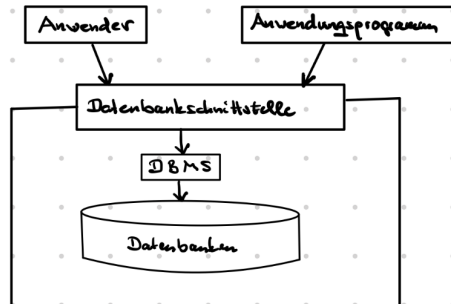


1. (4 Punkte) Nenne die Komponenten eines relationalen Datenbanksystems und beschreibe deren Funktion. Skizziere ein relationales Datenbanksystem und beschrifte die Komponenten.

Lösung: Ein relationales Datenbanksystem besteht aus einer in Tabellen strukturierten Sammlung von Daten (Datenbanken) und einer Software zur Verwaltung dieser Daten, dem sog. Datenbankmanagementsystem (DBMS). Das DBMS kontrolliert Zugriff und Verarbeitung der Daten in der Datenbasis. Anwender oder Anwendungsprogramme nutzen die Funktionalität des DBMS über eine Datenbankschnittstelle.



2. (4 Punkte) In einer Schulbibliothek werden Bücher verwaltet. Die Datenbank enthält die folgende Tabelle:

BuchID	Titel	Autor	Erscheinungsjahr
101	Die Physiker	Dürrenmatt	1962
102	Tschick	Herrndorf	2010
103	Krabat	Preußler	1971

Erläutere die Begriffe **Entität**, **Entitätstyp**, **Attribut**, **Tabelle**, **Datensatz** und **Datenfeld** anhand des Beispiels.

Lösung:

- **Entität:** Ein konkretes Objekt der realen Welt, z. B. das Buch „Tschick“.
- **Entitätstyp:** Die Klasse ähnlicher Objekte, hier „Buch“.
- **Attribut:** Eine Eigenschaft eines Entitätstyps, z. B. Titel oder Autor.
- **Tabelle:** Die strukturierte Darstellung eines Entitätstyps in Zeilen und Spalten.
- **Datensatz:** Eine Zeile der Tabelle, die alle Informationen zu einer Entität enthält.
- **Datenfeld:** Ein einzelner Wert innerhalb eines Datensatzes, z. B. „Tschick“ oder „2010“.

3. (4 Punkte) Eine Fahrschule verwaltet Informationen über Schüler und Fahrlehrer.

- Jeder Schüler besitzt eine Schülernummer, einen Namen und ein Geburtsdatum.
- Jeder Fahrlehrer besitzt eine Personalnummer und einen Namen.
- Ein Fahrlehrer unterrichtet mehrere Schüler.
- Jeder Schüler wird genau einem Fahrlehrer zugeordnet.

Bestimme die Entitätstypen und ihre Attribute. Beschreibe die Beziehung zwischen Schüler und Fahrlehrer und gib die Kardinalität an.

Lösung: Entitätstypen:

- Schüler - Attribute: Schülernummer, Name, Geburtsdatum
- Fahrlehrer - Attribute: Personalnummer, Name

Beziehung:

Ein Fahrlehrer unterrichtet Schüler.

Kardinalität:

1:n-Beziehung

Ein Fahrlehrer kann mehrere Schüler unterrichten, jeder Schüler hat jedoch genau einen Fahrlehrer.

4. (4 Punkte) Betrachte die folgende Tabelle:

KundenID	Name	Wohnort
101	Anna Meier	Stuttgart
102	Max Schulz	Böblingen
103	Lea Weber	Sindelfingen

Erläutere die Begriffe **Schlüssel** und **Primärschlüssel**. Bestimme außerdem den Primärschlüssel der Tabelle und begründe deine Wahl.

Lösung: Ein Schlüssel ist ein Attribut oder eine Attributkombination, mit der ein Datensatz eindeutig identifiziert werden kann.

Ein Primärschlüssel ist der ausgewählte Schlüssel, der jeden Datensatz einer Tabelle eindeutig kennzeichnet.

Der Primärschlüssel der Tabelle ist **KundenID**, da jede KundenID nur einmal vorkommt und jeden Kunden eindeutig identifiziert.

Die Attribute Name und Wohnort sind ungeeignet, da mehrere Kunden denselben Namen oder Wohnort besitzen könnten.

5. (4 Punkte) Eine Datenbank einer Schule enthält die folgenden Tabellen:

Schueler

SchuelerID	Name
1	Anna Meier
2	Max Schulz
3	Lea Weber

Kurs

KursID	Kursname	SchuelerID
10	Informatik	1
11	Mathematik	1
12	Physik	2

- Bestimme die Primärschlüssel beider Tabellen.
- Bestimme den Fremdschlüssel.
- Erläutere die Funktion eines Fremdschlüssels.

Lösung:

1. Primärschlüssel:

- Tabelle Schueler: SchuelerID
- Tabelle Kurs: KursID

2. Fremdschlüssel: SchuelerID in der Tabelle Kurs

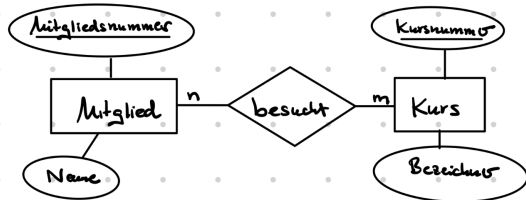
3. Ein Fremdschlüssel verweist auf den Primärschlüssel einer anderen Tabelle. Dadurch können Datensätze verschiedener Tabellen miteinander verknüpft werden. In diesem Beispiel zeigt die SchuelerID in der Tabelle Kurs auf einen Schüler in der Tabelle Schueler.

6. (4 Punkte) Ein Fitnessstudio verwaltet Informationen über seine Mitglieder und Kurse. Die Mitglieder können an mehreren Kursen teilnehmen.

- Jedes Mitglied besitzt eine Mitgliedsnummer und einen Namen.
- Jeder Kurs besitzt eine Kursnummer und eine Bezeichnung.

Modelliere den beschriebenen Ausschnitt der Realität als Entity-Relationship-Diagramm. Kennzeichne die Schlüssel und gib die Kardinalitäten an.

Lösung:

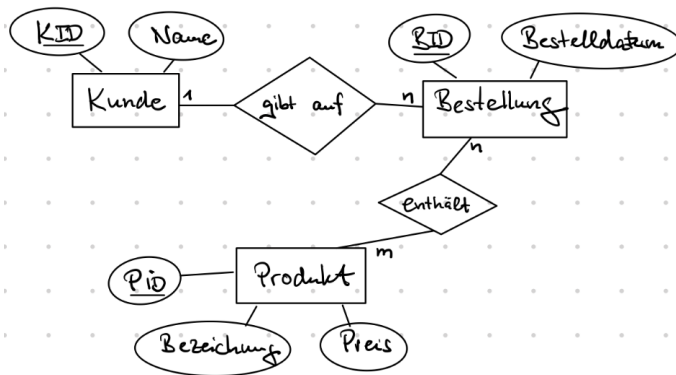


7. (5 Punkte) Ein Online-Shop verwaltet Informationen über Kunden, Bestellungen und Produkte.

- Jeder Kunde besitzt eine KundenID und einen Namen.
- Jede Bestellung besitzt eine BestellID und ein Bestelldatum.
- Jedes Produkt besitzt eine ProduktID, eine Bezeichnung und einen Preis.
- Ein Kunde kann mehrere Bestellungen aufgeben, jede Bestellung gehört jedoch genau zu einem Kunden.
- Eine Bestellung kann mehrere Produkte enthalten, und ein Produkt kann in mehreren Bestellungen enthalten sein.

Modelliere den beschriebenen Ausschnitt der Realität als Entity-Relationship-Diagramm. Kennzeichne die Schlüssel und gib die Kardinalitäten an.

Lösung:



8. (4 Punkte) Erläutere, was es bedeutet, dass eine Datenbank in der 1. Normalform vorliegt. Gib ein Beispiel an für eine Datenbank, die nicht in der 1. Normalform vorliegt, und erkläre, wie man sie in die 1. Normalform überführen kann.

Lösung: Eine Datenbank befindet sich in der 1. Normalform, wenn alle Attribute nur atomare Werte enthalten. Das bedeutet: In einer Zelle steht genau ein Wert, keine Listen, keine mehrfachen Werte und keine zusammengesetzten Felder.

Beispiel für keine 1. Normalform:

Name	Telefonnummern
Anna Meier	030-123456, 030-654321
Max Schulz	040-987654

Hier ist das Attribut *Telefonnummern* nicht atomar, weil mehrere Werte in einer Zelle stehen.

Überführung in die 1. Normalform:

Name	Telefonnummer
Anna Meier	030-123456
Anna Meier	030-654321
Max Schulz	040-987654

Nun hat jede Zeile genau einen Telefonwert, also liegt die Tabelle in der 1. Normalform vor.

9. (4 Punkte) Erläutere, was es bedeutet, dass eine Datenbank in der 2. Normalform vorliegt. Gib ein Beispiel an für eine Datenbank, die nicht in der 2. Normalform vorliegt, und erkläre, wie man sie in die 2. Normalform überführen kann.

Lösung: Eine Datenbank befindet sich in der 2. Normalform, wenn sie in der 1. Normalform vorliegt und jedes Nicht-Schlüsselattribut voll vom gesamten Schlüssel abhängt.

Beispiel für keine 2. Normalform:

KursID	SchülerID	SchülerName
10	1	Anna Meier
10	2	Max Schulz
11	1	Anna Meier

Hier ist der Schlüssel zusammengesetzt aus *KursID* und *SchülerID*. Das Attribut *SchülerName* hängt aber nur von *SchülerID* ab und damit nicht vom ganzen Schlüssel.

Überführung in die 2. Normalform:

SchülerID	SchülerName
1	Anna Meier
2	Max Schulz

KursID	SchülerID
10	1
10	2
11	1

Nun sind alle Nicht-Schlüsselattribute voll vom gesamten Schlüssel abhängig. Die Tabellen liegen damit in der 2. Normalform vor.

10. (4 Punkte) Erläutere, was es bedeutet, dass eine Datenbank in der 3. Normalform vorliegt. Gib ein Beispiel an für eine Datenbank, die nicht in der 3. Normalform vorliegt, und erkläre, wie man sie in die 3. Normalform überführen kann.

Lösung: Eine Datenbank befindet sich in der 3. Normalform, wenn sie in der 2. Normalform vorliegt und kein Nicht-Schlüsselattribut von einem anderen Nicht-Schlüsselattribut abhängt.

Beispiel für keine 3. Normalform:

MitarbeiterID	Name	AbteilungsID	AbteilungsName
1	Anna Meier	10	Buchhaltung
2	Max Schulz	20	IT
3	Lea Weber	10	Buchhaltung

Hier ist *MitarbeiterID* der Schlüssel. Das Attribut *AbteilungsName* hängt aber nicht direkt vom Schlüssel ab, sondern von *AbteilungsID*.

Überführung in die 3. Normalform:

MitarbeiterID	Name	AbteilungsID
1	Anna Meier	10
2	Max Schulz	20
3	Lea Weber	10

AbteilungsID	AbteilungsName
10	Buchhaltung
20	IT

Nun hängt jedes Nicht-Schlüsselattribut nur noch direkt vom Schlüssel ab. Die Tabellen liegen damit in der 3. Normalform vor.

11. (4 Punkte) Gegeben ist die folgende Tabelle einer Schule:

SchuelerID	Schuelername	Klassenname	Klassenlehrer
1	Anna Meier	10A	Herr Müller
2	Max Schulz	10A	Herr Müller
3	Lea Weber	10B	Frau Schmidt

Erläutere, warum die Tabelle Redundanzen enthält. Überführe die Tabelle anschließend in ein relationales Datenbankschema in der 3. Normalform.

Lösung: Die Information, wer der Klassenlehrer eine Klasse ist, wird mehrfach gespeichert. Die Information „10A – Herr Müller“ erscheint in mehreren Datensätzen. Dadurch können Redundanzen und Inkonsistenzen entstehen.

Relationales Schema:

- Schueler(SchuelerID, Schuelername, ↑KlassenID)
- Klasse(KlassenID, Klassenname, Klassenlehrer)

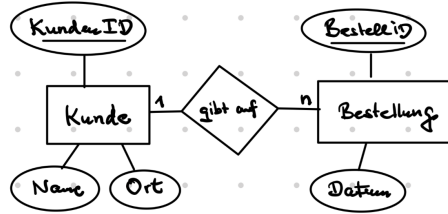
Die Informationen über Klassen werden nun nur noch einmal gespeichert.

12. (4 Punkte) Gegeben ist das folgende relationale Datenbankschema:

- Kunde(KundenID, Name, Ort)
- Bestellung(BestellNr, Datum, ↑KundenID)

Überführe das relationale Schema in ein Entity-Relationship-Diagramm. Mache die Kardinalitäten und Schlüssel deutlich.

Lösung:

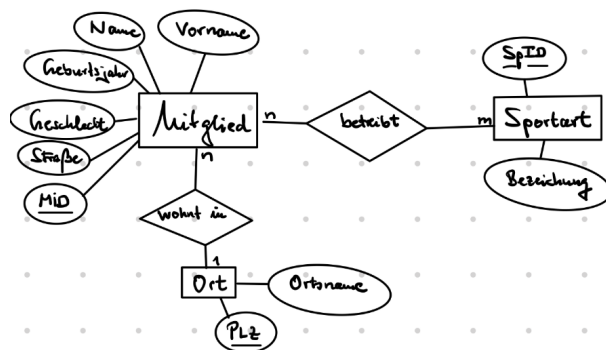


13. (5 Punkte) Gegeben ist folgendes relationale Datenbankschema:

- Mitglied (MID, Name, Vorname, Geburtsjahr, Geschlecht, Straße, ↑PLZ)
- Ort (PLZ, Ortsname)
- Sportart(SpID, Bezeichnung)
- betreibt_Sportart (↑MID, ↑SpID)

Entwurf zu dem Datenbankschemas ein passendes ER-Diagramm, das die Beziehungen zwischen den Entitäten darstellt. Gib dabei die Kardinalitäten der Beziehungen an.

Lösung:



14. (3 Punkte) Beschreibe, wie im Unterricht mit dem Datenbankmanagementsystem SQLite gearbeitet wurde.

Gehe dabei auf die folgenden Aspekte ein:

- Herstellen einer Datenbankverbindung
- Erstellen von Tabellen
- Einfügen von Daten
- Abfragen von Daten
- Löschen von Daten

Lösung: Wir verwenden das Datenbankmanagementsystem (DBMS) SQLite. Der Zugriff auf die Datenbank erfolgt aus Python heraus über das Modul sqlite3, das eine Schnittstelle zu SQLite bereitstellt.

- Mit `sqlite3.connect()` wurde eine Verbindung zu einer Datenbank hergestellt. Die Verbindung zur Datenbank gibt uns einen Cursor zurück, mit dem wir unsere SQL Anweisungen übermitteln.
- Mit dem SQL-Befehl `CREATE TABLE` wurden Tabellen angelegt.
- Mit `INSERT INTO` wurden Datensätze eingefügt.
- Mit `SELECT` wurden Daten aus Tabellen abgefragt.
- Mit `DELETE FROM` wurden Datensätze gelöscht.

15. (3 Punkte) Gegeben ist die folgende Tabelle:

Schueler

SchuelerID	Name	Klasse
1	Anna Meier	10A
2	Max Schulz	10B
3	Lea Weber	10A

Formuliere jeweils eine SQL-Abfrage für die folgenden Aufgaben:

1. Gib alle Daten aller Schülerinnen und Schüler aus.
2. Gib nur die Namen aller Schülerinnen und Schüler aus.
3. Gib alle Schülerinnen und Schüler der Klasse 10A aus.

Lösung:

1. `SELECT * FROM Schueler;`
2. `SELECT Name FROM Schueler;`
3. `SELECT * FROM Schueler
WHERE Klasse = '10A';`

16. (3 Punkte) Gegeben sind die folgenden Tabellen:

Lehrer

LehrerID	Name
1	Herr Müller
2	Frau Schmidt

Kurs

KursID	Bezeichnung	LehrerID
10	Informatik	1
11	Mathematik	2
12	Physik	1

Formuliere eine SQL-Abfrage, die die Namen der Lehrkräfte und die von ihnen unterrichteten Kurse ausgibt.

Lösung:

```
SELECT Lehrer.Name, Kurs.Bezeichnung
FROM Lehrer, Kurs
WHERE Lehrer.LehrerID = Kurs.LehrerID;
```

17. (3 Punkte) Gegeben ist die folgende Tabelle:

Schueler

SchuelerID	Name	Klasse
1	Anna Meier	10A
2	Max Schulz	10B

Formuliere eine SQL-Anweisung, um die folgende Schülerin einzufügen:

- SchuelerID: 3
- Name: Lea Weber
- Klasse: 10A

Lösung: INSERT INTO Schueler
VALUES (3, 'Lea Weber', '10A');

18. (3 Punkte) Gegeben ist die folgende Tabelle:

Produkt

ProduktID	Produktname	Preis
1	Maus	20
2	Tastatur	50

Der Preis der Tastatur soll auf 55 geändert werden.

Formuliere die passende SQL-Anweisung.

Lösung: UPDATE Produkt
SET Preis = 55
WHERE ProduktID = 2;

19. (3 Punkte) Gegeben ist die folgende Tabelle:

Kunde

KundenID	Name
1	Anna Meier
2	Max Schulz
3	Lea Weber

Der Datensatz von Max Schulz soll aus der Datenbank entfernt werden.

Formuliere die passende SQL-Anweisung.

Lösung: DELETE FROM Kunde
WHERE KundenID = 2;

20. (3 Punkte) Gegeben ist folgendes relationales Datenbankschema:

- Mitglied (MID, Name, Vorname, Geburtsjahr, Geschlecht, Straße, ↑PLZ)
- Ort (PLZ, Ortsname)
- Sportart(SpID, Bezeichnung)
- betreibt_Sportart (↑MID, ↑SpID)

Einige der Mitglieder sind Trainer einer oder mehrerer Sportarten und erhalten dafür eine monatliche Vergütung. Ergänze das Datenbankschema, so dass diese Informationen in der Datenbank gespeichert werden können.

Lösung:

- Trainer (TID, ↑MID, Vergütung, ↑SpID)

21. (8 Punkte) In einer Datenbank sollen Informationen zu Büchern, ihren Kapiteln und den jeweiligen Autoren gespeichert werden. Hierzu wird zunächst das folgende relationale Datenbankschema definiert:

Dabei gilt: Ein „Kapitel“ gehört genau zu einem Buch. Ein „Buch“ hat im Ausgangsmodell genau einen Autor.

- Kapitel (KID, Überschrift, ↑BID)
- Buch (BID, Titel, ↑AID, Erscheinungsjahr)
- Autor (AID, Name, Geburtsjahr)

- a) Erläutere anhand dieser Relationen die Begriffe „Primärschlüssel“ und „Fremdschlüssel“.
- b) Entwirf ein ER-Diagramm zum angegebenen Datenbankschema. Gib auch die Kardinalitäten aller vorkommenden Beziehungen an.
- c) Jemand schlägt vor, das Datenbankschema zu ändern und nur eine einzige Tabelle zu verwenden.

Kapitel(KID, Überschrift, Titel, Erscheinungsjahr, Name, Geburtsjahr)

Beurteile diesen Vorschlag.

Gib jeweils eine SQL-Anfrage an, die die folgenden Ergebnisse liefert:

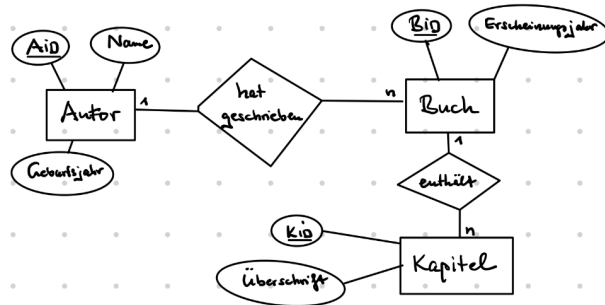
- d) Name aller Autoren, die im Jahr 1970 geboren wurden.
- e) Titel und Erscheinungsjahr aller Bücher, die vom Autor „J. K. Rowling“ veröffentlicht wurden, aufsteigend sortiert nach dem Erscheinungsjahr.

Das Datenbankschema soll so geändert werden, dass es auch möglich ist, einem Buch mehrere Autoren zuzuordnen (z. B. bei Sammelbänden).

- f) Erläutere, welche Änderungen an der Datenbank hierzu nötig sind.
- g) Gib eine zu deinen Änderungen aus Teil f) passende SQL-Abfrage an, die die Titel aller Bücher liefert, an denen der Autor „Sebastian Fitzek“ beteiligt war.

Lösung: a) Primärschlüssel: Ein Attribut oder eine Attributkombination, das bzw. die jeden Datensatz einer Relation eindeutig identifiziert. Fremdschlüssel: Ein Attribut in einer Relation, das auf den Primärschlüssel einer anderen Relation verweist, um Beziehungen zwischen Datensätzen herzustellen.

b)



c) Der Vorschlag ist nicht sinnvoll, da dadurch starke Redundanzen entstehen. Bei jedem Kapitel würden Buchdaten und Autoren Daten erneut gespeichert. Das erhöht den Speicherbedarf und führt leichter zu Inkonsistenzen bei Änderungen.

d)

```

SELECT Autor.Name
  FROM Autor
 WHERE Geburtsjahr = 1970
  
```

e)

```

SELECT Buch.Titel, Erscheinungsjahr
  FROM Buch, Autor
 WHERE Buch.AID = Autor.AID
       AND Autor.Name = 'J. K. Rowling'
 ORDER BY Erscheinungsjahr ASC
  
```

f) Durch die Änderung entsteht eine n:m-Beziehung zwischen Buch und Autor. Dazu wird der Fremdschlüssel AID aus der Tabelle Buch entfernt und eine zusätzliche Beziehungstabelle eingeführt:

Buch_Autor(↑BID, ↑AID)

g)

```

SELECT Buch.Titel
  FROM Buch, Buch_Autor, Autor
 WHERE Buch.BID = Buch_Autor.BID
       AND Buch_Autor.AID = Autor.AID
       AND Autor.Name = 'Sebastian Fitzek'
  
```