
2 Informationsmodellierung

Petra Sauer

Die Informationsmodellierung beschäftigt sich mit der (formalen) Beschreibung der Datenbestände eines Anwendungsbereiches. Als eine der Kerntätigkeiten des Datenbankentwurfs hat sie entscheidenden Einfluss auf die Qualität einer Datenbank (DB). Als Beschreibungsformalismus werden das Entity-Relationship-Modell (ERM) (→ 2.2) oder der Standard UML (Unified Modelling Language) /2.10/ verwendet.

2.1 Datenbankentwurf

Das Hauptanliegen des Datenbankentwurfs ist die Definition der Struktur der DB. Zur Umsetzung der **Drei-Ebenen-Architektur** (→ 1.4) gehört die Entwicklung der **externen** sowie des **konzeptuellen** und **internen Schemas**. Als weiteres Entwurfsdokument wird das logische Schema definiert. Die Entwicklung der Schemata unterliegt einer strengen Reihenfolge und führt von einer zunächst abstrakten zu einer zunehmend konkreten Beschreibung der DB. Die **Abhängigkeit von der konkreten Realisierung mit einem DBMS wird dabei immer größer**; **zunächst von dessen Datenmodell (DM)**, schließlich von **System und DB-Sprache**. Die **Unabhängigkeit der frühen Entwurfsdokumente** vom DBMS **ermöglicht eine stabile Bezugsbasis** für unterschiedliche Implementierungen der DB.

2.1.1 Phasenmodell

Der Entwurf einer DB findet meistens im Zusammenhang mit der Entwicklung eines Informationssystems (→ 1.6) statt und orientiert sich an den Entwurfsschritten des Phasenmodells des Software Engineering. **Die Phasen des Datenbankentwurfs sind angepasst an die Bedürfnisse der Datenbankentwicklung**: Anforderungsanalyse, konzeptioneller Entwurf, logischer Entwurf, Datendefinition auf Basis einer Datendefinitionssprache (Data Definition Language, DDL), physischer Entwurf, Implementierung und Wartung (→ Bild 2.1).

- *Hinweis:* In diesem Kapitel wird das grundsätzliche Vorgehen bei der Entwicklung einer DB behandelt. In Abhängigkeit von der geplanten Architektur der DB können weitere Phasen hinzukommen. Soll eine verteilte DB entwickelt werden, so ist ein Verteilungsentwurf vor dem logischen Entwurf vorzunehmen (→ 13.2). Ist eine DB auf Basis existierender Datenbestände zu entwerfen, ist eine Integration der vorhandenen Datenbanken vorzunehmen. Hierzu existieren

verschiedene Ansätze wie u. a. die DB-Föderation, die Gegenstand von Kapitel 13 sind.

Wie auch im Software Engineering werden die Phasen nicht streng sequenziell durchlaufen. Der **Datenbankentwurf** ist ein **iterativer Prozess**.

2

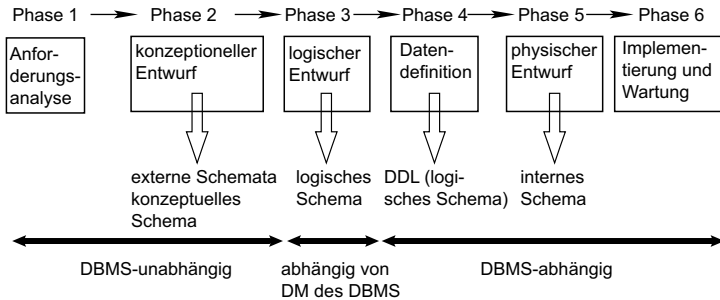


Bild 2.1 Phasenmodell des Datenbankentwurfs

2.1.2 Anforderungsanalyse

Die Anforderungen der potenziellen **Benutzer** an die DB **werden erfasst, analysiert** und **klassifiziert**. Die Beschreibung erfolgt *informal* in Form von Texten, Tabellen etc. Dabei werden **Informations-** von **Bearbeitungsanforderungen** **unterschieden**.

Informationsanforderungen werden im Rahmen der **Datenanalyse** erfasst und beschreiben die Daten unabhängig von der Auswertung.

Der DB-Entwurf beschäftigt sich mit den Informationsanforderungen.

Bearbeitungsanforderungen werden im Rahmen der **Funktionsanalyse** erfasst und beschreiben die Auswertungs- und Bearbeitungserfordernisse auf der DB.

Zu den **Bearbeitungsanforderungen** gehören Angaben über

- die **Art der** auszuführenden **Bearbeitungsprozesse**, wie z. B. Anfragen, Updates, Berichtsgenerierung,
- **Quantitäten und Prioritäten** der **auszuführenden Bearbeitungsprozesse**, z. B. Häufigkeit einzelner Prozesse, Datenvolumen, Zuwachsraten, Anforderungen an die Effizienz,
- **Sicherheits- und Schutzanforderungen**, z. B. Zugriffsrechte und Zugangsbestimmungen.

Der Funktionsentwurf auf Basis der Bearbeitungsanforderungen ist Gegenstand des Software Engineering /1.8/ und wird hier nicht behandelt.

2.1.3 Konzeptioneller Entwurf

Datenbanken stellen ein abstrahiertes Abbild der Realität dar. Da es sich dabei immer nur um Teilbereiche handelt, ist dafür auch der Begriff *Weltausschnitt* (**Universe of Discourse, UoD**) gebräuchlich.

- *Beispiel:* Weltausschnitte können sein: die Abteilung eines Unternehmens (z. B. Vertrieb), ein komplettes Unternehmen, eine bestimmte Funktion des Unternehmens (z. B. Web-Präsenz).

Auf Basis der Ergebnisse der Anforderungsanalyse erfolgt im konzeptionellen Entwurf eine *erste formale Beschreibung* des abzubildenden Weltausschnitts, auch als **Modellierung des Universe of Discourse** bezeichnet. Diese Beschreibung erfolgt **unabhängig von der Realisierung in einem konkreten DBMS und unabhängig von konkreten Anwendungen**, um eine stabile Basis für die weiteren Entwurfsphasen zu besitzen. Als Beschreibungsformalismus wird ein **abstraktes Datenmodell**, wie das **ERM** eingesetzt. Die Anforderungen der Benutzer werden in formale Schemabeschreibungen überführt. **Daraus resultieren das konzeptuelle und die externen Schemata, Letztere auch als Sichten (views) bezeichnet.**

Für die Schemaentwicklung in dieser Phase existieren zwei Ansätze:

- Beim **Top-Down-Ansatz** wird zunächst das konzeptuelle Schema modelliert. Externe Schemata werden als **nicht notwendig disjunkte Ausschnitte des konzeptuellen Schemas** abgeleitet.
- Beim **Bottom-Up-Ansatz** werden die externen Schemata modelliert. Das konzeptuelle Schema resultiert aus deren Integration. Externe Schemata spiegeln die Anforderungen einzelner Anwender wider. Diese können sich überschneiden und Widersprüche enthalten. Eine **wichtige Tätigkeit** im Rahmen der Schemaintegration ist die **Analyse der Schemata**, die **Erkennung von Konflikten und deren Auflösung**.

Bei der **Analyse der Sichten** werden **Konflikte zwischen Sichten, Redundanzen und Widersprüchlichkeiten** (Inkonsistenzen) **aufgedeckt**. Konflikte treten u. a. auf:

- als **Namenskonflikt**, bei **unbeabsichtigten Namensüberschneidungen oder -abweichungen durch** die Verwendung von **Synonymen und Homonymen**,
- als **Typkonflikt**, indem **verschiedene Datenstrukturen** für das **gleiche Element**, basierend auf unterschiedlichen Aussagen von Benutzern zum UoD, modelliert werden,

- als **Bedingungskonflikt**, durch die Verwendung unterschiedlicher Integritätsbedingungen in Sichten,
- als **Strukturkonflikt**, durch die Verwendung von verschiedenen Modellierungsvarianten zu einem Sachverhalt.

Bei der *Schemaintegration* werden diese Konflikte gelöst und Beziehungen zwischen den Sichten identifiziert (→ 13.7.3).

2.1.4 Logischer Entwurf

Auf Basis des konzeptuellen Schemas wird eine *weitere formale Beschreibung* der Datenstrukturen vorgenommen, wobei der Beschreibungsformalismus wechselt. Das logische Schema wird modelliert.

Das **logische Schema** beschreibt die Datenstrukturen des UoD in Abhängigkeit vom konkreten Datenmodell des DBMS (Datenbankmodell).

Beispielsweise ist in dieser Phase die Entscheidung zu treffen, ob die Realisierung der DB mit einem konventionellen, meist relationalen DBMS (→ 3) oder einem objektorientierten oder objektrelationalen DBMS (→ 10) erfolgen soll. Es werden damit Voreinstellungen hinsichtlich der Modellierungskonzepte des Zielsystems vorgenommen. Die Entscheidung für das konkrete Zielsystem kann später fallen.

Die Entwicklung des logischen Schemas erfolgt in zwei Schritten (→ Bild 2.2):

1. Im ersten Schritt kommt es zur **Transformation** des konzeptuellen Schemas auf ein (initiales) logisches Schema. Wird hier vom ERM auf das relationale Datenmodell abgebildet, kann dieser Schritt weitestgehend automatisiert werden, da ein etabliertes System von Transformationsregeln (→ 3.3) existiert.
2. Der zweite Schritt dient der Verbesserung bzw. **Nachbearbeitung** des logischen Schemas. Ein wichtiges Optimierungskriterium ist die Minimierung der Redundanz. Weitere Kriterien, wie schneller Zugriff auf die Daten, können teilweise konkurrieren. Liegt das logische Schema im relationalen Datenmodell vor, erfolgt in diesem Schritt die **Normalisierung** (→ 3.4). Mit Hilfe der Normalisierung können relationale Datenstrukturen so gestaltet werden, dass Einfüge-, Änderungs- und Löschoperationen auf den Datenbeständen ohne Probleme (*Anomalien*) ablaufen. Dazu werden Redundanzen durch Aufteilung von Relationen schrittweise reduziert.

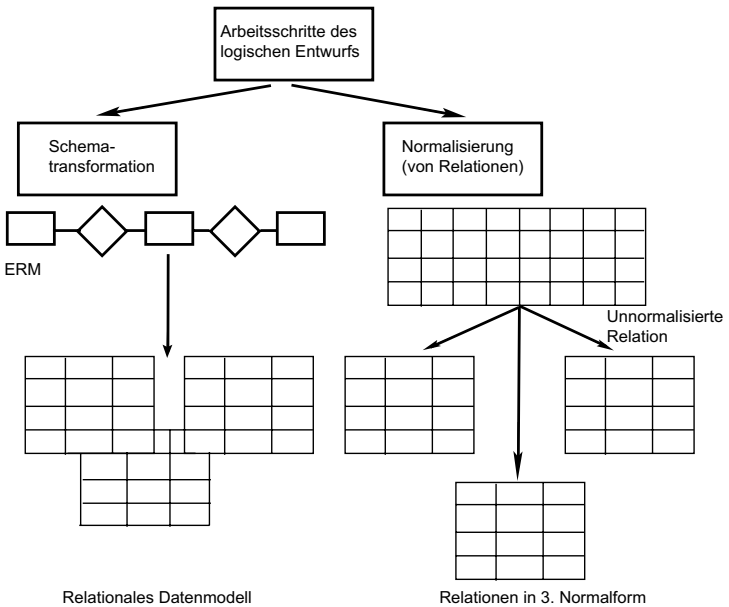


Bild 2.2 Arbeitsschritte des logischen Entwurfs

2.1.5 Datendefinition

In der **Datendefinitionsphase** werden die **entwickelten logischen und externen Schemata** mit Hilfe der **Datenbanksprache** des DBMS **beschrieben**. Für die **Beschreibung des logischen Schemas** steht die **DDL**, für die **Beschreibung der Sichten** die **Sichtdefinitionssprache** (View Definition Language, VDL) als Teil der DDL zur Verfügung.

Mit Hilfe der DDL werden beim relationalen Datenmodell Relationen, Attribute und Wertebereiche definiert. **DDL** und **VDL** sind **Bestandteile** der Sprache **SQL** (→ 4). Obwohl SQL standardisiert ist, existieren Dialekte, die in dieser Phase eingesetzt werden. Die **Datendefinitionen** sind also **vom DBMS abhängig**.

Neben den Datenstrukturen werden die Integritätsbedingungen mit den Mitteln der Datenbanksprache beschrieben, wie beispielsweise die Primär- und Fremdschlüsselbeziehungen beim relationalen Datenmodell. **Ergebnis** dieser Phase **sind** die **DB-Schemata** für das **konkrete DBMS**.

2.1.6 Physischer Entwurf

Gegenstand des physischen Entwurfs ist die **Entwicklung des internen Schemas** mit Hilfe der **Speicherstruktursprache** (Storage Structure Language, SSL) des DBMS. Dazu gehört die Definition der geeigneten Speicherform und möglicher Zugriffspfade für die Daten, z. B. der effektive Zugriff auf die Werte häufig genutzter Attribute über Indexdateien.

Ziel des physischen Entwurfs ist es, eine möglichst **hohe Leistungsfähigkeit des DBS** zu gewährleisten. Dazu gehören **geringe Zugriffszeiten, hohe Durchsatzraten von Transaktionen** und die **optimale Ausnutzung des Speicherplatzes**. In gewisser Weise erfolgt in dieser Phase eine **Optimierung der DB-Definition** (→ 9).

2.1.7 Implementierung und Wartung

Der **Entwurfsprozess schließt mit** der tatsächlichen **Installation des DBS** in der Umgebung ab, in der der Dauerbetrieb realisiert wird. Hierzu gehört **auch die Erfassung oder Generierung der Datenbestände** mit Hilfe der Datenmanipulationssprache (Data Manipulation Language, DML) des DBMS. **Bei Übernahme aus anderen (DB)-Systemen** kommen **weitere Migrationsaufgaben zum DB-Entwurf** hinzu. Hierzu gehört u. a. der **Abgleich der Datenstrukturen** sowie die **Entwicklung von Konvertierungsprogrammen**.

Mit der Inbetriebnahme der DB **beginnt** die **Wartung**. Das **DBS unterliegt einer fortlaufenden Anpassung an neue Anforderungen**, wie u. a. **strukturelle Veränderungen, Optimierung der physischen Realisierung der DB, Anpassung an neue Systemplattformen, Portierung auf neue DBMS**.

2.2 Grundlagen des Entity-Relationship-Modells (ERM)

Das ERM gehört zu den semantischen Datenmodellen. Als abstraktes Datenmodell wird es zum Entwurf des konzeptuellen Schemas sowie zur Modellierung von Benutzersichten in der Phase des konzeptionellen Entwurfs eingesetzt. Es ermöglicht eine **erste formale Beschreibung** des Weltausschnitts, noch unabhängig von Datenmodell und DBMS der Realisierung der DB-Anwendung. Als allgemeiner Beschreibungsformalismus für Datenstrukturen jedweder Anwendung ist es der De-facto-Standard der Informationsmodellierung.

Entwickler des ERM ist Peter P. Chen. Er stellte die grundlegenden Bestandteile des ERM im Jahre 1976 vor /2.2/. Diese ersten Ideen von Chen werden