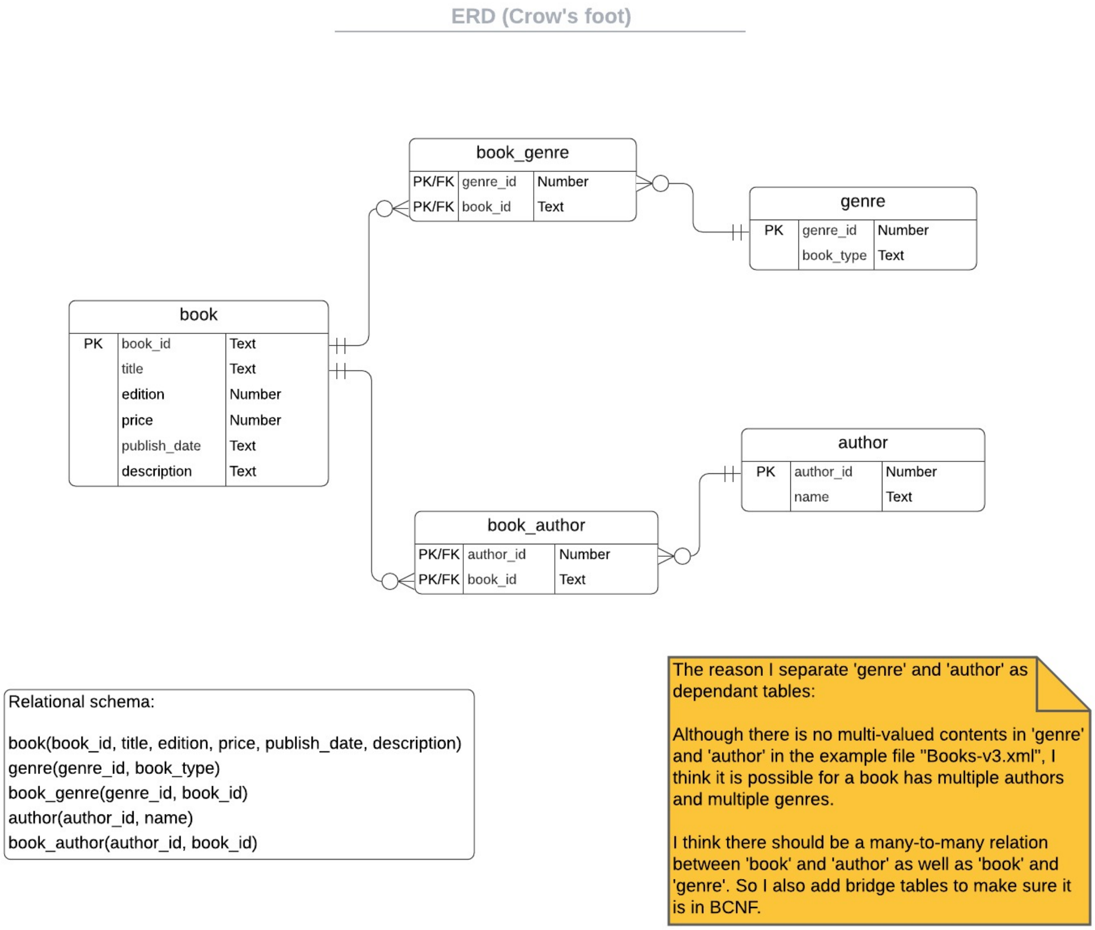
Code ▾

Assignment10

Yiman Liu

1.(25 pts) Create a normalized (BCNF) relational schema and visualize the schema in an ERD for the data in the XML file. Include the ERD in your R Notebook.



Relational schema:

book(book_id, title, edition, price, publish_date, description)
genre(genre_id, book_type)
book_genre(genre_id, book_id)
author(author_id, name)
book_author(author_id, book_id)

The reason I separate 'genre' and 'author' as dependant tables:

Although there is no multi-valued contents in 'genre' and 'author' in the example file "Books-v3.xml", I think it is possible for a book has multiple authors and multiple genres.

I think there should be a many-to-many relation between 'book' and 'author' as well as 'book' and 'genre'. So I also add bridge tables to make sure it is in BCNF.

2.(25 pts) Create a SQLite database that implement the schema, i.e., define the tables with CREATE TABLE. Use SQL chunks in your R Notebook.

Hide

```
library(RSQLite)

# connect to the SQLite database in the specified file
con <- dbConnect(SQLite(), dbname="/Users/liuyiman/database/my_db")
```

Hide

```
CREATE TABLE IF NOT EXISTS book (
  book_id TEXT PRIMARY KEY NOT NULL,
  title TEXT,
  edition INTEGER,
  price INTEGER,
  publish_date TEXT,
  description TEXT
);
```

Hide

```
CREATE TABLE IF NOT EXISTS genre (
  genre_id INTEGER PRIMARY KEY NOT NULL,
  book_type TEXT
);
```

Hide

```
CREATE TABLE IF NOT EXISTS book_genre (
  genre_id INTEGER NOT NULL,
  book_id TEXT NOT NULL,
  PRIMARY KEY (genre_id, book_id),
  FOREIGN KEY (genre_id)
    REFERENCES genre (genre_id)
        ON DELETE RESTRICT
        ON UPDATE RESTRICT,
  FOREIGN KEY (book_id)
    REFERENCES book (book_id)
        ON DELETE RESTRICT
        ON UPDATE RESTRICT
);
```

Hide

```
CREATE TABLE IF NOT EXISTS author (
  author_id INTEGER PRIMARY KEY NOT NULL,
  name TEXT
);
```

Hide

```
CREATE TABLE IF NOT EXISTS book_author (
  author_id INTEGER NOT NULL,
  book_id TEXT NOT NULL,
  PRIMARY KEY (author_id, book_id),
  FOREIGN KEY (author_id)
    REFERENCES author (author_id)
        ON DELETE RESTRICT
        ON UPDATE RESTRICT,
  FOREIGN KEY (book_id)
    REFERENCES book (book_id)
        ON DELETE RESTRICT
        ON UPDATE RESTRICT
);
```

3.(25 pts) Load the XML data from the file into R data frames; you will need to use either node-by-node traversal of the XML tree or a combination of node-by-node traversal with XPath; you likely will not be able to accomplish it with only XPath. Use surrogate keys and/or the ID attributes in the XML.

Hide

```
library(XML)
doc = xmlParse(file = "Books-v3.xml")
df <- xmlToDataFrame(doc)

library(XML)
library(methods)

# load xml file into R
data = xmlParse(file = "Books-v3.xml")
book_id_data <- xpathSApply(data, "/catalog/book/@id")
df$book_id = book_id_data

df <- df[c("book_id", "author", "title", "genre", "price", "publish_date", "description"
, "edition")]
print(df)
```

book_id	author	title	genre
<chr>	<chr>	<chr>	<chr>
bk101	Gambardella, Matthew	XML Developer's Guide	Computer
bk102	Ralls, Kim	Midnight Rain	Fantasy
bk103	Corets, Eva	Maeve Ascendant	Fantasy
bk104	Corets, Eva	Oberon's Legacy	Fantasy
bk105	Corets, Eva	The Sundered Grail	Fantasy
bk106	Randall, Cynthia	Lover Birds	Romance

book_id <chr>	author <chr>	title <chr>	genre <chr>
bk148	Galos, Mike	Visual Studio	Computer
bk107	Thurman, Paula	Splish Splash	Romance
bk108	Knorr, Stefan	Creepy Crawlies	Horror
bk109	Kress, Peter	Paradox Lost	Science Fiction

1-10 of 16 rows | 1-5 of 8 columns Previous **1** 2 Next

4.(25 pts) Transform data types as necessary and then write the data frames to the appropriate tables in the database.

Hide

```
library(dplyr)
library(magrittr)
# change data type
df %<>%
  mutate(publish_date= as.Date(publish_date))
df %<>%
  mutate(edition= as.numeric(edition))
df %<>%
  mutate(price= as.numeric(price))
print(df)
```

book_id <chr>	author <chr>	title <chr>	genre <chr>
bk101	Gambardella, Matthew	XML Developer's Guide	Computer
bk102	Ralls, Kim	Midnight Rain	Fantasy
bk103	Corets, Eva	Maeve Ascendant	Fantasy
bk104	Corets, Eva	Oberon's Legacy	Fantasy
bk105	Corets, Eva	The Sundered Grail	Fantasy
bk106	Randall, Cynthia	Lover Birds	Romance
bk148	Galos, Mike	Visual Studio	Computer
bk107	Thurman, Paula	Splish Splash	Romance
bk108	Knorr, Stefan	Creepy Crawlies	Horror
bk109	Kress, Peter	Paradox Lost	Science Fiction

1-10 of 16 rows | 1-5 of 8 columns Previous **1** 2 Next

Hide

```
# loading data from dataframe into book table
for(i in 1:nrow(df)){
  write_sql <- paste("Insert into book (book_id, title, edition, price, publish_date, description) values (", '"', df[i, "book_id"], '"', ",", '"', df[i, "title"], '"', ",", '"', df[i, "edition"], '"', ",", '"', df[i, "price"], '"', ",", '"', df[i, "publish_date"], '"', ",", '"', df[i, "description"], '"', ")",seq = "")
  dbSendQuery(con, write_sql)
}
```

Hide

```
x <- data.frame(c(df["author"]))
x <- unique(x)
```

Hide

```
# loading data from dataframe into author table
for(i in 1:nrow(x)){
  write_sql <- paste("Insert into author (name) values (", '"', x[i, "author"], '"', ")",seq = "")
  dbSendQuery(con, write_sql)
}
```

Hide

```
# loading data from dataframe into bridge table (book_author)
book_author <- data.frame(
  author_id = c(1,2,3,3,3,4,5,6,7,8,5,9,9,9,5,3),
  book_id = c(df["book_id"])
)
# loading data from dataframe into book_author table
for(i in 1:nrow(book_author)){
  write_sql <- paste("Insert into book_author (author_id, book_id) values (", book_author[i, "author_id"], ",", '"', book_author[i, "book_id"], '"', ")",seq = "")
  dbSendQuery(con, write_sql)
}
```

Hide

```
y <- data.frame(c(df["genre"]))
y <- unique(y)
```

Hide

```
# loading data from dataframe into genre table
for(i in 1:nrow(y)){
  write_sql <- paste("Insert into genre (book_type) values (", '"', y[i, "genre"], '"', ")",seq = "")
  dbSendQuery(con, write_sql)
}
```

Hide

```
# loading data from dataframe into bridge table (book_genre)
book_genre <- data.frame(
  genre_id = c(1,2,2,2,2,3,1,3,4,5,1,1,1,1,1,2),
  book_id = c(df["book_id"])
)
# loading data from dataframe into book_genre table
for(i in 1:nrow(book_genre)){
  write_sql <- paste("Insert into book_genre (genre_id, book_id) values (", book_genre
[i, "genre_id"], ",", '"', book_genre[i, "book_id"], '"', ")",seq = "")
  dbSendQuery(con, write_sql)
}
```

5.(25 pts) Once the data from the XML is in the database, build SQL chunks for the following queries:

- **What are the titles and prices of all books written by “Galos, Mike”? List the titles and the prices.**

Hide

```
SELECT title, price FROM book
INNER JOIN book_author ON book.book_id = book_author.book_id
INNER JOIN author ON author.author_id = book_author.author_id
WHERE author.name LIKE "%Galos, Mike%";
```

title <chr>	price <dbl>
Visual Studio	69.95
Visual Basic for Beginners	29.95
Visual Studio 7: A Comprehensive Guide	49.95

3 rows

- **What is the most recent year of publication of all books written by “O’Brien, Tim”.**

Hide

```
SELECT MAX(publish_date) as mostRecentYear FROM book
INNER JOIN book_author ON book.book_id = book_author.book_id
INNER JOIN author ON author.author_id = book_author.author_id
WHERE author.name LIKE "%O'Brien, Tim%";
```

mostRecentYear <chr>
2009-10-01

1 row

- **What is the average price of all books in the “Fantasy” genre.**

Hide

```
SELECT AVG(price) as averagePrice FROM book
INNER JOIN book_genre ON book.book_id = book_genre.book_id
INNER JOIN genre ON genre.genre_id = book_genre.genre_id
WHERE genre.book_type LIKE "%Fantasy%";
```

averagePrice <dbl>
6.35

1 row

- **Find the number of books in each genre.**

Hide

```
SELECT book_type, COUNT(*) as cnt FROM book
INNER JOIN book_genre ON book.book_id = book_genre.book_id
INNER JOIN genre ON genre.genre_id = book_genre.genre_id
GROUP BY book_type;
```

book_type <chr>	cnt <int>
Computer	7
Fantasy	5
Horror	1
Romance	2
Science Fiction	1

5 rows

- **List the title and author of all books that cost less than the average price of books.**

Hide

```
SELECT title, name as author FROM book
INNER JOIN book_author ON book.book_id = book_author.book_id
INNER JOIN author ON author.author_id = book_author.author_id
WHERE book.price < (
  SELECT AVG(price) FROM book
);
```

title <chr>	author <chr>
Midnight Rain	Ralls, Kim
Maeve Ascendant	Corets, Eva

title	author
<chr>	<chr>
Oberon's Legacy	Corets, Eva
The Sundered Grail	Corets, Eva
Lover Birds	Randall, Cynthia
Splish Splash	Thurman, Paula
Creepy Crawlies	Knorr, Stefan
Paradox Lost	Kress, Peter
Oberon's Revenge	Corets, Eva

9 rows