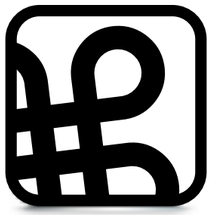


Macoun

Appelgriebsch

NoSQL auf Apple Systemen

Andreas Gerlach

Navigation

- Odyssee der Datenablage
- Jenseits von Stamm- & Bewegungsdaten
- Orientierungshilfe im “NoSQL Country”
- Sightseeings

Odyssee der Datenablage

Grundlagen der EDV

Eingabe Verarbeitung Ausgabe

Kontext von Daten

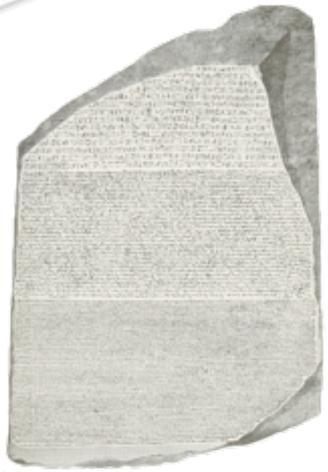
60311 Frankfurt a.M.

- Daten fundamentaler Bestandteil der EDV
- Informationsgewinnung steht im Vordergrund
- Daten sind zunächst flüchtig, sie sind für die weitere Verarbeitung dauerhaft abzulegen

Datenablage B.C.

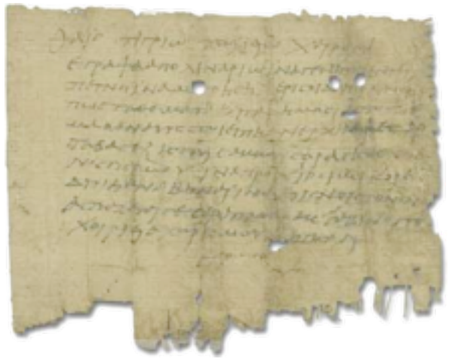
- Daten werden vom Programm in Dateien auf dem lokalen Magnetspeicher abgelegt
- Speicherplatz und Rechenzeit sind teuer
- optimierte, proprietäre Datenformate vorherrschend
- wenige Spezialisten als Lieferanten & Nutzer

Datenablage nach Codd



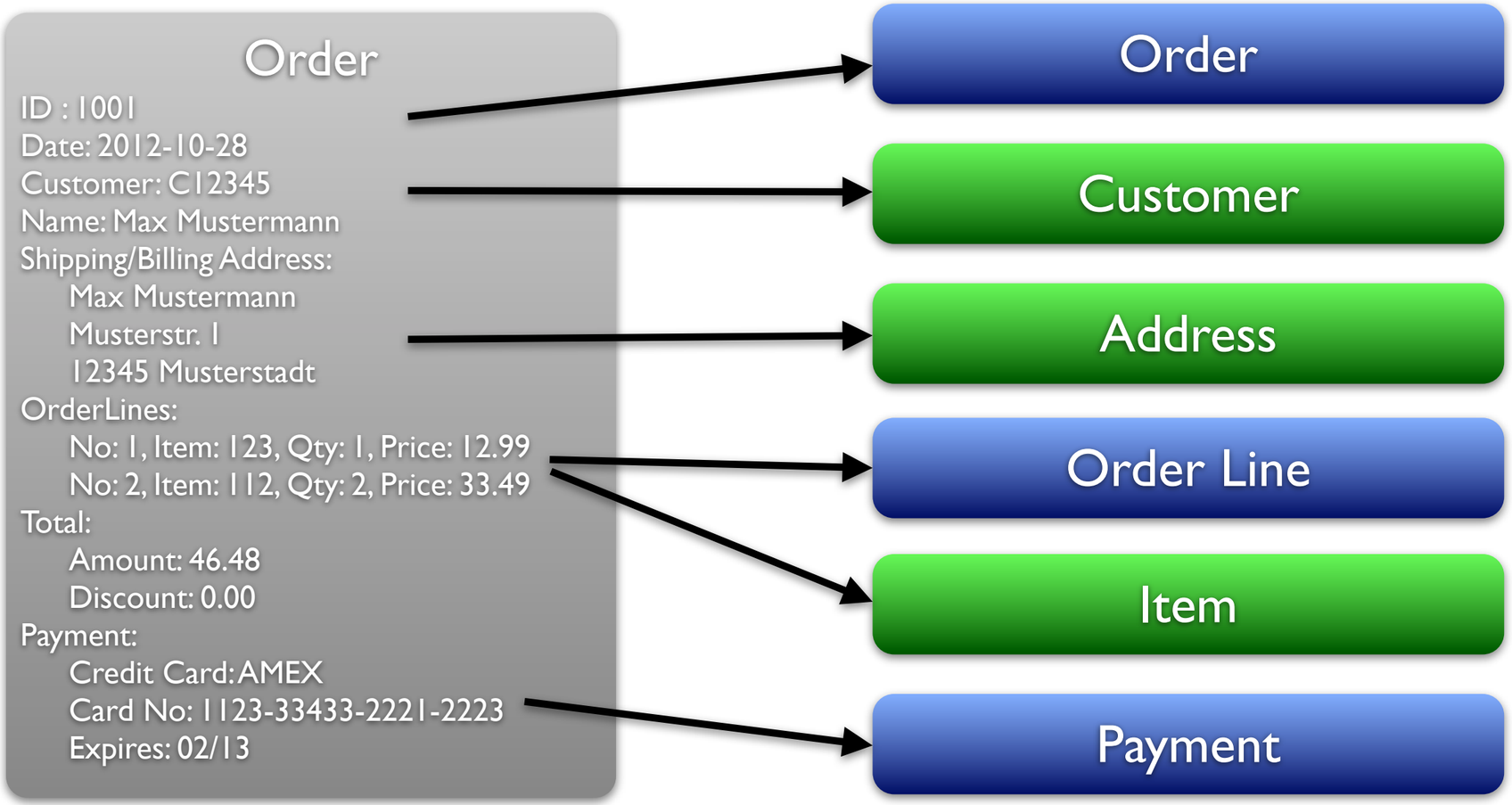
- Theorie: Ablagesystem basiert auf relationalem Mengenmodell
- strukturierte Erfassung der Entitäten, deren Attributen und Beziehungen
- Praxis: Normalisierung der Daten (3. NF) und Abbildung in Tabellenform
- standardisierter Datenzugriff (SQL)
- Integrationsfunktion für Anwendungen

Defizite bei Codd

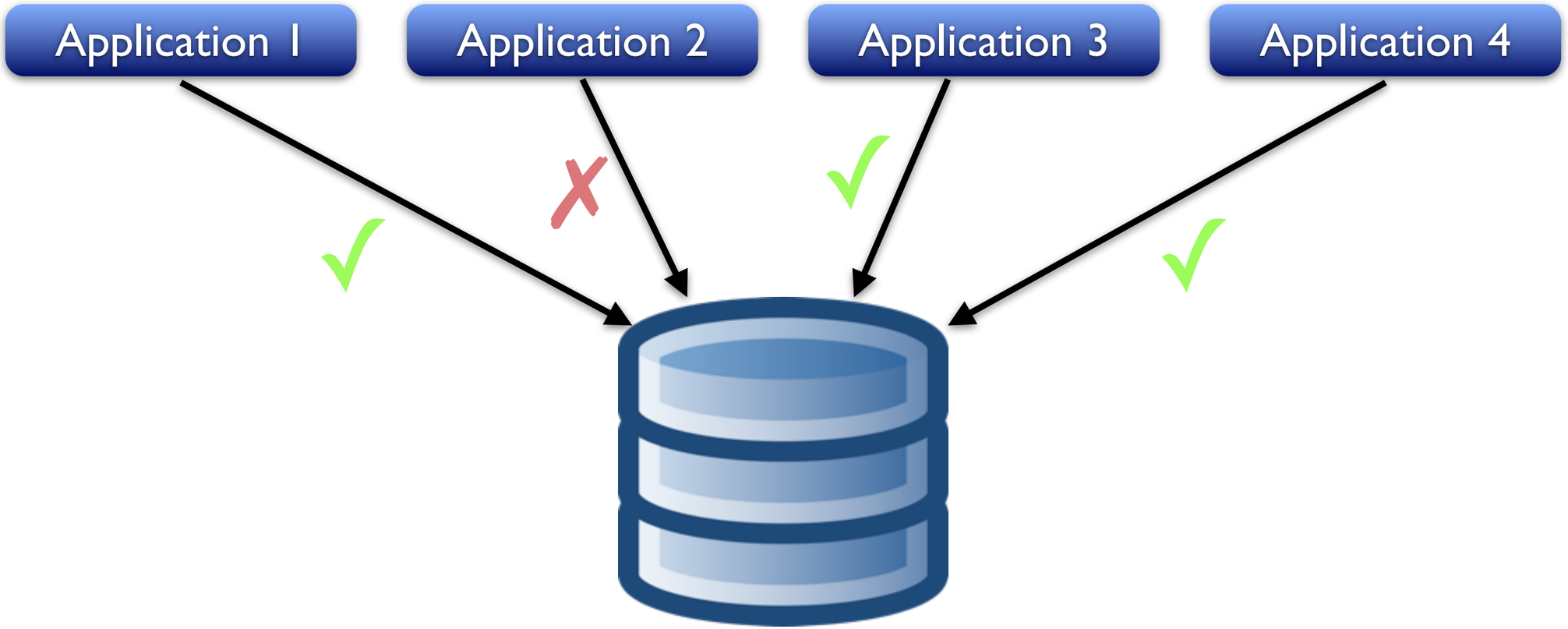


- “Impedance Mismatch”
- verbindliches Tabellenschema
- Skalierbarkeit des Systems

Impedance Mismatch



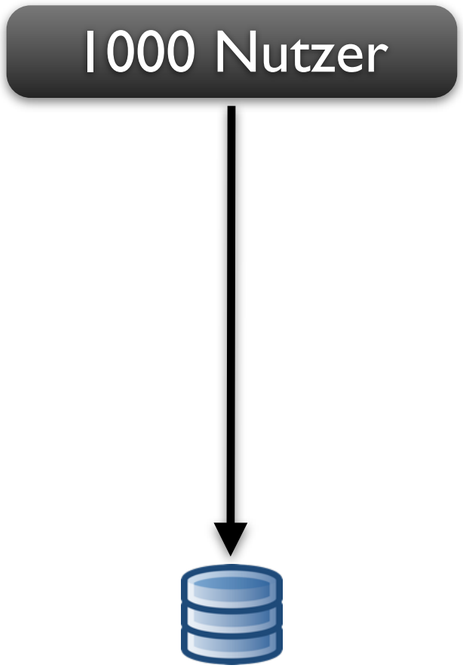
verbindliches Tabellenschema



Skalierbarkeit des Systems

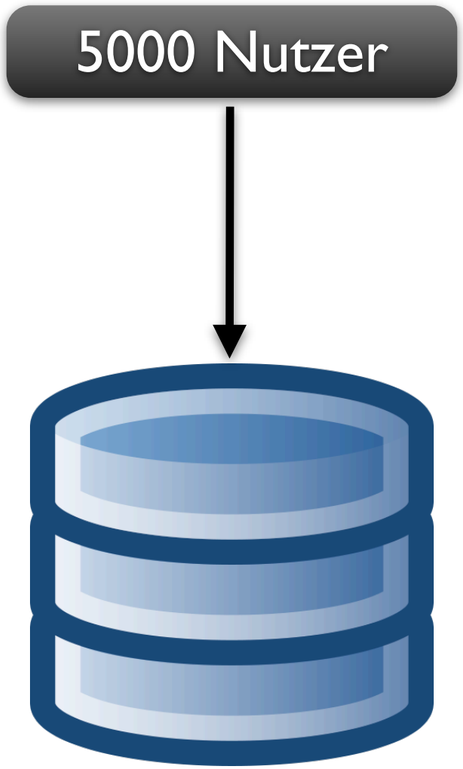
Scale-up

Scale-out

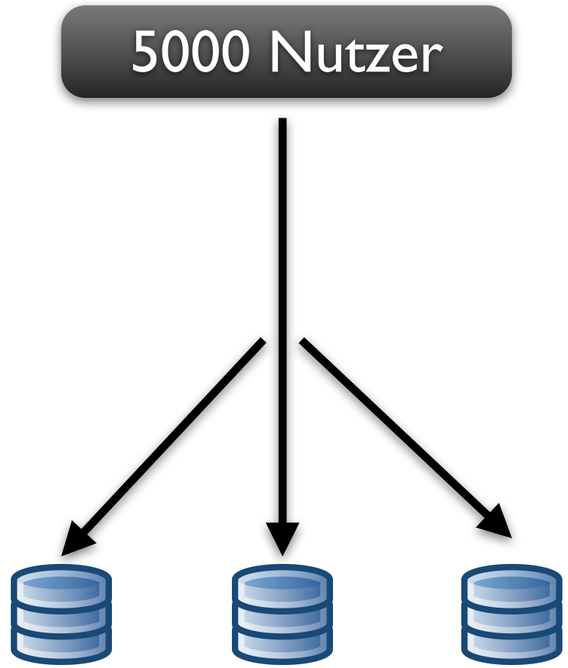


Skalierbarkeit des Systems

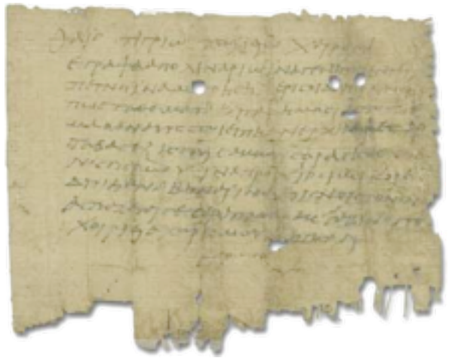
Scale-up



Scale-out



Datenablage A.C.



- “Die Information ist das Öl des 21. Jh.”
- “Datenmenge [ist] in den letzten 5 Jahren um Faktor 5 gewachsen”
- “weltweit [werden] täglich 2.5 Trillionen Byte Daten [erzeugt]”
- verschiedene Quellen mit Daten unterschiedlicher Qualität und Beschaffenheit
- bis zu 60% unstrukturierte Daten im Bestand

Erwartungen A.C.

- “scale-out” für maximale Skalierbarkeit
- Einsatz auf Commodity-Hardware
- kostengünstige Lösung
- schema-freie Datenbank
- Unterstützung komplexer Datenstrukturen
- Support von umfangreichen Datenanalysen

Pioniere A.C.

- Dynamo: Amazon's Highly Available Key-value Store (Amazon, 2007)
- Bigtable: A Distributed Storage System for Structured Data (Google, 2006)
- Cassandra - A Decentralized Structured Storage System (Facebook, 2009)

Impulse

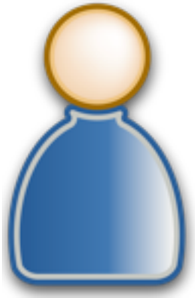
- massive, verteilte Datenhaltung ist möglich
- einfachere Abbildung komplexer Datenstrukturen inkl. stark verknüpfter Daten
- verteilte Analysen über MapReduce
- Achtung: CAP Theorem berücksichtigen!
- erhöhte Produktivität & Flexibilität in der Entwicklung durch "Polyglott Persistence"

Jenseits von Stamm- & Bewegungsdaten

Dimensionen von Daten

Format

Beschaffen-
heit



Rolle

Dynamik

Umfang

Zugriffs-
form

Reichweite

Orientierungshilfe im “NoSQL Country”

Schlüssel-Wert-Speicher

- einfachstes Datenmodell in “NoSQL Country”
- Zuordnung eines Werts (Daten) zu einem eindeutigen Schlüssel (analog: Dateisysteme)
- hohe Performance, da meist In-Memory
- daher gut für Caching, Einstellungen, Message Queuing geeignet
- Vertreter: Redis, Riak, Amazon DynamoDB

dokumentorientiert

- Betrachtung von Dokumenten als Gesamtheit
- Dokumente können in XML, JSON oder BSON Format vorliegen
- komplexe, auch hierarchische Strukturen lassen sich sehr einfach in der Datenbank verwalten
- daher gut für CMS, e-Commerce, ... (solange keine starken Abhängigkeiten vorhanden)
- Vertreter: MongoDB, CouchDB, Lotus Notes

spaltenorientiert

- Daten (Name:Wert) werden in Spaltenfamilien gruppiert
- jede Zeile kann unterschiedliche Anzahl von Spalten enthalten
- Zugriff über den eindeutigen Zeilenschlüssel, Spaltenfamilie und Spaltenname
- sehr gute Skalierung auch für Schreibvorgänge
- Vertreter: Cassandra, HBase, Google BigTable

graphenorientiert

- Verwaltung ganzer Graphen mit beliebig vielen Knoten und Kanten
- sehr gute Performance bei der Auswertung von Relationen zwischen Instanzen (Social Networks)
- Detailinformationen sind meist ausgelagert
- gut für Netzwerke, Vorschlagssysteme
- Vertreter: Neo4J, Infinite Graph, OrientDB

Sightseeings

Redis

- www.redis.io
- von vmware gesponsert
- in ANSI C entwickelt
- generell In-Memory Schlüssel-Wert-Speicher
- Unterstützung von numerischen/alpha-numerischen Daten, Listen, Hash-Werte
- eigener Publish / Subscribe Mechanismus

Redis

RDBMS	Redis
Datenbank Instanz	Redis Database
Tabelle	Bucket / Namespace
Zeile	Schlüssel-Wert Paar
Primärschlüssel	Schlüssel

Redis

- Demo

MongoDB

- www.mongodb.org
- in C++ entwickelt
- flexible dokumentenorientierte Datenbank mit erweiterten Indizes & eigener Abfragesprache
- automatisches Sharding, Master-Slave Replikation
- Unterstützung von MapReduce

MongoDB

RDBMS	MongoDB
Datenbank Instanz	MongoDB Instanz
Datenbank	Datenbank
Tabelle	Kollektion
Index	Index
Zeile	Dokument
Spalte	Feld
Primärschlüssel	_id Feld

http://www.mongodb.org/display/DOCS/SQL+to+Mongo+Mapping+Chart

MongoDB

- Demo

MongoDB

Operatoren	Beschreibung
$ne	ungleich
$lt	kleiner als
$lte	kleiner als oder gleich
$gt	größer als
$gte	größer als oder gleich
$exists	prüft auf Existenz eines Feldes
$all	prüft auf Existenz aller Elemente
$in	prüft auf Existenz eines Elements
$nin	prüft auf Nichtvorhandensein eines Elements
$or	oder
$nor	entweder oder
$not	negiert eine Bedingung

Riak

- www.basho.com
- auf Erlang basierende, massiv verteilte Lösung
- erweiterte Schlüssel-Wert Datenbank mit Unterstützung von JSON Daten und Anlage zusätzlicher Indizes
- automatisches Sharding, Master-Master Replikation (Riak Ring)
- Unterstützung von MapReduce

Riak

- Demo

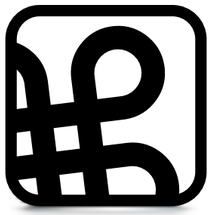
Ziel

Beispiele

http://codelabs.appelgriebsch.com

Fragen?

Vielen Dank



Macoun