

# SQL INTRO

# STRUCTURED QUERY LANGUAGE

# 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** lagras alltid  
tillfälligt

Alla variabler ligger enbart i minnet



# DATABAS

En bas för data, en låda med data

Fast sjukt strukturerad i tabeller

id	username	password
5	zero_cool	alligator1

# RELATIONSDATABAS

Data ordnad i olika relationer, tabeller

Relationera överlappar för att binda ihop data

Det går skitsnabbt att hämta och lagra data

Ett avancerat excel-blad



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

**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

Allting på hela internet baseras på CRUD

# PRIMARY KEY

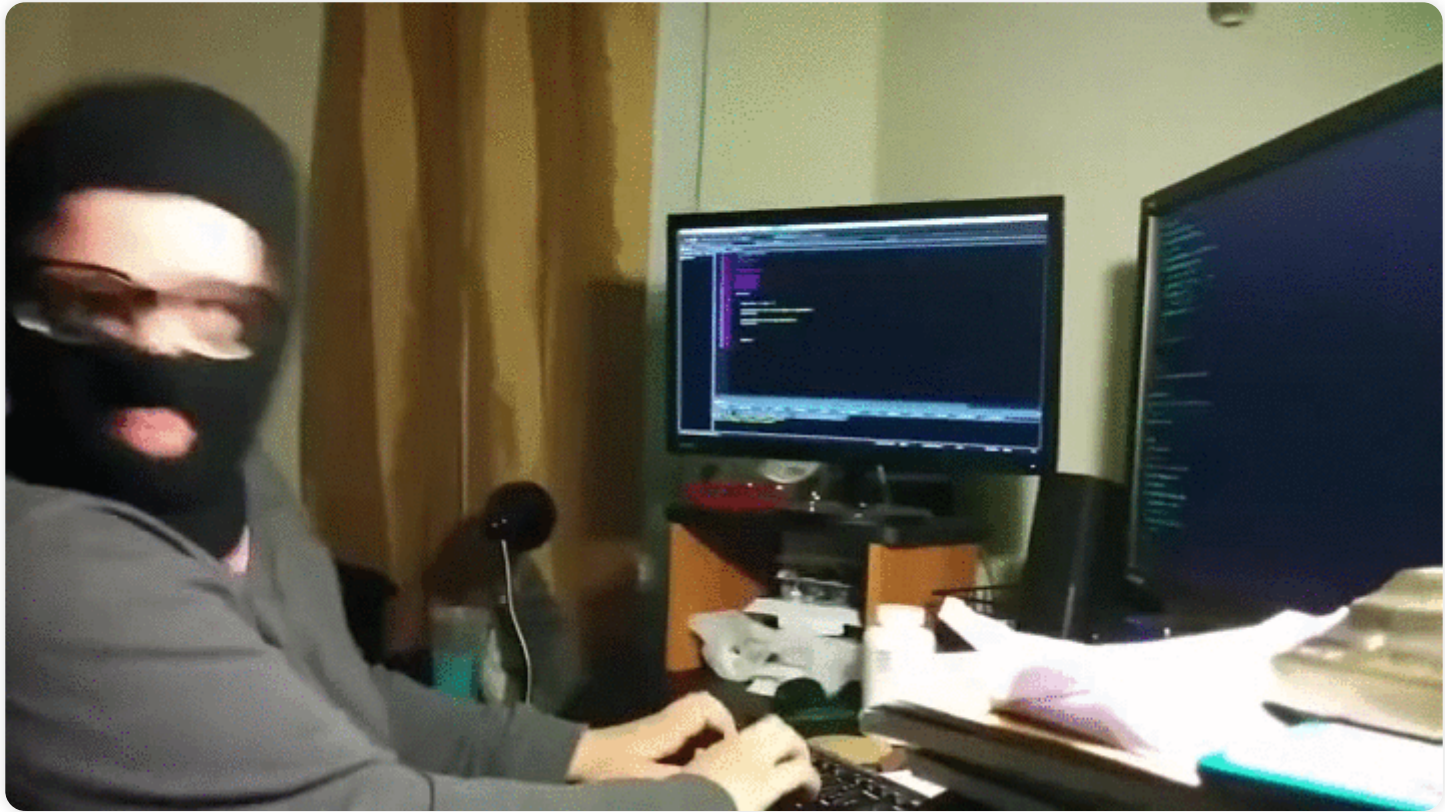
Varje rad i en tabell måste ha ett unikt id

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 sålla 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;
```