Beziehungen

in relationalen Datenbanken

Kardinalitäten

Die Kardinalitäten in relationalen Datenbanken beschreiben die Beziehung zwischen Tabellen und wie viele Datensätze in einer Tabelle mit Datensätzen in einer anderen Tabelle verknüpft sind. Es gibt verschiedene Typen von Kardinalitäten, die in Beziehungen auftreten können.

Hier sind die wichtigsten Kardinalitäten vorgestellt.

One-to-One (1:1)

Bei einer One-to-One-Beziehung entspricht ein Datensatz in der ersten Tabelle genau einem Datensatz in der zweiten Tabelle und umgekehrt.

Beispiel: Eine Tabelle für Mitarbeiter kann eine One-to-One-Beziehung zu einer Tabelle für Mitarbeiterausweise haben.

Mitarbeiter	
id	Name
1	Bob
2	Alice

Mitarbeiter-Ausweis	
Mitarbeiter-ID	Ausweisnummer
1	A343023430
2	A349203439

SQL Beispiel

```
CREATE TABLE mitarbeiter (
        id INT PRIMARY KEY AUTO_INCREMENT,
        Name VARCHAR(200)
);

CREATE TABLE mitarbeiter_ausweise (
        id INT PRIMARY KEY AUTO_INCREMENT,
        mitarbeiter_id INT UNIQUE,
        ausweisnummer VARCHAR(200),
        FOREIGN KEY (mitarbeiter_id) REFERENCES mitarbeiter(id)

);
```

One-to-Many (1:N)

Bei einer One-to-Many-Beziehung entspricht ein Datensatz in der ersten Tabelle einem oder mehreren Datensätzen in der zweiten Tabelle, während ein Datensatz in der zweiten Tabelle nur einem Datensatz in der ersten Tabelle entspricht.

Eine Tabelle Mitarbeiter kann eine One-to-Many-Beziehung zu einer Tabelle Tasks haben, da jeder Mitarbeiter mehrere Tasks haben kann. Jeder Taks hingegen wird genau von einem Mitarbeiter ausgeführt.

One-to-Many (1:N)

Mitarbeiter	
id	Name
1	Bob
2	Alice

Task		
Task ID	Mitarbeiter ID	Beschreibung
1	1	Hof kehren
2	1	Seminar Sicherheit in der IT
3	1	Daten sichern
4	2	Akten aufräumen

SQL Beispiel

```
CREATE TABLE mitarbeiter (
            id INT PRIMARY KEY AUTO_INCREMENT,
            Name VARCHAR(200)
);

CREATE TABLE tasks (
            task_id INT,
            mitarbeiter_id INT,
            beschreibung TEXT
            PRIMARY KEY (task_id),
            FOREIGN KEY (mitarbeiter_id) REFERENCES mitarbeiter(id)

);
```

Many-to-Many (M:N)

Bei einer Many-to-Many-Beziehung entspricht ein Datensatz in der ersten Tabelle mehreren Datensätzen in der zweiten Tabelle, und umgekehrt. Dies wird normalerweise durch eine Verknüpfungstabelle (Junction-Tabelle) realisiert, die die Verbindung zwischen den beiden Tabellen herstellt.

Ein Mitarbeiter kann Mitglied in vielen Teams sein, und ein Team besteht aus vielen Mitarbeitern.

Many-to-Many (N:M)

Mitarbeiter	
id	Name
1	Bob
2	Alice

Team	
id	Team-Name
1	DevOp
2	Data Science

Team_Mitarbeiter	
mitarbeiter_id	team_id
1	1
1	2
2	1

SQL Beispiele

```
CREATE TABLE team (
        id INT PRIMARY KEY AUTO_INCREMENT,
        team_name VARCHAR(200)
);

CREATE TABLE team_mitarbeiter (
         team_id,
        mitarbeiter_id,
        PRIMARY KEY (mitarbeiter_id, team_id),
        FOREIGN KEY (mitarbeiter_id) REFERENCES mitarbeiter(id),
        FOREIGN KEY (team_id) REFERENCES team(id)
);
```