

Datenbankverwaltungssystem (DBVS/DBMS)

Ein Datenbankverwaltungssystem (DBVS, data base management system : DBMS) ist die Gesamtheit aller Programme (Ressourcen) zur Erzeugung, Verwaltung (einschl. Daten- und Konsistenzsicherung) und zur Manipulation einer Datenbank.

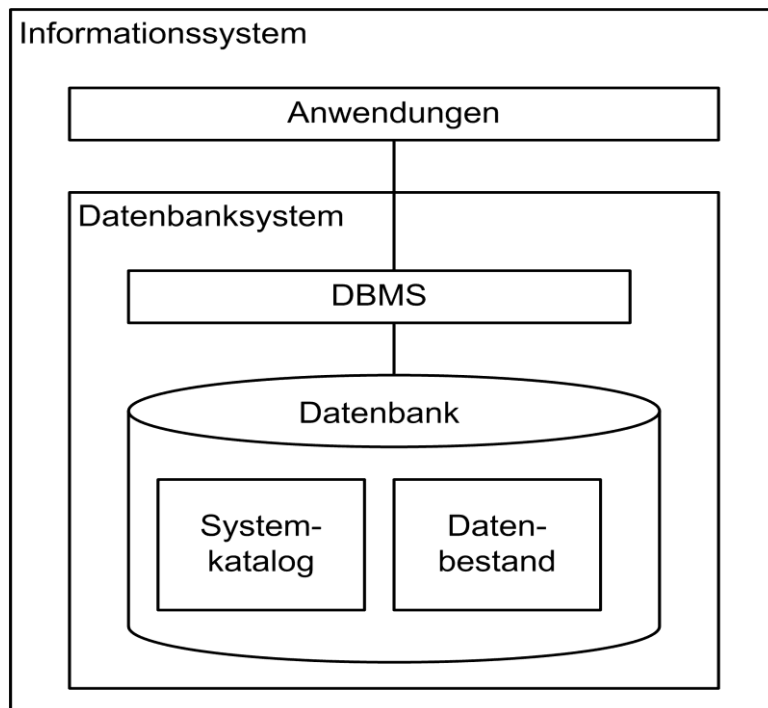
Datenbanksystem (DBS)

Als Datenbanksystem wird ein Datenbankverwaltungssystem zusammen mit einer oder mehrerer Datenbanken bezeichnet.

Informationssystem

Ein Informationssystem ist eine Datenbank zusammen mit allen Programmen, die die Verarbeitung der in der Datenbank gespeicherten Informationen ermöglichen; ein Datenbanksystem ist somit ein Softwarepaket zur Erzeugung von Informationssystemen (ein Informationssystem-Generator).

Wichtig!!!!



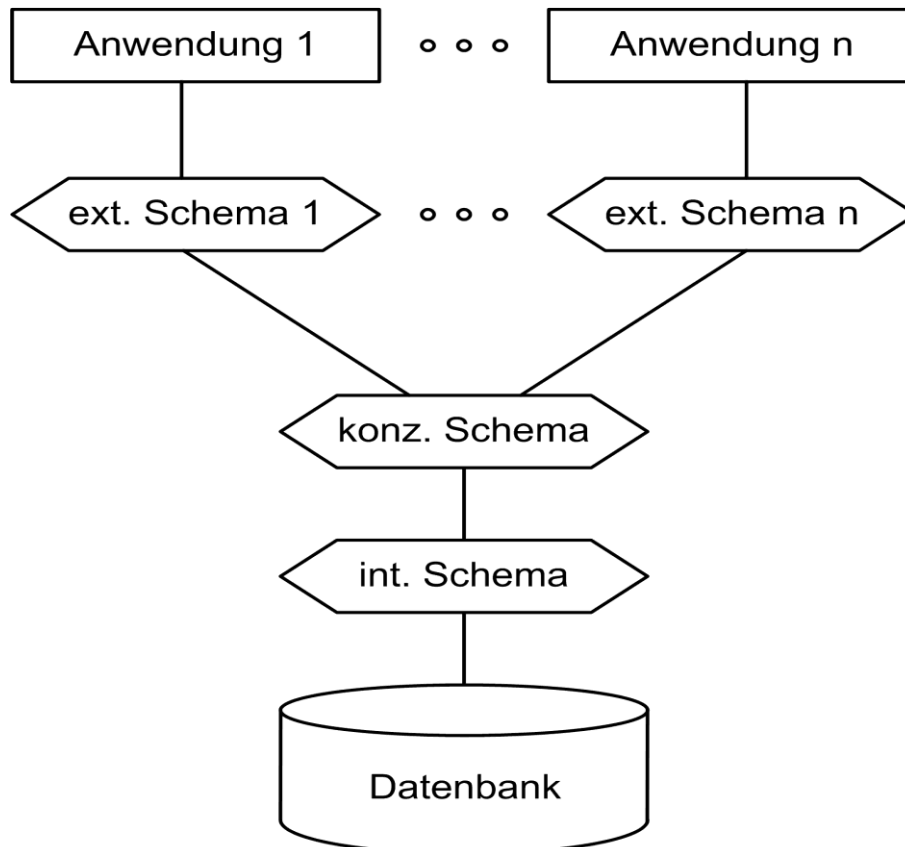
Systemkatalog

Der Systemkatalog (Data Dictionary: DD) ist die im Rechner gehaltene Verwaltungsinformation über einen Datenbestand, d.h. die physische und logische Beschreibung der Datenbank sowie alle Informationen hinsichtlich Benutzung und Benutzungsvorschriften, Speicherbelegung und physische Speicherplatzierung. In einem Datenbanksystem ist das Data Dictionary integriert.

Anforderungen an eine Datenbank

- Keine Redundanz (außer kontrollierbare)
- Gewährleistung von Integritätsbedingungen
- Einhaltung von Regeln

ANSI-SPARC Modell



Datenunabhängigkeit

Die **externe-konzeptuelle Transformation** stellt die Beziehung zwischen der externen und der konzeptuellen Ebene her. Sie sichert die logische Datenunabhängigkeit.

Die **konzeptuelle-interne Transformation** stellt die Beziehung zwischen der konzeptuellen und der internen Ebene her. Sie sichert die physische Datenunabhängigkeit.

Die Datenunabhängigkeit ist die Unabhängigkeit eines Schemas von darunterliegenden Schemata. Änderungen im tiefer liegenden Schema werden durch die Transformation gegenüber dem nächsthöheren Schema verborgen.

Datenunabhängigkeit physisch: Änderungen in der physischen Ebene (internes Schema, physische Abspeicherung) haben keinen Einfluss auf die konzeptuelle und die externen Schemata. Die Schnittstelle zum internen Schema bleibt unverändert. (z.B. neue Indexe oder Verschiebung auf eine andere Platte).

Datenunabhängigkeit logisch: Änderungen in der konzeptuellen Ebene (konz. Schema, logische Datensicht) haben keinen Einfluß auf die externen Schemata. Die Schnittstelle zum konzeptuellen Schema bleibt unverändert. (z.B. neue Objekte).

DML:

Die Datenmanipulationssprache (**data manipulation language : DML**) besteht aus einem Änderungsteil zum Eintragen, Ändern und Löschen sowie einem Anfrageteil zum Lesen der Daten.

Select, insert, update, delete

DDL: Die Datendefinitionssprache (**data definition language : DDL**) übernimmt die Umsetzung der Strukturen und Konsistenzbedingungen des logischen Datenmodells für ein konkretes Datenbanksystem.

Create / drop / alter

DAL: Die Datenadministrationssprache (**data administration language : DAL**) besteht aus einem Speicher-Definitions-Teil (storage description language : SDL) zur Beschreibung der Speicherorganisation und Festlegung von Zugriffen und aus Sprachkonstrukten für die Zugriffskontrolle und Ablaufsteuerung (data control language : DCL).

grant

natural join

Wird für den Verbund *natural* angegeben, so wird die Verbundbedingung als Gleichheit gleichnamiger Felder automatisch hergeleitet.

Achtung: Felder, die für den Verbund (automatisch) verwendet werden, sind danach Verbundfelder und gehören zu keiner der Tabellen mehr.

ACID Prinzip

Transaktion: ACID

- A - Atomicity (Atomizität)
- C - Consistency (Konsistenz)
- I - Isolation (Isolierung)
- D - Durability (Dauerhaftigkeit)

Atomicity (Atomizität)

Alles-oder-nichts-Eigenschaft

Eine Transaktion wird ganz oder gar nicht bearbeitet. Dies bedeutet Abbruch bei jeder Art von Fehler.

Consistency (Konsistenz)

Die Konsistenz-Anforderung garantiert Korrektheit durch Einhaltung semantischer Integritätsanforderungen.

Isolation (Isolierung)

Jede Transaktion läuft für sich so ab, als ob sie allein auf der Datenbank arbeiten würde. Sie sieht zu jeder Zeit eine konsistente Datenbank.

Durability (Dauerhaftigkeit)

Der erfolgreiche Abschluss einer Transaktion macht die durchgeführten Datenbankänderungen „dauerhaft“.

Deadlock

Werden dieselben Daten in 2 Transaktionen in unterschiedlicher Reihenfolge geändert, so kann es zu Deadlocks kommen.

Superschlüssel

Eine identifizierende Attributkombination einer Relation ist ein Superschlüssel der Relation.

Identifikation :

Jedes Tupel hat eindeutige Werte im Schlüsselkandidaten.

Schlüsselkandidat

Eine minimale identifizierende Attributkombination einer Relation ist ein Schlüsselkandidat der Relation.

Identifikation :

Jedes Tupel hat eindeutige Werte im Schlüsselkandidaten.

Minimal :

Wird ein Attribut entfernt, so ist die Kombination nicht mehr identifizierend .

Jede Relation hat einen oder mehrere Schlüsselkandidaten.

Primärschlüssel

Der Primärschlüssel ist ein ausgewählter Schlüsselkandidat der Relation.

Die Schlüsselattribute eines Primärschlüssel müssen belegt sein.
=> Sie dürfen nicht NULL sein.

Fremdschlüssel

Eine Attributkombination heißt Fremdschlüssel (foreign key), wenn die Werte aus demselben Wertebereich stammen wie diejenigen eines Schlüsselkandidaten in derselben oder einer anderen Relation der Datenbank.

Der Fremdschlüssel und der referenzierte Schlüsselkandidat definieren eine Fremdschlüsselbeziehung.

Fremdschlüsselattribute müssen nicht belegt sein (d.h. sie können NULL sein).

Referentielle Integrität

Ein Fremdschlüsselwert ist nur zulässig, wenn der zugehörige Wert des Schlüsselkandidaten (Referenzschlüssel) in der Datenbank aktuell vorhanden ist.

Ein Referenzschlüsselwert darf nicht gelöscht werden, solange noch ein zugehöriger Fremdschlüsselwert vorhanden ist.

Eine Fremdschlüsselbeziehung kann so definiert werden, dass referenzierende Datensätze mit gelöscht werden.

Funktionale Abhängigkeit (FD : Functional Dependency)

Eine Attributkombination C ($C \subseteq A$) heißt funktional abhängig von einer Attributkombination B ($B \subseteq A$), wenn in jedem möglichen Tupel von R die Ausprägung von C durch die Ausprägung von B eindeutig bestimmt ist : $B \rightarrow C$

Unterschied wissen zu :

Volle funktionale Abhängigkeit

Eine Attributkombination C ($C \subseteq A$) heißt voll funktional abhängig von einer Attributkombination B ($B \subseteq A$), wenn C funktional abhängig ist von B und es keine Teilmenge von B gibt, von der C funktional abhängig ist : $B \rightarrow C$.

Normalformen:

Schreibt sicherheitshalber die Normalformen auf die ihr überprüft in Kurzform auf, damit nachvollzogen werden kann, was ihr macht.

1.Normalform (1NF)

Eine Relation R ist in 1NF, wenn alle ihre Domänen nur einfache (elementare, atomare) Werte enthalten

2.Normalform (2NF) Bedingung 1 NF erfüllt

Eine Relation R ist in 2NF genau dann, wenn sie in 1NF ist und jedes nicht-Schlüssel-Attribut voll funktional abhängig von einem Schlüsselkandidaten ist.

3.Normalform (3NF) Bedingung 2 NF erfüllt

Eine Relation ist dann und nur dann in 3NF, wenn sie in 2NF ist und jedes nicht-Schlüssel-Attribut nicht transitiv abhängig ist von einem Schlüsselkandidaten. Sie ist auch dann in 3NF, wenn sich eine transitive Abhängigkeit nur über Schlüsselkandidaten herleitet.

Boyce-Codd-Normalform (BCNF)

Eine Relation R ist in BCNF, wenn jede Determinante von R auch Schlüsselkandidat von R ist.

- BCNF fordert mehr als 3NF
- BCNF basiert nicht auf 1NF und 2NF
- Ist eine Relation in der 3NF und ist der Schlüsselkandidat die einzige Determinante, so ist die Relation in BCNF.

4.Normalform (4NF)

Seien $B, C \subset A$ und $B \subset C := \emptyset$. R ist dann und nur dann in 4NF, wenn gilt : Gibt es in R eine mehrwertige Abhängigkeit, z.B. $B \twoheadrightarrow C$, dann handelt es sich um eine funktionale Abhängigkeit und alle Attribute von R sind von B funktional abhängig (d.h. $B \rightarrow X$ für alle Attribute X von R).

- Die einzige funktionale/mehrwertige Abhängigkeit in einer Relation hat einen Schlüsselkandidat als Determinante.
- 4NF ist eine strengere Forderung als BCNF : Jede 4NF-Relation ist auch BCNF-Relation.
- Jede Relation kann ohne Informationsverlust in einen äquivalenten Satz von 4NF-Relationen zerlegt werden.
- Das Rissanen-Theorem gilt auch für mehrwertige Abhängigkeit.

5.Normalform (5NF)

Eine Relation ist in 5NF, wenn sie in 4NF ist und keine Verbundabhängigkeit existiert.
Vorsicht bei Beseitigung der Verbundabhängigkeiten : Es muss stets der vollständige Satz der Projektion betrachtet werden.

Normalisierungsprozess

1. Führe die unnormalisierten Relationen in normalisierte Relationen über.
2. Bilde Projektionen auf diese Relationen, so dass alle funktionalen Abhängigkeiten eliminiert werden, in denen die Determinante nicht Schlüssel ist. => BCNF
3. Bilde Projektionen dieser BCNF-Relationen, um alle mehrwertigen Abhängigkeiten zu eliminieren, die nicht gleichzeitig funktionale Abhängigkeiten sind. => 4NF
4. Bilde Projektionen dieser 4NF-Relationen, um alle Verbundabhängigkeiten zu eliminieren.
=> 5NF

Anomalien

Einfüge-Anomalie

Es soll ein neuer Kurs eingetragen werden : KNR = 24, Kurs = DBV, Kurspreis = 5.000 => Dies geht jedoch erst, wenn er auch einen Teilnehmer hat.

Änderungs-Anomalie

Der Teilnehmer Müller (TNR = 2100) zieht nach Bochum um. => Alle Datensätze, die die TNR 2100 enthalten, müssen geändert werden

Lösch-Anomalie

Der C-Kurs (KNR = 17) fällt aus. => Wenn Krüger an keinem weiteren Kurs teilnimmt, verschwinden seine Teilnehmerinformationen.