

DB-SYSTEME

Manuel Blechschmidt
Nils Glück

PRÄAMBEL

- Denken in Systemen:

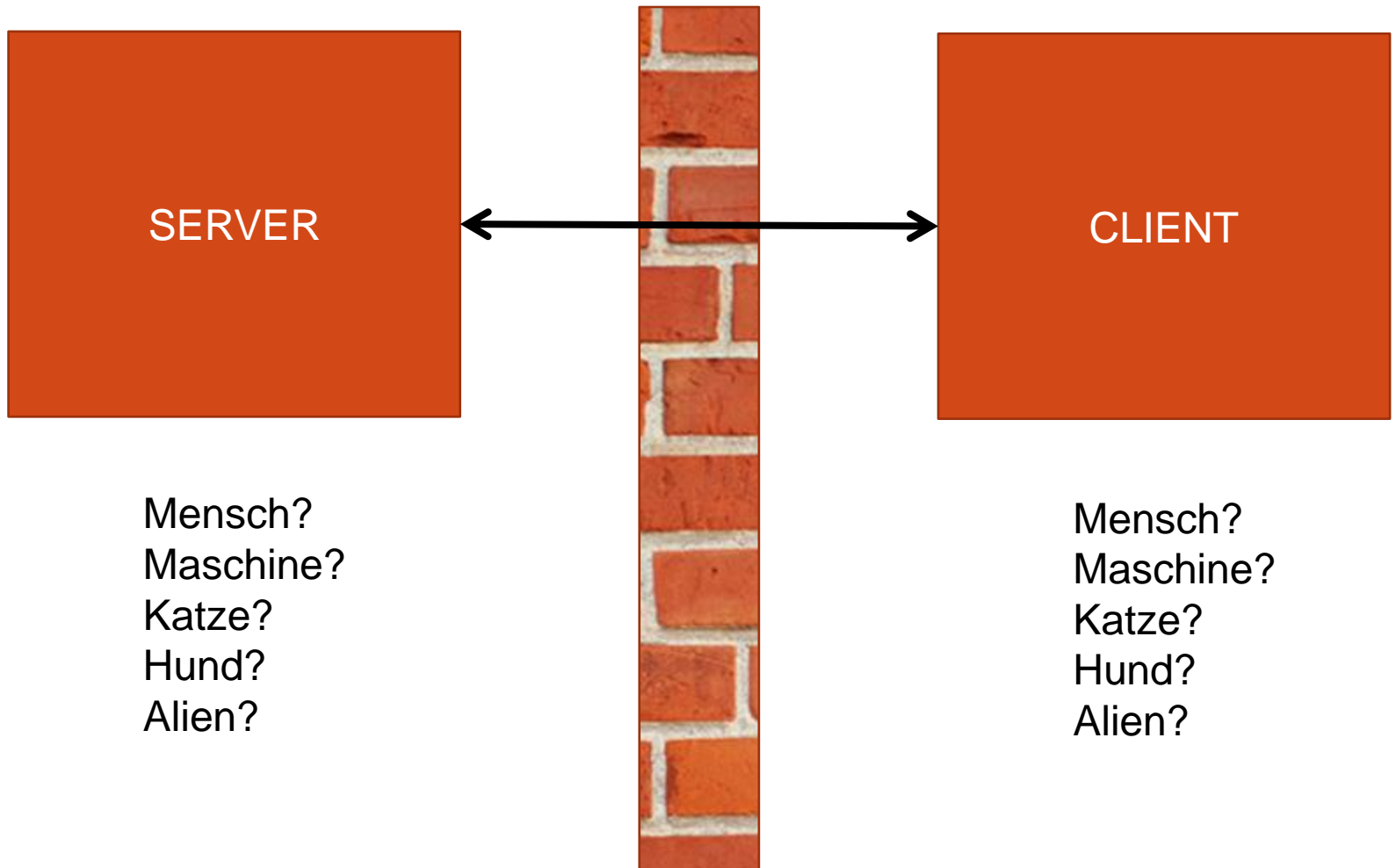
Ebene 1: Keine Ahnung!

Ebene 2: Da kenn ich mich aus!

Ebene 3: Keine Ahnung!

PROLOG 1: SERVER & CLIENT

- So nah, so fern.

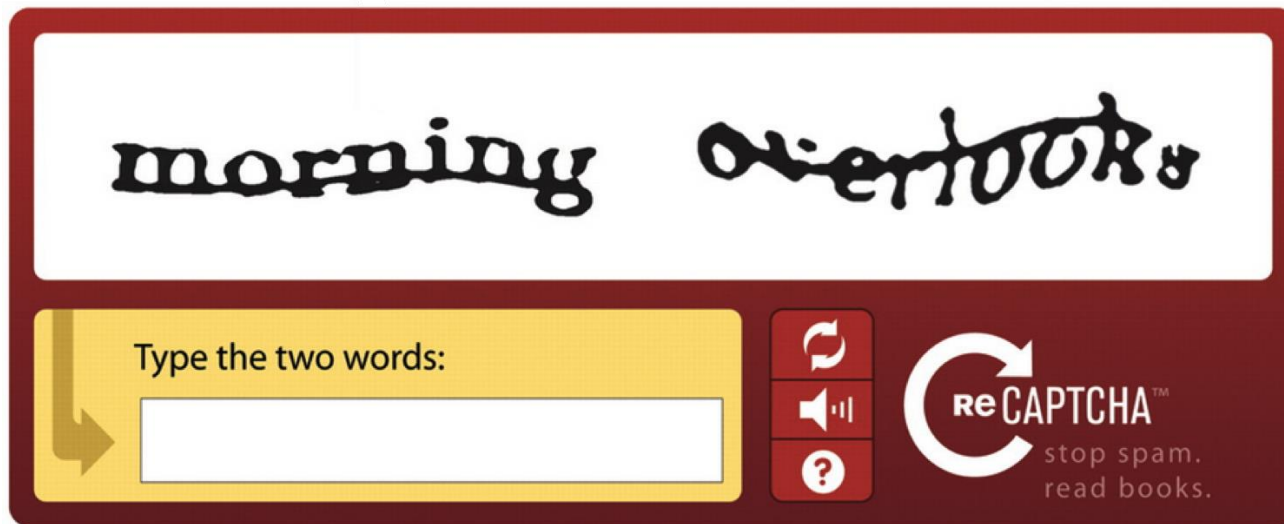


PROLOG 1: SERVER & CLIENT



PROLOG 1: SERVER & CLIENT

- Bitte abtippen.



morning overlook

Type the two words:

reCAPTCHA™
stop spam.
read books.

PROLOG 2: DAS GÄSTEBUCH

Was ist eine DB?

- Erschreckend simpel (Datenhaufen)
- Wie Excel-Daten, nur meistens als Server-Anwendung
- Strukturiert
- Ansprechbar über standardisierte Schnittstellen
- Sind zentral in unserer Wissensgesellschaft
 - Praktisch für alle, weil zentral. Spart Platz und vermeidet Bürokratie
 - Verfügbar
 - Leicht verständlich: Bewährt und konventionell. Bestens dokumentiert.

Relationale DBs

- Relation = Tabelle (= zB Datenblatt bei Excel)
- Wo ist meine DB? Auf dem Server.
- Wo sind meine Relationen? In der DB.
- Wo sind meine Objekte? In den Relationen.
- Relationales DBMS: mengenorientiert
- Alle Relationen liegen gleichberechtigt nebeneinander

Mein erster Arbeitstag

- Was sollte ich den Chef fragen (Pflicht):
 - Wo ist die Datenbank? (Host-Server / DB-Name)
 - Wie verbinde ich mich? (Adapter / Authentifikation)
- Was muss ich im Manual nachlesen?
 - Vier Basis-Operationen: C-R-U-D
 - Daten lesen
 - Daten manipulieren (verändern)
- Optional für später, wenn das erste Meeting überlebt ist. ;)
 - Wie tue ich das alles sicher?
 - So arbeiten, dass ich meine Ziele möglichst gut erreiche. Jetzt und später.

DBMS

- Gibt vor, wie die DB strukturiert und geregelt ist (zB Version, Vorschriften etc.)
- Wir nutzen MySQL (relationales DBMS = RDBMS)



DBMS-Admintools

- Anwendungen, mit denen man auf die DB zwecks Verwaltung zugreift
- Machen die DB sichtbar
- Liefern zB
 - GUI
 - Export- / Import-Tools
 - C-R-U-D-Helfer
 - Tools für Analyse und Error Tracking



DBMS != DBMS-Admin

- Nicht verwechseln:
 - MySQL (unsere DB)
 - PHPMyAdmin (unser Admin-Tool)

Genug Theorie.

LET'S DIVE INTO IT !!!!

- Datenbank auf dem Server erstellen
- Einloggen in PHPMysqlAdmin
- Rumspielen

C-R-U-D: SQL-Befehle

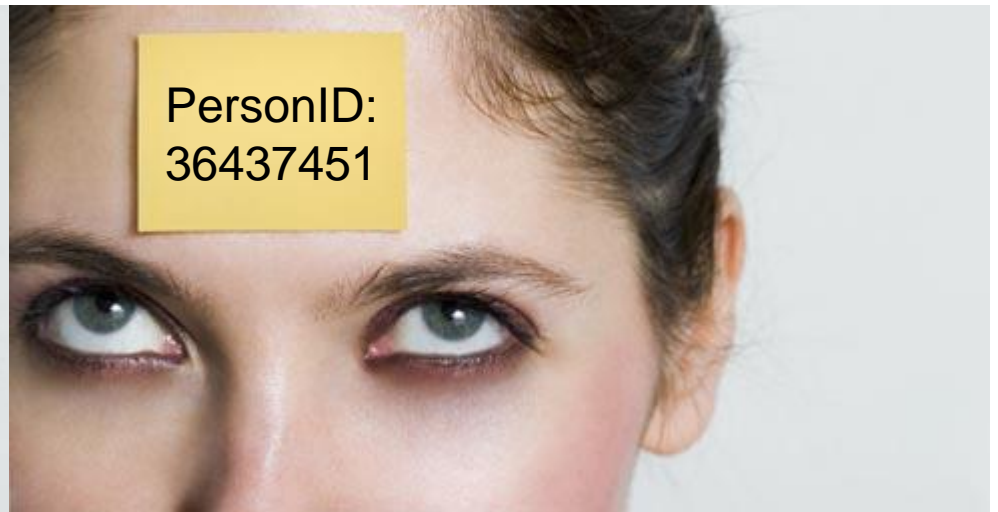
- C: INSERT
- R: SELECT
- U: UPDATE
- D: DELETE
- Zur Eingrenzung: WHERE Clauses
- Ergebnis-Rückgabe:
 - Bei SELECT: Eine Relation
 - Ansonsten: Anzahl der „affected rows“
- RTFM = Read The F***** Manual:
 - <http://dev.mysql.com/doc/refman/5.1/de/select.html>
 - <http://dev.mysql.com/doc/refman/5.1/de/insert.html>
 - <http://dev.mysql.com/doc/refman/5.1/de/update.html>
 - <http://dev.mysql.com/doc/refman/5.1/de/delete.html>

STRUKTUR-PRINZIPIEN

- Konsistenz anstreben
- Redundanz vermeiden
- Alles eindeutig identifizierbar machen
(**ID** = ungefähr „unique identification value“)



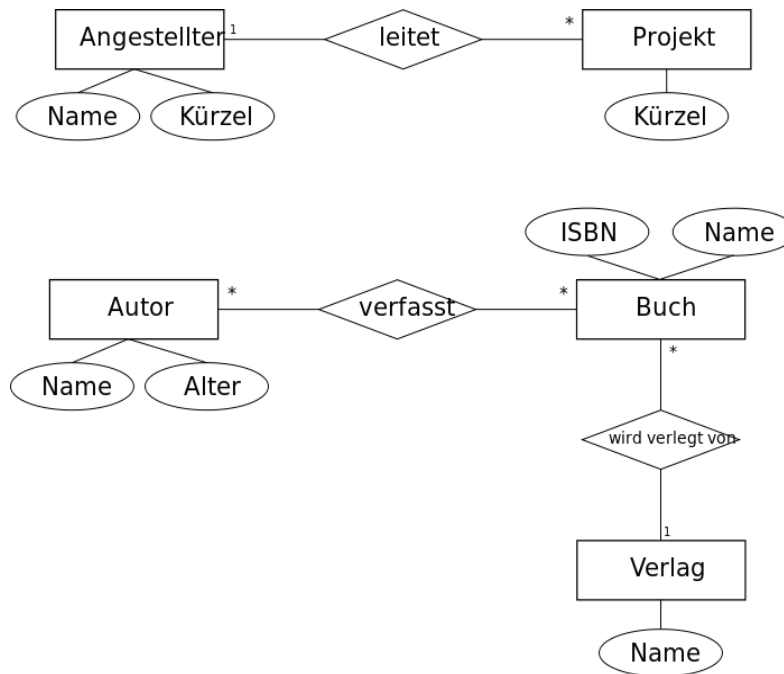
SCHLECHT



GUT

E/R-Schema

- Entity-Relationship
- Mensch will Bedeutung, DB will Struktur
- E/R „verheiratet“ beide Bedürfnisse



Angestellte, Projekte,
Autoren, Bücher, Verlage:
DIMENSIONEN
(Objekte)

Leiten, Verfassen,
Verlegen:
FAKTEN
(Sachverhalte)

DB-Normalformen

- Empfehlungen für E/R-Schemata
- Basieren auf viel Erfahrung der Community

| CD_Lied | | | | | |
|---------|--------------------|------------|-------------------|-------|----------------------------|
| CD_ID | Albumtitel | Interpret | Jahr der Gründung | Track | Titel |
| 4711 | Not That Kind | Anastacia | 1999 | 1 | Not That Kind |
| 4711 | Not That Kind | Anastacia | 1999 | 2 | I'm Outta Love |
| 4711 | Not That Kind | Anastacia | 1999 | 3 | Cowboys & Kisses |
| 4712 | Wish You Were Here | Pink Floyd | 1964 | 1 | Shine On You Crazy Diamond |
| 4713 | Freak of Nature | Anastacia | 1999 | 1 | Paid my Dues |

DB-Normalformen

| CD | | | | Lied | | |
|--------------|--------------------|------------|-------------------|--------------|-------|----------------------------|
| <i>CD_ID</i> | Albumtitel | Interpret | Jahr der Gründung | <i>CD_ID</i> | Track | Titel |
| 4711 | Not That Kind | Anastacia | 1999 | 4711 | 1 | Not That Kind |
| 4712 | Wish You Were Here | Pink Floyd | 1964 | 4711 | 2 | I'm Outta Love |
| 4713 | Freak of Nature | Anastacia | 1999 | 4711 | 3 | Cowboys & Kisses |
| | | | | 4712 | 1 | Shine On You Crazy Diamond |
| | | | | 4713 | 1 | Paid my Dues |

DB-Normalformen

CD

| <i>CD_ID</i> | Albumtitel | Interpret_ID |
|--------------|--------------------|--------------|
| 4711 | Not That Kind | 311 |
| 4712 | Wish You Were Here | 312 |
| 4713 | Freak of Nature | 311 |

Künstler

| <i>Interpret_ID</i> | Interpret | Jahr der Gründung |
|---------------------|------------|-------------------|
| 311 | Anastacia | 1999 |
| 312 | Pink Floyd | 1964 |

Lied

| <i>CD_ID</i> | Track | Titel |
|--------------|-------|----------------------------|
| 4711 | 1 | Not That Kind |
| 4711 | 2 | I'm Outta Love |
| 4711 | 3 | Cowboys & Kisses |
| 4712 | 1 | Shine On You Crazy Diamond |
| 4713 | 1 | Paid my Dues |

DB-Normalformen: Best Practice

- Welche Objekte gibt es? → Dimensionen
- Welche Sachverhalte gibt es? → Fakten
- Erfahrung: Lieber zu viele als zu wenige Tabellen
- Faktentabellen sind Sammlungen von IDs
- Musik-Beispiel:
 - Dimensionen: Interpret, Album, Lied
(= 3 **Dimensionstabellen**)
 - Fakten:
Interpret_Has_Album, Album_Has_Lied
(2 = **Faktentabellen** = **Zwischentabellen**)
 - ABER Konsistenzannahme: „Lieder auf Alben haben jeweils alle dieselben Interpreten“.
 - Alternativ hinzunehmen: Lied_Has_Interpret