

---

# **Datenbanken I**

Glossar

Autoren: Gunter Schlageter, Wolfgang Wilkes, Michael Balzer

©  
Fachbereich Informatik

Alle Rechte vorbehalten

---

An den einzelnen Kapiteln dieses Kurses haben unterschiedliche Personen mitgearbeitet. Die jeweiligen Mitautoren werden speziell genannt. Zum Autorenteam gehörten:

*M. Balzer, J. Bröking, T. Berkel, F. Laskowski, F. Lenzen, R. Meyer, W. Wilkes.*

Neben den Autoren setzt die Fertigstellung eines solchen Kurses immer auch die Mitarbeit einer ganzen Reihe weiterer Personen voraus: es müssen Zeichnungen erstellt und Texte gestaltet werden, der Autor sucht kritische Korrekturleser, viele technische Einzelaktivitäten sind zu koordinieren, manch ein Autor muß über seine verlorenen Kämpfe mit Winword hinweggetröstet werden ...

Zum Zustandekommen dieses Glossars haben beigetragen:

*R. Braun, K. Grahn, R. Hüwel, U. Manthey, G. Rademacher, U.-B. Reichberg, A. Volmerhaus.*

Sei die Schedule  $S$  eines Transaktionssystems gegeben. Wir nennen  $T_j$  abhängig von  $T_i$ , wenn es zwei Operationen  $a_i$  und  $b_j$  gibt, die in Konflikt zueinander stehen und  $a_i$  vor  $b_j$  in  $S$  ist. Zu  $S$  erzeugen wir den zugehörigen Abhängigkeitsgraphen als gerichteten Graphen wie folgt: eine gerichtete Kante von  $T_i$  nach  $T_j$  wird gezogen, wenn  $T_j$  abhängig von  $T_i$  ist.

### **Abort**

Der Abbruch einer Transaktion wird über den Abort-Befehl ausgelöst. Der Abort-Befehl kann von der Transaktion selber oder vom DBMS angestoßen werden.

### **Algebraische Optimierung**

Die Optimierung von Abfragen, die auf algebraischen Umformungen von Ausdrücken basiert, wird als algebraische Optimierung bezeichnet.

### **Attribut**

Im ER-Modell besitzt jeder Entity-Typ Attribute. Ein Attribut ist eine Eigenschaft, die jedem Entity dieses Typs zukommt. Jedes Attribut kann Werte aus einem bestimmten Wertebereich annehmen. Auch Beziehungstypen können Attribute zugeordnet werden.

*Graphische Darstellung:*

Attribute können in das Rechteck des zugehörigen Entity-Typs bzw. in die Raute des zugehörigen Beziehungstyps eingetragen werden, bei Platzmangel werden sie als Kreise gezeichnet und mit ungerichteten Kanten mit den zugehörigen Rechtecken bzw. Rauten verbunden.

### **Beziehung**

Im ER-Modell betrachtet man Entities und ihre Beziehungen zueinander.

### **Beziehungstyp**

Im ER-Modell werden gleichartige Beziehungen zwischen Entities als Beziehungstyp definiert. Für jeden Beziehungstyp ist seine Komplexität ein wichtiges Merkmal. Man spricht von einem 1:1-Beziehungstyp zwischen Entity-Typ  $E_1$  und Entity-Typ  $E_2$ , wenn zu jedem Entity vom Typ  $E_1$  genau ein Entity vom Typ  $E_2$  gehört und umgekehrt. Bei einem n:1-Beziehungstyp steht jedes Entity vom Typ  $E_2$  zu beliebig vielen Entities vom Typ  $E_1$ . Bei einem n:m-Beziehungstyp können einem Entity vom

Typ E1 beliebig viele Entities vom Typ E2 zugeordnet sein und einem Entity vom Typ E2 können ebenso beliebig viele Entities vom Typ E1 zugeordnet sein.

*Graphische Darstellung:*

Beziehungstypen werden durch Rauten dargestellt, die durch ungerichtete Kanten mit den zugehörigen Entitytypen verbunden werden.

### **Beziehungstyp, schwacher**

Beziehungstypen, an denen ein schwacher Entity-Typ teilnimmt, heißen schwache Beziehungstypen.

### **Binden**

Der Prozeß der Übersetzung von Objekten des externen Modells durch Objekte des konzeptuellen Modells und dann durch Objekte des internen Modells wird als Binden bezeichnet. Das Binden kann zur Übersetzungszeit oder zur Ausführungszeit stattfinden. Binden zur Übersetzungszeit wird als statisches Binden bezeichnet - Binden zur Ausführungszeit wird als dynamisches Binden bezeichnet.

### **Boyce-Codd Normalform (BCNF)**

Eine Relation R ist in Boyce-Codd Normalform, wenn für jede funktionale Abhängigkeit  $X \twoheadrightarrow A$  aus  $F^+$ , A nicht aus X, gilt: X enthält einen Schlüssel für R.

### **Checkpoint**

Ein Checkpoint ist eine Informationsmenge, die zu einem bestimmten Zeitpunkt vom System an das Datenbanksystem übergeben werden kann. Ein Checkpoint enthält im einfachsten Fall die Liste der zu diesem Zeitpunkt aktiven Transaktionen.

### **Closure**

Ist  $R(A_1, \dots, A_n)$  ein Relationenschema mit der Fd-Menge F (Menge aller funktionalen Abhängigkeiten), dann sagt man: F impliziert  $X \twoheadrightarrow Y$ , wenn  $X \twoheadrightarrow Y$  in allen Relationen eines Relationenschemas gültig ist, in denen auch F gültig ist. Die Menge  $F^+$  aller funktionalen Abhängigkeiten, die von F impliziert werden, heißt Closure von F.

### **Commit**

Die korrekte Beendigung einer Transaktion wird dem DBMS durch den Commit-Befehl von der Transaktion mitgeteilt.

### **Cursor**

Ein Cursor bezeichnet ein Konzept, mit dem eine Schnittstelle zwischen mengenorientierter Abfragesprache (z.B. SQL) und satzorientierter Programmiersprache geschaffen wird.

### **Data Definition Language (DDL)**

Die Definition von Daten im konzeptuellen Modell einer Datenbank wird mit Hilfe einer besonderen Sprache, der Data Definition Language durchgeführt.

**Data Dictionary**

Das Data Dictionary ist der zentrale, für alle Mitarbeiter verbindliche Katalog, in dem alle wesentlichen Informationen über die Daten und Programme des Informationssystems gespeichert werden.

Es kann etwa folgende Informationen verwalten: Beschreibungen der Daten, Angaben zu den Beziehungen zwischen den Daten, Beschreibungen der Programme (Transaktionen), Angaben darüber, welche Programme welche Daten nutzen, Konsistenzbedingungen, Angaben über Zugriffsbefugnisse etc.

**Dateisystem**

Ein Dateisystem ist ein Softwarepaket, das den Zugriff auf einzelne Sätze einer Datei besorgt, wenn das Anwendungsprogramm die entsprechenden Parameter zur Verfügung stellt.

**Datenbank**

Eine Datenbank ist eine integrierte Ansammlung von Daten, die allen Benutzern eines Anwendungsbereichs als gemeinsame Basis aktueller Informationen dient.

**Datenbankmanagementsystem:**

Das Datenbankmanagementsystem (DBMS) ist ein Softwaresystem, das es ermöglicht, eine Datenbank zu definieren, Daten zu speichern, zu verändern und zu löschen, so wie Anfragen an die Datenbank zu stellen.

**Datenbanksystem**

Datenbank und Datenbankmanagementsystem (DBMS) bilden zusammen ein Datenbanksystem.

**Datenmanager**

Bei der Implementierung eines Datenbanksystems teilt man die Gesamtfunktionalität in mehrere funktionale Komponenten auf, eine davon ist der Datenmanager. Der Datenmanager bietet eine Mengenschnittstelle an, die von Anwendungsprogrammen direkt genutzt werden kann, er bedient sich dabei der Dienste des Zugriffsmanagers.

**Datenmanipulationssprache**

Sprache für den Zugriff auf eine Datenbank, mit DML abgekürzt

**Datenmodell**

Ein Datenmodell beschreibt mit bestimmten Konstrukten und Bauelementen die Datenwelt eines Unternehmens. Gängige Datenmodelle sind das hierarchische Datenmodell, das Netzwerk-Datenmodell und das relationale Datenmodell. Diese Datenmodelle sind an verfügbare Datenbanksysteme angelehnt, bieten aber nicht ausreichende Ausdrucksmöglichkeiten, um alle wesentlichen Informationen und Beziehungen der Datenwelt abzubilden. Deshalb gibt es neben den genannten Datenmodellen sogenannte semantisch reichere Datenmodelle, wie z.B. das Entity-Relationship-Modell.

**Datenunabhängigkeit**

Datenunabhängigkeit bedeutet, daß Anwendungsprogramme von Änderungen auf der internen und der konzeptuellen Ebene unberührt bleiben. Man unterscheidet

physische und logische Datenunabhängigkeit.

*Physische Datenunabhängigkeit* bedeutet Isolierung der Anwendungsprogramme vor Änderungen der physischen Datenorganisation.

*Logische Datenunabhängigkeit* bedeutet Isolierung der Anwendungsprogramme vor Änderungen des konzeptuellen Modells.

Unter dem Gesichtspunkt des Bindens unterscheidet man statische Datenunabhängigkeit, wenn zur Übersetzungszeit gebunden wird, und dynamische Datenunabhängigkeit, wenn zur Ausführungszeit gebunden wird.

### **DC-System**

Das DC-System (Data Communication System) besorgt die Kommunikation mit den Terminals.

### **Deadlock**

Ein Deadlock tritt auf, wenn zwei Transaktionen jeweils auf die andere warten, um ein Objekt sperren zu können.

### **Entity**

Ein Entity ist ein Begriff aus dem ER-Modell. Ein Entity ist ein Objekt der realen Welt, das für sich allein identifizierbar ist und für den betrachteten Ausschnitt der realen Welt von Relevanz ist.

### **Entity-Typ**

Ein Entity-Typ repräsentiert die Menge aller Entities, die die gleichen charakteristischen Eigenschaften besitzen. Ein Entity-Typ bezeichnet sowohl einen Typ, also eine Klassifizierung von Objekten, als auch die Menge der Objekte.

*Graphische Darstellung:*

Entity-Typen werden durch Rechtecke repräsentiert.

### **Entity-Typ, schwacher**

Entity-Typen, die nur in Beziehung mit anderen Entity-Typen identifizierbar sind, nennt man schwache Entity-Typen.

*Graphische Darstellung:*

Doppelt umrandete Rechtecke bezeichnen schwache Entity-Typen.

### **Entity-Relationship-Modell**

Das Entity-Relationship-Modell (ER-Modell) ist ein semantisch reiches Datenmodell zur Beschreibung der Daten eines Unternehmens. Seine Basiskonstrukte sind: Entity-Typ, Beziehungstyp, Attribut, Instanz

### **Fehler für Recovery**

Fehler, die durch Recovery behoben werden, können folgender Art sein:

- Transaktionsfehler
- Systemfehler

### **hot spot**

Als *hot spots* bezeichnet man besonders intensiv genutzte Teile der Datenbank.

### **Inkonsistenz**

Ein Datenbanksystem kann inkonsistent nach Abbruch einer Transaktion oder Zusammenbruch des Betriebssystems sein:

- es können Zeiger auf falsche Objekte verweisen: *physische Inkonsistenz*
- es können Sätze widersprüchliche oder veraltete Daten enthalten: *logische Inkonsistenz*

### **Integrität, referentielle**

Referentielle Integrität bezeichnet die Eigenschaft einer Datenbank, nur in sich konsistente Werte zu enthalten: jeder Wert in einem Fremdschlüssel ist auch Wert

Integritätsbedingungen sind Bedingungen, die für die Daten in einer Datenbank immer gelten oder Vorschriften für das Ändern von Daten machen. Diese Bedingungen liefern somit die Möglichkeit zur Prüfung der Korrektheit der Daten, sie werden vom DBS automatisch durchgeführt.

### **Internes Schema**

Das interne Schema ist eine Ebene der Datenbankarchitektur, die die gewählte physische Datenorganisation beschreibt. Das interne Schema enthält alle Informationen über den Aufbau der abgespeicherten Daten, die Speicherung der Daten, die Zugriffspfade usw.

### **Kartesisches Produkt**

Unter dem Kartesischen Produkt zweier Mengen versteht man die Menge aller Tupel, deren erste Komponenten aus der ersten Menge und deren übrige Komponenten aus der zweiten Menge stammen.

### **in Konflikt stehende Transaktionen**

Zwei Operationen verschiedener Transaktionen stehen in Konflikt zueinander, wenn

sie auf dasselbe Objekt zugreifen und mindestens eine der Operationen ein *write* ist.

### **Livelock**

Wenn eine Transaktion nie zu Ende durchgeführt werden kann, weil die Lock-Zuteilung "unfair" ist, dann spricht man von einem Livelock.

### **Log**

Ein Log ist die Mitschrift (eines Teils) der Aktivitäten der Transaktionen auf der Datenbank. Wir unterscheiden physische und logische Logs.

- Ein *physisches Log* enthält Information über die Werte von Datenbankobjekten vor und nach den Änderungen durch Transaktionen.
- Ein *logisches Log* enthält Information über die durchgeführten Operationen

### **Lock-Manager**

Der Lock-Manager ist der Teil des DBMS, der die Sperren verwaltet.

### **lost update**

Mit *lost update* bezeichnet man eine Situation, in der wegen Synchronisationsfehlern eine Änderung im Datenbanksystem verloren gegangen ist.

### **Mengendifferenz**

Unter der Mengendifferenz  $R - S$  zweier Relationen  $R$  und  $S$  versteht man die Menge aller Tupel aus  $R$ , die nicht auch in  $S$  enthalten sind. Voraussetzung für die Bildung der Mengendifferenz ist, daß die Relationen  $R$  und  $S$  den gleichen Grad und die gleichen Wertmengen enthalten.

### **Normalform, erste**

Eine Relation heißt in erster Normalform, wenn die Werte der Wertebereiche jedes Attributs unteilbare Werte sind und nicht ihrerseits wieder aus Mengen oder Tupeln bestehen.

### **Normalform, zweite**

Eine Relation  $R$  ist in zweiter Normalform, wenn jedes Nichtschlüsselattribut  $A$  von  $R$  voll funktional abhängig von jedem Schlüssel  $X$  von  $R$  ist.

### **Normalform, dritte**

Ein Relationenschema  $R$  mit Fd-Menge  $F$  ist in dritter Normalform, wenn für alle  $X \rightarrow A$  aus  $F^+$  mit  $A$  nicht aus  $X$  gilt:  $X$  enthält einen Schlüssel für  $R$  oder  $A$  ist

### **Optimistische Verfahren**

Optimistische Verfahren gehören zu den Synchronisationsverfahren.

Anders als bei Sperrverfahren werden bei den optimistischen Verfahren Operationen niemals verzögert, sondern immer sofort zur Ausführung gebracht. Erst wenn eine Transaktion Commit ausführen will, wird geprüft, ob diese Transaktion an einer nicht-serialisierbaren Schedule beteiligt ist, falls ja, wird die Transaktion abgebrochen - sonst erfolgt Commit.

### **Phantom**

Ein Phantom ist ein Datenbankeintrag, der wegen Synchronisationsfehlern im Datenbanksystem nicht konsistent vorhanden ist, sondern unerwartet auftaucht.

**Primärschlüssel**

Gibt es mehrere Schlüssel für einen Entity-Typ im ER-Modell, so wird im allgemeinen einer von ihnen als Primärschlüssel ausgezeichnet.

**Projektion**

Sei  $R$  eine Relation vom Grad  $n$  mit den Attributen  $a_1, a_2, \dots, a_n$ .

$r[j]$  bezeichne die  $j$ -te Komponente von  $r$ . Ist

$L = (i_1, \dots, i_s)$ ,  $i_k$  aus  $\{1, \dots, n\}$ ,  $k = 1, \dots, s$

eine Liste von Spaltennummern, so ist  $r[L]$  definiert als

$r[L] = (r[i_1], \dots, r[i_s])$ .

Die Projektion von  $R$  auf  $L$  ist dann definiert als

$R[L] = \{r[L] \mid r \text{ aus } R\}$ .

Es werden also nur die Spalten mit den Nummern  $i_1, \dots, i_s$  der Relation  $R$  betrachtet.

**QbE, Query-by-Example**

Query-by-Example (QbE) ist eine Abfragesprache, bei der der Benutzer seine Abfrage nicht in einer linearen Notation aufschreibt, sondern im Dialog am Bildschirm erstellt, wobei er einen speziellen graphischen Editor benutzt.

Diese Art der Abfragesprache hat die Intention, den Endbenutzer, der über wenig formale Kenntnisse verfügt, an Abfragesprachen heranzuführen.

**Recovery**

Die Recovery bezeichnet die Wiederherstellung konsistenter Zustände nach Fehlern.

**Recovery-Manager**

Der Recovery-Manager ist der Teil eines DBMS, der die Recovery nach einem Fehler durchführt.

**Recovery-Strategie**

Eine Recovery-Strategie beschreibt, inwieweit im Fehlerfall REDO oder UNDO von Transaktionen notwendig wird.

**Redo-Protokoll**

Von einem Redo-Protokoll spricht man, wenn folgende Abarbeitungsreihenfolge eingehalten wird:

Bevor eine Transaktion Commit ausführen kann, müssen alle Objektveränderungen in der Datenbank oder im Log gespeichert sein.

**Relation**

Sind  $D_1, D_2, \dots, D_n$  Mengen von Werten, so ist

$R$  Teilmenge von  $D_1 \times D_2 \times \dots \times D_n$  eine  $n$ -stellige Relation über den Mengen

$D_1, D_2, \dots, D_n$

$n$  ist der Grad der Relation.

**Relationenalgebra**

In der Relationenalgebra spezifiziert man gewünschte Relationen durch Angabe einer Folge von Operationen, mit der die Relationen aufgebaut werden sollen. Der Benutzer wendet spezielle Operationen auf Relationen an, um seine gewünschte Relation zu konstruieren.

**Relationenkalkül**

Im Relationenkalkül spezifiziert man gewünschte Relationen in deskriptiver Weise, d.h. ohne Angabe, welche Operationen zum Aufbau der Relation verwendet werden sollen. Mit Hilfe des Prädikatenkalküls wird die Menge der gewünschten Tupel beschrieben: es wird ein Prädikat (eine Bedingung) angegeben, das die Tupel

**Relationsschema**

Ein Relationsschema  $R(A_1, \dots, A_n)$  spezifiziert eine Relation mit Namen  $R$  und mit den Attributen  $A_1, \dots, A_n$ . Jedem Attribut  $A_i$  ist ein Wertebereich  $\text{dom}(A_i)$  zugeordnet.

**Rollback, fortgeplanter**

Mit "Rollback" oder "Zurücksetzen" bezeichnet man die Operationen, die ein DBMS veranlaßt, um eine Transaktion ungeschehen zu machen. Wenn der Rollback nicht nur die verursachende Transaktion selbst betrifft, sondern noch weitere, spricht man von einem fortgeplanten Rollback.

**Rücksetzbarkeitsbedingung**

Um das Rücksetzen bereits abgeschlossener Transaktionen in einem Transaktionssystem zu vermeiden, definiert man Rücksetzbarkeitsbedingungen. Es werden folgende unterschieden:

- Rücksetzbarkeitsbedingung fürs Lesen
- Rücksetzbarkeitsbedingung für Schreiben
- Strikte Ausführung

**Satz**

Ein Satz besteht aus einzelnen Datenelementen, auch Felder genannt. Sätze sind die Zugriffseinheiten bzw. Verarbeitungseinheiten der Anwendungsprogramme.

**Schedule**

Ein System paralleler Transaktionen erzeugt eine Folge von Lese- und Schreiboperationen, die als Schedule oder auch Historie bezeichnet wird.

**Schlüssel**

Im ER-Modell sind Entities als voneinander unterscheidbare, identifizierbare Einheiten definiert. Die Merkmale eines Entities - d.h. Attribute -, die es identifizieren, werden als Schlüssel bezeichnet.

**Selektion**

Eine Selektion selektiert aus einer Relation alle Tupel, die eine gegebene Bedingung erfüllen. Die Bedingung ist für jedes Tupel für sich nachprüfbar.

**Serialisierbarkeit**

Ein paralleles System von Transaktionen heißt serialisierbar, wenn es mindestens eine (gedachte) serielle Ausführung derselben Transaktionen gibt, die

1. denselben Datenbankzustand und
2. dieselben Ausgabedaten der Transaktionen liefert

Ein serialisierbares System von Transaktionen ist korrekt synchronisiert.

### **Serialisierbarkeitskriterium**

Für die Serialisierbarkeit von parallelen Transaktionen gilt das Serialisierbarkeitskriterium mit: Ein Transaktionssystem ist dann serialisierbar, wenn der zugehörige Abhängigkeitsgraph zyklensfrei ist.

### **Shadowing**

Mit Shadowing wird ein Recovery-Verfahren bezeichnet, das mit Haupt- und Hilfskatalog für alle Objekte arbeitet, um UNDO und REDO-Abläufe zu vermeiden.

### **Sperre**

Eine Sperre sperrt den Zugriff auf Objekte in der Datenbank. Sperren sind ein Mittel zur Synchronisation. Man unterscheidet Lesesperren, *shared lock*, und Schreibsperren, *exclusive lock*.

### **Sperrhierarchie**

Wenn ein DBMS Sperren auf unterschiedlichen Ebenen, wie z.B. Tupel, Relation, Datenbank, zuläßt, dann spricht man von einer Sperrhierarchie.

### **Sperrprotokoll**

Ein Sperrprotokoll gibt an, nach welchem Verfahren Sperren für die Synchronisation eines Transaktionssystems vergeben und wieder entfernt werden. Beispiele für Sperrprotokolle sind

- Zwei-Phasen-Sperrprotokoll
- Striktes Zwei-Phasen-Sperrprotokoll
- Preclaiming

### **SQL, embedded SQL, dynamisches SQL**

SQL - Structured Query Language - ist die am meisten verbreitete Abfragesprache für relationale Datenbanksysteme. Sind SQL-Ausdrücke in eine Programmiersprache eingebettet, dann spricht man von embedded-SQL. Beim embedded-SQL unterscheidet man statisches und dynamisches SQL: beim statischen SQL wird jedes SQL-Statement fest vorgegeben, beim dynamischen SQL kann ein beliebiges SQL-Statement an das bereits kompilierte Programm übergeben werden.

### **Synchronisation**

Die Synchronisation bezeichnet die Kontrolle der Parallelarbeit mehrerer Benutzer.

### **Synchronisationsverfahren**

Ein Synchronisationsverfahren sorgt für die Serialisierbarkeit eines Transaktionssystems. Man unterscheidet folgende Vorgehensweisen:

- Verifizierende Verfahren, die erst mit Abbruch einer Transaktion eingreifen, wenn die Schedule nicht mehr serialisierbar ist.
- Präventive Verfahren, die dafür sorgen, daß keine nicht-serialisierbare Schedule entsteht.

### **Systempuffer-Manager**

Bei der Implementierung eines Datenbanksystems wird die Gesamtfunktionalität auf mehrere funktionale Komponenten verteilt, eine davon ist der Systempuffer-Manager. Der Systempuffer-Manager stellt den höheren Schichten des DBMS Seiten zur Verfügung, auf denen sich die benötigten Daten befinden.

### **Transaktion**

Eine Transaktion ist eine logische Arbeitseinheit in einem Anwendungsprogramm, die entweder ganz oder gar nicht durchgeführt wird. Aus Sicht des DBMS ist eine Transaktion eine Einheit, welche die Datenbank von einem konsistenten Zustand wieder in einen konsistenten Zustand überführt. Für Transaktionen gelten die Eigenschaften:

#### *Unteilbarkeit (atomicity)*

Eine Transaktion ist eine unteilbare Verarbeitungseinheit; sie wird entweder ganz oder überhaupt nicht ausgeführt.

#### *Konsistenz (consistency)*

Eine korrekte Ausführung der Transaktion führt die Datenbank von einem konsistenten zu einem konsistenten Zustand.

#### *Isolation (isolation)*

Eine Transaktion muß so ablaufen, als sei sie die einzige im System. Zwischenzustände (die ja inkonsistent sein können) dürfen für andere Transaktionen nicht sichtbar sein.

#### *Dauerhaftigkeit (durability)*

Ergebnisse einer erfolgreich beendeten Transaktion sind dauerhaft, d.h. überleben jeden nachfolgenden Fehler.

Diese Eigenschaften werden häufig mit dem Begriff ACID abgekürzt.

### **Transaktionsmanager**

Der Transaktionsmanager ist die Komponente im DBMS, die zuständig ist für die korrekte Synchronisation der Transaktionen.

### **Tupelvariable, frei oder gebunden**

Eine Tupelvariable ist eine Variable, die ein Tupel einer Relation bezeichnet.

Eine Tupelvariable ist in einem Ausdruck gebunden, wenn sie durch einen Existenz- oder Allquantor eingeführt wird, sonst heißt sie frei.

### **Undo-Protokoll**

Von einem Undo-Protokoll spricht man, wenn folgende Reihenfolge eingehalten wird: Bevor ein geändertes Objekt in die Datenbank geschrieben werden darf, muß sein Before-Image im Log gespeichert sein.

Das Gegenstück zum Undo-Protokoll ist das Redo-Protokoll.

### **Undo/Redo-Prinzip**

Das Undo- und das Redo-Prinzip bezeichnen zwei Arbeitsverfahren, die der

Recovery-Manager zur Verfügung stellen und nach Bedarf einsetzen muß:

#### UNDO

Der Recovery-Manager muß zu jedem Zeitpunkt alle bereits durchgeführten Änderungen einer aktiven Transaktion rückgängig machen.

#### REDO

Der Recovery-Manager muß zu jedem Zeitpunkt die von einer abgeschlossenen Transaktion durchgeführten Änderungen wiederherstellen können.

Es seien A in der Relation R und B in der Relation S Attribute, nach denen R und S verknüpft werden sollen. Sowohl A als auch B müssen den gleichen Wertebereich besitzen. Dann ist der Verbund  $R[A@B]S$  folgendermaßen definiert:  $R [A@B] S = \{rs \mid r \text{ aus } R, s \text{ aus } S, r[A] @ s[B] \}$  wobei rs die Konkatenation bezeichnet und @ ein arithmetischer Vergleichsoperator ist.

#### Vereinigung

Sind R und S zwei Relationen mit gleichem Grad und gleicher Wertemenge der Attribute, dann versteht man unter der Vereinigung von R und S die Vereinigung der Tupelmengen von R und S.

#### View

Die Sicht eines Benutzers auf die Daten im Datenbanksystem heißt View.

#### Zeitstempel-Verfahren

Das Zeitstempel-Verfahren ist ein mögliches Synchronisationsverfahren.

Bei den Zeitstempel-Verfahren (*timestamp ordering*) werden Transaktionen nicht über Sperren von Objekten synchronisiert, sondern auf der Basis der zeitlichen Reihenfolge, in der diese im DBMS ankommen.

#### Zugriffsmanager

Bei der Implementierung eines Datenbanksystems wird die Gesamtfunktionalität auf mehrere funktionale Komponenten verteilt, eine davon ist der Zugriffsmanager. Der Zugriffsmanager stellt eine Schnittstelle zur Verfügung, in der einzelne Tupel und logische Zugriffspfade angesprochen werden können. Er bildet die Schicht zwischen Systempuffer-Manager und Datenmanager.