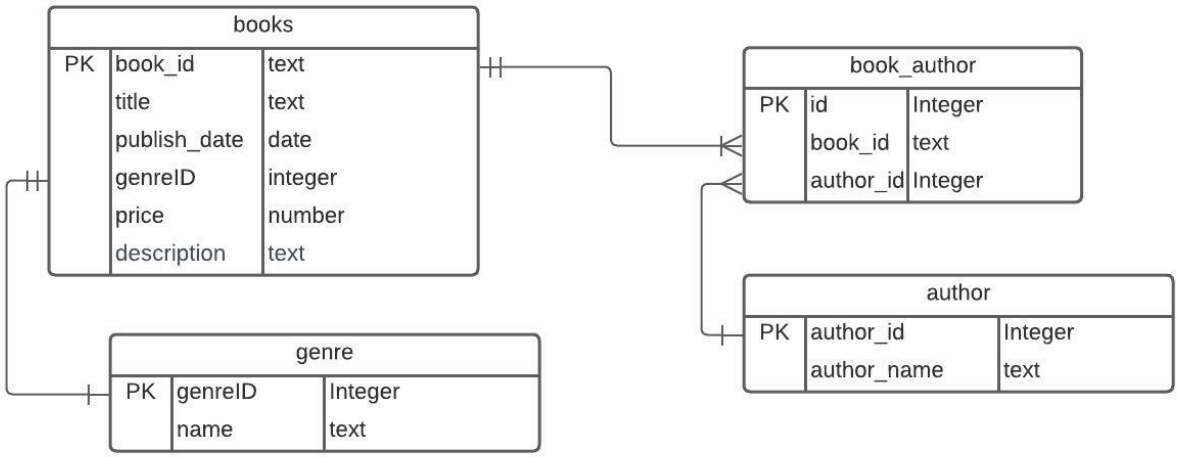
### Crowfoot diagram

```
download.file(url = "https://i.imgur.com/FhtMssI.jpg",
              destfile = "image.jpg",
              mode = 'wb')
knitr::include_graphics("image.jpg")
```




### Create SQLite database

```
library(RSQLite)

# Set the file path for the SQLite database (you can change the filename and path)
db_file <- "my_database.db"

# Connect to the database or create one if it doesn't exist
con <- dbConnect(RSQLite::SQLite(), dbname = db_file)


query_genre <- "CREATE TABLE IF NOT EXISTS genre (
  genreID INTEGER PRIMARY KEY,
  name TEXT
)"

dbExecute(con, statement = query_genre)
```

```
## [1] 0
```

```
query_books <- "CREATE TABLE IF NOT EXISTS books (
  book_id TEXT PRIMARY KEY,
  title TEXT,
  genreID INTEGER,
  price NUMBER,
```

```
  publish_date DATE,
  description Text,
  FOREIGN KEY (genreID) REFERENCES genre (genreID)
)"


dbExecute(con, statement = query_books)
```

```
## [1] 0
```

```
# Create a table for authors
query_authors <- "CREATE TABLE IF NOT EXISTS authors (
  author_id INTEGER PRIMARY KEY,
  author_name TEXT
)"

dbExecute(con, statement = query_authors)
```

```
## [1] 0
```

```
query_book_author <- "CREATE TABLE IF NOT EXISTS book_author (
  id INTEGER PRIMARY KEY,
  book_id INTEGER,
  author_id INTEGER,
  FOREIGN KEY (book_id) REFERENCES books (book_id),
  FOREIGN KEY (author_id) REFERENCES authors (author_id)
)"

dbExecute(con, statement = query_book_author)
```

```
## [1] 0
```

```
# Close the database connection
```

### Load data from xml

```
library("XML")
library(xml2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
xml_file <- "Books-v4.xml"
doc <- read_xml(xml_file)

# Extract book information using XPath
book_nodes <- xml_find_all(doc, "//book")

books_df <- data.frame(book_id = character(),
                       title = character(),
                       genreID = numeric(),
                       price = numeric(),
                       publish_date = character(),
                       description = character(),
                       stringsAsFactors = FALSE)

authors_df <- data.frame(author_name = character(),
                         author_id = numeric(),
                         stringsAsFactors = FALSE)

genre_df <- data.frame(name = character(),
                         genreID = numeric(),
                         stringsAsFactors = FALSE)

book_author_df <- data.frame(
                         id = numeric(),
                         book_id = character(),
                         author_id = numeric(),
                         stringsAsFactors = FALSE)

temp_genre_id <- 0
temp_author_id <- 0
temp_book_id <- 0
temp_book_author_id <- 0

authors_list <- list()

for (book_node in book_nodes) {
  book_id <- xml_attr(book_node, "id")
  title <- xml_text(xml_find_first(book_node, "./title"))
  genre <- xml_text(xml_find_first(book_node, "./genre"))
  price <- as.numeric(xml_text(xml_find_first(book_node, "./price")))
  publish_date <- xml_text(xml_find_first(book_node, "./publish_date"))
  description <- xml_text(xml_find_first(book_node, "./description"))


  # Extract author information and add it to the authors data frame
  authors <- xml_text(xml_find_all(book_node, "./author"))

  genre_mp_id <- 0;

  if (!genre %in% genre_df$name) {
    temp_genre_id = temp_genre_id + 1

    # Create a new genre row and add it to genre_df
```

```
    new_genre_row <- data.frame(name = genre, genreID = temp_genre_id)
    genre_df <- rbind(genre_df, new_genre_row)
    genre_mp_id<- temp_genre_id;
  } else {
    # If the genre already exists, get its genreID
    genre_mp_id<- genre_df$genreID[genre_df$name == genre]
  }

  new_book_row <- data.frame(book_id, title, price, publish_date, description, genreID =  genre_mp_id)
books_df <- rbind(books_df, new_book_row)

  authors_list <- union(authors_list,authors)


}

a_id <- 1;

for(author in authors_list){
  new_row <- data.frame(author_id = a_id , author_name = author)
  print(new_row)
  authors_df <- rbind(authors_df,new_row)
  a_id = a_id +1;
}
```

```
##   author_id          author_name
## 1         1 Gambardella, Matthew
##   author_id author_name
## 1         2  Ralls, Kim
##   author_id author_name
## 1         3 Corets, Eva
##   author_id      author_name
## 1         4 Randall, Cynthia
##   author_id author_name
## 1         5 Galos, Mike
##   author_id    author_name
## 1         6 Thurman, Paula
##   author_id   author_name
## 1         7 Knorr, Stefan
##   author_id  author_name
## 1         8 Kress, Peter
##   author_id       author_name
## 1         9 Katz, Christopher
##   author_id       author_name
## 1        10 Kamarovsky, Susan
##   author_id  author_name
## 1        11 O'Brien, Tim
##   author_id author_name
## 1        12  Lu, Xinyue
```

```
book_author_id <- 1;

for(book in book_nodes){
```

```
    author_names_to_find <- xml_text(xml_find_all(book, "./author"))
    matching_ids <- authors_df$author_id[match(author_names_to_find, authors_df$author_name)]
     book_id <- xml_attr(book, "id")

    for(id in matching_ids){
      new_book_author_row <- data.frame(id=book_author_id,book_id=book_id,author_id=id)
      book_author_df <- rbind(book_author_df,new_book_author_row)
      book_author_id = book_author_id + 1;
    }



}


dbWriteTable(con, "genre", genre_df, append = T,row.names=FALSE)
dbWriteTable(con, "authors", authors_df, append = T,row.names=FALSE)
dbWriteTable(con, "book_author", book_author_df, append = T,row.names=FALSE)
dbWriteTable(con, "books", books_df, append = T,row.names=FALSE)
```

### What is the number of genres have at least three books?

```
select count(*) as num_of_genres from (SELECT genreID , count(*) AS num_books
FROM books
GROUP BY genreID
HAVING num_books >= 3)
;
```

Table 1: 1 records

| num_of_genres |
|---|
| 2 |

### What is the oldest year in which a publication was published?

```
SELECT strftime('%Y', publish_date) AS oldest_year
FROM books
ORDER BY publish_date ASC
LIMIT 1;
```

Table 2: 1 records

| oldest_year |
|---|
| 2000 |

### Find the number of books and average price for each genre

```
select b.genreID , g.name, avg(b.price) as Average_Price ,
count(*) as num_books from books AS b join genre as g ON g.genreID = b.genreID
group by b.genreID
```

Table 3: 5 records

| genreID | name | Average_Price | num_books |
|---|---|---|---|
| 1 | Computer | 46.0875 | 8 |
| 2 | Fantasy | 6.3500 | 5 |
| 3 | Romance | 4.9500 | 2 |
| 4 | Horror | 4.9500 | 1 |
| 5 | Science Fiction | 6.9500 | 1 |

### Which books have more than one author? List the titles of those books

and their number of authors.

```
select b.title,count(a.author_id) as num_authors from books AS b join
book_author as ba ON b.book_id=ba.book_id join authors as a ON a.author_id = ba.author_id  group by b.b
```

Table 4: 3 records

| title | num_authors |
|---|---|
| MSXML3: A Comprehensive Guide | 2 |
| Designing Ontologies with XML | 2 |
| Visual Basic for Beginners | 3 |

### List the title and author of all books that are less than 0.8*AVG or more than 1.2*AVG, where AVG is the average price of all books.

```
select b.title , a.author_name from books AS b join
book_author as ba ON b.book_id=ba.book_id join authors as a ON a.author_id = ba.author_id  where b.price
b.price > 1.2 * (SELECT AVG(price) FROM books);
```

Table 5: Displaying records 1 - 10

| title | author_name |
|---|---|
| XML Developer's Guide | Gambardella, Matthew |
| Midnight Rain | Ralls, Kim |
| Maeve Ascendant | Corets, Eva |
| Oberon's Legacy | Corets, Eva |
| The Sundered Grail | Corets, Eva |

| title | author_name |
|---|---|
| Lover Birds | Randall, Cynthia |
| Visual Studio | Galos, Mike |
| Splish Splash | Thurman, Paula |
| Creepy Crawlies | Knorr, Stefan |
| Paradox Lost | Kress, Peter |