

ddi: Mini-Challenge LE2 – Relationale Datenbanken (SQL)

Lukas Reber

Aufgabenstellung:

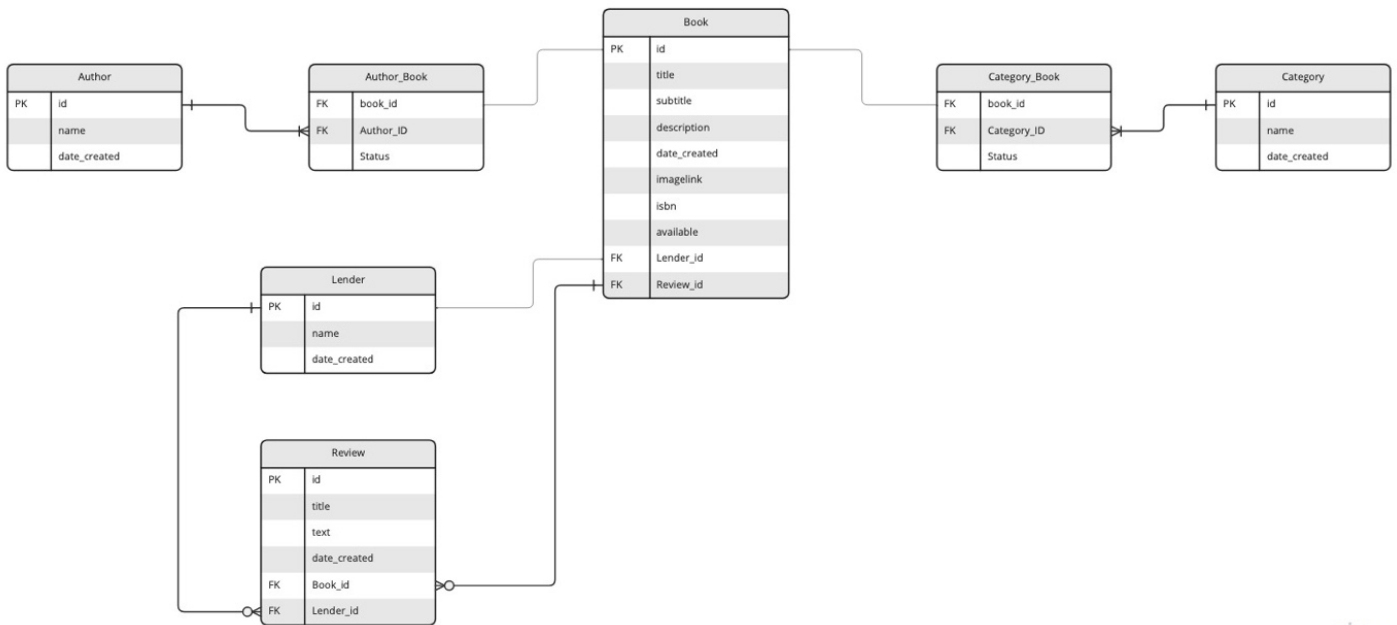
Erstelle ein ERD und ein Relationales Modell für die folgende Applikation:

Eine Bibliotheksapplikation für die Verwaltung von Büchern. Es soll damit möglich sein, Bücher zu erfassen, auszuleihen und für entsprechende Bücher Rezensionen zu schreiben.

- Jedes Buch verfügt über einen Titel, einen Untertitel, eine Beschreibung, ein Link zum Cover, einem oder mehreren Autoren, einer oder mehreren Kategorien, die ISBN Nummer, dem Status (Verfügbar: True/False) und dem Datum, wann das Buch in die Bibliothek aufgenommen wurde.
- Jede Kategorie kann ein oder mehrere Bücher enthalten.
- Jeder Autor kann ein oder mehrere Bücher geschrieben haben.
- Ein Buch kann durch eine Person ausgeliehen werden.
- Jede Person kann mehrere Bücher ausleihen.
- Von den Personen welche Bücher ausleihen, den Autoren/Autorinnen sowie der Buchkategorie ist einzig der Name sowie das Datum erfasst, wann die jeweilige Entität erstellt wurde.
- Für jedes Buch können Rezensionen geschrieben werden.
- Ein Buch kann mehrere Rezensionen erhalten.
- Jede Rezension ist von einer Person erfasst.
- Die Rezension beinhaltet einen Titel, den Text sowie das Datum, wann die Rezension geschrieben wurde.

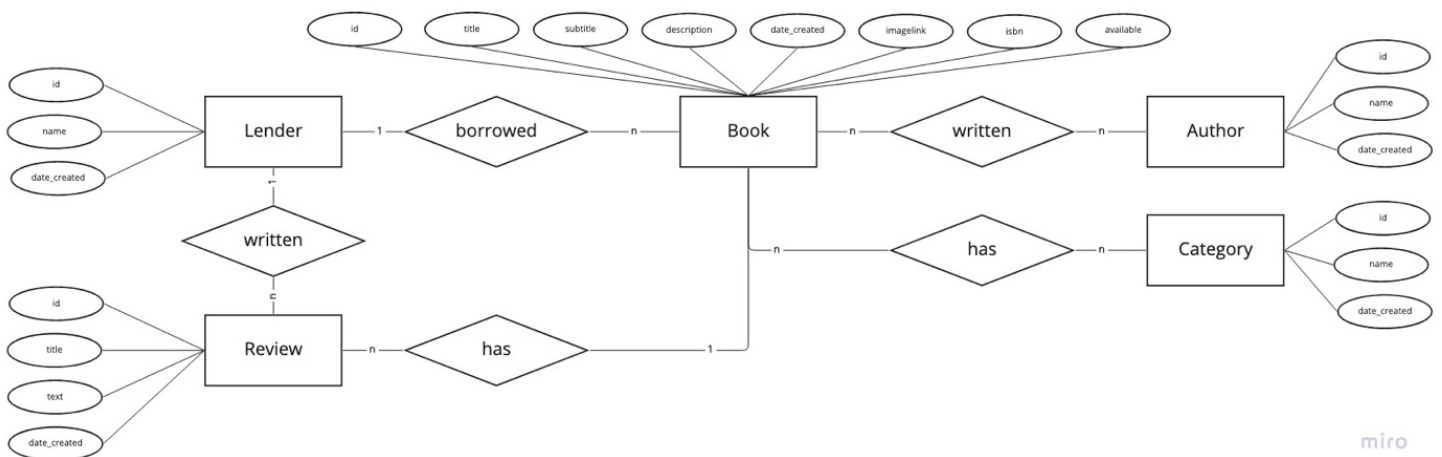
Musterlösung:

Relationales Modell



miro

ERD Modell



miro

SQL Script (Postgres)

```
CREATE TABLE "Author" (  
    "id" SERIAL,  
    "name" varchar(200) NOT NULL,  
    "date_created" TIMESTAMP NOT NULL,  
    PRIMARY KEY ("id")  
);
```

```
CREATE TABLE "Book" (  
    "id" SERIAL,  
    "title" varchar(200),  
    "subtitle" varchar(200),  
    "description" text,  
    "date_created" TIMESTAMP NOT NULL,  
    "imagelink" varchar(200),  
    "isbn" int8,  
    "available" bool,  
    PRIMARY KEY ("id")  
);
```

```
CREATE TABLE "Category" (  
    "id" SERIAL,  
    "name" varchar(200) NOT NULL,  
    "date_created" TIMESTAMP NOT NULL,  
    PRIMARY KEY ("id")  
);
```

```
CREATE TABLE "Lender" (  
    "id" SERIAL,  
    "name" varchar(200) NOT NULL,  
    "date_created" TIMESTAMP NOT NULL,  
    PRIMARY KEY ("id")  
);
```

```
CREATE TABLE "Book_Author" (  
    "id" SERIAL,  
    "book_id" int8 NOT NULL,  
    "author_id" int8 NOT NULL,  
    PRIMARY KEY ("id"),  
    CONSTRAINT fk_book  
        FOREIGN KEY ("book_id")  
        REFERENCES "Book"("id")  
        ON DELETE SET NULL,  
    CONSTRAINT fk_author  
        FOREIGN KEY ("author_id")  
        REFERENCES "Author"("id")  
        ON DELETE SET NULL  
);
```

```
CREATE TABLE "Book_Category" (  
  "id" SERIAL,  
  "book_id" int8 NOT NULL,  
  "category_id" int8 NOT NULL,  
  PRIMARY KEY ("id"),  
  CONSTRAINT fk_book  
    FOREIGN KEY ("book_id")  
    REFERENCES "Book"("id")  
    ON DELETE SET NULL,  
  CONSTRAINT fk_category  
    FOREIGN KEY ("category_id")  
    REFERENCES "Category"("id")  
    ON DELETE SET NULL  
);
```

```
CREATE TABLE "Review" (  
  "id" SERIAL,  
  "title" varchar(200),  
  "text" text,  
  "date_created" timestamptz NOT NULL,  
  "book_id" int8,  
  "lender_id" int8,  
  PRIMARY KEY ("id"),  
  CONSTRAINT fk_book  
    FOREIGN KEY ("book_id")  
    REFERENCES "Book"("id")  
    ON DELETE SET NULL,  
  CONSTRAINT fk_lender  
    FOREIGN KEY ("lender_id")  
    REFERENCES "Lender"("id")  
    ON DELETE SET NULL  
);
```