

SQL INTRO

STRUCTURED QUERY

LANGUAGE

EN INKÖRSDROG TILL DE TYNGRE GREJERNA

QUERY

A question, especially one expressing doubt or requesting information.

Strukturerat frågespråk

SQL

Språk för att hämta data från databaser på ett strukturerat sätt

SQL har flera olika **dialekter**

Den vi ska använda är **MySQL**, den mest använda

Frontend: aldrig **persistent data** allting lagras tillfälligt

Inget är heller dolt: api-nycklar, variabler, information



DATABAS

En bas för data, en låda med data

Fast sjukt strukturerad i tabeller

id	username	password
5	zero_cool	alligator1

MEN SHIT, VAD ÄR DET FÖR FEL PÅ **JSON?**

```
[  
  {  
    "id": "42",  
    "username" : "zero_cool",  
    "password" : "alligator1"  
  }  
]
```

MongoDB - Så kallad NoSQL

RELATIONSDATABAS

Data ordnad i olika relationer, tabeller

Relationera överlappar för att binda ihop data

Det går skitsnabbt att hämta och lagra data

id	title	createdBy
42	How to code	Jesper

id	title	createdBy	createdByEmail
42	How to code	Jesper	jesper@orb.se

id	title	createdBy
42	How to code	15

id	username	password	email
15	jesperorb	pass123	jesper@orb.se

```
{  
  "id"      : 42,  
  "title"   : "How To Code",  
  "createdBy" : 15  
}
```

```
{  
  "id"      : 15,  
  "username" : "jesperorb",  
  "password" : "pass123",  
  "email"    : "jesper@orb.se"  
}
```

```
{
  "id"      : 42,
  "title"   : "How To Code",
  "createdBy" : {
    "id"      : 15,
    "username" : "jesperorb",
    "email"    : "jesper@orb.se"
  }
}
```

Som det **typ** ser ut i **MongoDB**

PHPMYADMIN

RDBMS

Först måste vi ha ett
Relational Database Management System

Ett system som hanterar all data

Men det har vi redan: phpMyAdmin

DATATYPER

- `int` - heltal, 4 byte
- `float` - flyttal, 4 byte
- `varchar` - strängar, 1 byte per tecken
- `text` - använd för mycket långa strängar
- `datetime` - datum och tid, i formatet '2016-12-19 14:01:00'

CRUD!

- INSERT : CREATE
- SELECT : READ
- UPDATE : UPDATE
- DELETE : DELETE

Inget chockerande! Vi använder andra nyckelord.

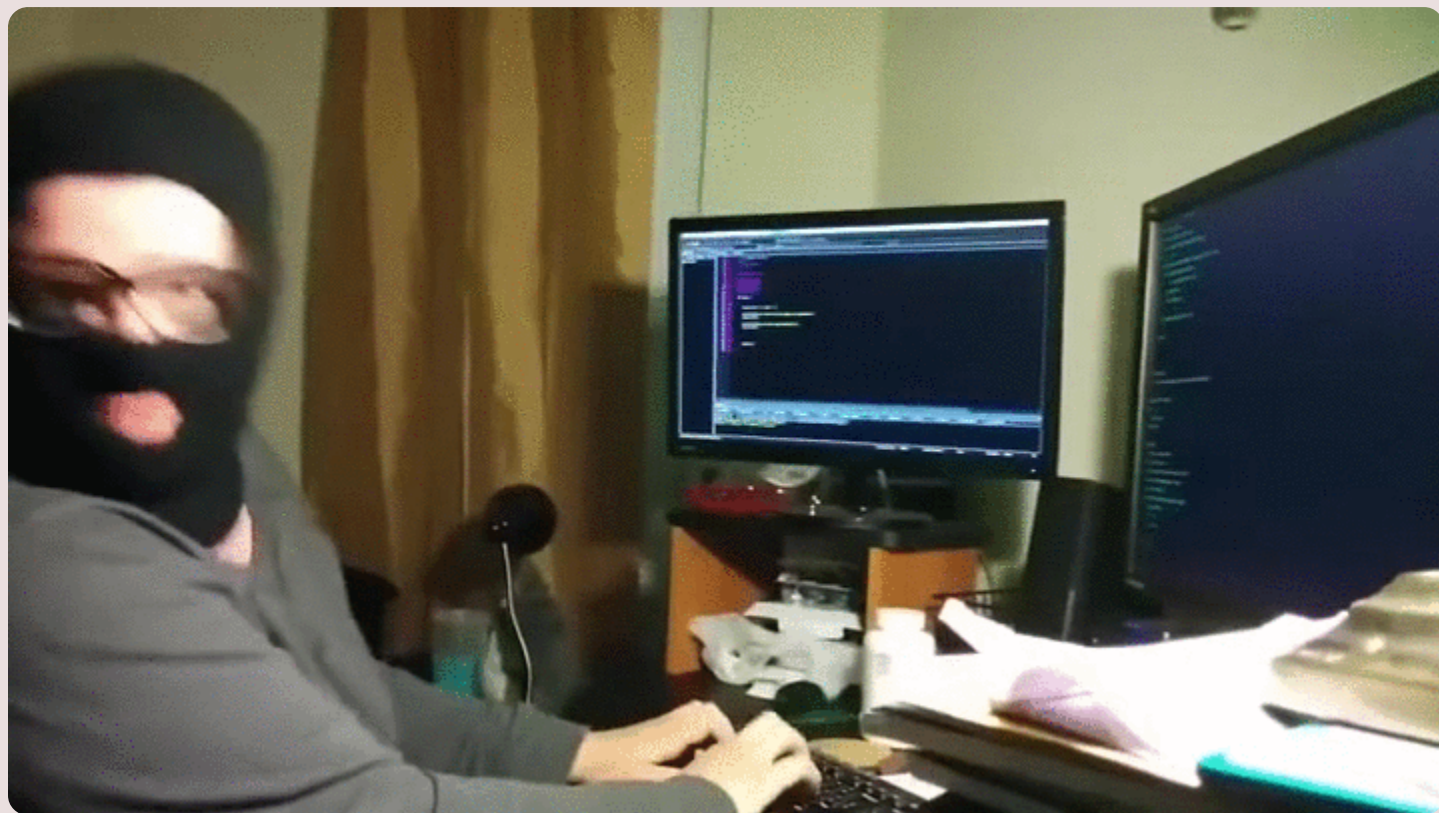
PRIMARY KEY

Varje rad i en tabell måste ha ett unikt id

id	title	createdBy
42	How to code	15

id	title	createdBy
105	How to code	15

VI SKAPAR EN TABELL



VI HÄMTAR FRÅN EN TABELL?

SQL QUERIES

SELECT FROM

```
SELECT * FROM table;
```

Asterisk säger att vi ska välja alla kolumner

```
SELECT name FROM table;
```

SELECT FROM WHERE

Vi vill ju kunna sälla innehållet. Detta gör man med WHERE

```
SELECT * FROM notes WHERE id = 1;
```

Alla kolumner är sökbara

```
SELECT * FROM notes WHERE title = "My Title";
```

SELECT DISTINCT

När man vill ta bort dubletter

```
SELECT DISTINCT name FROM notes;
```

Om det finns flera personer med samma namn t.ex. så plockar vi enbart namnen

Annars får vi kanske tillbaka 15 Magdalena

SELECT AS

Döper om en kolumn när vi hämtar den

Ändrar INTE i själva databasen

```
SELECT title AS BlogTitle from blog;
```

title blir till BlogTitle i resultatet men heter fortfarande title i tabellen

SELECT COUNT()

Räknar antalet rader (rows) i en tabell

```
SELECT COUNT(name) FROM notes
```

ORDER BY

Vi kan även direkt sortera data i en query

```
SELECT * FROM notes ORDER BY name ASC
```

ASC - Ascending == stigande

DESC - Descending == fallande

UPDATE

Vi uppdaterar tabeller enligt följande mönster (obs pseudokod)

```
UPDATE table  
SET column = value  
WHERE condition;
```

```
UPDATE notes  
SET title = "My New Shiny Title!"  
WHERE id = 1;
```

DELETE

DELETE används för att ta bort allt eller vissa rader

```
--Tar bort allt!  
DELETE FROM notes;
```

Vi måste använda `condition`

Tar bort en specifik rad

```
DELETE FROM notes  
WHERE id = 1;
```

Tar bort massa!

```
DELETE FROM notes  
WHERE id > 1;
```