Änderungsanomalie – siehe **Inkonsistenz\Inkonsistenz I**

Attribut / Merkmal – Spalten einer Tabelle

Attribute – Spalten einer Tabelle, weisen Namen und Wertebereich auf. Die einzelnen Daten heissen Attributwerte.

Attributwerte – Werte eines Attributes. (Werte in einer Spalte)

Beziehung – Verbindung von zwei Datenbanken, benützt Primärschlüssel und Fremdschlüssel.

Beziehung – verschiedene Arten, wie sich Daten aufeinander beziehen können. **Einfache Beziehung (1: /** ⟊ **)** – Jede Entität b-zieht sich auf genau **eine** andere Entität einer zweiten Entitätsmenge.

Konditionelle Beziehung (c: / ⟜ **)** – Jede Entität bezieht sich auf **maximal eine** andere Entität einer zweiten Entitätsmenge. Findet sich nicht in einer relationalen Datenbank. **Mehrfache Beziehung (m: /** ⟊⋔ **)** – Jede Entität bezieht sich auf **mindestens eine** andere Entität einer zweiten Entitätsmenge.

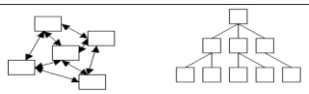
Mehrfachkonditionelle Beziehung (mc: / ⟜⋔ **)** – Jede Entität bezieht sich auf beliebig viele andere Entitäten einer zweiten Entitätsmenge. Findet sich nicht in einer relationalen Datenbank.

Datenmanagement – Der Gebrauch von Daten, weiter in Kategorien unterteilbar.

	Ziele	Instrument
Datenarchitektur	Pflegen und erweitern des Datenmodells. Unterstützung der auf Datenmodellierung basierenden Softwareentwicklung.	Instrumente der rechengestützten Datenmodellierung.
Datenadministration	Verwalten von Daten und Funktionen anhand von Standardisierungsrichtlinien und internationalen Normen. Beraten von Endbenutzern.	Data-Dictonarysysteme, Werkzeuge für den Verwendungsnachweis.
Datentechnik	Installieren, Reorganisieren von Datenbanken, Durchführen von Datenbankrestaurierungen nach einem Störfall.	Datenbankverwaltungssysteme, Hilfsmittel für die Wiederherstellung von Datenbanken und zur Leistungsoptimierung.
Datennutzung	Bereitstellung von Auswertungs- und Reportfunktionen unter Berücksichtigung des Datenschutzes resp. der Dateneignerschaft.	Sprache für die Datenbankabfragen und –manipulation, Reportingtools.

Datenmodelle – Daten können in verschiedenen Weisen dargestellt werden.

Netzwerkmodell und hierarchisches Modell
Sie sind Vorgänger des relationalen Modells. Sie bauen auf individuellen Datensätzen auf und können hierarchische Beziehungen oder auch allgemeinere netzartige Strukturen der Realwelt ausdrücken.



Netzwerk und hierarchisches Datenmodell

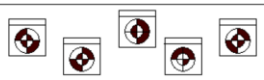
Relationales Modell
Es ist das bekannteste und in heutigen DBMS am weitesten verbreitete Datenbankmodell. Es stellt die Datenbank als eine Sammlung von Tabellen (Relationen) dar, in denen alle Daten angeordnet werden.

P-ID	Name	Vorname	Ort
1	Müller	Hans	Oberdorf
2	Meier	Jakob	Hinterwil
3	Keiser	Josef	Unterdorf
...	...	...	...

Ort-ID	Fläche	Art	...
5689	45865	Dorf	...
4758	23864	Weiler	...
2145	86541	Stadtteil	...
...	...	...	...

Relationales Datenbankmodell

Objektorientiertes Modell
Objektorientierte Modelle definieren eine Datenbank als Sammlung von Objekten mit Eigenschaften und Methoden.



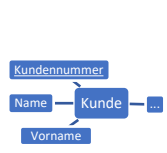
Schematische Darstellung eines objektorientierten Datenbankmodells

Objektrelationales Modell
Objektorientierte Modelle sind zwar sehr mächtig, aber auch recht komplex. Mit dem objektrelationalen Datenbankmodell wurde das einfache und weit verbreitete relationale Datenbankmodell um einige grundlegende objektorientierte Konzepte erweitert.

ID	XY	DF	ER
56		XXX	
45		YYY	
...	...	...	...

Schematische Darstellung des objektrelationalen Datenbankmodells

Datenobjekt – Gruppierung von zusammenhängenden oder passenden Informationen, eg. Kontakte auf dem Handy. Kann auf mehrere Arten dargestellt werden:



Stern;

Attributliste

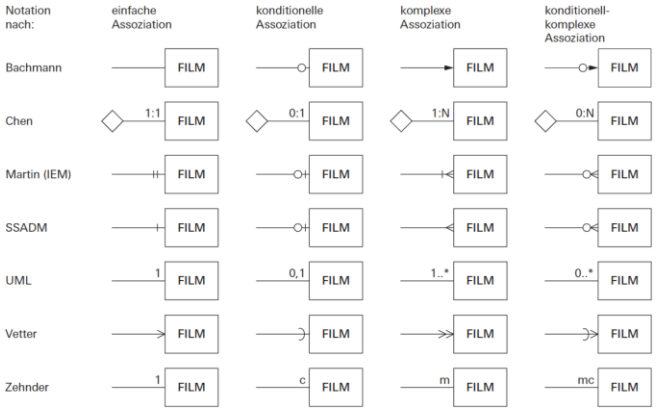
Box

Einfügeanomalie – siehe **Inkonsistenz\Inkonsistenz III**

Entitätsmenge – Menge aller Datenobjekte, welche zusammengehören und in einer Tabelle darstellbar sind.

Entitätstyp – Spezifikation eines Objektes

Entity-Relationship-Diagramm – **ERD**, Darstellungsweise von Beziehungen von Daten. Dabei verbreitete Notationen. Weitere Informationen unter **Beziehungen**.



FETCH – **FETCH** erlaubt eine zeilenweise Abfrage von Werten einer Tabelle.
CURSOR – Laufvariable:

```
DECLARE cursor CURSOR FOR SELECT Attr1 FROM TABLE;
OPEN cursor;
FETCH NEXT FROM cursor;
WHILE @@FETCH_STATUS = 0;
BEGIN
        FETCH NEXT FROM cursor;
END
CLOSE cursor;
DEALLOCATE cursor;
GO
```

Falls mit **SELECT** mehrere Werte ausgewählt, zuweisung zu Variablen mit: **FETCH NEXT FROM *cursor* INTO *@var1, @var2;***

PRINT – Ausgabe von Variabel-Werten.

Inkonsistenz – Verschiedene Arten von Inkonsistenzen
Inkonsistenz I: Wird bei einem Datensatz ein attributwert verändert, kann es sein das bei zwei verschiedenen Datensätzen zwei verschiedene Werte auf die selbe Eigenschaft verweisen. z.B.: Land: CH und Land: Schweiz
Inkonsistenz II: Z.B. Falls eine Personaltabelle zwei Personen denselben Namen besitzen und nun eine Person dieses Namens aus der Firma austritt. Entweder ist diese Person 2mal eingetragen, so werden beide Einträge gelöscht, oder es gibt 2 Personen mit demselben Namen, so ist unklar welchen Datensatz zu löschen ist.
Inkonsistenz III: Ähnlich zu **Inkonsistenz I**, falls

bei Eintrag ein Fehler unterläuft und nun wieder bei zwei Datensätzen zwei Werte auf dasselbe verweisen.

Löschanomalie – siehe **Inkonsistenz\Inkonsistenz II**

Normalform – Stufen von Normalisierungen
Unnormalisiert: Es gibt Zellen mit nicht atomaren Werten.
1. Normalform (1NF): Alle Attributwerte sind atomar.
2. Normalform (2NF): Die Attributwerte erfüllen die erste Normalform und alle Nicht-Schlüssel-Attribute sind voll funktionsfähig vom Primärschlüssel ableitbar. (Eindeutigkeit)
3. Normalform (3NF): Die Attributwerte erfüllen die zweite Normalform und keine der Nicht-Schlüssel-Attribute sind transitiv, z.B. keines der Attribute lässt sich durch ein anderes induzieren.

Redundante Daten - Information, welche doppelt in Datenbank vorliegt. Kann mithilfe von Auslagerung einiger Attribute verhindert werden (Wohnorte/PLZ).

SQL – Begriffsreferenz für SQL
Formale Tabellenbeschreibung:
TABELLE(#ID, Attr1, Attr2, #Attr3, #FID);
CREATE TABLE:
CREATE TABLE Tabelle(#ID, ...);
INSERT:
INSERT INTO TABELLE(Attr1, Attr3) VALUES (‘*txt*’, *3*);
UPDATE:
UPDATE Tabelle SET Attr1 = ‘bsp’, Attr3 = 5 WHERE ID = 1;
DELETE:
DELETE FROM Tabelle WHERE ID = 2;
SELECT:
SELECT */[ATTR] (Alle/Ausgewählte Attribute durch , getrennt)
SELECT DISTINCT ([ATTR]) (Gibt mehrfach vorkommende Datenwerte nur einmal aus)
JOIN:
INNER JOIN: SELECT * FROM Tabelle AS t JOIN Liste AS l ON t.ID = l.NR; $M \cap N$
LEFT JOIN (Analog RIGHT JOIN): SELECT * FROM Tabelle AS t LEFT JOIN Liste AS l ON t.ID = l.NR; $(M \cap N) \cup (M \setminus N)$
WHERE:
SELECT * FROM Tabelle WHERE (Attr3 = 3);
(Kombination durch **AND, OR** (AND>OR))
GROUP BY:
SELECT * FROM Tabelle GROUP BY Attr1;
ORDER BY:
SELECT * FROM Tabelle ORDER BY Attr1 ASC;
ASC Aufsteigend; **DESC** Absteigend
LIMIT / OFFSET: Ausgabe begrenzen und den Ausschnitt lokalisieren:
SELECT * FROM Tabelle LIMIT **10** OFFSET **3;**
ALIAS:
SELECT Attr1 AS “Attribut 1” FROM Tabelle;
IMPORT: (Importieren von Daten aus per Konsole)
.separator ”;“
.import **FILE.csv TABELLE.NAME**
Operatoren:
$= \widehat{=} =; < \widehat{=} <; <= \widehat{=} \leq; >= \widehat{=} \geq; <> \widehat{=} \neq; NOT \widehat{=} \neg$
BETWEEN ... AND ... (Zwischen 2 Werten)
(NOT) LIKE (Stringvergl.) **WILDCARDS: % = *; _ = ?**
IS (NOT) NULL (Prüft ob Wert NULL)
MAX / MIN:
SELECT MAX(Attr1) FROM Tabelle (Ausgabe Min/Max des Attributes)
SUM / COUNT:
SELECT SUM(Attr3) FROM Tabelle (Summe der Attribute)
SELECT COUNT(Attr1) FROM Tabelle (Anzahl Einträge)
ROUND / AVG:
SELECT ROUND(Attr3, N) FROM Tabelle (Runden auf N Stellen)
SELECT AVG(Attr3) FROM Tabelle (Durchschnitt)
CONSTRAINT:
CONSTRAINT beschränkt die möglichen Datenwerte eines Attributes: **NOT NULL** = Kann nicht NULL sein
UNIQUE = Muss eindeutig sein
PRIMARY KEY = Stellt einen Primärschlüssel dar
FOREIGN KEY = Stellt einen Fremdschlüssel dar
CHECK = Muss einen bool. TRUE ausgeben bei Conditional
FETCH: Siehe **FETCH**
VARIABEL: Siehe **Variabel**
PRINT: Siehe **FETCH\PRINT**

Variabel – Eine Variabel in SQL beginnt jeweils mit einem @: **@VAR1**

Wertebereich / Domäne – Zulässige Werte eines Attributes