

Seite 5. Relationales Datenbankmodell

Relationales Datenbankmodell

Diese Seite erklärt das **relationale Datenbankmodell** so, dass du es schnell für IHK-Aufgaben anwenden kannst.

Das relationale Datenbankmodell beschreibt, wie Daten in **Tabellen** gespeichert werden.

Dabei sind besonders wichtig:

Tabellen
Datensätze
Attribute / Spalten
Primärschlüssel
Fremdschlüssel
Beziehungen zwischen Tabellen

Grafik 1: Grundlagen des relationalen Datenbankmodells

Relationales Datenbankmodell – Grundlagen

Diese Grafik zeigt dir die wichtigsten Grundbegriffe:

- **Tabelle**
- **Datensatz**
- **Primärschlüssel**
- **Fremdschlüssel**
- Zusammenhang zwischen Zeilen, Spalten und Schlüsseln

Wofür braucht man das relationale Datenbankmodell?

Das relationale Datenbankmodell wird benutzt, um ein fachliches Datenmodell technisch näher an eine echte Datenbank zu bringen.

Ein ERM zeigt zuerst fachlich:

Kunde gibt Bestellung auf

Das relationale Modell macht daraus Tabellen:

Kunde

Bestellung

Und legt fest:

Welche Spalten gibt es?

Was ist der Primärschlüssel?

Wo wird ein Fremdschlüssel benötigt?

Wie hängen die Tabellen zusammen?

Woran erkenne ich in einer Aufgabe, dass ein relationales Datenbankmodell gemeint ist?

Typische Signalwörter:

Signalwort / Formulierung	Hinweis
Tabelle	Daten sollen tabellarisch dargestellt werden
Datensatz	Es geht um Zeilen in einer Tabelle
Attribut / Spalte	Es geht um Felder einer Tabelle
Primärschlüssel	Eindeutige Kennzeichnung eines Datensatzes
Fremdschlüssel	Verweis auf eine andere Tabelle
Relation	Tabelle oder Beziehung zwischen Tabellen
Normalisierung	Tabellen sollen sauber strukturiert werden
n:m auflösen	Zwischentabelle wird benötigt
ERM in Tabellen überführen	Relationales Modell ist gemeint

Grundbegriffe

Begriff	Bedeutung	Beispiel
Tabelle	Sammlung gleichartiger Datensätze	Kunde, Bestellung, Produkt
Datensatz	Eine Zeile in einer Tabelle	ein bestimmter Kunde
Attribut / Spalte	Eigenschaft eines Datensatzes	name, email, preis
Primärschlüssel	eindeutige ID einer Tabelle	kunden_id
Fremdschlüssel	Verweis auf Primärschlüssel einer anderen Tabelle	kunden_id in Bestellung

Begriff	Bedeutung	Beispiel
Relation	Tabelle oder Beziehung zwischen Tabellen	Kunde 1:n Bestellung

Tabelle

Eine Tabelle speichert gleichartige Daten.

Beispiel Tabelle **Kunde**:

kunden_id	name	email
1	Müller	mueller@example.de
2	Schmidt	schmidt@example.de

Die Tabelle hat:

- Spalten
 - Zeilen
 - Datensätze
 - einen Primärschlüssel
-

Datensatz

Ein Datensatz ist eine einzelne Zeile in einer Tabelle.

Beispiel:

kunden_id	name	email
1	Müller	mueller@example.de

Dieser eine Kunde ist ein Datensatz.

Merksatz:

Datensatz = eine vollständige Zeile einer Tabelle

Attribut / Spalte

Ein Attribut ist im relationalen Modell eine Spalte.

Beispiel Tabelle **Kunde**:

```
kunden_id  
name  
email  
telefon
```

Diese Spalten beschreiben den Kunden genauer.

Merksatz:

Attribut im ERM wird meistens zur Spalte in der Tabelle.

Primärschlüssel

Ein Primärschlüssel identifiziert jeden Datensatz eindeutig.

Beispiel:

```
kunden_id  
produkt_id  
bestellung_id
```

Warum ist das wichtig?

Namen sind nicht eindeutig.

Beispiel:

```
Müller
```

kann mehrfach vorkommen.

Aber:

```
kunden_id = 1
```

ist eindeutig.

Merksatz:

Jede Tabelle sollte einen Primärschlüssel haben.

Fremdschlüssel

Ein Fremdschlüssel verweist auf den Primärschlüssel einer anderen Tabelle.

Beispiel:

Tabelle **Kunde**:

kunden_id	name
1	Müller
2	Schmidt

Tabelle **Bestellung**:

bestellung_id	bestelldatum	kunden_id
101	01.02.2026	1
102	03.02.2026	1

Hier ist:

kunden_id

in der Tabelle **Bestellung** ein Fremdschlüssel.

Warum?

Weil er auf die Tabelle **Kunde** verweist.

Bestellung.kunden_id → Kunde.kunden_id

Grafik 2: Vom ERM zum relationalen Modell

Vom ERM zum relationalen Modell

Diese Grafik zeigt:

- Aus der Entität **Kunde** wird die Tabelle **Kunde**
- Aus der Entität **Bestellung** wird die Tabelle **Bestellung**
- Die 1:n-Beziehung wird durch einen **Fremdschlüssel** auf der n-Seite abgebildet
- In der Tabelle **Bestellung** steht deshalb `kunden_id` als Fremdschlüssel

ERM vs. relationales Datenbankmodell

ERM	Relationales Datenbankmodell
fachliches Modell	tabellarisches Modell
zeigt Entitäten	zeigt Tabellen
zeigt Attribute	zeigt Spalten
zeigt Beziehungen	zeigt Fremdschlüssel
zeigt Kardinalitäten	zeigt Schlüsselbeziehungen
eher Planung	näher an Datenbank / SQL

Merksatz:

ERM = Was gibt es fachlich?

Relationales Modell = Wie wird es tabellarisch gespeichert?

Aus Entitäten werden Tabellen

Im ERM hast du zum Beispiel:

Kunde
Bestellung
Produkt

Im relationalen Modell werden daraus Tabellen:

Tabelle Kunde
Tabelle Bestellung
Tabelle Produkt

Beispiel:

Entität Kunde

wird zu:

Kunde(kunden_id, name, email)

Aus Attributen werden Spalten

ERM:

```
Kunde
- kunden_id
- name
- email
```

Relationales Modell:

kunden_id	name	email
-----------	------	-------

Merksatz:

Attribute werden im relationalen Modell zu Spalten.

1:1-Beziehung im relationalen Modell

Beispiel:

```
Person 1 --- 1 Personalausweis
```

Mögliche Tabellen:

```
Person(person_id, name)
Personalausweis(ausweis_id, ausweisnummer, person_id)
```

Hier kann `person_id` in **Personalausweis** als Fremdschlüssel gespeichert werden.

Wichtig:

Bei 1:1 ist fachlich zu prüfen, ob man beide Tabellen wirklich getrennt braucht.

1:n-Beziehung im relationalen Modell

Beispiel:

```
Kunde 1 --- n Bestellung
```

Das bedeutet:

Ein Kunde kann mehrere Bestellungen haben.
Eine Bestellung gehört genau zu einem Kunden.

Tabellen:

Kunde(kunden_id, name, email)

Bestellung(bestellung_id, bestelldatum, kunden_id)

Wichtig:

Der Fremdschlüssel kommt auf die **n-Seite**.

Also:

kunden_id

steht in:

Bestellung

Merksatz:

Bei 1:n steht der Fremdschlüssel auf der n-Seite.

n:m-Beziehung im relationalen Modell

Beispiel:

Bestellung n --- m Produkt

Das bedeutet:

Eine Bestellung kann mehrere Produkte enthalten.
Ein Produkt kann in mehreren Bestellungen vorkommen.

Eine n:m-Beziehung kann man nicht sauber direkt mit nur einem Fremdschlüssel abbilden.

Deshalb braucht man eine Zwischentabelle.

Aus:

Bestellung n:m Produkt

wird:

Bestellung 1:n Bestellposition n:1 Produkt

Tabellen:

Bestellung(bestellung_id, bestelldatum)

Produkt(produkt_id, bezeichnung, preis)

Bestellposition(bestellung_id, produkt_id, menge)

Die Tabelle **Bestellposition** enthält mindestens:

Spalte	Bedeutung
bestellung_id	Fremdschlüssel auf Bestellung
produkt_id	Fremdschlüssel auf Produkt
menge	Zusatzinformation zur Beziehung

Warum braucht man bei n:m eine Zwischentabelle?

Beispiel:

Eine Bestellung enthält mehrere Produkte.

Ein Produkt kann in mehreren Bestellungen vorkommen.

Zusätzlich muss man speichern:

Wie oft wurde ein Produkt bestellt?

Welcher Preis galt zum Bestellzeitpunkt?

Gab es Rabatt?

Diese Informationen gehören nicht nur zur Bestellung und nicht nur zum Produkt.

Sie gehören zur Verbindung zwischen Bestellung und Produkt.

Deshalb braucht man:

Bestellposition

Typische Umwandlung von ERM zu relationalem Modell

ERM	Relationales Modell
Entität	Tabelle
Attribut	Spalte
Primärschlüssel	Primärschlüssel
1:n-Beziehung	Fremdschlüssel auf n-Seite
n:m-Beziehung	Zwischentabelle
Kardinalität	Schlüsselbeziehung

Beispiel: Kunde und Bestellung

Fachliche Beschreibung:

Ein Kunde kann mehrere Bestellungen aufgeben.

Eine Bestellung gehört genau zu einem Kunden.

ERM:

Kunde 1 --- n Bestellung

Relationales Modell:

Kunde(kunden_id, name, email)

Bestellung(bestellung_id, bestelldatum, kunden_id)

Erklärung:

kunden_id ist Primärschlüssel in Kunde.
kunden_id ist Fremdschlüssel in Bestellung.

Beispiel: Schüler und Kurs

Fachliche Beschreibung:

Ein Schüler kann mehrere Kurse belegen.
Ein Kurs kann von mehreren Schülern belegt werden.

ERM:

Schüler n --- m Kurs

Relationales Modell:

Schüler(schueler_id, vorname, nachname)

Kurs(kurs_id, kursname)

Belegung(schueler_id, kurs_id, anmeldedatum)

Erklärung:

Belegung ist die Zwischentabelle.
schueler_id ist Fremdschlüssel auf Schüler.
kurs_id ist Fremdschlüssel auf Kurs.

Primärschlüssel und Fremdschlüssel markieren

In der Prüfung kannst du Schlüssel zum Beispiel so kennzeichnen:

PK = Primary Key = Primärschlüssel
FK = Foreign Key = Fremdschlüssel

Beispiel:

```
Kunde(  
  kunden_id PK,  
  name,  
  email  
)
```

```
Bestellung(  
  bestellung_id PK,  
  bestelldatum,  
  kunden_id FK  
)
```

Oder kompakt:

```
Kunde(kunden_id PK, name, email)
```

```
Bestellung(bestellung_id PK, bestelldatum, kunden_id FK)
```

Zusammengesetzter Primärschlüssel

Bei Zwischentabellen kann der Primärschlüssel aus mehreren Spalten bestehen.

Beispiel:

```
Belegung(schueler_id, kurs_id, anmeldedatum)
```

Hier könnten zusammen den Datensatz eindeutig machen:

```
schueler_id + kurs_id
```

Das nennt man einen **zusammengesetzten Primärschlüssel**.

Alternative:

Man kann auch eine eigene ID verwenden:

```
belegung_id
```

Dann sieht die Tabelle so aus:

Belegung(belegung_id PK, schueler_id FK, kurs_id FK, anmeldedatum)

Beides kann je nach Aufgabenstellung sinnvoll sein.

Normalisierung kurz erklärt

Normalisierung bedeutet:

Daten so strukturieren, dass Wiederholungen und Fehler vermieden werden.

Einfach gesagt:

Gleiche Informationen sollen nicht unnötig mehrfach gespeichert werden.

Schlechtes Beispiel:

bestellung_id	kundenname	kundenemail	produkt
1	Müller	mueller@example.de	Monitor
2	Müller	mueller@example.de	Tastatur

Problem:

Kundendaten werden mehrfach gespeichert.

Besser:

Kunde(kunden_id, name, email)

Bestellung(bestellung_id, kunden_id)

Dann stehen die Kundendaten nur einmal in der Tabelle **Kunde**.

Vorgehensweise in der Prüfung

Wenn du ein relationales Datenbankmodell erstellen sollst, gehe so vor:

Schritt	Frage
1	Welche Entitäten gibt es?

Schritt	Frage
2	Welche Tabellen entstehen daraus?
3	Welche Attribute werden zu Spalten?
4	Was ist der Primärschlüssel jeder Tabelle?
5	Gibt es eine 1:n-Beziehung?
6	Wo muss der Fremdschlüssel stehen?
7	Gibt es eine n:m-Beziehung?
8	Welche Zwischentabelle wird benötigt?
9	Welche Zusatzattribute gehören in die Zwischentabelle?

Typische IHK-Fehler

Fehler	Warum problematisch?
Primärschlüssel vergessen	Datensätze sind nicht eindeutig erkennbar
Fremdschlüssel auf falscher Seite	Beziehung wird falsch abgebildet
n:m ohne Zwischentabelle	Relational unsauber
Attribute doppelt speichern	führt zu Redundanz
Entität und Attribut verwechseln	falsche Tabellenstruktur
Zwischentabelle ohne Fremdschlüssel	Verbindung zwischen Tabellen fehlt
Kardinalität nicht beachtet	falsche Modellierung

Prüfungs-Merksätze

Tabelle = Sammlung gleichartiger Datensätze
 Datensatz = eine Tabellenzeile
 Attribut = Spalte
 Primärschlüssel = eindeutige ID
 Fremdschlüssel = Verweis auf andere Tabelle
 1:n = Fremdschlüssel auf der n-Seite
 n:m = Zwischentabelle nötig
 ERM = fachliche Sicht
 Relationales Modell = Tabellen- und Schlüssel-Sicht

Mini-Beispiel 1

Aufgabe:

Ein Kunde kann mehrere Bestellungen haben.
Eine Bestellung gehört genau zu einem Kunden.

Lösung:

Kunde(kunden_id PK, name, email)

Bestellung(bestellung_id PK, bestelldatum, kunden_id FK)

Begründung:

1:n-Beziehung.
Der Fremdschlüssel kunden_id steht auf der n-Seite, also in Bestellung.

Mini-Beispiel 2

Aufgabe:

Ein Schüler kann mehrere Kurse belegen.
Ein Kurs kann von mehreren Schülern belegt werden.

Lösung:

Schüler(schueler_id PK, vorname, nachname)

Kurs(kurs_id PK, kursname)

Belegung(schueler_id FK, kurs_id FK, anmeldedatum)

Begründung:

n:m-Beziehung.
Deshalb braucht man eine Zwischentabelle.

Mini-Beispiel 3

Aufgabe:

Eine Bestellung kann mehrere Produkte enthalten.
Ein Produkt kann in mehreren Bestellungen vorkommen.

Lösung:

Bestellung(bestellung_id PK, bestelldatum)

Produkt(produkt_id PK, bezeichnung, preis)

Bestellposition(bestellung_id FK, produkt_id FK, menge)

Begründung:

n:m-Beziehung.
Bestellposition löst die Beziehung auf und speichert zusätzlich die Menge.

Mini-Testfragen

1. Was ist eine Tabelle?

Eine Tabelle speichert gleichartige Datensätze.

Beispiel:

Kunde
Produkt
Bestellung

2. Was ist ein Datensatz?

Ein Datensatz ist eine einzelne Zeile in einer Tabelle.

Beispiel:

```
kunden_id = 1, name = Müller, email = mueller@example.de
```

3. Was ist ein Primärschlüssel?

Ein Primärschlüssel identifiziert jeden Datensatz eindeutig.

Beispiel:

```
kunden_id  
produkt_id  
bestellung_id
```

4. Was ist ein Fremdschlüssel?

Ein Fremdschlüssel verweist auf den Primärschlüssel einer anderen Tabelle.

Beispiel:

```
Bestellung.kunden_id verweist auf Kunde.kunden_id
```

5. Wo steht der Fremdschlüssel bei einer 1:n-Beziehung?

Der Fremdschlüssel steht auf der **n-Seite**.

Beispiel:

```
Kunde 1:n Bestellung
```

Dann steht:

```
kunden_id
```

in der Tabelle:

Bestellung

6. Was braucht man bei einer n:m-Beziehung?

Eine n:m-Beziehung braucht eine **Zwischentabelle**.

Beispiel:

Schüler n:m Kurs

wird zu:

Schüler 1:n Belegung n:1 Kurs

7. Warum ist eine Zwischentabelle bei Bestellung und Produkt sinnvoll?

Weil eine Bestellung mehrere Produkte enthalten kann und ein Produkt in mehreren Bestellungen vorkommen kann.

Außerdem können Zusatzinformationen gespeichert werden:

menge
einzelpreis
rabatt

Diese gehören zur Verbindung zwischen Bestellung und Produkt.

8. Was bedeutet Normalisierung ganz einfach?

Normalisierung bedeutet, Daten so zu strukturieren, dass unnötige Wiederholungen und Fehler vermieden werden.

Merksatz:

Informationen möglichst nur einmal speichern.

Nächste Seite

Danach kommt die eigene Trainer-Seite:

Relationales-Datenbankmodell-Trainer

Dort bauen wir den interaktiven Trainer mit Aufgaben zu:

Tabellen ableiten

Primärschlüssel bestimmen

Fremdschlüssel richtig platzieren

1:n-Beziehungen umsetzen

n:m-Beziehungen mit Zwischentabelle auflösen

Revision #2

Created 27 May 2026 22:54:42 by Admin

Updated 27 May 2026 22:55:48 by Admin