

## Grundwissen zu relationalen Datenbanken

Die Funktion einer relationalen Datenbank besteht in der elektronischen Verwaltung von Daten in Computersystemen. Die Basis für relationale Datenbanken bildet das relationale Datenbankmodell, das 1970 von Edgar F. Codd vorgestellt wurde und bis heute zu den etablierten Standards für Datenbanken gehört. Das Datenbankmanagementsystem, das zu einer relationalen Datenbank gehört, wird relationales Datenbankmanagementsystem oder RDBMS, was für Relationale Database Management System steht, genannt und um die Daten abzufragen oder zu manipulieren, wird üblicherweise die Datenbanksprache SQL verwendet.

Das Konzept von relationalen Datenbanken basiert auf der Relation, die im Sinne der Mathematik klar definiert die Beschreibung einer Tabelle darstellt. Die Operationen beruhen auf der relationalen Algebra, die damit zeitgleich auch die theoretische Basis der Datenbanksprache SQL darstellt. Obwohl dem Datenbankmodell eine mathematische, abstrakte Definition zugrunde liegt, kennzeichnen sich relationale Datenbanken durch eine flexible und im Vergleich zu anderen Datenbankmodellen einfache Handhabung.

### **Das Konzept von relationalen Datenbanken**

Vereinfacht erklärt handelt es sich bei einer relationalen Datenbank um eine Ansammlung von Tabellen mit darin gespeicherten Datensätzen. Jede Tabelle ist dabei eine Relation und jede Zeile, die als Tupel bezeichnet wird, stellt einen Datensatz dar.

Ein Tupel wiederum gliedert sich in mehrere Tabellenspalten, die jeweils einen Wert einer Eigenschaft enthalten. Das Relationenschema definiert, durch wie viele Attribute eine Relation beschrieben wird und welcher Art diese Eigenschaften sind.

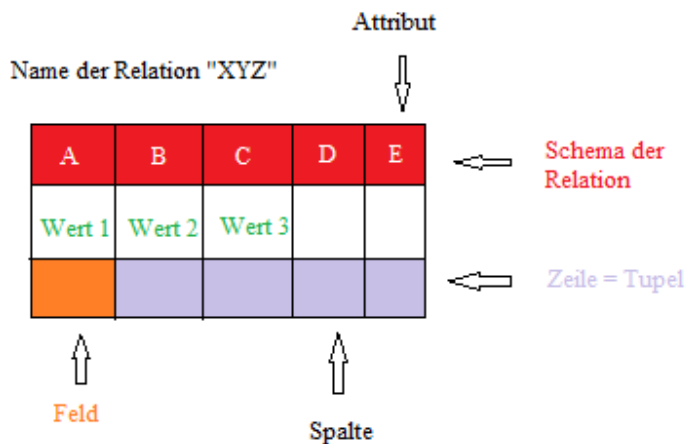
Im Fall einer Personaldatenbank würde eine Relation beispielsweise wie folgt aussehen:

Relation „Personal“

<b>Person anr.</b>	<b>Name, Vorname</b>	<b>Abteilung</b>	<b>Position</b>	<b>Vertragsart</b>	<b>Eintritt am</b>
1	Martin Müller	Produktion	Helfer	Unbefristet	03.04.1999
2	Lisa Schmidt	Sekretariat	Angestellte	Unbefristet	15.08.2008
3	Peter Klein	Versand	Abteilungsleiter	Unbefristet	02.01.2009
4	Tina Maier	Versand	Hilfskraft	Befristet	15.07.2009

Ein Datensatz muss immer eindeutig identifizierbar sein, was mithilfe eines oder mehrerer Schlüssel sichergestellt wird. Im obigen Beispiel wäre die Personalnummer ein solcher Schlüssel. Ein Schlüssel darf nicht verändert werden, da er sich nicht auf die Position in der Tabelle, sondern auf den jeweiligen Datensatz bezieht.

Allgemein gliedert sich eine relationale Datenbank somit folgendermaßen:



## Die Beziehungen zwischen den Tabellen

Um die Tabellen in entsprechende Beziehungen zueinander zu setzen, können Verknüpfungen erstellt werden.

Im Fall einer Personaldatenbank könnte es beispielsweise eine Tabelle „Arbeitszeit“ geben, die neben der Personalnummer die Anzahl der Soll-Arbeitszeit, der Arbeitstage, der Urlaubstage und der Krankheitstage erfasst. Zudem könnte es eine Tabelle „Lohn/Gehalt“ geben, die neben der Personalnummer die Lohngruppe sowie die ausgezahlte Vergütung enthält.

Die Sollarbeitszeit sowie die Lohngruppe wären die Schlüssel in diesen Relationen und würden die Tabellen als Fremdschlüssel mithilfe der Personalnummer mit einem Eintrag aus der ersten Tabelle verbinden. Enthält eine Tabelle keine Fremdschlüssel, wird sie als flache Tabelle bezeichnet.

## **Die Theorie von relationalen Datenbanken**

Die theoretischen Grundlagen entwickelte Edgar F. Codd und beschrieb sie in seiner Arbeit „A Relational Model of Data for Large Shared Data Banks“. Alle Operationen beruhen auf der relationalen Algebra, die als algebraisches Modell beschreibt, in welcher Form Daten gespeichert, abgefragt und bearbeitet werden können. Die grundlegenden Operationen, die zeitgleich auch die Basis für alle anderen Operationen bilden, sind

- die Projektion,
- die Selektion,
- das Kreuzprodukt,
- die Umbenennung,
- die Vereinigung und
- die Differenz.

Wird mittels SQL eine Anfrage an eine relationale Datenbank gestellt, übersetzt das Datenbankmanagement die Anfrage auf diese Operatoren. In der praktischen Anwendung gibt es zwar noch weitere Operatoren, allerdings handelt es sich bei diesen letztlich auch nur um eine Kombination aus mehreren der grundlegenden Operationen.

Allerdings ist es mithilfe der relationalen Algebra nicht möglich, rekursive Anfragen zu berechnen. So wäre es beispielsweise nicht möglich, alle Personen zu berechnen, die jemals in einer bestimmten Position tätig waren, wenn die einzelnen Positionen in einer Relation mit der jeweiligen Personalnummer gespeichert sind und es nur eine Verbindung zum jeweiligen Vorgänger gibt.

In diesem Fall müsste die Ermittlung der Informationen durch mehrere Abfragen erfolgen. Durch die Einführung von SQL-99 wurde jedoch eine erweiterte relationale Algebra generiert, die nun auch eine Operation vorsieht, durch die die Berechnung der transitiven Hülle realisierbar wird.

## **Das Schema von relationalen Datenbanken**

Das Schema einer relationalen Datenbank definiert die gespeicherten Daten in der Datenbank sowie deren Beziehungen zueinander. Das Erstellen eines Schemas wird als Datenmodellierung bezeichnet. Anwendung hierbei findet beispielsweise das Entity-Relationship-Modell, durch das ein Konzept des Schemas entworfen und anschließend mithilfe eines Datenbankmanagementsystems implementiert werden kann.

Ein solcher Arbeitsschritt heißt logischer Entwurf oder Datenmodellabbildung. Ein wesentlicher Schritt bei der Modellierung ist die Normalisierung, deren Ziel darin liegt, Anomalien zu verhindern und Redundanzen zu reduzieren.

Dadurch soll einerseits die Konsistenz der Daten sichergestellt und andererseits die Wartung der Datenbank vereinfacht werden. Nach Codd gibt es hierfür vier Normalformen, die bis heute bei Entwürfen von relationalen Datenbanken verwendet werden, zwischenzeitlich jedoch um weitere Normalformen erweitert wurden.

### **Die zwölf Regeln für relationale Datenbanken**

1985 hat Codd zwölf Regeln formuliert, die definieren, wann es sich um eine relationale Datenbank im strengen Sinne handelt.

**Darstellung von Informationen:** In einer relationalen Datenbank müssen alle Informationen wie beispielsweise Daten, Attribute oder Sicherheitsinformationen logisch in Tabellen aufgeführt sein.

### **Zugriff auf Daten:**

Jeder Wert muss durch die Kombination von Tabellename, Schlüssel und Attribut gefunden werden können. Insofern darf sich in einer Tabelle an einem Schnittpunkt zwischen der Zeile und der Spalte nur ein Wert befinden.

### **Systematische Behandlung von Nullwerten:**

Sind Nullwerte in Attributen kein Teil des Schlüssels, bezeichnen sie fehlende Informationen und müssen durchgehend gleich behandelt werden.

### **Struktur der Datenbank:**

Nicht nur die Daten, sondern auch die Struktur der Datenbank wird in Tabellen gespeichert. Sämtliche Tabellen, die zu der Datenbank gehören, werden dazu in einer Tabelle als Katalog erfasst und müssen zugänglich sein.

### **Die Abfragesprache:**

Üblicherweise wird in relationalen Datenbanken SQL verwendet. Generell muss die verwendete Abfragesprache jedoch mindestens die Funktionen Datendefinition, Definition von Views, Definition von Integritätsbedingungen, Definition von Transaktionen und Definition von Berechtigungen unterstützen.

**Aktualisieren von Views:**

Grundsätzlich können alle Views auch durch das System aktualisiert werden.  
Abfrage und Edition von ganzen Tabellen: Operationen müssen ermöglichen, nicht nur einzelne Sätze, sondern auch ganze Tabellen abzufragen und zu editieren.

**Physikalische Unabhängigkeit:**

Der Nutzer muss unabhängig davon, wie Daten gespeichert sind oder in welcher Form der physikalische Zugriff erfolgt, auf die gespeicherten Daten zugreifen können. Insofern dürfen Anwendungen nur die logische Systemstruktur nutzen.

**Logische Unabhängigkeit von Daten:**

Es darf keine logische Änderungen von Anwendungen oder Zugriffen geben, wenn alle Informationen nach einer Änderung erhalten bleiben, beispielsweise wenn eine Tabelle in zwei Tabellen aufgeteilt wird.

**Unabhängigkeit der Integrität.**

Die Integritätsbedingungen müssen durch die Abfragesprache definiert und in Tabellen aufgeführt werden. Dabei muss das System mindestens überprüfen, ob die Vollständigkeits- sowie die Beziehungsintegrität gegeben sind und es einen eindeutigen Schlüssel ohne Nullwert gibt.

**Verteilung der Daten:**

Die Anwendungen aus einer Datenbank dürfen sich nach der Aufspaltung in eine verteilte Datenbank hinsichtlich ihrer Logik nicht ändern.

### **Unterlaufen der Abfragesprache:**

Wird in einer Datenbank eine Low-Level-Abfragesprache verwendet, darf es nicht zu einem Unterlaufen der Integritätsbedingungen der High-Level-Sprache kommen.

Nicht alle Regeln werden gleich gewichtet, wenn ein Datenbanksystem beurteilt wird. Daher sprechen einige Hersteller schon dann von einer relationalen Datenbank, wenn von ihrer Datenbank die wichtigsten Kriterien erfüllt werden.

Ende 1990 differenzierte Codd seine zwölf Regeln in einem Buch über relationale Datenbanken. Durch die Erweiterung auf 333 Regeln legte er damit die Relationalität einer Datenbank bis ins kleinste Details fest.

### **Alternative Konzepte zur Verwaltung von Daten**

Zur Verwaltung von Daten in anderen Strukturen gibt es neben dem relationalen Datenbankmodell weitere alternative Konzepte. Diese sind zwar weniger bedeutend oder konnten sich nicht durchsetzen, ermöglichen jedoch in einigen Fällen eine einfachere Datenanbindung.

- In den 1960er- und 70er-Jahren wurden hierarchische Datenbanksysteme und Netzwerk-Datenbanksysteme zur betrieblichen Verarbeitung von Daten genutzt. Die Daten- und Tabellestruktur wird hierbei in der Entwurfsphase festgelegt und kann bei einer Abfrage nicht verändert werden. In einigen Sonderfällen werden solche Datenbanksysteme auch heute noch eingesetzt.
- Infolge des Aufkommens von objektorientierten Programmiersprachen stieg auch das Angebot an Objektdatenbanken. Dadurch können Objekte direkt in die Datenbank integriert werden, die Übersetzung der Objekte auf eine relationale Tabellenstruktur entfällt. Dies ist dann von Vorteil, wenn komplexe Daten gespeichert werden sollen, die nicht oder nur bedingt in flachen relationalen Tabellenstrukturen dargestellt werden können. Nachteile gegenüber relationalen Datenbanken entstehen jedoch, wenn große Datenmengen verarbeitet werden müssen, da in diesem Fall über mehrere Pfadarten auf Objekte zugegriffen wird.

Von objektrelationalen Datenbanken wird gesprochen, wenn einer relationalen Datenbank objektorientierte Eigenschaften hinzugefügt werden. Objektrelationale Datenbanken bilden Objekte jedoch nicht direkt in der Programmiersprache ab, sondern verwenden lediglich das Vererbungskonzept bei Definitionen und Anfragen von Tabellen mit ähnlich strukturierten Feldern, wodurch deren Handhabung dann vereinfacht wird.

Semistrukturierte Datenbanken gehören zu den neueren Konzepten. Im Unterschied zu herkömmlichen relationalen Datenbanken haben semistrukturierte Datenbanken kein streng festgelegtes Schema. Eine solche Datenbank ist dann baumartig aufgebaut und jede Einheit

der Datenbank mit dem gleichen Typ kann eine unterschiedliche Anzahl von Attributen haben.

Ein Beispiel für eine solche Datenbank ist eine XML-Datenbank, die hierarchisch aufgebaut ist und nach XML-Definition geformte XML-Daten mit beliebigen Strukturen enthalten kann. Die Abfrage der Daten erfolgt beispielsweise über XQuery oder Xpath, die Manipulation der Daten mittels proprietären Spracherweiterungen. Im Vergleich zu relationalen Datenbanken ist die Performance von XML-Datenbank jedoch geringer.

### **Kritikpunkte zu relationalen Datenbanken**

Trotz der vielen positiven Eigenschaften gibt es auch einige Kritikpunkte am relationalen Datenbankmodell.

Wird ein Objekt gespeichert, wird es im Rahmen der relationalen Darstellung auf mehrere unterschiedliche Relationen aufgeteilt. Üblicherweise sind Anwendungsobjekte komplex, was bedeutet, dass sie sich selbst ebenfalls aus mehreren Objekten oder Listen zusammensetzen. Das relationale Modell kennt jedoch nur aus Werten zusammengesetzte Tupelmengen.

Daher wird es bei einer Abfrage durch das Datenbankmanagementsystem notwendig, komplexe Anwendungsobjekte durch zahlreiche, hintereinander ausgeführte Operationen aus den einzelnen Relationen wiederherzustellen.

Dadurch kann es zu unübersichtlichen Abfragen kommen, die jedes Mal auf ihre Anpassungsnotwendigkeit hin überprüft werden müssen, wenn das Anwendungsobjekt strukturell verändert wird. Der Einsatz von hintereinander ausgeführten Operationen, den sogenannten Joins, der eine Unterstützung durch passende Datenbank-Indizes erforderlich macht, führt dazu, dass der Zugriff auf ein Objekt sowohl im Zusammenhang mit dem Ressourcenbedarf als auch hinsichtlich des Entwicklungsaufwandes deutlich aufwändiger ist.

Um Tupel eindeutig zu identifizieren, wird es teilweise erforderlich, künstliche Schlüssel einzusetzen. Auf diese Weise kann beispielsweise die Größe eines Schlüssels, der als Fremdschlüssel definiert werden soll, verkleinert oder es können Gehört-zu-Beziehungen einbettet werden. Durch künstliche Schlüssel werden somit Attribute in eine Relation integriert, die keine Bedeutung für die abstrakte Beschreibung eines Anwendungsobjektes sind, sondern lediglich die Funktion von Verwaltungsinformationen haben.

In vielen relationalen Datenbanken werden Datenmanipulationssprachen verwendet, deren Mächtigkeit eingeschränkt ist. Dadurch entsteht vielfach die Notwendigkeit von Schnittschnellen mit mächtigeren Programmiersprachen. Dies kann aber wiederum zu einer schwierigeren Handhabung führen, beispielsweise wenn das mengenorientierte SQL in dem satzorientierten C++ verarbeitet werden soll. Allerdings gibt es relationale Datenbanken, in denen mit beispielsweise PL/pgSQL in PostgreSQL oder PL/SQL in Oracle mächtige Programmiersprachen vorhanden sind.

In einer relationalen Datenbank ist es nicht möglich, das typische Verhalten von einem Objekt in einer Anwendung zu beschreiben. Eine solche Beschreibung ist damit nur

außerhalb der Datenbank möglich und erfordert eine entsprechende Anwendungssoftware. Nutzen jedoch mehrere Anwendungen einen identischen Datenbestand, kann eine redundante Implementierung die Folge hiervon sein.

**Weiterführende Informationen für Programmierer und Anwender:**

**Fachzeitschriften für Computeranwendungen und Internet**

[http://www.deutsche-fachzeitschriften.de/fachzeitschriften\\_computeranwendungen\\_und\\_internet\\_.html](http://www.deutsche-fachzeitschriften.de/fachzeitschriften_computeranwendungen_und_internet_.html)

**Ajax Lernprogramme**

<http://www.online-lernprogramme.de/index.php/Lernprogramm-Ajax/>

**Projektplanung mit PHP**

[http://www.projektarbeit-projektplanung.de/projektplanung\\_php/index.php](http://www.projektarbeit-projektplanung.de/projektplanung_php/index.php)

Copyright by [www.eu-datenbank.de](http://www.eu-datenbank.de)