

Databasmotorer

Alla sätt är bra utom de dåliga

Databasmotorer

- Två viktiga frågor:
 - Vad ska lagras?
 - Hur ska det användas?
- Alla exempel skriver vi i PHP

CSV

- Ren text
- Vanligt exportformat
- Varje post börjar på en egen rad
- Ofta en rad per post
- Kommatecken mellan egenskaper
- Enkelt, men känsligt



CSV

BRA

- Snabbt att lägga till en post
- Enkelt att implementera

DÅLIGT

- Långsamt att hitta, ändra och ta bort
- Lätt att göra bort sig

Att skriva och läsa CSV

```
$db = fopen("database.csv", "a");  
$data = array(3, "Nisse", 34);  
fputcsv($db, $data);
```

```
$db = fopen("database.csv", "r");  
while ($d = fgetcsv($db))  
    print("ID: $d[0], name: $d[1], age: $d[2]\n");
```

```
$db = fopen("database.csv", "r");  
while (list($id, $name, $age) = fgetcsv($db))  
    print("ID: $id, name: $name, age: $age\n");
```

JSON

- Ren text (unicode)
- Vanligt som serialisering
- Används ofta för API-kommunikation
- Måsvingar för objekt, hakparenteser för listor
- Robust och väldefinierat



JSON

BRA

- Svårt att göra bort sig
- Modernt, alla förstår

DÅLIGT

- Tar mer plats
- Långsamt att hitta, ändra och ta bort

Att skriva och läsa JSON

```
$db = fopen("database.json", "a");  
$data = array("id" => 3,  
              "name" => "Nisse",  
              "age" => 34);  
fwrite($db, json_encode($data)."\n");
```

```
$db = fopen("database.json", "r");  
while ($d = json_decode(fgets($db), true))  
    print("Name: ". $d["name"]."\n");
```


Nyckel/värde

- Om man vill lagra objekt som har en given nyckel
- Själva värdet kan lagras som exempelvis JSON
- Vanligt när databaser måste vara snabba
- NoSQL



Nyckel/värde

BRA

DÅLIGT

- Väldigt snabbt
- Enkelt att resonera om
- Stödjer bara en sorts fråga

Att skriva och läsa nyckel/värde

```
$db = dba_open("database.db4", "c", "db4");  
dba_insert("3", "Nisse", $db);  
dba_insert("4", '{"name" : "Nisse",  
                "age"   : "34"}', $db);
```

```
$db = dba_open("database.db4", "r", "db4");
```

```
$name = dba_fetch("3", $db);
```

```
$person = json_decode(dba_fetch("4", $db), true);
```

Relationsdatabaser

- Tabeller med kolumner och rader
- Rader kan ha relationer till andra rader
- Man menar oftast databaser som har ett frågespråk
- SQL



Relationsdatabaser

BRA

DÅLIGT

- Snabba
- Bra för komplexa databaser
- Kraftfulla frågespråk
- Komplicerade att använda
- Kraftfulla frågespråk

SQLite med PDO

```
$dogDB = new PDO('sqlite:dogdb.db');  
  
$data = array($_REQUEST['dogname'],  
             $_REQUEST['dogid']);  
  
$setDogName = $dogDB->prepare('UPDATE Dog  
                                SET name = ?  
                                WHERE id = ?');  
  
$setDogName->execute($data);
```

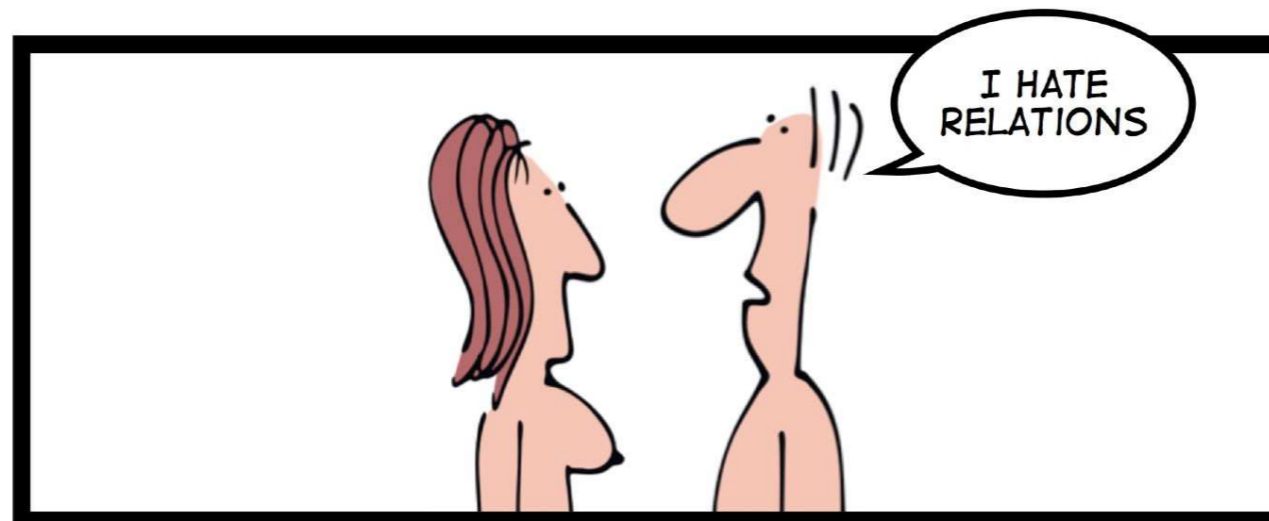
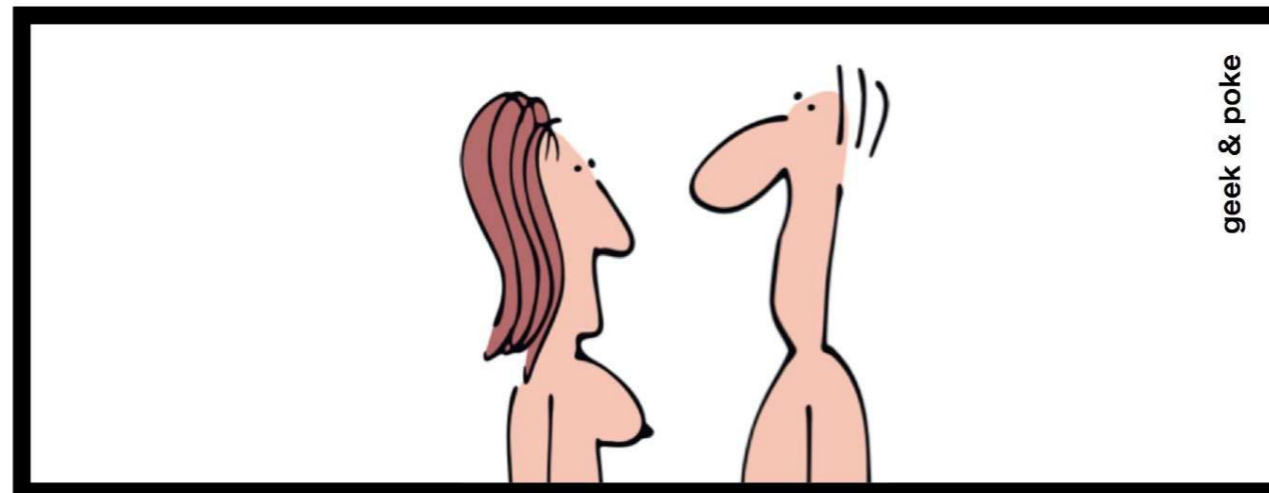
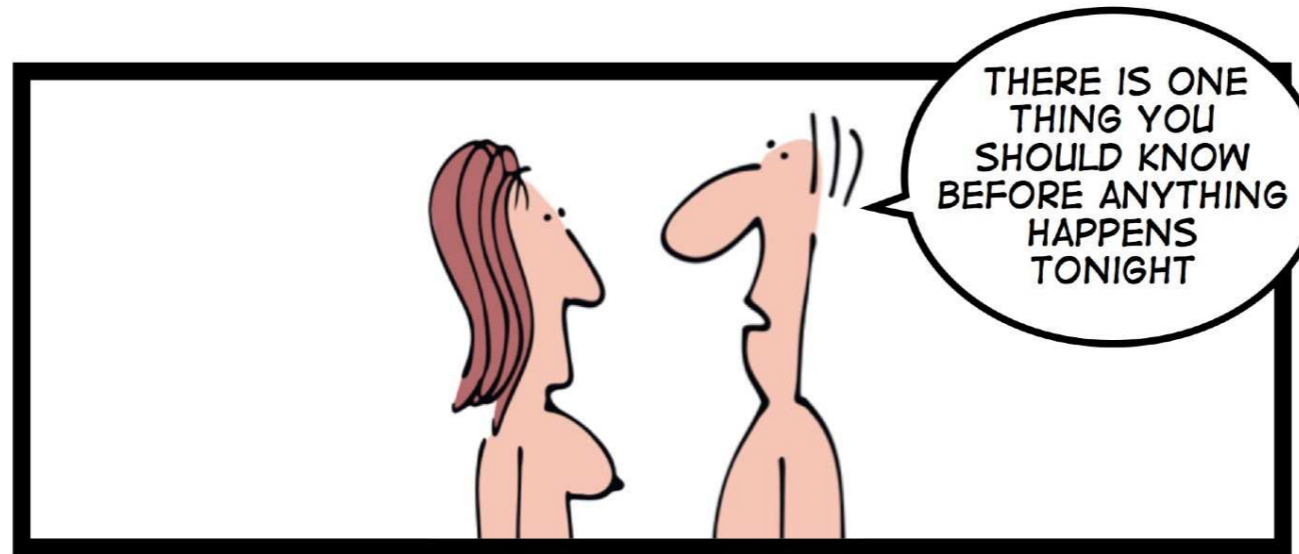
MySQL med PDO

```
$dogDB =  
    new PDO( 'mysql:host=localhost;dbname=dogdb' ,  
            'admin' , 'bajs' );  
  
$data = array( $_REQUEST[ 'dogname' ] ,  
               $_REQUEST[ 'dogid' ] );  
  
$setDogName = $dogDB->prepare( 'UPDATE Dog  
                                SET name = ?  
                                WHERE id = ?' );  
  
$setDogName->execute( $data );
```

SQLite - ett alternativ

- Lättare än MySQL
- Hela databasen är en fil på disken
- Ingen användarhantering
- Vanligt i lokala applikationer

The Hard Life of a NoSQL Coder



Part 1: The Outing