

Aufgabe: Datenbankmodellierung – Fitnessstudio

Aufgabe 2: (Datenbank, Datenmodellierung und Normalisierung)

Im Verwaltungssystem eines Fitnessstudios werden Mitglieder- und Trainerdaten in einer relationalen Datenbank gespeichert. Die ursprüngliche Tabellenstruktur sieht wie folgt aus:

Ursprünglich nicht normalisierte Tabelle

Mitgl_ID	Vorname	Nachname	Adresse (Str., Stadt, PLZ)	Vertrag	Trainer_ID	Name_Trainer	Tel_Trainer
1	Anna	Müller	Marktstr. 10, Berlin, 10115	Premium	T01	Lara König	0301234567
2	Tim	Weber	Allee 4, Hamburg, 20095	Basis	T02	Max Reiter	0408765432
3	Anna	Müller	Marktstr. 10, Berlin, 10115	Premium	T02	Max Reiter	0408765432
4	Lisa	Schmitt	Parkweg 2, München, 80331	Standard	T03	Julia Haas	0893344556

Teilaufgaben

- Untersuchen Sie Redundanzen und Anomalien in der Tabelle.
(10 BE)
- Analysieren Sie die nicht normalisierte Tabelle und normalisieren Sie sie bis zur 3. Normalform. Geben Sie die neu entstandenen Tabellen an (ohne Daten).
(25 BE)
- Erläutern Sie die konkreten Vorteile, die die Normalisierung in diesem Fall für das Fitnessstudio bringt.
(5 BE)

Lösung Teil a) – Redundanzen und Anomalien

Die vorliegende Tabelle enthält mehrere **Redundanzen** und ist **nicht normalisiert**, was zu **Anomalien** führen kann. Eine detaillierte Analyse:

1. Redundanzen (Datenwiederholungen)

- Die Daten von **Mitglied Anna Müller** (ID 1 und 3) erscheinen mehrfach – gleiche Adresse, gleicher Vertrag.
- Die Informationen über **Trainer Max Reiter** (T02) erscheinen mehrfach (bei Mitglied Tim Weber und Anna Müller).

2. Anomalien

- **Einfügeanomalie:** Ein neuer Trainer kann nicht gespeichert werden, wenn er noch kein Mitglied betreut. Beispiel: Ein neuer Trainer „Nina Braun“ kann nicht erfasst werden, ohne dass ihr ein Mitglied zugeordnet wird.
- **Löschanomalie:** Wenn das letzte Mitglied, das einem Trainer zugeordnet ist, gelöscht wird, gehen auch die Trainerdaten verloren. Beispiel: Wenn Lisa Schmitt gelöscht wird, verschwindet auch Trainerin Julia Haas aus dem System.
- **Änderungsanomalie:** Wenn sich die Telefonnummer eines Trainers ändert, muss diese an mehreren Stellen manuell aktualisiert werden. Wird z.B. die Nummer von Max Reiter geändert, muss sie bei jedem Mitglied, das ihn als Trainer hat, separat geändert werden.

3. Verstoß gegen die Normalformen

- Die Tabelle ist **nicht in 1. NF**, da die Adresse als zusammengesetztes Attribut gespeichert ist („Marktstr. 10, Berlin, 10115“).
- Sie ist auch **nicht in 2. NF**, da z.B. die Trainerdaten funktional nur von **TrainerID** abhängen, nicht vom vollständigen Primärschlüssel **MitglID**.
- Die **3NF** ist ebenfalls verletzt, da es transitive Abhängigkeiten zwischen Trainerdaten und Mitgliederdaten gibt.

Verstoß gegen die 2. Normalform (2.NF): Die 2.NF verlangt: Die Tabelle muss in 1.NF sein und jedes Nichtschlüsselattribut muss voll funktional abhängig vom Primärschlüssel sein. Beispielproblem: Die Trainerdaten (NameTrainer, TelTrainer) sind nicht vom Primärschlüssel **MitglID** abhängig, sondern allein von der **TrainerID**. Das bedeutet: Mehrere Mitglieder können denselben Trainer haben, aber die Trainerinformationen wiederholen sich – das ist eine partielle Abhängigkeit. Verstoß, da die 2.NF partielle Abhängigkeiten ausschließt. Lösung: Auslagerung der Trainerinformationen in eine eigene Tabelle.

Verstoß gegen die 3. Normalform (3NF): Die 3NF verlangt: Keine transitiven Abhängigkeiten – also Nichtschlüsselattribute dürfen nicht voneinander abhängig sein. Beispielproblem: Wenn **TrainerID** den **NameTrainer** bestimmt, und **NameTrainer** wiederum **TelTrainer**, dann liegt eine transitive Abhängigkeit vor. Diese Verbindungen zwischen Nichtschlüsselattributen müssen durch weitere Tabellen aufgelöst werden. Verstoß, da z.B. **TelTrainer** nicht direkt von **MitglID** abhängt, sondern indirekt über **TrainerID**.

Lösung Teil b – Normalisierte Tabellen (bis 3. Normalform)

Tabelle: Mitglied

MitglID	Vorname	Nachname
---------	---------	----------

Tabelle: Adresse

AdresseID	Straße	Stadt	PLZ
-----------	--------	-------	-----

Tabelle: MitgliedAdresse

MitglID (FK)	AdresseID (FK)
--------------	----------------

Tabelle: Vertrag

VertragID	Vertragsname
-----------	--------------

Tabelle: MitgliedVertrag

MitglID (FK)	VertragID (FK)
--------------	----------------

Tabelle: Trainer

TrainerID	Name_Trainer	Tel_Trainer
-----------	--------------	-------------

Tabelle: MitgliedTrainer

MitglID (FK)	TrainerID (FK)
--------------	----------------

Lösung Teil c – Vorteile der Normalisierung für das Fitnessstudio

Die Normalisierung bringt dem Fitnessstudio erhebliche Vorteile, sowohl im täglichen Betrieb als auch langfristig bei der Wartung und Erweiterung des Datenbanksystems:

1. Vermeidung von Redundanzen:

Daten wie die Adresse eines Mitglieds oder die Telefonnummer eines Trainers werden nicht mehrfach gespeichert. Das reduziert den Speicherbedarf und vermeidet unnötige Wiederholungen.

2. Schutz vor Anomalien:

Durch die Zerlegung in konsistente Tabellen werden Einfüge-, Änderungs- und Löschanomalien vermieden. Beispielsweise kann ein neuer Trainer angelegt werden, ohne dass sofort ein Mitglied zugeordnet werden muss.

3. Datenkonsistenz:

Da Informationen nur an einer Stelle gepflegt werden, ist die Wahrscheinlichkeit von Widersprüchen geringer. Ändert sich z. B. die Telefonnummer eines Trainers, muss sie nur einmal geändert werden – und ist danach überall korrekt.

4. Erleichterte Wartung:

Durch die klare Struktur und Aufteilung ist die Datenbank leichter zu verstehen und zu pflegen. Änderungen an einer Tabelle haben keinen Einfluss auf andere Tabellen, solange die Beziehungen erhalten bleiben.

5. Skalierbarkeit und Erweiterbarkeit:

Neue Anforderungen wie die Erfassung zusätzlicher Trainingsangebote oder Vertragsarten lassen sich einfacher umsetzen, da das Modell flexibel und modular aufgebaut ist.

6. Bessere Performance bei Abfragen:

Normalisierte Tabellen enthalten weniger redundante Daten, wodurch Abfragen effizienter verarbeitet werden können – insbesondere bei großen Datenmengen.

7. Verbesserte Datenintegrität:

Durch den Einsatz von Primär- und Fremdschlüsseln wird sichergestellt, dass Beziehungen zwischen Tabellen korrekt eingehalten werden. Dadurch steigt die Qualität der gespeicherten Informationen.

Fazit:

Die Normalisierung sorgt für ein robustes, konsistentes und zukunftssicheres Datenbanksystem, das den Verwaltungsaufwand reduziert und eine verlässliche Datenbasis für das Fitnessstudio schafft.