vår webserver.

Har man inte skapat en databas sedan tidigare borde man se fem databaser.

Steg 3:

Nu ska vi skapa en egen databas som vi kallar **intro** med kommandot:

CREATE DATABASE *intro*;

OBS! Om ni redan har en databas som heter **intro** så får ni ta bort den först med följande kommando: **DROP DATABASE *intro*;**

Sedan ska vi kontrollera att databasen skapades genom att skriva:

SHOW DATABASES;

Sedan ska vi välja vår nyskapade databas så vi kan arbeta med den. Skriv:

USE *intro*;

Steg 4:

Vi ska nu öva på att skapa tabeller och fylla dem med information. Skriv:

CREATE TABLE *test1* (*namn* varchar(32));

För att kolla att tabellen nu finns så skriver vi:

SHOW TABLES;

Vi skapar en till tabell med kommandot:

CREATE TABLE *test2* (*förnamn* varchar(32), *efternamn* varchar(32));

Kolla nu att båda tabellerna finns med.

Sedan vill vi även titta på strukturen för dessa tabeller. Skriv:

EXPLAIN *test1*;

EXPLAIN *test2*;

Då får man upp en någorlunda grafisk vy för hur de två olika tabellerna är uppbyggda med kolumner och datatyper.

Steg 5:

Nu ska vi fylla våra tabeller med data! Börja med att skriva:

INSERT INTO *test1* (*namn*) VALUES ('Kalle');

Och sedan:

INSERT INTO *test2* (*förnamn*, *efternamn*) VALUES ('Kalle', 'Anka');

För att titta på allt innehåll i **test1** skriver vi:

SELECT * FROM *test1*;

Titta även på **test2** med samma kommando.

Fyll på **test2** med **Kajsa Anka**, **Knatte Anka**, **Fnatte Anka** och **Tjatte Anka**.

Titta sedan på det nya innehållet i **test2** med kommandot **SELECT**.

Fyll nu på **test2** med: **Musse Pigg** och **Mimmi Pigg**.

Lägg även till **Pluto**, **Piff** och **Puff** genom att skriva:

```
INSERT INTO test2 (förnamn) VALUES ('Pluto');
```

```
INSERT INTO test2 (förnamn) VALUES ('Piff');
```

```
INSERT INTO test2 (förnamn) VALUES ('Puff');
```

Kolla så att du fått med alla karaktärer med kommandot **SELECT**.

OBS! Du ska ha 10 rader med karaktärer i din tabell om du allt gjort rätt!

Observera att efternamnen på **Pluto**, **Piff** och **Puff** har värdet NULL.

Det är helt rätt och beror på att vi inte angav något värde för efternamn och då får det automatiskt värdet NULL, som betyder att det inte finns något värde.

Steg 6:

Lägg till en ny kolumn i **test2** genom att skriva:

```
ALTER TABLE test2 ADD (ålder int(3));
```

Kolla hur det ser ut i **test2** med **SELECT ***.

Det vi gjorde var att vi ändrade tabellen **test2** och lade till en kolumn som heter **ålder** och innehåller datatypen int.

Nu ska vi ge alla karaktärer en ålder. Då måste vi uppdatera en post på följande vis:

```
UPDATE test2 SET ålder=24 WHERE förnamn='Kalle';
```

Uppdatera även **Kajsas** ålder till 24.

Man kan även ändra fler fält samtidigt om man ställer rätt fråga:

```
UPDATE test2 SET ålder=28 WHERE efternamn='Pigg';
```

Vi ändrade alltså alla poster med efternamn **Pigg** och satte ålder till 28.

Nu ska vi uppdatera **Knatte**, **Fnatte** och **Tjattes** ålder i en och samma fråga. Men vi vill ju inte ändra alla med **Anka** som efternamn för då kommer **Kalle** och **Kajsa** även få samma ålder...

Då kan vi skriva med ett extra villkor i vår fråga separerat av den logiska operatör AND. Sådär ser det ut:

```
UPDATE test2 SET ålder=13 WHERE efternamn='Anka' AND ålder IS null;
```

Nu ska vi bara uppdatera **Pluto**, **Piff** och **Puffs** åldrar. Det som dem har gemensamt och som skiljer sig ifrån de andra är att de börjar på bokstaven P. Vi kan då skriva:

```
UPDATE test2 SET ålder=8 WHERE förnamn LIKE 'P%';
```

Databasen tolkar % som vad som helst och i detta fall betyder det att förnamn ska följa kravet att första bokstaven är P följt av vad som helst. Observera att denna typ av jämförelse enbart funkar om man skriver LIKE innan!

Nu kan vi kolla så att alla har fått rätt ålder med kommandot:

```
SELECT * FROM test2;
```

Steg 7:

Ifall vi har massor av information i vår databas är det bra att kunna få endast den information vi är intresserade av. Det gör vi med **SELECT** men vi får skriva till ett eller flera villkor som talar om vad vi är intresserade av. Om vi vill ha all information om alla som heter **Anka** i efternamn skriver vi:

```
SELECT * FROM test2 WHERE efternamn='Anka';
```

Skriv en fråga så att du får all information om alla som heter **Pigg** i efternamn.

Om vi bara vill ha viss information om alla som heter **Anka** i efternamn så får vi specificera den själva genom att skriva efter **SELECT** vilka fält vi vill kolla på:

```
SELECT förnamn,ålder FROM test2 WHERE efternamn='Anka';
```

Vi kan även kolla på alla som till exempel är äldre än 10 år:

```
SELECT förnamn FROM test2 WHERE ålder>10;
```

Steg 8:

NU SKA NI FÅ TÄNKA SJÄLVA OCH GÖRA NÅGRA UPPGIFTER.
SVARA PÅ FRÅGORNA I ETT DOKUMENT OCH VISA MIG NÄR NI ÄR KLARA
ALTERNATIVT SKICKA IN PÅ PROGRESS!

Uppgift 1-4 är obligatoriska. Uppgift 5 är valfri.

1. *Vad skriver jag för fråga om jag från test2 vill hämta;*

- a) Ålder på alla karaktärer med 'tt' i sitt förnamn?
- b) Förnamn på alla som inte har något efternamn?
- c) All information om de karaktärer vars förnamn börjar på 'K'?
- d) Förnamn på alla som är under 13?
- e) Alla efternamn i test2?
- f) Förnamn och ålder på alla som heter Anka i efternamn?
- g) All information om de karaktärer som har ett U i sitt förnamn?

2. *Skapa en ny tabell som heter städer med följande kolumner*

stad - varchar(32)

kommun - varchar(32)

landskap - varchar(32)

län - varchar(32)

postkod - int(5)

invånare - int(8)

3. *Lägg in information i tabellen städer*

Skriv in följande [kod](#) genom att kopiera den och högerklicka i kommandoprompten och sedan klistra in.

OBS! Om ni skapat städer fel kommer ni få errors när ni klistrar in!

Om det händer, skriv då: **DROP TABLE städer;** Och gör om, gör rätt!

Lägg sedan själv till minst 3st svenska städer i din tabell med rätt information.

4. *Få en lista med namn och befolkning på alla städer som:*

- a) Ligger i Dalarna.
- b) Har en befolkning på över 50 000 invånare.
- c) Har en befolkning på över 200 000 invånare.
- d) Har en postkod som börjar på siffran 4.
- e) Ligger i Västra Götalands län.
- f) Ligger i en kommun vars namn börjar på A.
- g) Har en befolkning på under 50 000 invånare.
- h) Har ett e i sitt namn.
- i) Ligger i Hälsingland och har över 20 000 invånare.

- j) Ligger i en kommun som börjar på bokstaven B har ett stadsnamn som börjar på bokstaven B och som har över 25 000 invånare.

5. *Ställ en fråga så du får all information om alla städer sorterat efter...*

- a) Antal invånare i fallande ordning.*
- b) Bokstavsordning från A-Ö.*
- c) Postnummer i stigande ordning.*

** Detta får du söka upp själv hur man gör ;) **