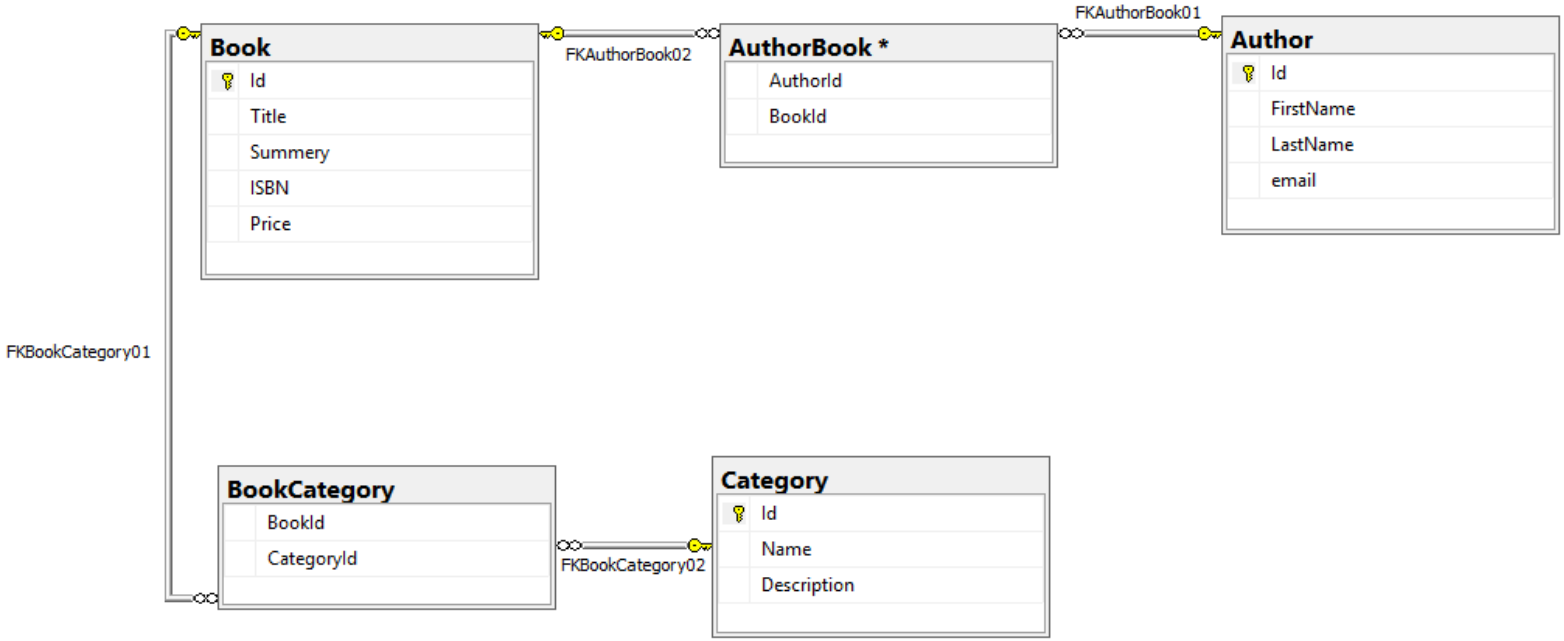
FanzyBooks database

Database

Database tabeller:

FanzyBooks databasen består af 3 tabeller til data: **Author**, **Book** og **Category**.
Der ud over er der 2 link tabeller/junktions tables/foreign key tables.
De anvendes til at håndtere mange-til-mange relationerne mellem forfatter & bog og mellem bog & kategori.
Tabellen **AuthorBook** linker forfatter-id til bog-id.
Tabellen **BookCategory** linker bog-id til kategori-id.

Tabel diagram



https://github.com/simonpeterrasmussen/Database2021/blob/master/Uge03/FanzyBooks.png

Overvejelser

- Det er ikke nok at lave en fremmednøgle i tabellen Book til Author, eller i Book til Category, fordi relationerne er mange-til-mange.
- ISBN er unik og kunne anvendes som primær nøgle i tabellen Book, men dette er ikke gjort, fordi det kunne være at en bogs ISBN skulle ændres. Dette vil så medføre en kaskade opdatering alle de steder bogen er refereret.
- Ved *kun* at lade ISBN være unique kan det ændres.

SQL kode

Der er 4 kodeblokke:

1. Oprettelse af database og tabeller.
2. Oprettelse af Stored Procedures.
3. Indsættelse af data i tabellerne.
4. Udtræk af data.

```sql
-- Create the new database 'FanzyBooks' if it does not exist already and create tables.
IF NOT EXISTS (
    SELECT [name]
        FROM sys.databases
        WHERE [name] = N'FanzyBooks'
)
CREATE DATABASE FanzyBooks
GO

USE FanzyBooks;
GO

-- References must be deleted first if they exist:
IF OBJECT_ID('[dbo].[BookCategory]', 'U') IS NOT NULL
DROP TABLE [dbo].[BookCategory]
GO
IF OBJECT_ID('[dbo].[AuthorBook]', 'U') IS NOT NULL
DROP TABLE [dbo].[AuthorBook]
GO

IF OBJECT_ID('[dbo].[Author]', 'U') IS NOT NULL
DROP TABLE [dbo].[Author]
GO
CREATE TABLE [dbo].[Author]
(
    [Id] INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [FirstName] NVARCHAR(50) NOT NULL,
    [LastName] NVARCHAR(50) NOT NULL,
    [email] NVARCHAR(50) UNIQUE NOT NULL
);
GO

IF OBJECT_ID('[dbo].[Book]', 'U') IS NOT NULL
DROP TABLE [dbo].[Book]
GO
CREATE TABLE [dbo].[Book]
(
    [Id] INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [Title] NVARCHAR(100) NOT NULL,
    [Summery] NVARCHAR(255) NOT NULL,
```

```
    [ISBN] NVARCHAR(13) UNIQUE NOT NULL,
    [Price] SMALLMONEY NOT NULL
);
GO

IF OBJECT_ID('[dbo].[Category]', 'U') IS NOT NULL
DROP TABLE [dbo].[Category]
GO
CREATE TABLE [dbo].[Category]
(
    [Id] INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    [Name] NVARCHAR(50) NOT NULL,
    [Description] NVARCHAR(255) NULL
);
GO

IF OBJECT_ID('[dbo].[AuthorBook]', 'U') IS NOT NULL
DROP TABLE [dbo].[AuthorBook]
GO
CREATE TABLE [dbo].[AuthorBook]
(
    [AuthorId] INT NOT NULL,
    [BookId] INT NOT NULL,
    CONSTRAINT [UniqueAuthorIdBookId] UNIQUE ([AuthorID], [BookId]),
    CONSTRAINT [FKAuthorBook01] FOREIGN KEY ([AuthorId]) REFERENCES [Author]([Id]),
    CONSTRAINT [FKAuthorBook02] FOREIGN KEY ([BookId]) REFERENCES [Book]([Id])
);
GO

IF OBJECT_ID('[dbo].[BookCategory]', 'U') IS NOT NULL
DROP TABLE [dbo].[BookCategory]
GO
CREATE TABLE [dbo].[BookCategory]
(
    [BookId] INT NOT NULL,
    [CategoryId] INT NOT NULL,
    CONSTRAINT [UniqueBookIdCategoryID] UNIQUE ([BookId], [CategoryId]),
    CONSTRAINT [FKBookCategory01] FOREIGN KEY([BookId]) REFERENCES [Book]([Id]),
    CONSTRAINT [FKBookCategory02] FOREIGN KEY ([CategoryId]) REFERENCES [Category]([ID])
);
```

GO

```sql
-- Stored Procedures used with FanzyBooks.
USE FanzyBooks;
GO

IF EXISTS   (SELECT *
                FROM INFORMATION_SCHEMA.ROUTINES
                WHERE SPECIFIC_SCHEMA = N'dbo'
                    AND SPECIFIC_NAME = N'CreateCategory'
                    AND ROUTINE_TYPE = N'PROCEDURE'
            )
    DROP PROCEDURE dbo.CreateCategory
GO
CREATE or ALTER PROCEDURE [dbo].[CreateCategory]
    @CategoryName NVARCHAR(50) = 'New Cat',
    @Description NVARCHAR(50) = 'New Description'
AS
BEGIN
    INSERT INTO [dbo].[Category]
    ( [Name], [Description] )
    VALUES
    (
    @CategoryName, @Description
    )
END
GO


CREATE or ALTER PROCEDURE [dbo].[CreateAuthor]
    @AuthorFirstName NVARCHAR(50) = 'New first',
    @AuthorLastName NVARCHAR(50) = 'New last',
    @email NVARCHAR(50) = 'New email'
AS
BEGIN
    INSERT INTO [dbo].[Author]
    ( [FirstName], [LastName], [email] )
    VALUES
    ( @AuthorFirstName, @AuthorLastName, @email )
END;
GO
```

```sql
CREATE or ALTER PROCEDURE [dbo].[CreateBook]
    @Title NVARCHAR(50) = 'New title',
    @Summery NVARCHAR(50) = 'New summary',
    @ISBN NVARCHAR(13) = 'New ISBN',
    @Price SMALLMONEY = '0.00'
AS
BEGIN
    INSERT INTO [dbo].[Book]
    ( [Title], [Summery], [ISBN], [Price] )
    VALUES
    ( @Title, @Summery, @ISBN, @Price)
END;
GO


CREATE or ALTER PROCEDURE [dbo].[GetBooks]
-- Bøger der matcher parametrene:
    @Title NVARCHAR(50) = '',
    @Catagory NVARCHAR(50) = '',
    @AuthorFirstName NVARCHAR(50) = '',
    @AuthorLastName NVARCHAR(50) = '',
    @ISBN NVARCHAR(13) = '',
    @Summary NVARCHAR(255) = '',
    @Price NVARCHAR(50) = ''
AS
BEGIN
    SELECT DISTINCT b.[Title] AS 'Title', b.[ISBN] AS 'ISBN', CAST(b.[Price] AS nvarchar) AS 'Price',
        STUFF ((SELECT '; ' + [FirstName] + ' ' + [LastName]
                    FROM [dbo].[Author] AS aNames
                    JOIN [dbo].[AuthorBook] as abNames on abNames.[AuthorId] = aNames.[Id]
                    JOIN [dbo].[Book] as bNames on bNames.[Id] = abNames.[BookId]
                    WHERE bNames.[Id] = b.[Id]
                    ORDER BY aNames.LastName, aNames.FirstName
                    FOR XML PATH ('')
                ) , 1, 1, '') AS 'Author(s)',

        STUFF ((SELECT '; ' + [Name]
                    FROM [dbo].[Category] AS cCat
                    JOIN [dbo].[BookCategory] as bcCat on bcCat.[CategoryId] = cCat.[Id]
```

```
                    JOIN [dbo].[Book] as bCat on bCat.[Id] = bcCat.[BookId]
                    WHERE bCat.[Id] = b.[Id]
                    ORDER BY cCat.[Name]
                    FOR XML PATH ('')
                ) , 1, 1, '') AS 'Category(ies)',
        b.[Summery] AS 'Summery'
        FROM [dbo].[Book] as b
        -- Author data:
        JOIN [AuthorBook] AS ab ON ab.[BookId] = b.[Id]
        JOIN [Author] AS a ON a.[Id] = ab.[AuthorId]
        -- Category data:
        JOIN [dbo].[BookCategory] as bc on bc.[BookId] = b.[Id]
        JOIN [dbo].[Category] AS c ON c.[Id] = bc.[CategoryId]

        WHERE b.[Title] LIKE ('%' + @Title + '%')
            AND b.[ISBN] LIKE  ('%' + @ISBN + '%')
            AND b.[Summery] LIKE ('%' + @Summary + '%')
            AND a.[FirstName] LIKE ('%' + @AuthorFirstName + '%')
            AND a.[LastName] LIKE ('%' + @AuthorLastName + '%')
            AND c.[Name] LIKE ('%' + @Catagory + '%' )
            AND b.[Price] LIKE ( '%' + @Price + '%')
        ORDER BY [Title]
END;
GO


CREATE or ALTER PROCEDURE [dbo].[GetAllBooks]
-- Alle bøger med info:
AS
BEGIN
    SELECT DISTINCT b.[Title] AS 'Title', b.[ISBN] AS 'ISBN', CAST(b.[Price] AS nvarchar) AS 'Price',
        STUFF ((SELECT '; ' + [FirstName] + ' ' + [LastName]
                    FROM [dbo].[Author] AS aNames
                    JOIN [dbo].[AuthorBook] as abNames on abNames.[AuthorId] = aNames.[Id]
                    JOIN [dbo].[Book] as bNames on bNames.[Id] = abNames.[BookId]
                    WHERE bNames.[Id] = b.[Id]
                    ORDER BY aNames.LastName, aNames.FirstName
                    FOR XML PATH ('')
                ) , 1, 1, '') AS 'Author(s)',
```

```
        STUFF ((SELECT '; ' + [Name]
                    FROM [dbo].[Category] AS cCat
                    JOIN [dbo].[BookCategory] as bcCat on bcCat.[CategoryId] = cCat.[Id]
                    JOIN [dbo].[Book] as bCat on bCat.[Id] = bcCat.[BookId]
                    WHERE bCat.[Id] = b.[Id]
                    ORDER BY cCat.[Name]
                    FOR XML PATH ('')
                ) , 1, 1, '') AS 'Category(ies)'
        FROM [dbo].[Book] as b
        -- Author data:
        JOIN [AuthorBook] AS ab ON ab.[BookId] = b.[Id]
        JOIN [Author] AS a ON a.[Id] = ab.[AuthorId]
        -- Category data:
        JOIN [dbo].[BookCategory] as bc on bc.[BookId] = b.[Id]
        JOIN [dbo].[Category] AS c ON c.[Id] = bc.[CategoryId]
        ORDER BY [Title]
END;
GO
```

```
-- Insert default data into the database FanzyBooks.
USE FanzyBooks;
GO

-- Create Category values:
EXECUTE dbo.CreateCategory 'Kogebøger', 'Bøger der indeholder opskrifter.'
EXECUTE dbo.CreateCategory 'Krimi', 'Kriminalhistorier'
EXECUTE dbo.CreateCategory 'SciFi', 'Science Fiction'
EXECUTE dbo.CreateCategory 'Gyser', 'Historier der giver et gys.'
EXECUTE dbo.CreateCategory 'Hobby', 'Bøger om hobbies'
EXECUTE dbo.CreateCategory 'Biler', 'Bilbøger'
EXECUTE dbo.CreateCategory 'Årbog', 'Årbøger'
EXECUTE dbo.CreateCategory 'XML', 'Bøger XML formatet.'
EXECUTE dbo.CreateCategory 'Programmering', 'Bøger der beskriver programmering.'
EXECUTE dbo.CreateCategory 'SQL', 'Bøger der omhandler SQL programmering.'
EXECUTE dbo.CreateCategory 'JavaScript', 'Bøger der omhandler JavaScript programmering.';
GO

-- Create Authors:
EXECUTE [dbo].[CreateAuthor] 'Kathi', 'Kellenberg', 'kk@apress.com';
EXECUTE [dbo].[CreateAuthor] 'Lee', 'Everest', 'le@apress.com';
EXECUTE [dbo].[CreateAuthor] 'Mike', 'McGrath', 'mm@ineasysteps.com';
EXECUTE [dbo].[CreateAuthor] 'Thomas', 'Connolly', 'tc@pearson.com';
EXECUTE [dbo].[CreateAuthor] 'Carolyn', 'Begg', 'cb@pearson.com';
EXECUTE [dbo].[CreateAuthor] 'Elliotte Rusty', 'Harold', 'erh@idgbooks.com';
GO

-- Create Books:
EXECUTE [dbo].[CreateBook] 'Beginning T-SQL', 'A Step-by-Step Approach', '9781484266052', 299.00
EXECUTE [dbo].[CreateBook] 'SQL in easy steps', 'SQL for web developers, programmers & students', '9781840785432', 139.00
EXECUTE [dbo].[CreateBook] 'JavaScript in easy steps', 'Create functions for the web', '9781840785702', 139.00
EXECUTE [dbo].[CreateBook] 'Database Systems', 'A Practical Approach to Design, implementation and Management',
'9781292061184', 399.95;
GO

-- Insert rows into table 'BookCategory' in schema '[dbo]'
INSERT INTO [dbo].[BookCategory]
( [BookId], [CategoryId] )
VALUES
( 1, 9 ),
```

```
( 1, 10),
( 2, 9 ),
( 2, 10),
( 3, 9 ),
( 3, 11),
( 4, 10)
GO

-- Insert rows into table 'AuthorBook' in schema '[dbo]'
INSERT INTO [dbo].[AuthorBook]
( [AuthorId], [BookId] )
VALUES
( 1, 1 ),
( 2, 1 ),
( 3, 2 ),
( 3, 3 ),
( 4, 4 ),
( 5, 4 )
GO
```

```sql
-- Query data from FanzyBooks
USE FanzyBooks;
SET NOCOUNT ON;
--GO

EXECUTE dbo.GetBooks @Title = 'Be', @Summary = 'by'  ;
EXECUTE dbo.GetBooks @AuthorFirstName = 'Lee';
EXECUTE dbo.GetBooks @AuthorLastName = 'eve';
EXECUTE dbo.GetBooks @ISBN = '02';
EXECUTE dbo.GetBooks @Catagory = 'Prog';
EXECUTE dbo.GetBooks @Catagory = 'SQL';
EXECUTE dbo.GetBooks @Price = 13;
EXECUTE dbo.GetBooks;

EXECUTE [dbo].[GetAllBooks];
GO
```