BOOK RATING DATABASE

A DATABASE PROJECT FOR CSC 4402 CLASS SPRING 2017 AT LOUISIANA STATE UNIVERSITY

Group Members:
Pu Du
Pengfei Pan
Min Cui
Tianyi Zhang

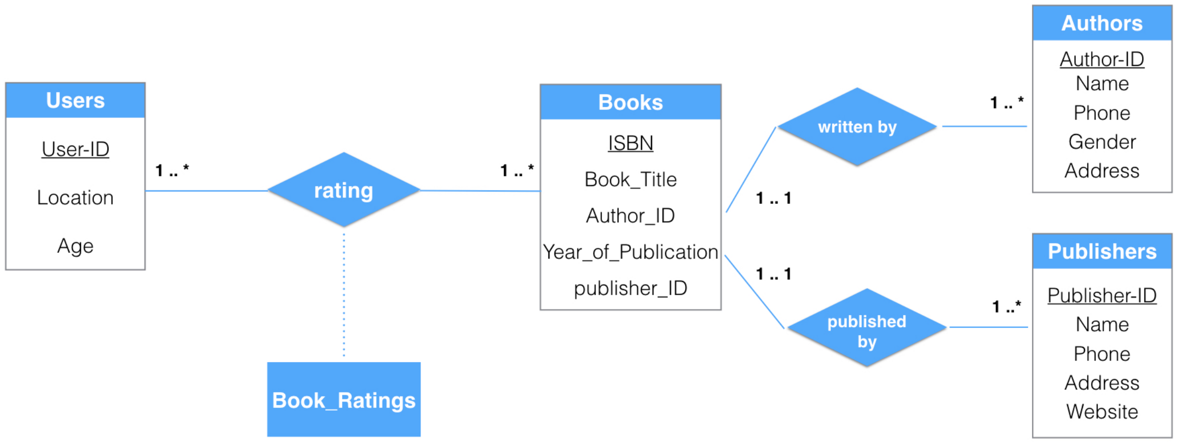
OUTLINE FOR THIS PRESENTATION

- Introduction
- E-R diagram
- Create tables
- SQL queries
  - Count
  - group by and order by
  - natural join
  - subquery

INTRODUCTION

- Database Name: **Book crossing dataset**
  - collected by Cai-Nicolas Ziegler in a 4-week crawl from bookcrossing.com.
  - contains 1,149,780 ratings of 271,379 books by 278,858 users.

E-R DIAGRAM



Users

User-ID

Location

Age

1 .. *

rating

1 .. *

Book_Ratings

Books

ISBN

Book_Title

Author_ID

Year_of_Publication

publisher_ID

1 .. 1

written by

1 .. *

Authors

Author-ID

Name

Phone

Gender

Address

1 .. 1

published by

1 ..*

Publishers

Publisher-ID

Name

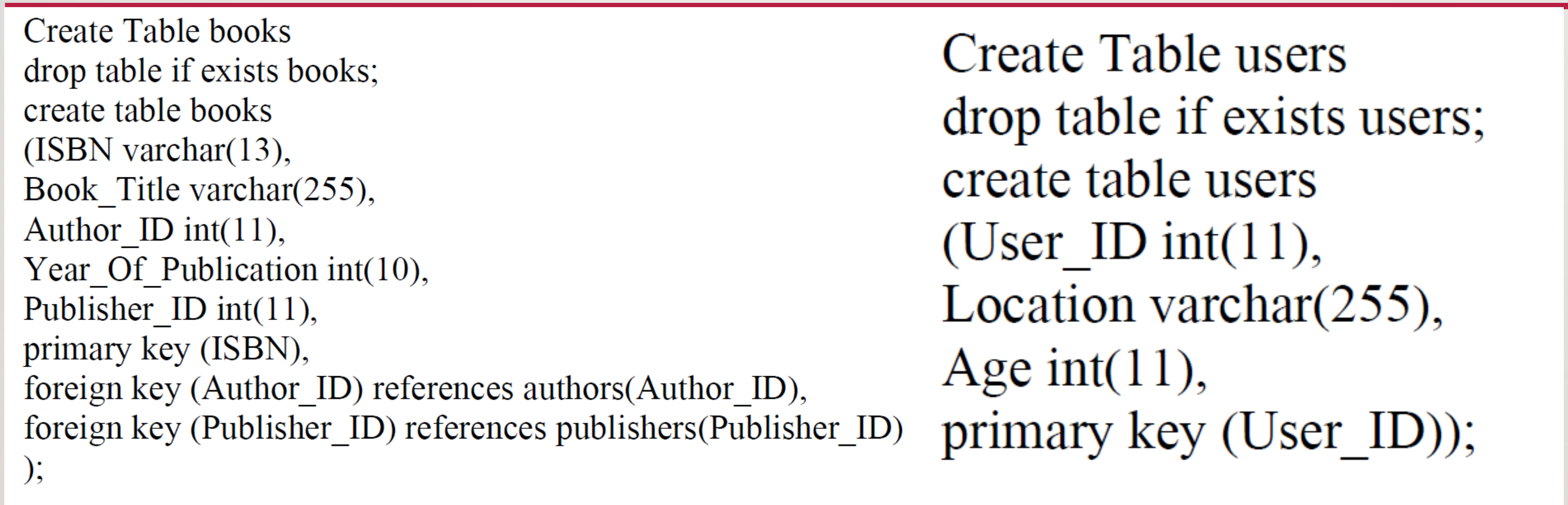
Phone

Address

Website

```
Create table Book_Ratings
drop table if exists book_ratings;
create table book_ratings
(User_ID int(11),
ISBN varchar(13),
Rating int(11),
primary key (User_ID, ISBN),
foreign key (User_ID) references users(User_ID),
foreign key (ISBN) references books(ISBN)
);
```

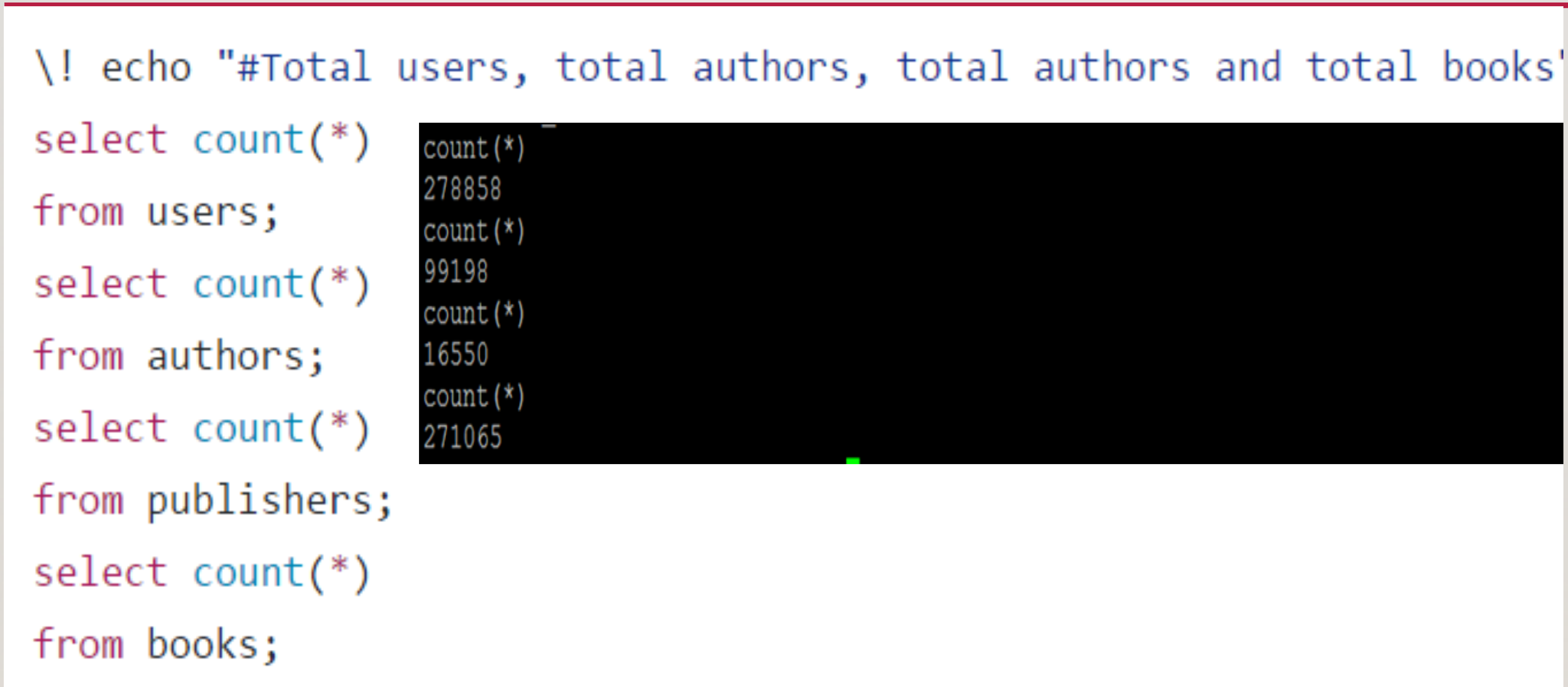
CREATE TABLES



```
Create Table books
drop table if exists books;
create table books
(ISBN varchar(13),
Book_Title varchar(255),
Author_ID int(11),
Year_Of_Publication int(10),
Publisher_ID int(11),
primary key (ISBN),
foreign key (Author_ID) references authors(Author_ID),
foreign key (Publisher_ID) references publishers(Publisher_ID)
);
```

```
Create Table users
drop table if exists users;
create table users
(User_ID int(11),
Location varchar(255),
Age int(11),
primary key (User_ID));
```

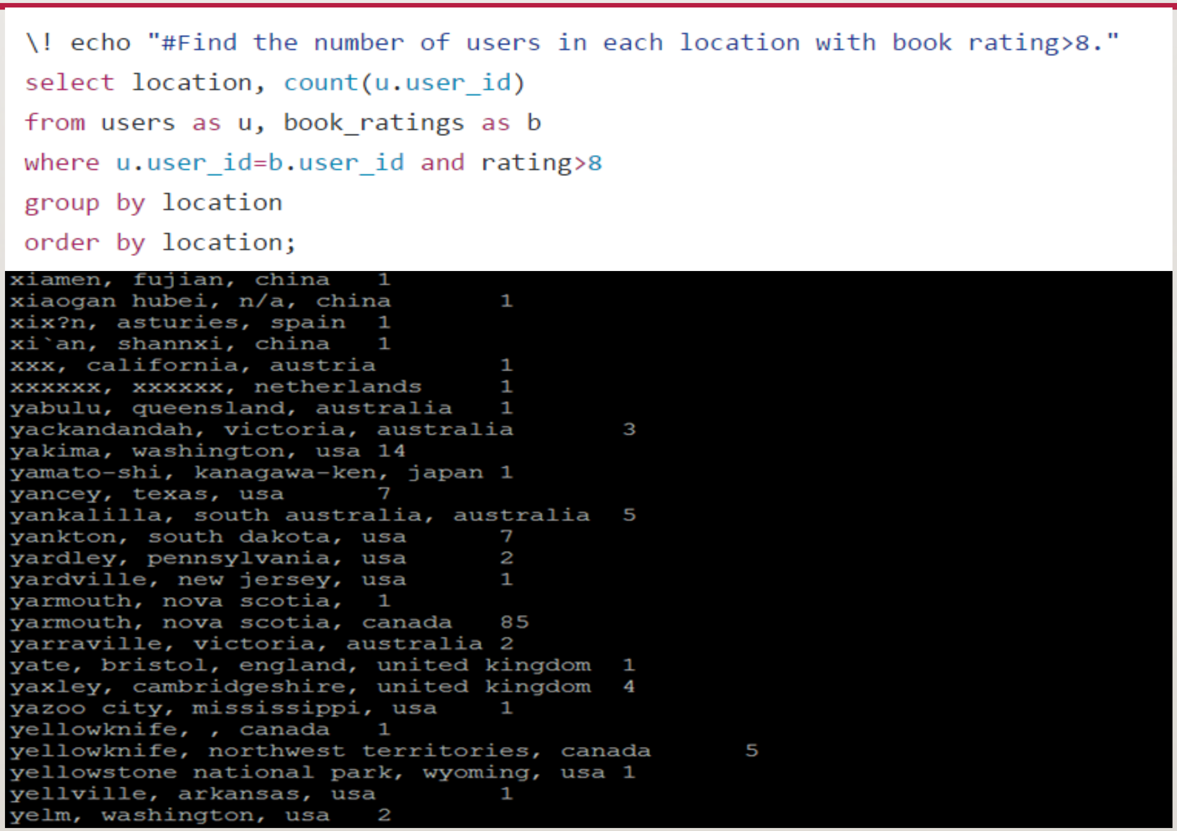
SQL QUERIES--COUNT



```
\! echo "#Total users, total authors, total authors and total books'
select count(*)
from users;
select count(*)
from authors;
select count(*)
from publishers;
select count(*)
from books;
```

```
count(*)
278858
count(*)
99198
count(*)
16550
count(*)
271065
```

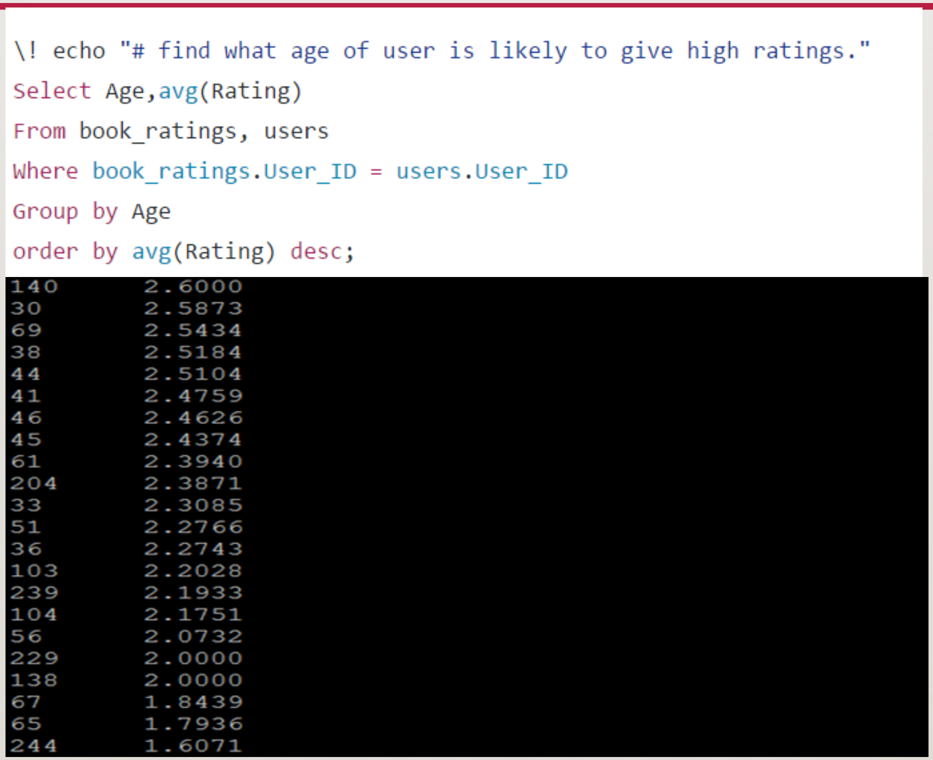
SQL QUERIES—GROUP BY AND ORDER BY



```
\! echo "#Find the number of users in each location with book rating>8."
select location, count(u.user_id)
from users as u, book_ratings as b
where u.user_id=b.user_id and rating>8
group by location
order by location;
```

```
xiamen, fujian, china   1
xiaogan hubei, n/a, china       1
xix?n, asturies, spain  1
xi`an, shannxi, china   1
xxx, california, austria        1
xxxxxx, xxxxxx, netherlands     1
yabulu, queensland, australia   1
yackandandah, victoria, australia       3
yakima, washington, usa 14
yamato-shi, kanagawa-ken, japan 1
yancey, texas, usa      7
yankalilla, south australia, australia  5
yankton, south dakota, usa      7
yardley, pennsylvania, usa      2
yardville, new jersey, usa      1
yarmouth, nova scotia,  1
yarmouth, nova scotia, canada   85
yarraville, victoria, australia 2
yate, bristol, england, united kingdom  1
yaxley, cambridgeshire, united kingdom  4
yazoo city, mississippi, usa    1
yellowknife, , canada   1
yellowknife, northwest territories, canada      5
yellowstone national park, wyoming, usa 1
yellville, arkansas, usa        1
yelm, washington, usa   2
```

SQL QUERIES– INTERESTING EXAMPLE



```
\! echo "# find what age of user is likely to give high ratings."
Select Age,avg(Rating)
From book_ratings, users
Where book_ratings.User_ID = users.User_ID
Group by Age
order by avg(Rating) desc;
```

```
140       2.6000
30        2.5873
69        2.5434
38        2.5184
44        2.5104
41        2.4759
46        2.4626
45        2.4374
61        2.3940
204       2.3871
33        2.3085
51        2.2766
36        2.2743
103       2.2028
239       2.1933
104       2.1751
56        2.0732
229       2.0000
138       2.0000
67        1.8439
65        1.7936
244       1.6071
```

SQL QUERIES– NATURAL JOIN

- ## select the books whose rating is grater than 5 , published by author id = 1, order by rating

```
Select distinct Book_Title,authors.Name,Rating Year_Of_Publication

From books NATURAL JOIN authors NATURAL JOIN book_ratings

Where authors.Author_ID =1 and book_ratings.ISBN = books.ISBN and Rating>5

order by Rating desc;
```

SQL QUERIES– SUBQUERY

```
## select the book rated by user from Sweden and rating = 10
select  A.Name
from books as B, Authors as A
where A.Author_ID = B.Author_ID and ISBN in
(select ISBN
from book_ratings
where Rating = 10 and User_ID in
(select User_ID
from users
where Location = 'sweden'))
```

THANKS FOR WATCHING