



Innehåll MySQL Intro

- ✓ Allmänt om Lagrade Procedurer
- ✓ Enkel utformning
- ✓ Skapa en lagrad procedur
- ✓ Använda parameter som indata



Lagrad procedur / Stored Procedure

- ✓ Lagrad procedur har många namn, förkortningen sproc används ofta.
- ✓ En lagrad procedur är ett avsnitt av SQL/program-kod.
- ✓ Oftast använder man sproc för att automatisera sina databasapplikationer.
- ✓ Man skickar inte data direkt till tabellerna utan låter alla data gå från webbmiljön via sproc till tabellerna.
- ✓ Lagrade procedurer är mycket kraftfulla och kan lösa många problem.
- ✓ En lagrad procedur används ofta för att lösa säkerhetsproblem exempelvis SQL-injections.
- ✓ I MySQL se hjälpen i Query Browser.
- ✓ I MSSQL se länken som leder till [SQL Server Developer Center](#)



Lagrad procedur / Stored Procedure

- ✓ Går snabbare att köra än vanlig SQL, koden är redan kompilerad.
- ✓ Återanvändning av kod.
- ✓ Ökad säkerhet, det går att styra rättigheter för procedurer.
- ✓ Man kan låta affärslogiken ligga i databasen.
- ✓ Reducerad nätverkstrafik.
- ✓ Bättre säkerhet.
- ✓ Enklare installation. Endast en kopia på servern – ingen på klienten.



Lagrad procedur

En enkel lagrad procedur kan se ut enligt nedan. Uppgiften för den är att visa en lista på kunderna.

```
DELIMITER $$
CREATE PROCEDURE KollaKunder()
BEGIN
    SELECT Namn, Adress, Postnr, Ort
    FROM Kund;
END $$
DELIMITER ;
```

Först kör du koden enligt ovan vilket gör att din lagrade procedur skapas så den finns tillgänglig sedan.

Kör sedan med följande rad. Du måste börja med anropet **CALL**. Resultatet får du som en lista på det du har i SQL-satsen.

```
CALL Kollakunder
```

	Namn	Adress	Postnr	Ort
1	Adamssons El	Lingonstigen 13	393 52	KALMAR
2	Berglunds Bilskola	Stigen 71	393 65	KALMAR
3	Erikssons Måleri	Adamsstigen 99	393 64	KALMAR
4	Johansson Kalle	Marsstigen 7B	393 51	KALMAR



Skapa en Lagrad Procedur

1

Högerklicka på databasen som du vill knyta den lagrade proceduren till.

2

Välj
Create New Procedure / Function

3

Du får en del färdig kod. Anpassa innehållet.
Exempelvis:

```
DELIMITER $$
DROP PROCEDURE IF EXISTS KollaKunder $$
CREATE PROCEDURE KollaKunder ()
BEGIN
    SELECT Namn, Adress, Postnr, Ort
    FROM Kund;
END $$
DELIMITER ;
```

4

Kör koden så skapas den lagrade proceduren och färdig att användas. OBS!
Du kanske måste se till att din databas är Default

5

```
CALL Kollakunder ()
```

Edit Schema
Drop Schema
Copy SQL to Clipboard

Create New Schema
Create New Table
Create New View
Create New Procedure / Function

Refresh

Make Default Schema



Några nyckelord i MySQL SQL

DELIMITER \$\$	Avgränsare mellan avsnitt är \$\$.
DROP PROCEDURE IF EXISTS	Istället för att använda ALTER så raderas proceduren (om den finns) innan den skapas på nytt.
BEGIN / END	Börjar o slutar SQL genomförande avsnittet.
USE	Anger vilken databas som ska användas USE Faktura
DECLARE	Definierar en variabel. DECLARE antal INTEGER
SET	Set värde i variabel. SET antal=4
Med default värde	DECLARE antal INTEGER Default 25;
LAST_INSERT	Returnerar sista/högsta värdet I Pk. Räknare.
FOUND_ROWS	Hur många rader har returnerats
/* bla bla */	Kommentera flerradigt
Tilldela en variabel värde med SELECT.	Variabeln får inte heta samma som fältet.

Exempel:

```
SELECT Pris From Artikel WHERE Artikelid=Artid INTO Priset;
```



Använda Parameter som indata

Antag att du vill skicka med data så att endast en viss kund visas med sina uppgifter.

Komplettera med en parametern Kundnr och vilken datatyp den har.

OBS! om du ska ändra befintlig – högerklicka på den och välj Edit Procedure.

IN-data till sproc OUT/INOUT)

```
DELIMITER $$
DROP PROCEDURE IF EXISTS KollaKunder $$
CREATE PROCEDURE KollaKunder (IN id integer)
BEGIN
    SELECT Namn, Adress, Postnr, Ort
    FROM Kund
    WHERE Kundid=id;
END $$
DELIMITER ;
```

Namn på param

datatyp för param

Parametervärde

```
CALL KollakunderEnViss (1)
```

OBS! Tänk på datatypen. Ett textfält/sträng omgärdas av apostrof (')



Alla eller en speciell?

```
DELIMITER $$
```

```
DROP PROCEDURE IF EXISTS KollaKunder $$
```

```
CREATE PROCEDURE KollaKunder (IN id integer)
```

```
BEGIN
```

```
    DECLARE Kundnr INTEGER Default id; ←
```

```
    IF Kundnr=0 THEN /* Lista alla kunder */
```

```
        SELECT Namn, Adress, Postnr, Ort  
        FROM Kund;
```

```
    ELSE /* Lista en kund */
```

```
        SELECT Namn, Adress, Postnr, Ort  
        FROM Kund  
        WHERE Kundid=Kundnr;
```

```
    END IF;
```

```
END $$
```

```
DELIMITER ;
```

Kundnr tilldelas
värdet av id

```
CALL Kollakunder (0) Ger alla kunderna
```

```
CALL Kollakunder (2) visar kund med kundid=2
```




Ny fakturarad

Lägga till en ny fakturarad. Hämta data ur artikel och uppdaterar artikel. Eftersom uppdatering måste ske i två tabeller används transactions.

```
DELIMITER $$
CREATE PROCEDURE FakturaRadNy $$
(
IN Faktid int, IN Artid int, IN Ant int, IN Rabatt Decimal(2,2), IN Momsid int
)
BEGIN
    DECLARE Priset Decimal(6,2) Default 0;

    SELECT Pris From Artikel WHERE Artikelid=Artid INTO Priset;

    START TRANSACTION;      /* OBS! Start istället för BEGIN*/
        INSERT INTO Fakturarad (Fakturaid, Artikelid, Antal, Pris, rabatt, momsid)
        VALUES (Faktid, Artid, Ant, Priset, Rabatt, Momsid);

        UPDATE Artikel Set Antal=Antal-Ant
        WHERE Artikelid=Artid;
    COMMIT;

END $$
DELIMITER ;

CALL FakturaradNy(6,104,2,0.05,1)
```



Radera Fakturarad

Radera en fakturarad och samtidigt uppdatera artikel med antal. Två tabeller ska uppdateras därför använder vi transactions.

```
DELIMITER $$
CREATE PROCEDURE FakturaRadRadera $$
(
IN Faktid int, IN Artid int, IN ANT int
)
BEGIN
    START TRANSACTION;
    UPDATE Artikel Set Antal=Antal+Ant
    WHERE Artikelid=Artid;

    DELETE From Fakturarad
    WHERE Fakturaid=Faktid AND Artikelid=Artid
    COMMIT;
END $$
DELIMITER ;

CALL FakturaRadRadera (6,104,2)
```



Ett exempel

Det är ofta som man vill göra en simulering av värden från flera tabeller. Då skapar man en ny tabell, temporär, och för över data från andra tabeller. Därefter behandlar man data i den nya tabellen för att till sist kasta tabellen.

Exempel: ***Vi vill ha en lista på hur mycket våra kunder har handlat och när det har handlat. Vilken kund har handlat vad och hur mycket.***

Följande steg gör man då med en lagrad procedur:

1. Skapar den nya tabellen (CREATE....
2. Överför data från andra tabeller (INSERT INTO.....
3. Behandlar data (SELECT
4. Slutligen raderar tabellen (DROP.....

En kortversion av den lagrade proceduren finns på nästa sida.....



spoc KundStatistikLista

```
DELIMITER $$
DROP PROCEDURE IF EXISTS KundStatistikLista $$
CREATE PROCEDURE KundStatistikLista ()
BEGIN
    CREATE Temporary Table Temp
    (
        Eid int Primary Key AUTO_INCREMENT,
        Namn varchar(50) NOT NULL,
        Ort varchar(30) NOT NULL,
        Datum date NOT NULL,
        ArtikelID int NOT NULL,
        Antal smallint,
        Pris Decimal(8,2)
    );
    INSERT INTO Temp (namn,ort,datum,artikelid, antal, pris)
    SELECT k.namn, k.ort, f.datum, fa.artikelid, fa.antal, fa.pris
    FROM Kund as k INNER JOIN Faktura as f ON k.kundid=f.kundid
    INNER JOIN Fakturarad as fa ON f.fakturaid=fa.fakturaid;

    SELECT * From Temp;
    DROP Table Temp;
END $$
DELIMITER ;
```