

# DBS ÜB 1

Julian Habib, Tutorium: Burtscher, Felicia Tue 12:00 pm, 2:00 pm K 040/T9

Boyan Hristov, Tutorium: Lehmann, Nicolas Thu 4:00 pm, 6:00 pm SR 025/026/A6

Code auf Git/gitlab:

LINK:

## Aufgabe 1

- a) Ein Datenbanksystem bietet die Funktionalität einer Datenbank (Sammlung von Daten wie Kundeninformationen oder Messdaten) mit Hilfe eines Datenbankmanagementsystems (Softwareprogramme für Definition der Datenstruktur sowie Zugriff, Erstellung und Veränderung der Datenbank) nach dem Prinzip der Modularität zu verwalten. Die Sammlung der Daten kann unter Umständen sehr viele Daten beinhalten (Big Data). Bei dem Datenbankmanagementsystem sind viele Anforderungen wie Benutzerfreundlichkeit, Datensicherheit (Konsistenz, Persistenz, Authentizität, Integrität, Verfügbarkeit, Vertraulichkeit) Effizienz, Speicherbedarf, Redundanz, Anschaffungskosten und weitere Aspekte zu berücksichtigen.
- Datenbank (DB) + Datenbankmanagementsystem (DBMS) = Datenbanksystem (DBS)
- b) Zusammensetzung aus drei Teilen:
- Konzeptuelles Datenbankschema: Darstellung des Modells meistens mit Hilfe eines ER-Diagramm oder UML-Diagramm. Soll die reale Welt als Modell abbilden und anschaulich sein aus der nicht-technischen Sichtweise. Das heißt Vereinfachung der Wirklichkeit.
  - Logisches Datenbankschema: Implementierung des konzeptuellen Modells mit der gewählten Sprache passend zum Datenbankmanagementsystems (z.B. SQL). Soll das Konzept-Modell in Software umsetzen.
  - Physische Datenbankschema: technischnahes Modell, d.h. Hardware spezifisch oder zumindest hardwarenahe z.B. werden "Indexstrukturen" (Hashing, Binärbäume usw.) festgelegt. Soll die Hardware für den Betrieb bereitstellen und effizient sein.
- c) Physische Datenunabhängigkeit heißt, dass die DBMS weiter funktionieren kann, wenn die Struktur des physischen Speichers sich ändert (z.B. SSD anstatt von HDD, unterschiedliche Speichergröße).
- d) Eine logische Datenunabhängigkeit gibt es, wenn die Datenbankmanagementsystem weiter fehlerlos funktionieren können, nachdem sich die Art des Modells ändert und dazu keine Daten auf dem Speicher geändert werden.

- e) Anforderungen
  - i. Funktionale
    - i. Effizienz – Operationen auf dem DB müssen schnell ausführbar sein
    - ii. Redundanz – Daten sind einigermaßen mehrmals vorhanden und erlauben Fehlerkorrektur
  - ii. Nicht-funktionale
    - i. Standarte benutzen – ein DBS muss die schon existierende Standarte benutzen, damit dieses besser verständlich und einheitlich ist.
    - ii. Geringe Kosten – das DBS muss günstig genug sein, damit es in mehrere Fälle sinnvoll verwandbar ist.
- f) Meta-Daten sind administrative Daten, die die andere Daten beschreiben, wie z.B. Indexierung der DB, Datum der letzten Veränderung usw.

## Aufgabe 2

- a) Arten von Datenbanksysteme nach dem Arten von DBMS
  - i. Hierarchische – Daten werden in einem Baumstruktur gespeichert, wobei Relationen zwischen ein Knoten und seine Kinder stattfinden

Bsp. aus Quelle [1]

Vorgänger A

-----Nachfolger 1

----- Nachfolger 1.1

----- Nachfolger 1.2

.....

-----Nachfolger 2

-----Nachfolger 3

- ii. Netzwerk Datenbanksysteme – ähnlich zu hierarchische DBS, ein Knoten kann aber mehrere Elternknoten haben, damit sind auch many-to-many Relationen erlaubt. Das ist eine Oberklasse von den hierarchischen Datenbaksystemen.

Bsp. aus Quelle [1]

Vorgänger A

Zwischenfolger 1-----Nachfolger 1

Zwischenfolger 2-----Nachfolger 2

Zwischenfolger 3-----Nachfolger 3

.....

Vorgänger B

Zwischenfolger 1-----Nachfolger 4

Zwischenfolger 2-----Nachfolger 5

Zwischenfolger 3-----Nachfolger 6

- iii. Relationale Datenbanksysteme – diese, die in der Vorlesung vorgestellt wurden. Daten werden in Tabellen gespeichert, wobei die Spalten die Attributen definieren (jeder Attribut hat ein Datentyp) und die Zeilen die Entities (Exemplare) enthalten. Relationen können zwischen Tabellen stattfinden.
- iv. Objektorientierte Datenbanksysteme – Daten werden als Objekte in dem DB gespeichert. Die Idee ist das Code und Daten in objektorientierte Sprachen näher zusammen gehören. Daten werden nach einem Pointer gespeichert. DBMS können somit mit Daten ähnlich wie mit Code umgehen.
- v. NoSQL – Flexible DBS, wo Attributen dynamisch eingefügt und gelöscht werden können, Entities (Exemplare) müssen nicht unbedingt alle Attributen haben. In diese DBS gibt es aber keine Relationen zwischen die Tabellen in dem DB.
- b) NoSQL (kurz für „not only SQL“) sind DBS, die sich von SQL DBS trennen wollen, in dem sie mehr Freiheit geben (durch dynamisch einfügbare Attributen) und keine Relationen zwischen den Datentabellen anbieten. Wie die Name schon sagt, kann man mit diesen DBS nicht nur mit SQL arbeiten, sondern auch mit alternative Sprachen, die näher zu den Entwicklern sein sollen. Ein Beispiel für solches System ist MongoDB, wo Daten in JSON (JavaScript object notation, Daten sehen wie anonyme Objekte in JS), wo man mit dem DBS arbeiten kann ohne SQL zu benutzen (es gibt verschiedene Frameworks).

Die verschiedenen Typen verwenden verschiedene Strukturen für die Mustern

- i. Column (Spalten)
- ii. Document verwenden Dokumente (Dateien) um Daten in verschiedene Formate zu speichern (XML, JSON, YAML, ...)
- iii. Key-value (Schlüssel-wert Strukturen a.k.a Wörterbücher)
- iv. Graph (Graphen)
- v. Multi-Model (mehrere von den oben benannten Strukturen können verwendet werden)
- vi.
- c) RDF (Resource Description Framework) ist eine Konzeption die von dem World Wide Web Consortium (W3C) stammt und ist ein grundlegender Baustein des semantischen Webs. Das RDF-Modell ist ein Modell, dass nur die Syntax zum Datenaustausch festlegt, um eine Anwendungsdomäne zu beschreiben z.B. Metadaten im Web Aufbau und Funktionsweise: URI einer Verlinkung im Web oder URI einer Mailadresse zur eindeutigen Identifikation. RDF-Modell ist ein Modell, dass nur die Syntax zum Datenaustausch festlegt, um eine Anwendungsdomäne zu beschreiben z.B. Metadaten im Web Aufbau und Funktionsweise: URI einer Verlinkung im Web oder URI einer Mailadresse zur eindeutigen Identifikation

d) Beispiel SPARQL

PREFIX rdf: <<http://www.w3.org/1999/02/22-rdf-syntax-ns#>>

PREFIX dc: <<http://purl.org/dc/elements/1.1/>>

SELECT ?title

WHERE{

?res dc:publisher ?pub .

?res dc:title ?title

FILTER (

sameTerm(?pub, "Wikipedia - Die freie Enzyklopädie")

)

}

Funktionsweise: Stellt eine Anfrage an die Website mit dem Thema RDF, wählt den Eintrag Titel aus und wählt daraus den Herausgeber für den Titel aus.

Beispiel für RDF und SPARQL aus Quelle [3]

### Aufgabe 3:

- (a) kvv: Zeige mir alle Studenten, die letztes Jahr in Mathe durchgefallen sind.
- (b) Videothek: Zeige mir die Top-Ten Liste der PC-Spiele, die für das Mindestalter 18 freigegeben sind.
- (c) Photo-Sharing-Plattform: Zeige mir alle Photos von Name xy aus letzter Woche.
- (d) Mikobloggingdienst: Füge meinen Namen, zum Thema xy hinzu und zeige mir alle neuen Nachrichten dazu an.
- (e) soziales Netzwerk: Zeige mir alle Nutzer an, die ich bis jetzt blockiert habe und entferne User xy, falls dieser in der Liste (versehentlich) auftaucht.

## Quellen:

Vorlesung DBS

<https://de.wikipedia.org/wiki/Datenbank>

<https://de.wikipedia.org/wiki/Datenmodell>

<https://de.wikipedia.org/wiki/Datenunabh%C3%A4ngigkeit>

<https://www.johner-institut.de/blog/iec-62304-medizinische-software/funktionale-und-nicht-funktionale-anforderungen/>

<https://de.wikipedia.org/wiki/Metadaten>

[1] [http://www.scoberlin.de/content/media/http/informatik/dbgin32/dbarten\\_yifan.htm](http://www.scoberlin.de/content/media/http/informatik/dbgin32/dbarten_yifan.htm)

<https://de.wikipedia.org/wiki/NoSQL>

[2] <https://de.wikipedia.org/wiki/MongoDB>

<https://de.wikipedia.org/wiki/RDF-Schema>

[3] [https://de.wikipedia.org/wiki/Resource\\_Description\\_Framework](https://de.wikipedia.org/wiki/Resource_Description_Framework)