**Abdifatah Dahiye**

**CS-300**

**Dr. Nathan**

**09/26/2022**

**Queries**

1. Retrieve the first name and last name of each author in the author relation. Order does not matter. **(1 points)**

**SQL Statement:**

SELECT atr.FirstName "Author First Name",atr.LastName "Author Last Name"  
FROM author atr;

**Output: (Only few records are shown)**

**Author First Name** **Author Last Name**

Toni Morrison

Paul Soloaroff

Vernor Vintage

Dick Francis

Peter Straub

Stephen King

Philip Pratt

Truddi Chase

Bradley Collins

Joseph Heller

Gary Wills

Douglas R. Hofstadter

Harper Lee

1. Retrieve the title and book type in the book relation. Order does not matter. **(1 points)**

**SQL Statement:**

SELECT bk.title "Book Title",bk.type "Book Type"

FROM Book bk;

**Output:**

**Book Title Book Type**

A Deepness in the Sky SFI

Magic Terror HOR

The Stranger FIC

Venice ART

Second Wind MYS

The Edge MYS

Dreamcatcher : A Novel HOR

Treasure Chests ART

Beloved FIC

Harry Potter and the Prisoner of Azkaban SFI

Van Gogh and Gauguin ART

Of Mice and Men FIC

Electric Light POE

Group : Six People in Search of a Life PSY

Nine Stories FIC

The Soul of a New Machine SCI

Travels with Charley TRA

Catch – 22 FIC

1. Retrieve the publisherCode in the book relation. List each publisherCode only once in the result. Order does not matter. **(1 points)**

**SQL Statement:**

SELECT distinct bk.publisherCode

FROM book bk;

**Output:**

**publisherCode**

BA

BP

BY

CT

FA

FS

HC

PE

PL

PU

RH

SC

SS

ST

ΤΑ

TO

1. Retrieve the title and price of each book in the book relation. Further add a calculated column named ‘discount’ that shows the price the book with 25% discount. Order does not matter. Show the first five rows of the result. **(2 points)**

**SQL Statement:**

SELECT bk.title "Book Title",bk.Price "Book Price",bk.price-bk.price\*0.25 "discount"

FROM book bk limit 5;

**Output:**

**Book Title Book Price Discount**

A Deepness in the Sky 7.19 5.3925

Magic Terror 7.99 5.9925

The Stranger 8.00 6.0000

Venice 24.50 18.7125

Second Wind 24.95 18.7125

1. Retrieve the title and price for any book whose price is higher than $20.00 in the book relation. Show the full result. **(2 points)**

**SQL Statement:**

SELECT bk.title "Book Title",bk.Price "Book Price"   
FROM book bk   
WHERE bk.Price>20.00;

**Output:**

**Book Title Book Price**

Venice 24.50

Second Wind 24.95

Treasure Chests 2.46

Van Gogh and Gauguin 21.00

A Guide to SQL 37.95

1. Retrieve the publisherName of all publishers that are in New York only in the publisher relation. Order does not matter. **(2 points)**

**SQL Statement:**

SELECT plr.PublisherName "Publisher Name"  
FROM PUBLISHER plr   
WHERE plr.city like 'New York';

**Output:**

**Publisher Name**

Arcade Publishing

Back Bay books

Fawcett books

Farrar Straus and Giroux

HarperCollins publishers

Jove Publications

Lb books

Penguin USA

Plume

Putnam Publishing Group

Random House

Schoken books

Scribner

Simon and Schuster

SchoLASTic Trade

Tor books

Thames and Hudson

Vintage books

W.W. Norton

1. Retrieve the publisherName of all publishers that are not in New York in the publisher relation. (use != for inequality). Order does not matter. Show the full result. **(3 points)**

**SQL Statement:**

SELECT plr.PublisherName "Publisher Name"  
FROM PUBLISHER plr   
WHERE plr.city != 'New York';

**Output:**

**Publisher Name**

Arkham House

Basic books

Berkley Publishing

Course Technology

eremy P. Tarcher

McPherson and Co.

Taunton Press

Touchstone books

Westview Press

1. Retrieve the bookCode and onHand for each book for which a branch has between 2 and 4 copies in the inventory relation. **Use the BETWEEN keyword in this query.** Order does not matter. Show the full result. **(3 points)**

**SQL Statement:**

SELECT ivt.BookCode "Book Code",ivt.onHand "Quantity on Hand"  
FROM inventory ivt   
WHERE ivt.onHand between 2 and 4;

**Output:**

**Book Code Quantity on hand**

0180 2

0189 2

0200 3

0378 2

079X 2

079X 3

1351 4

1351 2

138X 3

2226 3

2226 2

2281 3

2766 2

2908 3

3350 2

3906 2

5790 2

6128 4

1. Retrieve the title and type for each book in the book relation in which the type is SFI, HOR, ART or PSY. **Use the IN operator for this query.** Order the result by type ascending. Show the first five rows of the result. **(3 points)**

**SQL Statement:**

SELECT bk.Title "Book Title",bk.type "Book Type" from book bk   
WHERE bk.type in ('SFI','HOR','ART','PSY')  
ORDER BY bk.Title asc limit 5;

**Output:**

**Book Title Book Type**

A Deepness in the Sky SFI

Black House HOR

Dreamcatcher : A Novel HOR

Group : Six People in Search of a Life PSY

Harry Potter and the Goblet of Fire SFI

1. Retrieve the title for each book in the book relation that begins with the word “The”. Order the result by title ascending. Show the full result. **(3 points)**

**SQL Statement:**

SELECT bk.Title "Book Title" from book bk   
WHERE bk.Title like 'The%'  
ORDER BY bk.Title asc;

**Output:**

**Book Title**

The Catcher in the Rye

The Edge

The Fall

The Grapes of Wrath

The Soul of a New Machine

The Stranger

1. Retrieve the title for each book in the book relation that doesn’t have a type. Order the result by title ascending. Show the full result. **(3 points)**

**SQL Statement:**

SELECT bk.Title "Book Title" from book bk   
WHERE bk.type is null  
ORDER BY bk.Title asc;

**Output:**

**Book Title**

Null (Empty)

1. Retrieve all of the columns from the author relation. Order the results by the authors lastName descending. **(3 points)**

**SQL Statement:**

SELECT atr.authorNum "Author Number",  
atr.FirstName "Author First Name",atr.LastName "Author Last Name"  
FROM author atr  
ORDER BY atr.LastName desc;

**Output:**

**Author Number Author First Name Author Last Name**

11 Gary Wills

3 Vernor Vintage

5 Peter Straub

20 John Steinbeck

2 Paul Solotaroff

25 Lon Schleining

16 J.D Salinger

15 J.K Rowling

7 Philip Pratt

22 Barbara Owen

23 Randy O’Rourke

1 Toni Morrison

13 Harper Lee

6 Stephen King

24 Tracy Kidder

12 Douglas R. Hofstadter

10 Joseph Heller

17 Seamus Heaney

1. Retrieve the title and type from the book relation. Order the results first by type and then by title. Both sort keys should be ascending order. Show the first five rows of the result. **(3 points)**

**SQL Statement:**

SELECT bk.Title "Book Title",bk.type "Book Type" from book bk   
ORDER BY bk.type asc,  bk.Title asc limit 5;

**Output:**

**Book Title Book Type**

Treasure Chests ART

Van Gogh and Gauguin ART

Venice ART

A Guide to SQL CMP

Beloved FIC

1. Retrieve a count of the number of books published by Penguin USA. Name the column ‘Penguin Books’. Order does not matter. Show your full result. **(3 points)**

**SQL Statement:**

SELECT count(bk.BookCode) "Penguin Books"  
 FROM book bk,publisher plr  
 WHERE bk.publishercode=plr.publishercode  
 AND plr.publisherName like 'Penguin USA';

**Output:**

**Penguin Books**

4

1. Retrieve the publisherCode and the number of books from from publisher from the book relation. Order the result by the count in descending order. Show the first five rows of the result. **(3 points)**

**SQL Statement:**

SELECT publisherCode, count(\*) as books\_count  
FROM Book  
GROUP BY publisherCode  
ORDER BY books\_count DESC LIMIT 5;

**Output:**

**PublisherCode Book\_Count**

PE 4

PL 3

LB 3

JP 3

VB 2

1. Retrieve the publisherCode and the number of books from that publisher from the book relation. Order the results by publisherCode and only show publishers who have 3 or more books in the relation. **Hint: will need to use the HAVING keyword.** Show your full result. **(3 points)**

**SQL Statement:**

SELECT publisherCode, count(\*) as books\_count  
FROM Book  
GROUP BY publisherCode  
HAVING count(\*)>2  
ORDER BY publisherCode;

**Output:**

**PublisherCode Books\_Count**

JP 3

LB 3

PE 4

PL 3

1. Retrieve the number of books in the book relation whose prices is $20.00 or lower. Order does not matter. Show your full result. **(3 points)**

**SQL Statement:**

SELECT \*  
FROM Book  
GROUP BY price  
HAVING price<21;

**Output:**

1. Retrieve the title of the most expensive book in the book relation. Use a subquery for this problem. Show your full result. **(3 points)**

**SQL Statement:**

SELECT title as Most\_Expensive\_Book  
FROM Book  
WHERE bookCode  IN (SELECT bookCode   
   FROM Book  
WHERE price >35);

**Output:**

**Most\_Expensive\_Book**

A Guide to SQL

1. Retrieve the title of the least expensive book in the book relation. Use a subquery for this problem. Show your full result. **(3 points)**

**SQL Statement:**

SELECT title as Least\_Expensive\_Book  
FROM Book  
WHERE bookCode  IN (SELECT bookCode   
   FROM Book  
WHERE price <6);

**Output:**

**Least Expensive\_Book**

Nine Stories

Franny and Zooey

The Catcher in the Rye

1. Retrieve the title of all books in the book relation that are not published in New York. Order result by the city ascending. Use a subquery for this query. **(5 points)**

**SQL Statement:**

SELECT title   
FROM Book  
WHERE publisherCode  IN (SELECT publisherCode   
   FROM Publisher  
WHERE  NOT city ="New York");

**Output:**

**Title PublisherCode**

Treasure Chests TA

Van Gogh and Gauguin WP

Group : Six People in Search of a Life BP

Band of Brothers TO

A Guide to SQL CT

Godel , Escher , Bach BA

1. Retrieve all of the columns from the book and publisher relations in one result. Use aliases in your query and use the simple JOIN syntax. Order does not matter. **(5 points)**

**SQL Statement:**

SELECT \*  
FROM Book  
FULL OUTER JOIN Publisher ON Book.publisherCode=Publisher.publisherCode;

**Output:**

Try to check the code above since i cant check because the platform that i use does not support join queries

Error : RIGHT and FULL OUTER JOINS are not currently supported

1. Rewrite the previous query using the ON keyword. **(5 points)**

**SQL Statement:**

SELECT \*  
FROM Book JOIN Publisher ON Book.publisherCode=Publisher.publisherCode;

**Output:**

1. Retrieve the title from the book relation and the city from the publisher relation using a JOIN query. Use aliases in your query. Order the result by title. **(5 points)**

**SQL Statement:**

SELECT atr.Title , ,atr.City  
FROM book atr;

**Output:**

1. Retrieve the title from the book relation and the author lastName from the author relation. Order by author lastName. Use aliases in your query. **This will involve JOINING the book, author and wrote relations.** **(5 points)**

**SQL Statement:**

SELECT title

atr.FirstName "Author First Name",atr.LastName "Author Last Name"  
FROM author atr;

**Output:**

1. Retrieve the title from the book relation and branchNum and onHand from the inventory relation. Use aliases in your query. Order the result by title. **(5 points)**

**SQL Statement:**

SELECT title, branchNum, onHand

FROM Book;

ORDER BY tittle;

**Output:**

1. Retrieve the title from the book relation, the branchName from the branch relation and number of copies onHand from the inventory relation. Use aliases in your query. Order the result by title ascending. **(5 points)**

**SQL Statement:**

SELECT title, branchNum

HAVING Count()

FROM Book

ORDER BY asc;

**Output:**

1. Retrieve the title from the book relation and compute the number of copies of the title that all branches have on hand. Name this computed column ‘Inventory’ **Hint: You will need to join book and inventory and do an aggregate query.** Use aliases in your query. Order the result by the total number of copies of the book in descending order. Show the first two rows of your result. **(5 points)**

**SQL Statement:**

**Output:**

1. Retrieve the first name and last name from the author relation and the title from the book relation for all paperback books in the book relation. Order the result by the author last name and title. **(5 points)**

**SQL Statement:**

SELECT atr.FirstName "Author First Name",atr.LastName "Author Last Name", title  
FROM Book ;

HAVING Count ()

ORDER BY Author last name and tittle;

**Output:**

1. Insert a new branch into the branch relation with the following data branch number = 5, branch name = Henry Lexington Green, branch location = 127 South Road, numEmployees = 7. Then write the query to show all of the branches in the branch relation. **(4 points)**

**SQL Statement:**

INSERT INTO branch  
VALUES  
(5,'Henry Lexington Green','127 South Road',7);

SELECT \* FROM Branch;

**Output:**

branchNum BranchName BranchLocation NumEmployee

1 Henry Downtown 16 Riverview 10

2 Henry on the Hill 1289 Bedford 6

3 Henry Brentwood Brentwood Mall 15

4 Henry Eastshore Eastshore Mall 9

5 Henry Lexington Green 127 South Road 7

1. The Henry Downtown branch moved to 184 St. John’s Way. Update the branch table with the new address. **(3 points)**

**SQL Statement:**

UPDATE Branch  
SET branchLocation="184 St. John's Way"  
WHERE branchNum = 1;

SELECT \* FROM Branch;

**Output:**

**SQL query successfully executed**

branchNum BranchName BranchLocation NumEmployee

1 Henry Downtown 184 St. John’s Way 10

2 Henry on the Hill 1289 Bedford 6

3 Henry Brentwood Brentwood Mall 15

4 Henry Eastshore Eastshore Mall 9

5 Henry Lexington Green 127 South Road 7