

Infoblatt : Datenbankmodelle

Logische Datenmodelle

- Hierarchisches Modell
- Netzwerk Modell
- Relationales Modell
- Objektorientiertes Modell

Hierarchische Datenmodell

Das hierarchische Datenmodell entstand aus konventionellen Dateisystemen zur Verwaltung von Dateien mit komplex-variabel langen Sätzen, d. h. Sätzen, bei denen beliebig viele Wiederholungsgruppen mit variabler Anzahl von Wiederholungen und beliebig geschachtelt vorkommen. Die Grundidee ist nun, dass man alle diese Wiederholungsgruppen vom eigentlichen Satz trennt und in jeweils eine eigene neue sequentielle Datei einstellt; so erhält man eine Menge sequentieller Dateien aus Sätzen unterschiedlichen Typs, die einander hierarchisch untergeordnet und entsprechend untereinander verzeigert (verpointert) sind, eine so genannte Dateihierarchie. Zugriffspfade dienen dem eindeutigen Zugriff auf Entities in einer Hierarchie: Dies besagt also, dass ein Entity immer und nur erreichbar ist über das zugehörige Wurzel-Entity als Einstiegspunkt in die Hierarchie und über einen von hier ausgehenden gerichteten Pfad, in dem keine Lücke sein darf.

Die grundsätzliche Schwierigkeit beim hierarchischen Modell liegt darin, dass sich zwischen übergeordneten und untergeordneten Entity-Typen immer nur 1:1- oder 1:n-Beziehungen darstellen lassen. Jede m:n-Beziehung muss unter Einführung von Wiederholungen (Redundanzen) in m getrennte 1:n-Beziehungen aufgelöst werden.

Netzwerk Modell

Das netzwerkorientierte Datenbankmodell ist aus dem hierarchischen Datenbankmodell hervorgegangen.

Jeder Entity-Typ kann wie bei der Baumstruktur mehrere Nachfolger haben.

Im Gegensatz zum hierarchischen Modell kann er aber mehrere Vorgänger besitzen.

Es kann mehrere Entity-Typen geben, die keinen Vorgänger haben (Wurzeln).

Im Netzwerkmodell lassen sich daher alle Beziehungen zwischen Entity-Typen in einfacher Weise darstellen. Besteht zwischen den Entities zweier Entity-Typen eine m:n-Beziehung, wird ein sogenannter "Verbindungs-Entity-Typ" (Komplexer Objekttyp) eingerichtet, z. B. "arbeitet für" zwischen Mitarbeiter und Projekt.

Objektorientiertes Datenbankmodell

Ein objektorientiertes Datenbanksystem organisiert und speichert Informationen als Objekte. Die Inhalte der Datenbank können so viel realitätsnäher und v. a. direkter Benutzern und Anwendungsprogrammen zur Verfügung gestellt werden: Ein Auftrag, eine Abteilung, ein Kunde, ein Projekt usw. ist jeweils ein Objekt (und nicht eine Datensammlung, die sich über mehrere Tabellen verteilt). Objektorientierte Datenbanken eignen sich daher in besonderer Weise für komplexe Anwendungen mit viel Semantik im Datenmodell.

Die wichtigsten Forderungen an ein objektorientiertes Datenbanksystem lauten:

- Unterstützung komplexer, d.h. zusammengesetzter Objekte
- Unterstützung von Objekttypen und Klassenbildung
- Vererbung von Objekteigenschaften über die Klassenhierarchien
- Kapselung der Objekte (= Schutz!) durch Methoden, d.h. Zugriff nur über vorher definierte Schnittstellen
- Polymorphie der Methoden (die gleiche Methode kann bei verschiedenen Objekten verschiedene bewirken, z.B. die Methode "stornieren" bei einem laufenden Kundenauftrag oder einer eigenen Bestellung)