Zusammenfassung Datenbank EDD

Dienstag, 17. Februar 2015

08:36

Inhalt

[2. Notwendigkeit von Datenbanken 2](#_Toc422414414)

[3. Grundbegriffe 2](#_Toc422414415)

[4. Anomalien 3](#_Toc422414416)

[5. Datenbankarchitektur 4](#_Toc422414417)

[6. Modellierung 6](#_Toc422414418)

[7. ERM (Entitäten-Beziehungsmodell) 6](#_Toc422414419)

[8. UML 8](#_Toc422414420)

[9. Relationenmodell / ERD (Entity Relationship Diagram) 10](#_Toc422414421)

[10. Rezept 13](#_Toc422414422)

[11. Normalisierung 13](#_Toc422414423)

[12. Joins (Pyramiden-System) 15](#_Toc422414424)

[13. Access 16](#_Toc422414425)

# Notwendigkeit von Datenbanken

* Daten und Prozesse strukturiert abbilden
* Redundanzen vermeiden (Mehrfachspeicherung von identischen Daten)
* Inkonsistenzen vermeiden (wenn mehrere Benutzer gleichzeitig benutzen bzw. zeitgleich arbeiten)
* Datenschutz gewährleisten (Zugriffsrechte oder Verschlüsselungen um vor unbefugtem Zugriff zu schützen)
* Datensicherheit (Vorkehrungen gegen Datenverlust)
* Datenunabhängigkeit (keine Abhängigkeit von Anwendungsprogrammen, sondern sind zentral in der DBMS)

# Grundbegriffe

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Datenbank | Ist eine **Sammlung von nicht-redundanten Daten**, die von mehreren Applikationen effizient benützt werden kann. Sie ist **eigenständig und flexibel**. Sie hat eine Datenbasis (also alle Daten) und das dazugehörende Datenbankverwaltungssystem (wie die Sachen abgespeichert werden) dazu enthält die Datenbank eine **Modellierung von realen Daten.**   |  |  | | --- | --- | | Menschen (Anwender) | Definiert Datenbanken und Funktionen, manipuliert und gibt Daten ein. | | Anwendungsprogrammen (Applikationen) | Durch Prozesse wird der Zugriff auf die Datenbestände veranlasst. Sie sind direkte Benutzer des Systems. | | Metadaten: (Eigene Definitionen) | Sind statisch solange Gültigkeitsdauer. Sind Namen und Datentypen der Felder (also Spalten-/ Zeilenbeschriftung). Werden von Menschen definiert mit der Sprache Data Definition Language (DDL). „Die Datenbank über die Datenbank“ nennt man dann Metadaten. | | Systemkatalog oder Data Dictionary | Die Gesamtheit aller gespeicherten Metadaten. | | Datenbasis (Eigene Kundendaten) | Ist dynamisch, kann also beliebig verändert werden. Sind die eigentlichen Nutzdaten. Das Manipulieren dieser Nutzdaten (löschen, einfügen) wird mit der Sprache Data Manipulations Language (DML) vollzogen. | | Database Management System (DBMS): (Access) | Vermittelt zwischen dem Anwender und der Datenbankmaschine. Es ist das Betriebssystem der Datenbank. Es ermöglicht den Mehrbenutzerbetrieb mit verschiedenen Zugriffsrechten und synchronisiert die Datenbank gleichzeitig. Sie bietet als Sprache SQL (Structured Query Language) mit dem die Befehle ausgeführt werden. | | Datenbankmaschine (Database Engine): (Hintergrund im Access) | Umhüllt die Datenbasis und die Metadaten. Regelt jegliche Manipulation (Veränderung) der Daten. Beinhaltet eine automatische Sicherungsfunktion und schreibt (persistiert) die Daten auf den Datenträger. Macht durch den Benutzer veranlasste Abfragen (Queries). Macht jede (Um-)Definition in den Metadaten und bewirtschaftet sie. | |
| Operationale Datenbanken - Online Transaction Processing (OLTP): | Sind dynamisch und ändern sich laufend. Reflektion: Zeitpunktbezogen |
| Analytische Datenbanken - Online Analytical Processing (OLAP): | Sind statisch, werden rückblickend angeschaut, also in der Vergangenheit erstellt. |

# Anomalien

|  |  |
| --- | --- |
| Flat Tables | **Eine Tabelle mit allen Informationen**, obwohl es sinnvoll wäre sie zu trennen. Solche Tabellen besitzen keine Fremdschlüssel und sind **anfällig auf Anomalien bzw. Inkonsistenzen**. |
| Mutationsanomalie | **Gleiche Attribute eines Datensatzes werden nicht automatisch geändert**. Man muss per Hand alle Einträge mühsam aktualisieren und es darf kein Fehler unterlaufen ansonsten **führt es zur Inkonsistenz**.  Computergenerierter Alternativtext: i d_pe r s  03.01.2005  425  05.10.2005  425  42s  11.11.2005  45 8  12.12.2005  10.02  n ame  505 Müller  Meier  Meier  Schulze  Schmid  15  ah t  h u dge t  kurs#  kurs name  Verkauf  Verkauf  Rhetorik  Führung  Rhetorik  Verkauf  Oracle  15  15.  2005  09. 2005  2005  72'000 XI 7  72 '000 Æ1  72 '000 L9S  33 '500  61 '900 XI 7  61 '900 1,31  Um z.B. die Abteilung von Meier zu ändern, braucht es 3 Änderungen, damit die Daten konsistent bleiben. |
| Einfügeanomalie | Da **nicht immer alle Werte eingegeben werden können**, zwingt es uns folgende Werte zu nehmen:   * NULL-Werte (oder Null-Marken) * Redundante-Werte * Phantasie-Werte   Computergenerierter Alternativtext: i d_pers  42 S  05.10.2005  425  425  11.11.2005  45 g  15.09.2005  703  10.02.2005  812  12.03.2005  name  505 Müller  Me er  Mei  Schulze  S chrnid  Knecht  abt #  15  18  18  abt  budget  kurs*  kurs name  Verkauf  verkauf  Rhetorik  Führung  Rhetorik  Verkauf  Oracle  Logisti  da tum  03  15  12  01.2005  og  2005  12  2005  12 '000 x17  72 '000 Lgl  72'ooa  33'500  61 '900 XI 7  61  3'500 T26 |
| Löschanomalie | Löschung eines Datensatzes **zwingt uns zu weiteren Löschungen** von nicht gewollten Informationen.  Computergenerierter Alternativtext: i d_per s  505  03.01.2005  425  05.10.2005  425  15.09.2005  11.11.2005  458  15.09.2005  703  703  10.02.2005  name  Me i er  Meier  425 Meier  Schulze  Schmid  15  15  26  ah t  h u dge t  kurst  kurs name  Verkauf  Rhetorik  Führung  Rhetorik  Verkauf  Oracle  da t um  12  12  2005  2'000X1  72 '000 L91  72 '000 1,98  33 ' soc  61 '900 XI 7  7331  Wenn Meier gelöscht wird, verschwindet gleichzeitig kurs\_name "Führung". |

# Datenbankarchitektur

Drei-Schichten-Architektur bzw. ANSI/SPARC-Modell

Computergenerierter Alternativtext:


|  |  |
| --- | --- |
| Externe Ebene | **Benutzerschicht**  Eingabemaske für Benutzer, welches einen massgeschneiderten Ausschnitt der Daten zur Verfügung stellt. |
| Konzeptionelle Ebene | **Struktur der Daten**  Konzept der Informationsstruktur aus sachlogischer Sicht.  Computergenerierter Alternativtext: benutzer  ent aus  Aus dieser Form entsteht ein relationales Modell, aus welchem später die Tabellen mit den entsprechenden Daten entstehen. |
| Interne Ebene | **Physikalische Speicherstruktur und Zugriffspfade der Datensätze**  Da viele Personen an vielen Computern genau eine Datenbank benutzen, muss eine entsprechende Systemarchitektur entworfen werden. Man nennt dies eine **verteilte Umgebung** (distributed Environment).   |  |  | | --- | --- | | Client (Frontend) | Computerprogramm nimmt Befehle entgegen, prüft es und **gibt es weiter**. Ausserdem stellt es die **Ergebnisse in geeigneter Form dar**.  => dienstfragende | | Server (Backend) | Nimmt die Befehle des Clients und **verarbeitet es** und gibt die **Resultate und Quittung zurück.**  => dienstleistende | | Middleware | Wenn zwischen DB-Server und dem Webserver aufwendige Berechnungen gemacht werden (Rabattsätze, automatisches Nachbestellen), so muss noch **ein Anwendungsserver** oder **Applikationsserver** her. Das nennt man **Multi-Tier C/S-Architektur**. Die Applikationsdienste tragen auch die Sammelbezeichnung **Middleware.** |     Computergenerierter Alternativtext: Abbildung 1(): Multi-Tier Client/Server    **Client und Server sind Prozesse** und keine Geräte! Können auf gleichen Computern laufen, aber ist in der Regel auf verschiedenen Systemen. |

# Modellierung

Der Verwendungszweck bestimmt, welche Informationen in das Modell einfliessen und bestimmt auch ob ein entsprechendes Modell geeignet ist oder nicht. Qualität eines Modells wird an folgenden semiotischen (sprachlichen Zeichen) Aspekten gemessen.

Bsp. ÖV-Streckenplan

|  |  |  |
| --- | --- | --- |
| Syntax | Darstellung der Sprache (Modellierungssprache)  => Qualität der Form | Farbcode, Kriese, Linien, Zonen, Texte |
| Semantik | Bedeutung der Sprache  => Qualität der Bedeutung | Linienbezeichnung, Haltestation, Strecke der Bahnen |
| Pragmatik | Wirkung der Sprache  => Qualität der Verwendung | Fortbewegung |

# ERM (Entitäten-Beziehungsmodell)

|  |  |
| --- | --- |
| Abstraktion | Weg von der Realwelt in eine Modellwelt. Wichtige Eigenschaften werden vom Modell extrahiert, verallgemeinert und dargestellt.  => in eine einfachere Form abbilden |
| Modelle | Vereinfachte Abbildung der Realität. Zweck ist es, dass eine experimentelle Manipulation der abgebildeten Strukturen und Zustände erstellt wird. Abstraktion ist der Prozess und Modell das Ergebnis. |
| Schemata | Grafische Darstellung eines Modells. Helfen dem Nutzer das Innenleben einer Datenbank zu verstehen. |
| Intension | Statische Datenbankmodelle |
| Extension | Dynamischer Zustand der Datenbasis |
| Schritte zu einer Datenbank | Der Computer arbeitet ausschliesslich in einer Modellwelt. Reale Informationen müssen in Daten umgewandelt werden. Dafür könnte man folgendermassen vorgehen:   1. **Analyse:** Aufnahme der realen Gegebenheiten 2. **Erstellung des konzeptionellen Schemas:** Mit Werkzeugen das Modell grafisch visualisieren 3. **Umsetzung in ein internes Schema:** Datenbank wird im Computersystem erstellt 4. **Abbilden der Geschäftslogik:** Programme schreiben welche Aktionen an er DBMS ausführen |
| Weg zum ERM | Das ERM besteht aus Informations-Beziehungen. Man soll untersuchen, wie viele minimale und maximale Beziehungen zu den einzelnen Entitäten vorhanden sind. Für die Planung ist eine grafische Darstellung zwingend. |
| Entitäten,  Entitätstypen | Individuelle, einzelne Objekte wie z.B. verschiedene Bücher oder Autoren, welche isoliert werden, um später zu modellieren.  Computergenerierter Alternativtext: Buch «Datenbanken», Buch «Visual C# 2012 — Kochbuch», Buch «Datenbankpro-  grammierung mit .NET 4.5» u.v.a.m.  Autor «Elmar Dehler», Autorin «Johanna Wiebusch», Autor «Gunter Saake»  u.v.a.m.  Verlag «Europa Lehrmittel», Verlag «mitp», Verlag «Galileo Press» u.v.a.m.  Entitäten werden in Entitätstypen oder Entitätsmengen zusammengefasst und repräsentieren die Menge aller real vorkommenden Objekte.  Computergenerierter Alternativtext: Buch  Au/or  Verl ag |
| Assoziationen,  Assoziationstypen,  Beziehungen,  Beziehungstypen | Die Informationsflüsse, welche zwischen den Entitäten laufen, nennen wir Assoziationstypen (nur in eine Richtung) oder Beziehung/-typen/-mengen (in beide Richtungen). Zwischen zwei Entitäten gibt es immer zwei Assoziationstypen. Werden in Metadaten (System Tables) verwaltet und sind in der physischen Ablage der DB nicht sichtbar. |
| Multiplizität | Drückt das erlaubte Minimum (0 oder 1) sowie das erlaubte Maximum (1 oder n) aus. |
| Kardinalität (Komplexität) | Zahleneigenschaft (Multiplizität) von Beziehungen. Muss beidseitig angegeben werden.  Computergenerierter Alternativtext: A 三 コ 0 ← I イ ・ こ ← マ re 朝 姦 ” : コ 【 0 新 z 朝 9  Xettr 1 コ = -  8  e  ヾ  ヾ  0  0  ズ 0 ー 。 ギ 一 、 0  っ 0 ↓ ま 0 ・ お  キ 宅 ュ  一 3 ミ 1-  : - よ よ 0 コ 切 1 : ニ 1 【 0 ち 9  よ 10 ネ 0 ー -  I ネ ^ き ミ : : 製 |
| Attribute | Eigenschaften, die jede Entität eines Entitätstyps individuell charakterisieren (Strukturschreibweise). |
| Modellierungsregeln | Computergenerierter Alternativtext: Das bildet den relevanten Realitätsausschnitt ab in ein Modellsys-  tem mit Entitätstypen, Beziehungstypen, Multiplizitäten und Attributen.  Entitätstypen erhalten sprechende Namen und stehen im Singular.  Beschriften Sie mit Kleinbuchstaben, ohne Umlaute und in männlicher Form.  Vermeiden Sie Homonyme, also die Namensgleichheit fiir mögliche, unterschiedliche Sachver-  halte: Lieferung (An- Oder Auslieferung?)  Venneiden Sie Synonyme, also die mögliche Bedeutungsgleichheit bei unterschiedlichen Na-  men: mit Adressat und Empfänger sind möglicherweise die gleichen Entitäten gemeint.  Beziehungstypen erhalten in beide Richtungen je eine sprechende Bezeichnung. Diese drückt  den sachlogischen Zusammenhang zwischen den Entitätstypen aus. Aus genanntem Grund  sollte nie «hat» und «ist'» verwendet werden.  Die beiden Assoziationstypen geben mittels Angaben der Multiplizität Auskunft über Optiona-  lität (minimal O Oder I) und mögliche Häufigkeit (maximal I Oder mehrere) der konkreten Be-  ziehungen an.  Die Attributslisten der Entitätstypen werden in der sogenannten Strukturschreibweise «autor  (name, vorname, festgehalten. |
| Übersicht | |  |  |  | | --- | --- | --- | | **Realität** | **ERM** | **Beispiel** | | Objekt | Entität | Autor "Dürrenmatt" | | Objekttyp/-klasse | Entitätstyp/-menge | Autor | | Beziehung | Beziehung | "Dürrenmatt" schreibt "Die Physiker" | | Beziehungstyp/-klasse | Beziehungstyp/-menge | Autor schreibt Buch | | Zahleneigenschaft | Kardinalität | Jeder Autor schreibt eine oder mehrere Bücher.  Jedes Buch wird von einem Autor geschrieben. | | Eigenschaft | Attribut | autor (name, vorname, geburtsdatum..) | |

# UML

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Basiselemente | Im Gegensatz zum klassischen Methode (Krähenfüsschen), wird noch die Richtung und nummerische Anzahl angegeben. Ausserdem wird Entitätstyp Klasse und Entität Objekt genannt.  Computergenerierter Alternativtext: besteht aus  lerngruppe  10,  Student  33  eine krngruppe besteht aus min. und max. 33 studenten   |  |  | | --- | --- | | Assoziationsklasse | Wird mit gestrichelten Linien dargestellt.  Computergenerierter Alternativtext: kunde  kauft  kauf  anzahl  artikel  kunde kauft anzahl artikel | | Rollen | Angabe der Rolle einer Klasse in Bezug auf die Assoziation.  Computergenerierter Alternativtext: angsteltter  *Mitarbeiter  abteilung | |
| Erweiterungen | |  |  |  |  | | --- | --- | --- | --- | | Generalisierung/ Spezialisierung | Zur Darstellung einer hierarchischen Beziehung zwischen Entitätstypen bzw. Klassen. Der Subtyp erbt alle Merkmale des Supertyp und kann um einige Attribute ergänzt werden => is-a Beziehung  Computergenerierter Alternativtext: kunde  person  (Generalisierungsmenge: Rolle der Person)  angestellter  vollständig und disjunkt?  lieferant   |  | | --- | | **Vollständig**  Sämtliche Elemente sind in den Subtypen enthalten und der Supertyp besitzt selber keine Elemente. | | **Disjunk**t  Einzelne Subtypen haben keine gemeinsamen Elemente. | | | Arten |  | | Aggregation | Um Zusammenhänge als "Teil-Ganzes"-Beziehungen darzustellen. Strukturiertes Zusammenfügen von Klassen zu einer komplexen Klasse.  Computergenerierter Alternativtext: has  mitarbeiter  has  is part o  freiberufler | | Komposition (starke Aggregation) | Wenn sich bei der "Teil-Ganzes"-Beziehung um eine Existenzaussage handelt. Teile können ohne das Ganze nicht existieren. Finger können ohne Hand nicht existieren.  Computergenerierter Alternativtext: 」 35u4  pueu | |

# Relationenmodell / ERD (Entity Relationship Diagram)

Das Relationsmodell packt alle Entitätstypen und Beziehungstypen in flache Tabellen. Diese Tabellen werden Relationen genannt. Eine relationale Datenbank ist eine nach dem Relationsmodell aufgebaute Ansammlung regelkonform verknüpfter Tabellen (=> immer über Primärschlüssel Fremdschlüssel-Paar).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Relationen (Tabellen) | Computergenerierter Alternativtext: tb1 b  v name  o r t: j ahr  1 998  Aarau 1998  v adr  mtertechnlk I Sauerländet• AG Laurenzenvorstaclt 89 Postfach  3 — 9-4143825 Glatz  Ll Sauerlander AG Lavu enzenvorstadt 89 Postf•acli  Glatz   |  |  | | --- | --- | | Tupel | Vollständiger Datensatz in einer Zeile (sind nie sortiert) => Entität | | Attribut | Attribute haben in der Tabelle eine eindeutige Benennung (Spaltenname) und ein Entitätstyp (Tabelle) hat mindestens zwei Attribute (Spalten) => PK und eine andere Info  Beziehen ihre Werte aus einer Domäne und sind atomar (ein Wort pro Feld). | | Attributslisten | C:\339B4F45\FFD82C26-6719-437B-83FF-1359CE788529-Dateien\image018.png | | Feld | Besitzt nur einen Attributwert oder ist leer (Null-Wert/Marke) | | Domäne | Wertbereich/Gültigkeitsbereich (sämtliche zugelassen Werte einer Tabelle)   |  |  | | --- | --- | | Primärschlüssel und normale Attribute | Statischer Wertebereich | | Fremdschlüssel | Dynamischer Wertebereich (jede Änderung bei der Referenztabelle führt zur Veränderung des Wertebereichs) | | | Projektion | nur gewisse Attribute und Ihre Werte | | Selektion | nur gewisse Tupel und Ihre Werte | | Breite (Degree) | Waagrechte Ausdehnung einer Relation (Tabelle) => Anzahl Attribute | | Höhe (Card) | Senkrechte Ausdehnung einer Relation (Tabelle) => Anzahl Tupel | | natürlich | In der realen Welt vorhanden | | künstlich | In der realen Welt nicht vorhanden | | atomar | Nicht zusammengesetzt, einfach | | zusammengesetzt | Mehrere Attributswerte zusammengesetzt | | einwertig | Nur 1 Speicher möglich | | mehrwertig | Datensatz ist nicht atomar. Mehrspeicher möglich | |
| Tupelintegrität | Da ein Tupel in einer Tabelle einzigartig vorkommen sollte, wird es mit einer Tupelintegrität identifiziert (auch Primärschlüsselintegrität genannt). Dies ermöglicht der Primärschlüssel (Identifkationsschlüssel), welches immer ein Attribut oder eine geordnete Menge mehrerer Attribute ist. Zur Unterscheidung sollte man den Primärschlüssel unterstreichen. Er ist eindeutig, beinhaltet keine Mehrfachnennung (SQL: UNIQUE) und sind nicht leer (SQL: NOT NULL). Kennzeichnung mit präfix "pk\_" oder "id\_" verwenden. |
| Referenzielle Integrität | Es dürfen nur Werte aus der Fremdschlüsseltabelle (Herkunftstabelle) vorkommen d.h. am Bezugsort vorhanden sein und besitzen die gleiche Domäne wie beim Bezugsort. Die Domäne kann sich dynamisch verändern, weil jeder neuer Wert aus der Herkunftstabelle den Wertbereich ändert. Dürfen ausserdem mehrfach vorkommen und werden umkreist.    1-seitige Tabelle = primärschlüsselseitige Tabelle  n-seitige Tabelle = fremdschlüsselseitige Tabelle    Bei einer n:1-Verknüpfung kommt der Fremdschlüssel auf der n-Seite. |
| Domänenintegrität | Überprüfung ob es in die definierte Domäne passt (z.B. nur Zahlen als PLZ erlaubt). |
| Beziehungstypen | Wenn ein Datensatz redundant ist, dann wird die Tabelle aufgebrochen (dekomponiert) und in zwei Tabellen mit einer Beziehung (Relation) gebrochen.   |  |  | | --- | --- | | m:m | Computergenerierter Alternativtext: Assoziative Entitätstyp: autorenscha  autor (id, nachname, vorname)  buch (isbn titel)  -auto fi isbn | | mc:mc | Computergenerierter Alternativtext: Assoziativer Entitâtstyp: verl lieg_besitz  d fi verl  fi 11e  verlag (id verlag verlagsort)  liegeschaft (Lligg)  lag | | mc:m | Computergenerierter Alternativtext: Assoziativer Entitäts  auto_vertr_pos d fi auto  d fi vertr  autor (jd_a.u.gu, nachname, vorname)  vertrag (id vertrag titel)  nsatz) | | 1:1 | Computergenerierter Alternativtext: Variante 1:  autor id auto fi web ame, ...)  webauuftritt (id  Variante 2:  autor (id autor, name  webauftritt id we fi_autor rl, ...)  Diese beiden Varianten ergeben einen Sinn dann, wenn beide Tabellen z.B. aus historischen, verwaltungs- oder  systemtechnischen Gründen getrennt bestehen  Variante 3  gehet t  autor (id autor nam, url,  Variante 3 wird genommen, wenn zwischen dem Autor und seinem Webauftritt schon eine 1:1 Beziehung be-  steht und keine anderen Sachzwänge vorliegen. | | 1:c | Computergenerierter Alternativtext: Variante 1  buch (isbn  Cip id ci  Variante 2  buch isbn  ci fi isbn  titel preis,  isbn, ip_titel, „  titel, preis, „ )  ip_titel)  cip_tit | | m:c | Computergenerierter Alternativtext: Assoziative Entitâtstyp:  (i fi la  verl  âche, tor) | | Rekursiv | Man kann einen Entitätstyp mit sich selber in eine Beziehung setzen.  Computergenerierter Alternativtext: besteht aus  ist Bestandteil von  produkt | |
| Tabellentypen (Entitätstypen) | |  |  |  | | --- | --- | --- | | Attributiv | tkey\_ | Eine Art Lookup-Tabellen (z.B. für Währungen), um eine Tabelle zu attributieren. Hilft auch Fehler zu vermeiden, weil die Inhalte von der attributiven Tabelle vordefiniert sind. Besitzen selber nie Fremdschlüssel und somit nie auf der n-Seite.  => Attributlieferanten | | Fundamental | tbl\_ | Objekte mit denen gearbeitet werden (Buch, Kunde usw.) | | Assoziativ | tass\_ | Zwischentabellen zur Auflösung von Mehrfachbeziehungen (meist künstlich) | |
| Übersicht | |  |  |  | | --- | --- | --- | | **ERM** | **Relationenmodell** | **Alternativ** | | Entität | Tupel | Zeile, Datensatz, Record | | Entitätstyp/-menge | Relation | Tabelle | | Beziehung | Beziehung |  | | Beziehungstyp/-menge | Beziehungstyp |  | | Kardinalität | Kardinalität | Anzahl (wenig sinnvoll) | | Attribut | Eigenschaft | Spalte | |

# Rezept

1. **Sicht bestimmen**
2. **Entitäten bestimmen**

Analysiere das reale System (oder den beschriebenen Fall oder die vorliegende flache Tabelle

usw.) und isoliere dessen Objekte als Entitäten. Fasse gedanklich gleichartige Entitäten zusammen und abstrahiere sie zu Entitätstypen.

=> Entwirf jeden Entitätstyp je als ein Rechteck und benenne dieses.

1. **Beziehungen definieren**

Analysiere die Beziehungen zwischen Entitäten. Fasse gedanklich gleichartige Beziehungen zusammen und abstrahiere sie zu Beziehungstypen zwischen Entitätstypen.

=> Entwirf jeden Beziehungstyp je als eine Linie. Benenne die beiden Assoziationstypen des Beziehungstyps in beide Leserichtungen korrekt.

1. **Häufigkeiten der Beziehung bestimmen (Kardinalität)**

Analysiere die Kardinalitäten an beiden "Enden" des Beziehungstyps.

=> Entwirf beidseitig je die minimale und maximale Kardinalität mit der "Krähenfüsschenmethode".

1. **Attribute definieren**

Ermittle die Eigenschaften der Entitäten. Fasse gedanklich gleichartige Eigenschaften zu Attributen zusammen.

=> Entwirf die Attributsliste aller Entitätstypen. Markiere für jeden Entitätstyp den Primärschlüssel durch Unterstreichung und die Fremdschlüssel durch Einkreisen.

# Normalisierung

Gewährleisten die langfristige Redundanzfreiheit von Datenbanken. Bei strikter Einhaltung der Schritte zur relationalen Datenbank, dienen die Normalformen nur zur Überprüfung. Alle Tabellen müssen in der 3. NF sein.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| * 1. NF | Eine Tabelle ist in erster Normalform, falls die Wertebereiche der Attribute **atomar** sind. Stadt und Postleitzahl nicht in einer Entität abspeichern, sondern trennen!   |  |  | | --- | --- | | Vorher | Computergenerierter Alternativtext: willi  ttern knattern  Buch hat mehrere Typen. | | Normalisiert | buch(ISBN, autor, titel, typ1, typ2)  => Falls es noch mehr Typen gibt, würde es nicht mehr funktionieren! | |
| * 1. NF | Wenn jedes Nichtschlüsselattribut vom zugehörigen Primärschlüssel voll funktional abhängig (wenn jeder Wert aus einer Domäne A genau einen Wert aus einer Domäne B, dann ist B funktional abhängig von A) ist.   |  |  | | --- | --- | | Vorher | Computergenerierter Alternativtext: Kcaccl  1234567890123  Nattern knatte  Aus den Primärschlüsseln BstNr können wir nicht auf die Anzahl kommen. | | Normalisiert | Eine assoziative Zwischentabelle (tass) erstellen.  tass\_buch\_bestellung(BstNr, ISBN, anzahl)  buch(ISBN, titel)  bestellung(bst\_id, kunde, ..) | |
| * 1. NF | Wenn kein Nichtschlüsselattribut von irgendeinem Schlüssel transitiv abhängig (wenn ein Wert eines Nichtschlüsselattribut einen anderen Attributswert bestimmt) ist. Die transitiv abhängigen Datenfelder werden in weitere Tabellen ausgelagert, da sie nicht direkt vom Schlüsselkandidaten abhängen, sondern nur indirekt.   |  |  | | --- | --- | | Vorher | Computergenerierter Alternativtext: e n Nattern  Vern d  VerlId schliesst auf Verlag und wäre somit nicht nötig | | Normalisiert | buch(isbn, verlid, titel)  verlag(verlid, verlag) | |
| Beispiel | Computergenerierter Alternativtext: 03.01.2005  Os. LO. 2005  L s .39.2005  425  LI. LI. 2005  458  12.12.2005  303  10.02.2005  SOS Muller  429 Me er  425 Me er  Schulze  Schmid  SchrnLd  kur s #  kurs  Verkauf  2CÆ5  72•OCO L91  72 • 000 L98  61  L31  Verkauf  Rhetor f k  E uh rung  Ehe rar i k  Verkauf  Oracle   |  |  | | --- | --- | | * 1. NF | Erfüllt | | * 1. NF | kursverwaltung(**id\_pers**, name, abt, abt\_budget, kurs, kursname, datum)  Alle Nichtschlüsselattribute müssen auf Abhängigkeit mit Schlüsselattributen überprüft werden.  Auflösen in 3 Tabellen:  pers(**id\_pers**, name, abt, abt\_budget)  kurs(**id\_kurs**, kurs\_name)  kursverwaltung(**fk\_pers**, **fk\_kurs**, **datum**) | | * 1. NF | Jetzt werden die einzelnen Tabellen auf Abhängigkeit zum Primärschlüssel geprüft.  pers(**id\_pers**, name, abt, abt\_budget)  id\_pers kann nicht über abt\_budget aussagen und muss somit ausgelagert werden.  abt(**id\_abt, abt\_budget**)  pers(**id\_pers**, name, fk\_abt) | |

weiteres Beispiel: <https://www.hdm-stuttgart.de/~riekert/lehre/db-kelz/chap4.htm>

# Joins (Pyramiden-System)

|  |  |
| --- | --- |
| Cross-Join | Kartesisches Produkt  Alle Kombinationen der ausgewählten Spalten (jeder mit jedem). Jeden Datensatz aus der einen Tabelle mit der anderen Tabelle verbinden, wobei es keine Beziehungen bzw. Verknüpfung zwischen den Tabellen gibt. Wird sehr selten gebraucht, weil alle Datensätze angezeigt werden.  Bsp. Jeder Kunde verfügt über jede Zahlungsmoral  Computergenerierter Alternativtext: über jede Zahlungsmoral.  Zürich  gut  gut |
| Inner-Join (inkl. Cross) | Eine ganz normale Beziehung zwischen zwei verschiedenen Tabellen, wo PK=FK übereinstimmen.  Computergenerierter Alternativtext: |
| Outer-Join (inkl. Cross + Inner) | Es werden alle Tupel von einer Tabelle angezeigt und die entsprechenden Tupel der anderen Tabelle, welche die Bedingungen erfüllen. Man unterscheidet zwischen Left-Outer Join und Right-Outer Join:   |  |  | | --- | --- | | Left | Computergenerierter Alternativtext: 'bl kunde  tass potice  '"_—s_at  _nfe  Alle aus der Tabelle „Kunden" sowie die, welche die  Bedingung erfüllen aus der Tabelle „Police" | | Right | Computergenerierter Alternativtext: tass_police  'bl versicherung  Alle aus der Tabelle „Versicherung" sowie die, welche  die Bedingung erfüllen aus der Tabelle „Policen". | |

# Access

|  |  |
| --- | --- |
| RDBMS | **Relationales Datenbanksystem**   * packt alle **Entitätstypen** (eines relationalen und normalisierten ERD) in **Tabellen** * sichert die **Tupelintegrität** in jeder Tabelle durch die Definition eines **Primärschlüssels** ab * drückt alle **Beziehungen** durch die Definition eines **Primär-/Fremdschlüsselpaars** aus * sichert **referenzielle Integrität** durch die Definition der **Lösch- bzw. Mutationsaktionen** **ab** * legt über alle **Definitionen** automatisch eine eigene relationale Datenbank ab: die Metadaten der Systemtabellen bzw. des **Systemkatalogs**.     Die benutzerdefinierten Tabellen samt ihren Inhalten bilden die **Datenbasis**, der Systemkatalog enthält die Metadaten. Die "grossen", kommerziellen RDBMS sind mehrbenutzerfähig, das heisst, sie   * regeln die **Authentisierung** und **Autorisierung** von Benutzenden * sichern **Anomalien** durch **gleichzeitige Zugriffe** ab * verhindern so **Phantomeffekte**.     Sie sind ferner in der Lage, Befehlseinheiten nach dem "Alles-oder-Nichts" Prinzip atomar abzuwickeln und gegebenenfalls zu stornieren - wir sprechen in diesem Zusammenhang von **Transaktionen** und **Rollback**. |
| Datenbank | * + trägt einen **systemweit eindeutigen Namen**   + besteht aus **zwei oder mehreren Tabellen** und in der Regel weiteren Objekten   + besteht aus den (automatisch geführten) **Metadaten**. |
| Tabelle | * + trägt einen **datenbankweit eindeutigen Namen**   + hat **zwei oder mehr Attribute** (Spalten, Felder)   + hat einen **Primärschlüssel** aus **einem oder mehreren Attributen**   + hat eine (relativ statische) **Breite**, bestimmt durch die Anzahl Attribute   + hat eine (relativ dynamische) **Höhe**, bestimmt durch die Anzahl Tupel (Datensätze).   + Ist das **wichtigste Objekt** einer Datenbank |
| Attribut | * + trägt einen **tabellenweit eindeutigen Namen**   + entnimmt seine **Werte** einer bestimmten **Domäne**, die wir hier Datentyp nennen   + hat eine **bestimmte Feldgrösse** die u.U. durch den **Datentyp** bestimmt wird |
| Datentypen | Computergenerierter Alternativtext: Alphanumerische Datentypen  Zeichentypen (Werte mit Buchstaben, Ziffern, Sonderzeichen)Das System kann mit alphanumerischen Typen  nicht rechnen! Sobald ein Buchstabe oder Sonderzeichen enthalten sind, muss die Domäne in einen Alpha-  numerischen Datentyp verwandelt werden. (1 Byte pro Zeichen)  Text  Memo  Länge der Zeichen kann definiert werden.  Für Bemerkungen usw., mit sehr variabler Länge (belastet das System erheblich, weil sie  separat abgelegt werden.  Numerische Typen  Mit Ihnen ist das Rechnen möglich.  Byte  Integers  Long Integer  Single  Double  Autowerte  Datum/Uhrzeit  Währung  Ja/Nein  O Ganzzahl 255 (1 Byte pro Zahl)  -32'768 Ganzzahl 32'767 (16 Bits 2 Bytes pro zahl)  -2'147'483'648 Ganzzahl 2'147'483'647 (32 Bits 4 Bytes pro zahl)  pos./neg. Fliesskommazahl mit einfacher Genauigkeit (4 Bytes pro Zahl)  pos. /neg. Fliesskommazahl mit doppelter Genauigkeit (8 Bytes pro Zahl).  ist ein Long Integer, der bei O beginnt. Mit jedem neuen Tupel vergibt das System eine um  1 grössere natürliche Zahl. Einmal vergebene Zahlen kommen nicht wieder, auch wenn das  mit ihm identifizierte Tupel gelöscht wurde. Autowerte werden oft für Primärschlüssel-  attribute verwendet.  akzeptiert nur Werte, die in die Domäne passen und formatiert sie gemäss den Landesein-  stellungen von Windows.  Währung akzeptiert nur Ganz- oder Fliesskommazahlen und formatiert sie gemäss den  Landeseinstellungen von Windows.  ist ein so genannter Wahrheitswert, der nur die Eintragung "ja" und "nein" erlaubt. Bei-  spiel: Attribut "Ehrenmitglied" in einer Vereinsverwaltung.  Wichtig:  Ein Fremdschlüsselattribut muss den gleichen Datentyp haben wie das zugehörige Primärschlüsselattribut. |
| Indizes | Computergenerierter Alternativtext: Die Datenbank sortiert die Tupel nicht. Wenn man sortierte Tupel haben möchte, muss man einen Index auf  das gewünschte Attribut erstellen. Damit erstellt das DBMS vom gesamten Attribut eine Kopie, die bei jeder  Erfassung, Mutation und Löschung sortiert wird.  Beispiel für einen Ablauf einer Indexsuche ist auf Seite 71 in unserem Skript.  Vorteile:  - Suche kann dramatisch beschleunigt werden  - System wird leicht belastet  Nachteile:  - Zusatzaufwand bei Tupelmutationen aller Art  - Mehr Speicher  Es sollten wenige Indizes vergeben werden. Man sollte Indizes an Attribute vergeben, welche oft als Such-  oder Sortierkriterium dienen oder eine hohe Selektivität aufweisen.  Wichtig:  Die Definition eines Primärschlüssels legt automatisch einen Index ohne Duplikate an!  Unter der Entwurfsansicht beim jeweiligen Attribut kann ein  Index erstellt werden:    Bsp. Suche nach Tuor:   1. Index auslesen   Computergenerierter Alternativtext: Bieri  Eigensatz  Grünin r  Keller  Luth- er  Ritschard  Rbbsli  Schibli  Schick  Schmid-ToniOlO  Schmucki  Senn  Stalder  Tuor  Wuillemin  3333  5107  2453  4716  3456  4721  4718  2467  471 g  1367  472E  4715   1. Sprung zum Tupel mit dem entsprechenden Index   Computergenerierter Alternativtext: |
| Beziehungen | Computergenerierter Alternativtext: Access kennt nur zwei Typen von Beziehungen:  (1 : O oder mehreren) Wenn Fremdschlüsselattribut kein Primärschlüssel ist (Normalfall), macht  1 : mc  Access eine solche Beziehung.  (1 : O oder 1) Ist Fremdschlüssel auch alleiniger Primärschlüssel, macht Access daraus eine solche  Beziehung.  Beziehung erstellen: Attribut mit Primärschlüssel auf das Fremdschlüsselattribut einer fremden Tabelle fal-  lenlassen.  Mit referentieller Integrität: Wird sicher gestellt, dass die Fremdschlüssel nur Werte annehmen können, die in  der referenzierten „Tabellel" bereits als Attributwerte des  Primärschlüssels vorhanden sind  Mit aktivierter Aktualisierungsweitergabe sichert das DBMS  die referenzielle Integrität durch kaskadiertes Mutieren ab:  Jede Änderung eines Primärschlüsselwerts in der Elterntabelle  mutiert die zugehörigen Fremdschlüsselwerte in der/den Kind-  tabelle/n. Dies kann seinerseits wieder Kaskadierungen auslö-  sen. Eine Mutation kann unumkehrbar eine Grosszahl weiterer  Mutationen auslösen!  d_rntg  r— .n verwarae  Veinopfwostyp... I  Nicht aktivierte Aktualisierungsweitergabe (auch genannt: bedingtes Mutieren) verhindert Mutationen an  Primärschlüsselwerten in der Elterntabelle, solange entsprechende Fremdschlüsselwerte existieren.  Mit aktivierter Löschweitergabe sichert das DBMS die referenzielle Integrität durch kaskadiertes Löschen ab:  Jede Löschung eines Primärschlüsselwerts in der Elterntabelle löscht die Tupel mit den zugehörigen Fremd-  schlüsselwerten in der/den Kindtabelle/n. Dies kann seinerseits wieder Kaskadierungen auslösen. Eine Lö-  schung kann also (unumkehrbar) eine Grosszahl weiterer Löschungen auslösen!  Nicht aktivierte Löschweitergabe (auch genannt: bedingtes Löschen) verhindert Löschungen von Primär-  schlüsselwerten in der Elterntabelle, solange entsprechende Fremdschlüsselwerte existieren. |
| Abfragen (Queries) | Computergenerierter Alternativtext: Abfragen Sind unverzichtbare Datenbankobjekte_ Mit Abfr4en erhält man übersichtliche Einblicke in die Da-  tenbasis_ Mit ihnen können auch Berechnungen gemacht werden. Die Reihenfolge der Tupa kann variieren,  jedoch aber die Anzahl der Ergebnistupel nicht.  Ta einer  Sind Datenbankobjekte, die Wir definieren und somit anlegen müssen  selektieren Daten aus einer Oder mehreren Tabellen (genannt: Verbünde Oder Joins)  liefern als Resultat immer Tabellen  werden ad hoc definiert oder als Definition gespeichert (dann eigentlich Sichten oder  Views genannt)  können auf Tabellen und/Oder ihrerseits auf Skhten basieren  können Wiederum Abfragen enthalten (genannt: Unterabfragen Oder Subqueries).  ist eine Menge (Set) mit O, I Oder n (Resultat-l und den verlangten Attributen  ist ohne unsere Veranlassung nie sortiert  Wird bei jedem Aktivieren der Abfrage neu berechnet  ist also nicht gespeichert, Wir sprechen deshalb Von einer temporåren Virtuellen  Tabelle  kennt keine Primårschlüssel  kan n Redundanzen enthalten.  7. I Abfragen aus einer Tabelle Oder Mehreren Tabellen  Verbund definieren — welcher Tabelle brauche ich welche Daten?  Projektion definieren — Welche Attribute brauche ich?  Selektion definieren — Wekhe Tupel Will ich sehen?  Tabellenüberschriftname: Kann definiert werden. ES können auch Berechnungen Vorgenommen  werden mit den normalen Operatoren.  Beispiel:  Moral und Versicherungsgebiet: lid_moral " &  Zu welcher Tabelle gehört das obere Attribut? Besonders aufpassen bei rekursiven Beziehungs-  typen.  gei Betätigung wird das Attribut bei der Abfrage angezeigt oder nicht-  Aufsteigend: V on A nach Z  Absteigend: Von Z nach A  Wichtig!  Numerische Werte werden als Zahl geschrieben: 3.45  Werte werden in Anführungs- und Schlusszeichen geschrieben:  Tabelle  Anzeigen  Sortierung   |  |  | | --- | --- | | Kriterien und Operatoren | Computergenerierter Alternativtext: Operator  Beschreibung  —"Zür.ch"  Kann meistens weggelassen werden, da Access  Wie „Zürich"  das Kriterium eh schon als Gleichung erkennt.  like „Zürich"  Alle Werte grösser-gleich 2.  Alle Werte grösser als 2  Alle Werte kleiner-gleich 2  Alle Werte kleiner als 2 | | Funktionen | Computergenerierter Alternativtext: Funktion  Gruppierung  Mittelwert  Anzahl  Beschreibung  Fasst a le gleichen Attributwerten Zusammen  Rechnet die Summe aller Attributswerten  Rechnet den Durchschnitt aller Attributswerten  Findet das Minimum aller Attributswerte  Findet das Maximum aller Attributswerte  Zählt Wieviel vorhanden Sind. Meist in VerbW1dungvon der Funktion „Gruppierung" | | Aktions-abfragen | Computergenerierter Alternativtext: Mit Abfragen kann man nicht nur Sachen abfragen, sondern auch Veränderungen an der Datenbank vorneh-  men. Veränderungen können nicht Rückgängig gemacht werden und werden in der Datenbank gespeichert'  lun gsabfrage  Aktualisierungs-  A nfügeabfra ge  Löschabfrage  Erstellt aus dem Ergebnis einer Abfrage eine neue  Tabelle  Macht bei allen selektionierten Tupa ein Update  auf einen neuen Wert.  Fügt die selektbnierten Tupel an eine andere Tabel-  Löscht alle selektionierten Tupel aus der Datenbank  tEL kund. SET  1 :ggo  (12212, "  11 ggc:  D E LEE | |
| Formulare und Berichte | Formulare operieren auf –Tabellen oder [bearbeitbaren] Abfragen ([Updatable]Views).   |  |  | | --- | --- | | Hauptformular | Da das Skript nur auf die Arbeit in Access und die Vorgehensweise eingeht, verzichte ich dieses Kapitel mit Copy & Paste hinein zu kopieren. Es startet ab Seite 96. Die Namensübernahme (Kombinationsfeld) vom Fremd- und Primärschlüssel startet ab Seite 97 in der unteren Hälfte. | | Haupt- und Unterformular | das **Hauptformular** ist die **1-Seite**: Ein Kunde ...  das **Unterformular** ist die **m-Seite**: ... hat mehrere Policen.  Computergenerierter Alternativtext: Hauptformtllar I Kunde  Unterfolnilllar policen  Hanptfornilllar: I Versichenmgsgebiet  Unterfollnular ni Verncluenlngen | | Berichte | Berichte sind selektive und statische Datenbankauszüge für die Präsentation und Dokumentation. Sie basieren in der Regel auf Abfragen. Es können aber auch Tabellen als Basis genommen werden. Sie sind meist mit bestimmten Kriterien sortiert oder aggregiert, also gruppiert ausgewertet: Summen, Maxima, Minima; Anzahl  usw.  Da das Skript nur auf die Arbeit in Access und die Vorgehensweise eingeht, verzichte ich dieses Kapitel mit Copy & Paste hinein zu kopieren. Es startet auf Seite 107. | |