

LANDESBERUFSSCHULE 4 SALZBURG

Informatik

Entity-Relationship-Modell

LBS 4

Inhalt

Einleitung	4
Daten-Informationen-Wissen.....	4
Datenmodelle	4
Elemente des ERM.....	5
Entitätstypen (Entity-Typen)	5
Beziehungstypen (Relationship)	5
Attribute	5
Schlüsselattribut	6
Kardinalität	6
1:1 Beziehung	6
1:N Beziehung.....	7
M:N Beziehung.....	7
Übung:.....	7
Kardinalität	8
min-max Notation	8
Grundbegriffe relationales Datenbankmodell.....	9
Relation, Tupel, Spalte	9
Schlüsselbegriffe	10
Normalisierung	11
Fehlermöglichkeiten:	12
1. Normalform (NF1).....	12
2. Normalform (NF2).....	13
Vorgangsweise für die Erreichung der NF2:.....	13
Funktionale Abhängigkeit	13

3. Normalform (NF3).....	15
Vorgangsweise für die Erreichung der NF3:.....	15
Transitiv:.....	16
Aufgabe:.....	16

Einleitung

Jeder der schon einmal etwas im Internet gekauft hat, kennt die Bedeutung von Daten. Bei der Bestellung muss der Name, die Adresse und ggf. eine Bankverbindung angegeben werden. Zusätzlich werden die Bestelldaten (Artikel, Menge, Preise, ...) erfasst. Diese Daten werden von einem Informationssystem verarbeitet und bilden die Grundlage der Bestellung. Große Plattformen analysieren Kundendaten zur Echtzeit und schlagen dem Kunden während der Bestellung verwandte Produkte vor. Damit die Daten nicht mehrmals erfasst werden müssen, werden diese strukturiert gespeichert. Datenbanken bilden hier die optimale Grundlage für diese Informationen.

Daten-Informationen-Wissen

Daten sind Zeichen, die einer Syntax folgen (Zeichen, Ziffern), z.B. eine Auftragsliste.

Informationen sind Daten, denen eine Bedeutung beigemessen wird. Von **Wissen** wird gesprochen, wenn die vorliegenden Informationen für Entscheidungsprozesse genutzt werden. Die Unterschiede lassen sich am nachstehenden Beispiel erklären:

- Ein Lagerarbeiter betrachtet eine Liste mit Tagesumsätzen, diese kann als Datensammlung angenommen werden.
- Wenn diese Liste nach Artikel und Menge sortiert ist, so kann man diese Informationen verwenden.
- Nach Analyse der Lagerbestände, werden Zusammenhänge zwischen Artikeln und Menge festgestellt. Der Lagerarbeiter hat nun das **Wissen** über den genauen Lagerbestand.

Datenmodelle

Der Prozess zur Strukturierung von Daten wird als Datenmodellierung bezeichnet und ist die Basis zur elektronischen Datenverarbeitung. Bei der Modellierung müssen Informationsobjekte (Bestellmenge) sowie auf die Beziehungen (Kunden bestellen Artikel) zwischen den einzelnen Objekten beschrieben werden.

Elemente des ERM

Das ERM wurde 1976 von Peter Chen vorgestellt und baut auf drei grundlegende Elemente zur Beschreibung der Daten.

Entitätstypen (Entity-Typen)

Sind Aspekte aus der realen Welt. Ein konkreter Entitätstyp ist ein Kunde oder ein bestimmter Artikel der namentlich benannt wird. Entitätstypen werden durch ein Rechteck beschrieben.

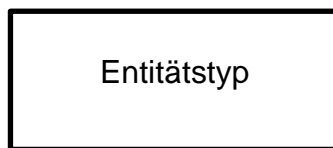


Abbildung 1: Entität

Beziehungstypen (Relationship)

Diese beschreiben den Zusammenhang zwischen den einzelnen Entitätstypen. Kunden können verschiedene Artikel bestellen. Die Beziehung wird mit einer Raute und Kanten zu den Entitätstypen dargestellt. Die Beschriftung der Beziehung erfolgt oft in Plural.

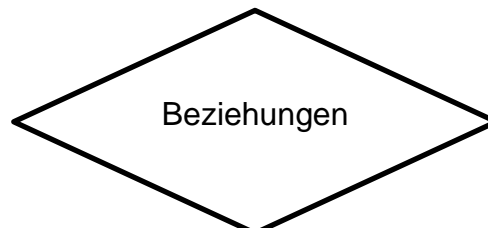


Abbildung 2: Beziehung

Attribute

Diese beschreiben die Entitätstypen und die Beziehungen näher. Ein Kunde besitzt eine Kundennummer, einen Namen usw..

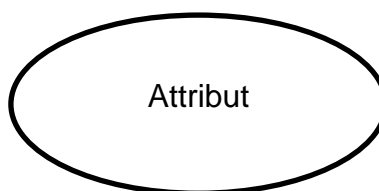


Abbildung 3: Attribut

Schlüsselattribut

Ein Schlüsselattribut ist ein Attribut, welches eine Entität identifiziert. Dieses kann aus einem Attribut – einfacher Schlüssel – oder mehreren Attributen – zusammengesetzter Schlüssel – bestehen, allerdings muss es eindeutig sein. Wenn kein eindeutiger Schlüssel zur Verfügung steht, kann ein künstlicher Schlüssel verwendet werden. Beispiele sind die Kundennummer oder eine Artikelnummer. Schlüsselattribute werden im ERM unterstrichen.

Ein Beispiel für ein einfaches ER-Diagramm:

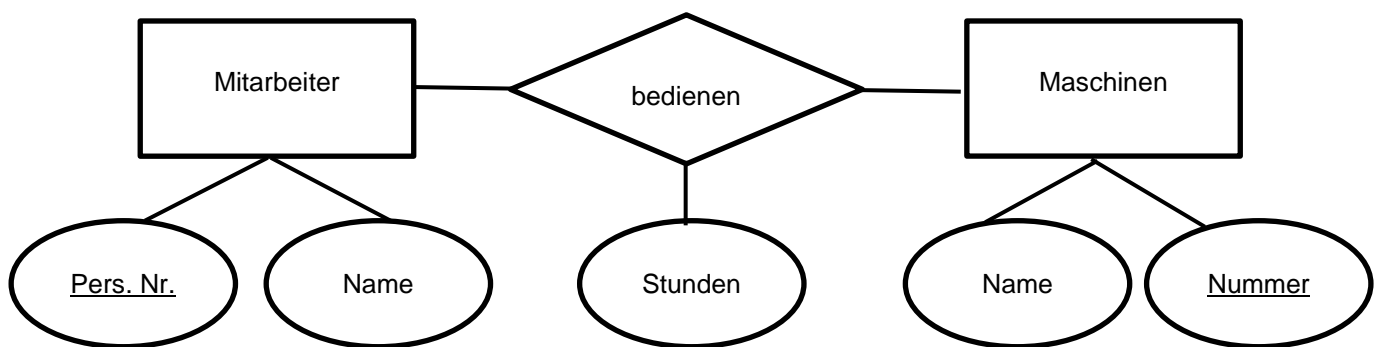


Abbildung 4: ER-Diagramm

Kardinalität

Die Beziehungen zwischen Entitäten können einfach oder komplex sein. Man kann zwischen drei grundlegenden Beziehungstypen unterscheiden:

1:1 Beziehung

Diese Beziehung beschreibt eine eindeutige Zuordnung zwischen den Entitäten. In Abbildung 5: 1:1 Beziehung ist ein Beispiel angeführt. Ein Schüler darf nur einen Ausweis besitzen, muss aber nicht.



Abbildung 5: 1:1 Beziehung

1:N Beziehung

In (Abbildung 6: 1:N Beziehung) darf ein Schüler nur eine Schule besuchen, eine Schule kann aber mehrere Schüler verwalten.

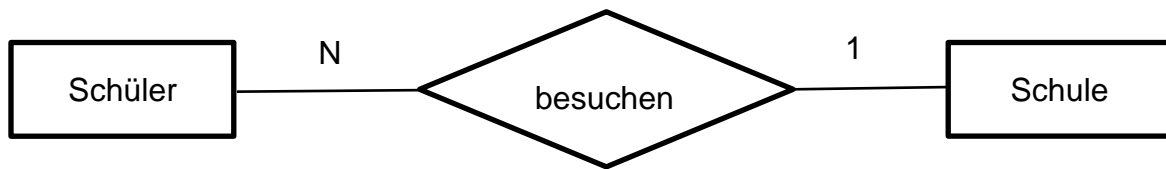


Abbildung 6: 1:N Beziehung

M:N Beziehung

In (Abbildung 7: M:N Beziehung) kann ein Schüler mehrere Unterrichtsfächer besuchen, ein Unterrichtsfach wird von mehreren Schülern besucht.

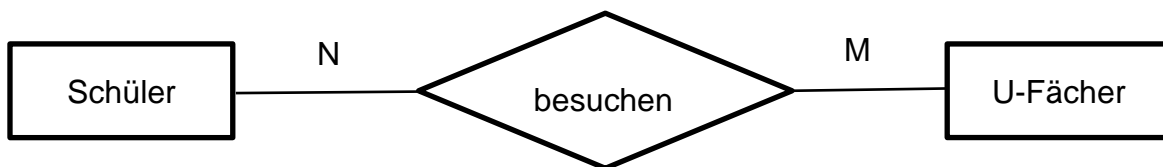


Abbildung 7: M:N Beziehung

Übung:

Modellieren Sie ein ERM mit den nachstehenden Anforderungen. Jeder Entitätstyp muss mit einem Schlüsselattribut identifiziert werden.

- Ein Mitarbeiter hat einen Namen sowie einen Wohnort
- Ein Mitarbeiter arbeitet in einer Abteilung
- Ein Mitarbeiter arbeitet an einem oder mehreren Projekten
- Die Arbeitszeit für ein Projekt soll erfasst werden
- Eine Abteilung hat einen Namen
- Ein Projekt besitzt eine Bezeichnung
- In einer Abteilung sind mehrere Mitarbeiter beschäftigt
- In einem Projekt arbeiten mehrere Mitarbeiter

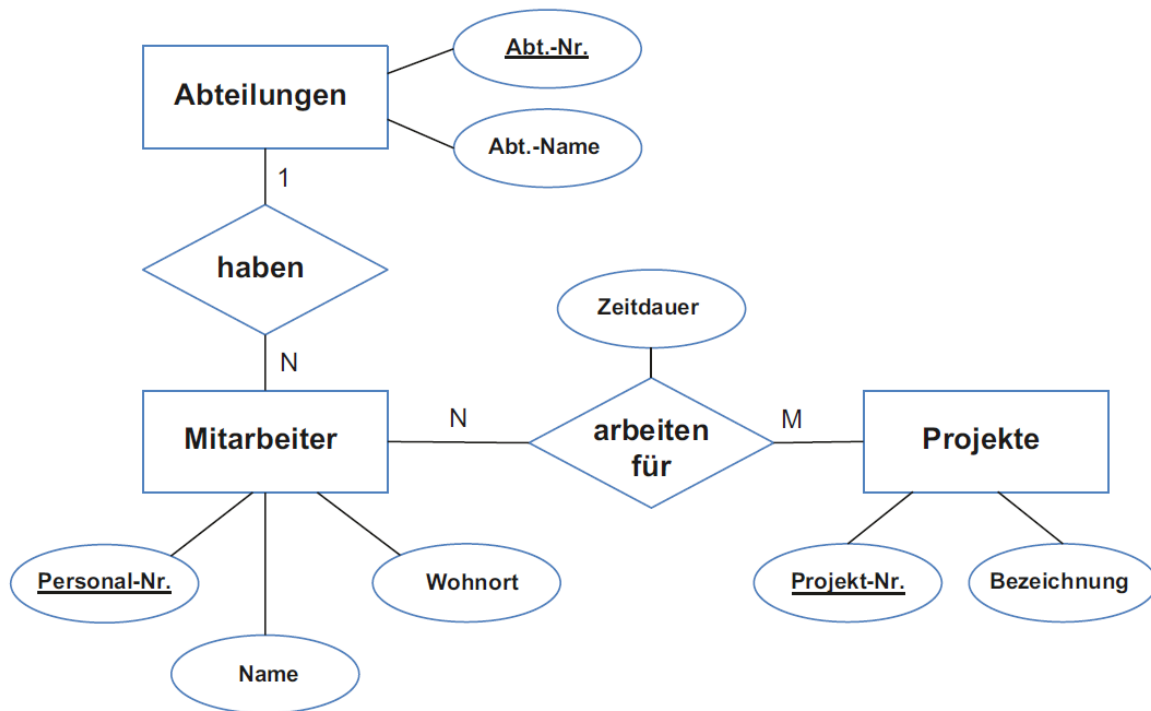


Abbildung 8: Lösungsvorschlag

Kardinalität

Die bisherige Notation hat nur Maximalkardinalitäten berücksichtigt. Sehr oft ist es auch nötig minimale Angaben zu machen. Wenn ein Objekt mindestens einmal eine Beziehung zu einer anderen Entität haben muss, wird das mit Minimalkardinalitäten angegeben.

min-max Notation

Die min-max Notation wird umgekehrt zur Chen-Notation geschrieben. Sie besteht aus zwei durch Komma getrennte Zahlen. Damit wird der minimale und maximale Wert der Beziehung dargestellt. Im nachstehenden Beispiel Abbildung 9: min-max Notation sind nachstehende Bedingungen erfüllt.

- Ein Mitarbeiter arbeitet in einer Abteilung
- Eine Abteilung hat einen oder mehrere Mitarbeiter
- Ein Mitarbeiter arbeitet in einem und höchstens in drei Projekten mit
- Ein Projekt hat mindestens einen, aber höchstens sechs Mitarbeiter

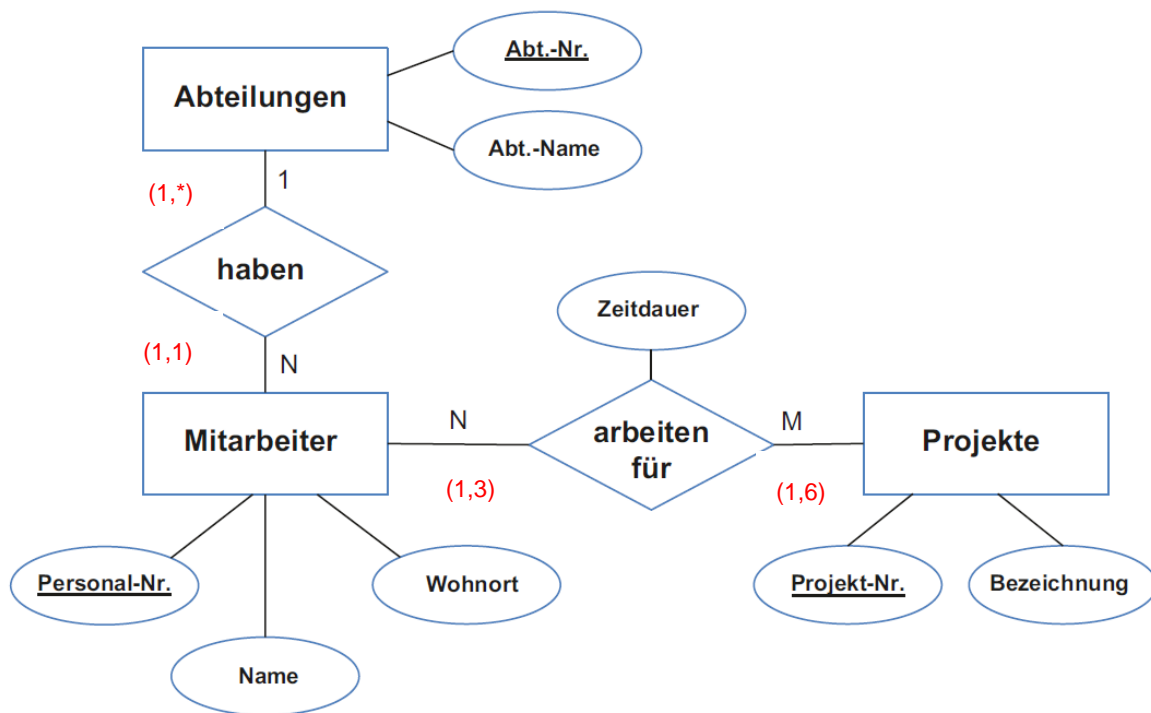


Abbildung 9: min-max Notation

Grundbegriffe relationales Datenbankmodell

Das Relationenmodell (logischer Entwurf) wurde von Edgar F. Codd entwickelt um die verteilte Datenhaltung zu verbessern. Mit diesem Modell wird das ERM (konzeptioneller Entwurf) weiter verfeinert um daraus Relationen (Tabellen) zu erhalten. Normalisierung ist somit eine weitere Möglichkeit, relationale Datenbanken zu erhalten. Ein exakter Modellentwurf macht eine Normalisierung eigentlich überflüssig, kann aber eingesetzt werden, um Anomalien zu beseitigen oder Redundanzen zu minimieren.

Relation, Tupel, Spalte

Eine Relation (Tabelle) ist eine Menge an Tupel, ein Tupel (Datensatz) ist eine Liste von Werten. Eine Relation hat einen eindeutigen Namen, 0 bis n Tupel und 1 bis m Spalten. Eine Spalte entspricht einem Attribut des ER-Modells. Jedes Tupel muss eindeutig identifizierbar sein.

Das relationale Modell hat Gemeinsamkeiten mit dem ER-Modell. Die Relation entspricht dabei einem Entitätstyp, die Tupel entsprechen den Entitäten. Nur für den Beziehungstyp gibt es keine direkte Abbildungsmöglichkeit.

Die (Abbildung 10: zusätzliche Relation Bestellung) zeigt einen Ausschnitt aus einem Datenmodell.

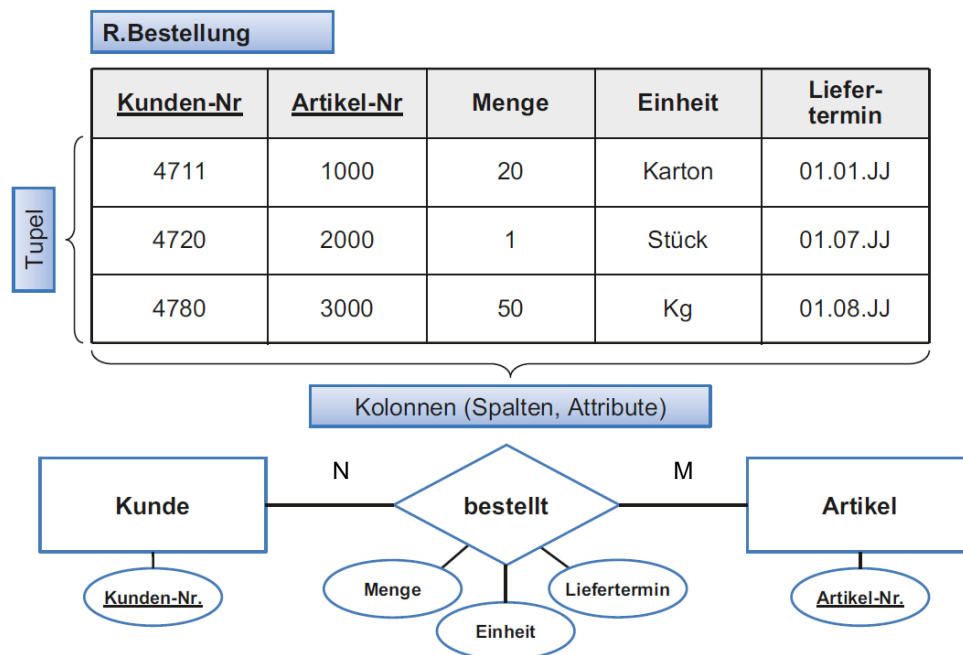


Abbildung 10: zusätzliche Relation Bestellung

Schlüsselbegriffe

Von besonderer Bedeutung ist der Primärschlüssel und der Fremdschlüssel. Jede Relation besitzt mindestens einen Primärschlüssel, der diese eindeutig identifiziert. Der Fremdschlüssel referenziert auf den Primärschlüssel einer anderen Relation. In (Abbildung 11: Schlüssel) wird die Beziehung zwischen den einzelnen Relationen abgebildet.

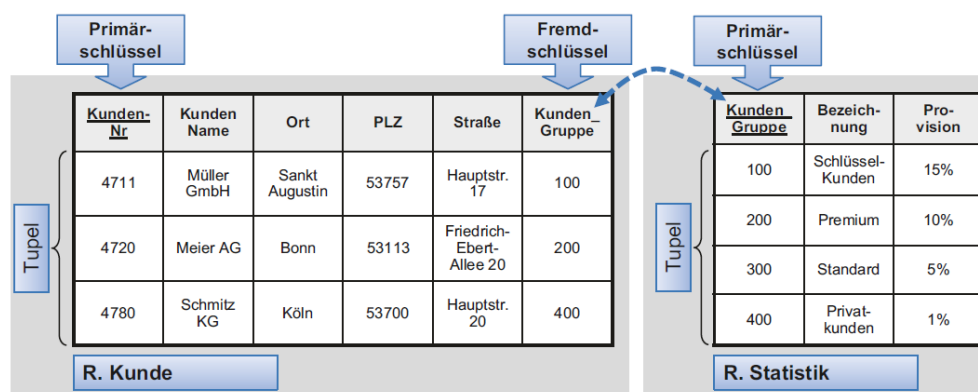


Abbildung 11: Schlüssel

Bei der Überführung vom ER-Modell (Abbildung 8: Lösungsvorschlag) in das Relationenmodell wird der Entitätstyp als Name der Relation dargestellt. Die Attribute

werden im Klammern nachfolgend geschrieben. Der Primärschlüssel wird unterstrichen.

Abteilungen (Abt.Nr., Abt-Name)

Mitarbeiter (Personal-Nr., Name, Wohnort)

Projekte (Projekt-Nr., Bezeichnung)

Projektzeit (Personal-Nr., Projekt-Nr., Zeitdauer)

Normalisierung

Ziel der Normalisierung ist die Vermeidung von Redundanzen. Jede Relation soll nur Attribute besitzen, die nötig sind. Es muss eine klare Struktur entstehen. Wenn Informationen welche nicht zusammenpassen in einer Relation gespeichert werden, können Anomalien auftreten.

Überlegen Sie, ob das nachstehende Beispiel eine gute Relation ist. Beschreiben Sie Verbesserungsvorschläge. Welche Probleme treten in Bezug auf das Sortieren nach Kundennummer oder Berechnung der Gesamtsumme der Bestellungen auf?

Beispiel:

Bestellung	Kunde
13.01.2019, 23.60	Josef Muster, Kundennummer 1234
14.01.2019, 101,99	Max Schuster, Kundennummer 123

bestellungen (bestellung, kunde)

Nach Aufteilen in einzelne Datenfelder entsteht in der Relation Redundanz.

Beispiel:

Bestelldatum	Nachname	Vorname	Betrag	Kundennr
10.01.2019	Schuster	Max	23,56	1234
11.01.2019	Schuster	Max	67,90	1234
11.01.2019	Muster	Josef	99.90	123

bestellungen (datum, nachname, vorname, betrag, kundennr)

Welche Probleme kann man in dieser Relation beobachten?

- Kundendaten von Schuster Max sind mehrfach vorhanden,
- müssen bei Änderung mehrfach geändert werden

Als Ausgangszustand für die Normalisierung wird nachstehende Tabelle verwendet.

Bestelldatum	Kunde	Rechnung
10.01.2019	Max Schuster, Salzburg, KundenNr 1234	32,99, Kreditkarte
11.01.2019	Josef Muster, Elixhausen, KundenNr 123	33,90 Vorkasse
11.01.2019	Hannes Huber, Freilassing, KundenNr 456	29,90 PayPal
12.01.2019	Max Schuster, Salzburg, KundenNr 1234	129,99 Vorkasse

ungeordnete, schlechte Datenstruktur! → fehleranfällig

Fehlermöglichkeiten:

Einfügeanomalie (unvollständiger Datensatz)

Änderungsanomalie (jedes Vorkommen muss geändert werden; Dateninkonsistenz)

Löschanomalie (wird die Bestellung von Hannes Huber storniert, sind alle Daten verloren)

1. Normalform (NF1)

Alle Attribute müssen atomar sein. Jede Information muss in einer eigenen Tabellenspalte definiert sein.

Bestelldatum	Nachname	Vorname	Wohnort	Kundennr	Betrag	Zahlungsart
10.01.2019	Schuster	Max	Salzburg	1234	32,99	Kreditkarte
11.01.2019	Muster	Josef	Elixhausen	123	33,90	Vorkasse

11.01.2019	Huber	Hannes	Freilassing	456	29,90	PayPal
12.01.2019	Schuster	Max	Salzburg	1234	129,99	Vorkasse

Tabelle 1: NF1

bestellungen (bestelldatum, nachname, vorname, wohnort, kundennr, betrag, zahlungsart)

In dieser Relation hängen Kunden nur von einem Teil des Primärschlüssels ab. Das Bestelldatum ist nicht relevant für den Namen. In nächsten Schritt wird die Relation aufgeteilt.

2. Normalform (NF2)

Die zweite Normalform (NF2) ist gegeben, wenn sich die Relation in der ersten Normalform (NF1) befindet und alle Nichtschlüsselattribute von einem Schlüsselkandidaten abhängen. Wenn Attribute von einem Teil des Schlüssels eindeutig identifiziert werden, dann liegt keine 2. Normalform vor!

Vorgangsweise für die Erreichung der NF2:

- Primärschlüssel der gegebenen Relation festlegen, falls dieser nur aus einem Attribut besteht, so liegt bereits 2. NF vor.
- Prüfen ob aus Teilschlüsselattributen weitere Attribute folgen. Falls nicht liegt bereits die 2. NF vor.
- Neue Relation bilden, die das Teilschlüsselattribut und alle von diesem abhängigen Nichtschlüsselattribute enthalten. Das Teilschlüsselattribut wird in der neuen Relation der Primärschlüssel.
- Löschen der ausgelagerten Nichtschlüsselattribute in der Ausgangsrelation.
- Vorgang ab 2. wiederholen, bis alle Nichtschlüsselattribute vom gesamten Schlüssel funktional abhängig sind.

Funktionale Abhängigkeit

Wenn ein Schlüsselkandidat eindeutig andere Attribute identifiziert spricht man von funktionaler Abhängigkeit. Die volle funktionale Abhängigkeit erreicht man wenn

$A \longrightarrow B$ und A minimal ist.

Bsp.: kundennr \rightarrow nachname, vorname, ort

bestelldatum \rightarrow betrag, zahlungsart

Es gibt funktionale Abhängigkeiten von beiden Teilschlüsseln, deshalb ist die Relation nicht in der NF2.

Aufteilen in zwei Relationen:

Kunden			
kundennr	vorname	nachname	wohntort
1234	Max	Schuster	Salzburg
123	Josef	Muster	Elixhausen
456	Hannes	Huber	Freilassing
1234	Max	Schuster	Salzburg

kunden (kundennr nachname, vorname, wohntort)

Bestellungen				
b_id	bestelldatum	kundennr	betrag	zahlungsart
1	10.01.2019	1234	32,99	Kreditkarte
2	11.01.2019	123	33,90	Vorkasse
3	11.01.2019	456	29,90	PayPal
4	12.01.2019	1234	129,99	Vorkasse

bestellungen (b_id, bestelldatum, kundennr, betrag, zahlungsart)

Jedes Nichtschlüsselattribut hängt vom gesamten Primärschlüssel ab!

Die Transformation in das Realtionenmodell hat dann nachstehende Form:

bestellungen (bestelldatum, nachname, vorname, wohntort, kundennr, betrag, zahlungsart)



kunden (kundennr nachname, vorname, wohntort)

bestellungen (b_id, bestelldatum, kundennr, betrag, zahlungsart)

3. Normalform (NF3)

Ein Relationenschema befindet sich in der 3. Normalform (NF3), wenn

- diese in der NF2 ist und
- jedes Nichtschlüsselattribut nicht transistiv vom Primärschlüssel abhängig ist

Wenn aus einem Nichtschlüsselattribut ein anderes Nichtschlüsselattribut folgt, dann liegt keine 3. Normalform vor!

Vorgangsweise für die Erreichung der NF3:

- Untersuchung, ob aus Nichtschlüsselattributen andere Nichtschlüsselattribute folgen. Falls nicht liegt bereits die NF3 vor
- Neue Relation bilden, die das Nichtschlüsselattribut (wird nun Primärschlüssel der neuen Relation) und die von ihm abhängigen Attribute enthält.
- Löschen der ausgelagerten Nichtschlüsselattribute mit Ausnahme des Attributes, das in der neuen Relation Primärschlüssel ist.
- Vorgang ab 2. wiederholen, bis keine Abhängigkeiten mehr bestehen

Bestellungen				
b_id	bestelldatum	kundennr	betrag	zahlungsart
1	10.01.2019	1234	32,99	Kreditkarte
2	11.01.2019	123	33,90	Vorkasse
3	11.01.2019	456	29,90	PayPal
4	12.01.2019	1234	129,99	Vorkasse

Abbildung 12: 3. Normalform

Kunden			
kundenr	vorname	nachname	wohntort
1234	Max	Schuster	Salzburg
123	Josef	Muster	Elixhausen
456	Hannes	Huber	Freilassing
1234	Max	Schuster	Salzburg

Abbildung 13: 3. Normalform

Transitiv:

Beispiel: es gibt drei Attribute y (Primärschlüssel), x, z

Wenn $x \rightarrow y$ (x funktional von y abhängig) und $z \rightarrow x$ (z abhängig von x ist), dann kann von $z \rightarrow x \rightarrow y$ bestimmt werden.

Die Transformation in die 3. Normalform ist nachstehend angeführt

bestellungen (b_id, bestelldatum, ↑kundenr, betrag, zahlungsart)

kunden (kundenr, nachname, vorname, wohnort)

Aufgabe:

Ein Auftraggeber verwaltet eine Tabelle in der ein Kunde mit Wohnort, Straße, PLZ gespeichert wird. Zusätzlich gibt es eine Tabelle in der die Artikelnummer, der Artikelname, Menge und Preis gespeichert werden.

Kunde	Bestellung
Megabit GmbH, Kurgasse 2, 6691 Jungholz	1234, SSD 250GB, 8 Stück je 59,90€

- erstellen Sie als erstes ein ER-Modell
- transformieren Sie dieses in ein Relationenmodell
- führen Sie die Schritte für die 3. Normalform durch