

Entwurf einer Datenbank

Beispiel Tierpark¹

1 Anforderungsanalyse

Entitäten (mit Attributen):

Tiere (TID, Name, Art), Pfleger, Gebäude, Futter, ...

Beziehungen:

Tier lebt in Gebäude, Tier frisst Futter, Futter lagert in Gebäude, Tier wird von Pfleger versorgt.

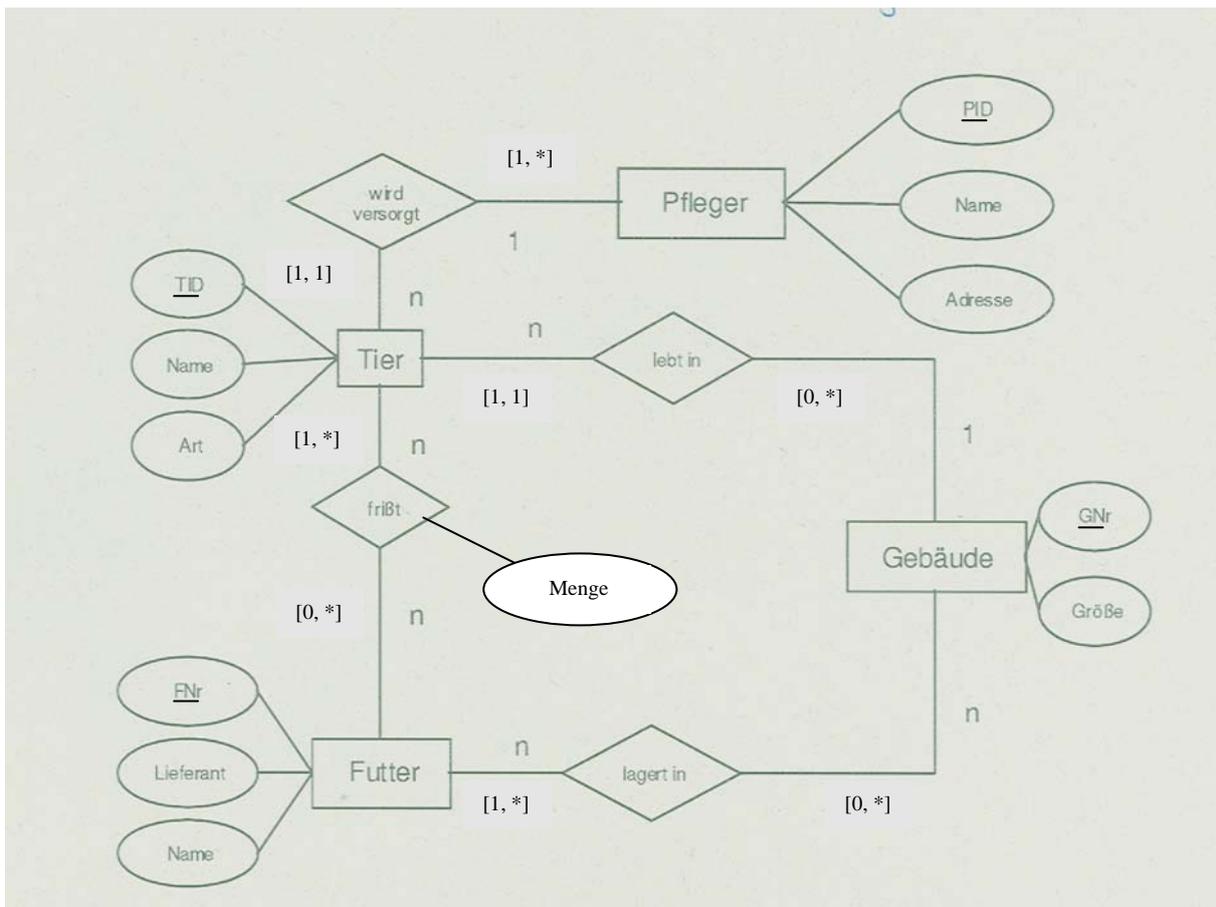
Kardinalitäten

Jedes Tier wird von genau einem Pfleger versorgt.

Jedes Tier wohnt in genau einem Gebäude.

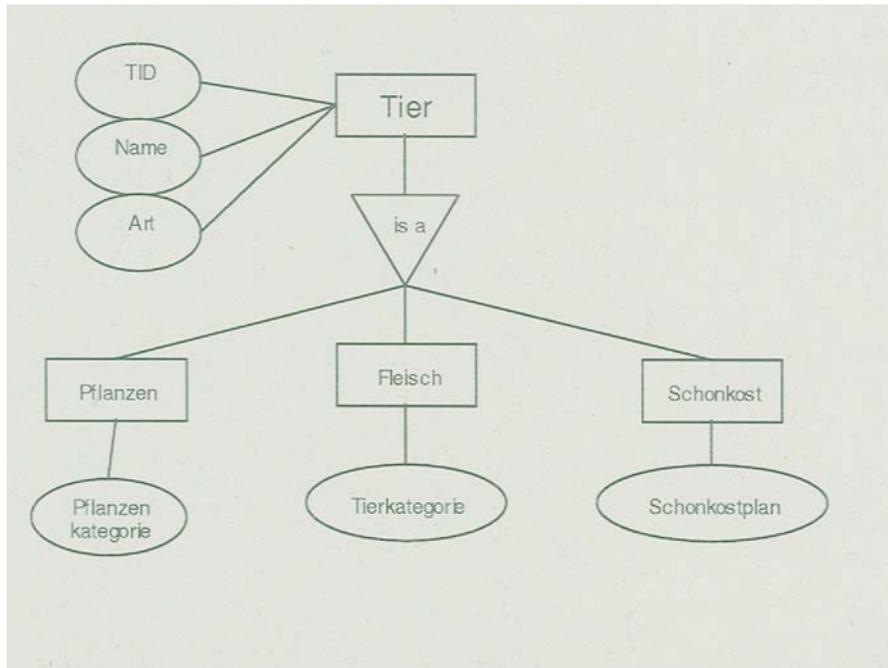
2. Modellierung

2.1 ER-Modell



¹ Nach einer Klausuraufgabe der FU Hagen

2.2 Erweitertes ER-Modell



3. ER-Modell → Relationales Schema

1. Schritt

Jeder Entity-Typ bekommt eine gleichnamige Relation mit demselben Schlüssel wie im ER-Modell:

```
Tier(TID, Name, Art)
Gebäude(GNr, Größe)
Pfleger(PID, Name, Adresse)
Futter(FNr, Lieferant, Name)
```

2. Schritt:

Jede m:n-Beziehung bekommt eine eigene Relation mit den Schlüsseln aus den beteiligten Relationen als zusammengesetzten Schlüssel:

```
frisst(TID, FNr, Menge)
lagert_in(GNr, FNr)
```

3. Schritt

1:n-Beziehungen (erhalten entweder eigene Relationen wie im 2. Schritt oder) werden durch ihre Schlüsselattribute (als Fremdschlüssel) in den entsprechenden Entity-Typen auf der Seite mit der Kardinalität [1:1] (bzw. [0, 1]) dargestellt.

```
Tier(TID, Name, Art, PID, GNr)
```

Damit ergibt sich folgendes Datenbankschema:

```
Gebäude(GNr, Größe)
Pfleger(PID, Name, Adresse)
Futter(FNr, Lieferant, Name)
lagert_in(GNr, FNr)
Tier(TID, Name, Art, PID, GNr)
frisst(TID, FNr, Menge)
```

4. Definition der Datenbank und der Tabellen (z. B.) mit der DDL von SQL

```
CREATE DATABASE IF NOT EXISTS Tierpark;
```

```
USE Tierpark;
```

```
CREATE TABLE Gebaeude (  
  GNr INTEGER NOT NULL PRIMARY KEY,  
  Groesse INTEGER NOT NULL);
```

Aufgabe: Definieren Sie die Tabellen *Pfleger* und *Futter*.

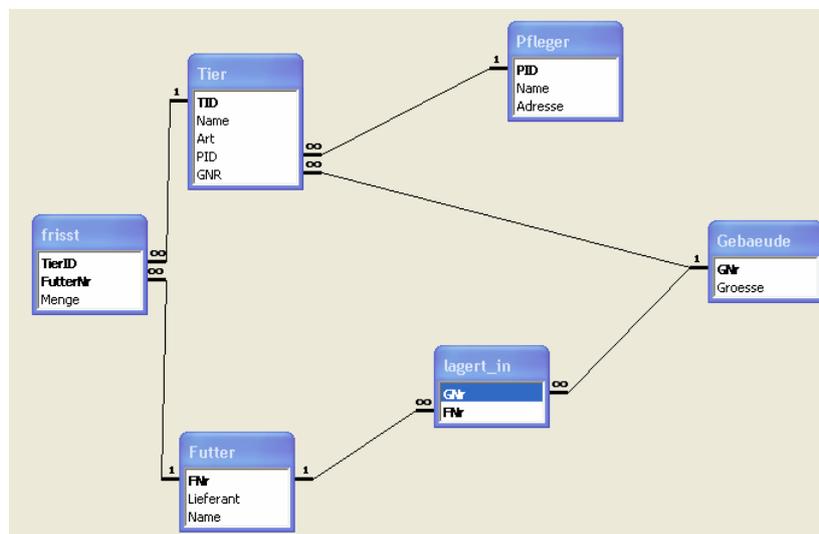
```
CREATE TABLE frisst (  
  TierID INTEGER NOT NULL,  
  FutterNr INTEGER NOT NULL,  
  Menge INTEGER,  
  PRIMARY KEY (TierID, FutterNr),  
  FOREIGN KEY (TierID) REFERENCES Tier(TID),  
  FOREIGN KEY (FutterNr) REFERENCES Futter(FNr));
```

Aufgabe: Definieren Sie die Tabellen *lagert_in* und *Tier*.

5. Erfassen der Daten (z. B. mit SQL von MS Access):

Aufgabe:

- Starten Sie *MS Access*. Legen Sie die neue Datenbank *Tierpark.mdb* an.
- Wechseln Sie wie folgt in die SQL-Ansicht:
 - Entwurfsansicht einer Abfrage öffnen
 - Fenster *Tabelle anzeigen* schließen
 - Symbol *SQL* anklicken
- Geben Sie die SQL-Anweisung zur Definition der Tabelle *Gebaeude* ein und führen Sie die Anweisung aus (auf das Ausrufezeichen klicken).
- Definieren Sie alle weiteren Tabellen mittels SQL-Anweisungen. Beachten Sie dabei die richtige Reihenfolge (z. B. muss die Tabelle *Tier* vor der Tabelle *frisst* definiert werden).
- Wechseln Sie in das Menü *Extras / Beziehungen* und sehen Sie sich das Ergebnis Ihrer Arbeit an.



6. Datenmanipulation (z. B.) mit der DML von SQL

```
INSERT INTO Gebaeude VALUES(1, 50);
```

```
INSERT INTO Gebaeude VALUES(2, 20);
```

usw.

```
INSERT INTO Pfleger VALUES(2, "Schulze", NULL);
```

usw.

```
INSERT INTO Futter VALUES (3, "Großhandel", "Hafer");
```

usw.

Aufgabe:

- Geben Sie in jede Tabelle mindestens 3 Datensätze mittels SQL-Anweisungen ein.
- Testen Sie die referenzielle Integrität, indem Sie versuchen, ein und demselben Tier zwei verschiedene Pfleger zuzuweisen.

Aktualisierung von Daten:

```
UPDATE Pfleger
```

```
SET Adresse = "Steinstr. 1" WHERE Name = "Schulze";
```

Löschen von Daten:

```
DELETE FROM Futter WHERE Name = "Hafer";
```

Abfragen formulieren:

Aufgabe:

Was bewirken folgende Anweisungen? Formulieren Sie ähnliche Anweisungen, die zu Ihrer Datenbank passen. Ergänzen Sie gegebenenfalls Daten in Ihren Tabellen.

```
SELECT Lieferant FROM Futter
```

```
WHERE Name = "Bananen";
```

```
SELECT SUM(Menge)AS Summe
```

```
FROM frisst
```

```
WHERE FutterNr = 2;
```

```
SELECT Pfleger.Name, Pfleger.Adresse, Tier.Name
```

```
FROM Pfleger, Tier
```

```
WHERE Tier.Art LIKE "*Rab*" AND Tier.PID = Pfleger.PID;
```

```
SELECT *
```

```
FROM Pfleger, Tier
```

```
ORDER BY Pfleger.PID;
```

```
SELECT *
```

```
FROM Pfleger
```

```
WHERE PID IN(SELECT PID from Tier);
```