

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 firstName, lastName
FROM author;

Output:

Morrison, Toni

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

SQL Statement: SELECT title, type
FROM book;

Output:

A Deepness in the Sky SFI

3. 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 publisherCode
FROM book;

Output:

'BA'

4. 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 title, price, price * 0.75 AS discount
FROM book

Output:

A Deepness in the Sky	7.19	5.3925
Magic Terror	7.99	5.9925
The Stranger	8.00	6.0000
Venice	24.50	18.3750
Second Wind	24.95	18.7125

5. 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 title, price
FROM book
WHERE price > 20.00;

Output:

'Venice', '24.50'
'Second Wind', '24.95'

'Treasure Chests', '24.46'
'Van Gogh and Gauguin', '21.00'
'A Guide to SQL', '37.95'

6. 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 publisherName
FROM publisher
WHERE city = "New York";

Output: 'Arcade Publishing'

7. 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 publisherName
FROM publisher
WHERE city != "New York";

Output:

Arkham House
Basic books
Berkley Publishing
Course Technology
Jeremy P. Tarcher
McPherson and Co.
Taunton Press
Touchstone books
Westview Press

8. 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 bookCode, onHand
FROM inventory
WHERE onHand BETWEEN 2 AND 4;

Output:

'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'
 '6128', '3'
 '6328', '2'
 '6908', '2'
 '7405', '2'
 '7559', '2'
 '8720', '3'
 '9611', '2'
 '9627', '2'
 '9701', '2'
 '9701', '3'
 '9701', '2'
 '9882', '3'
 '9883', '3'
 '9883', '2'
 '9931', '2'

9. 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 title, type
 FROM book
 WHERE type IN ("SFI", "HOR", "ART", "PSY")
 ORDER BY type ASC;

Output:

Venice ART
 Treasure Chests ART
 Van Gogh and Gauguin ART
 Magic Terror HOR
 Dreamcatcher: A Novel HOR

10. 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 title
 FROM book
 WHERE title LIKE "The%"
 ORDER BY title ASC;

Output:

The Catcher in the Rye
 The Edge
 The Fall

The Grapes of Wrath
The Soul of a New Machine
The Stranger

11. 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 title
FROM book
WHERE type IS NULL
ORDER BY title ASC;

Output: N/A

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

SQL Statement: SELECT *
FROM author
ORDER BY lastName DESC;

Output:

11	Wills	Gary
3	Vintage	Vernor
5	Straub	Peter
20	Steinbeck	John

13. 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 title, type
FROM book
ORDER BY type ASC, title ASC;

Output:

'Treasure Chests', 'ART'
'Van Gogh and Gauguin', 'ART'
'Venice', 'ART'
'A Guide to SQL', 'CMP'
'Beloved', 'FIC'

14. 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(*) AS "Penguin Books"
FROM book
WHERE publisherCode = "PE";

Output:

4

15. 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 bookCount
FROM book
GROUP BY publisherCode
ORDER BY bookCount DESC;

Output:

PE	4
JP	3
LB	3
PL	3
SC	2

16. 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 bookCount
FROM book
GROUP BY publisherCode
HAVING bookCount >= 3
ORDER BY publisherCode;

Output:

JP	3
LB	3
PE	4
PL	3

17. 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 COUNT(*) AS NumberOfBooks
FROM book
WHERE price <= 20.00;

Output:

28

18. 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
FROM book
WHERE price = (SELECT MAX(price) FROM book);

Output:

'A Guide to SQL'

19. 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
FROM book
WHERE price = (SELECT MIN(price) FROM book);

Output:

Nine Stories
Franny and Zooey
The Catcher in the Rye

20. 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 NOT IN (SELECT publisherCode FROM publisher WHERE city = "New York")
ORDER BY publisherCode ASC;

Output:

'Godel, Escher, Bach'
'Group: Six People in Search of a Life'
'A Guide to SQL'
'Treasure Chests'
'Band of Brothers'
'Van Gogh and Gauguin'

21. 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 b.*, p.*
FROM book AS b
JOIN publisher AS p
WHERE b.publisherCode = p.publisherCode;

Output:

0180	A Deepness in the Sky	TB	SFI	7.19	Y	TB	Tor books	New York
------	-----------------------	----	-----	------	---	----	-----------	----------

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

SQL Statement: SELECT b.*, p.*
FROM book AS b
JOIN publisher AS p
ON b.publisherCode = p.publisherCode;

Output:

0180	A Deepness in the Sky	TB	SFI	7.19	Y	TB	Tor books	New York
------	-----------------------	----	-----	------	---	----	-----------	----------

23. 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 b.title, p.city
FROM book b
JOIN publisher p
ON b.publisherCode = p.publisherCode
ORDER BY b.title;

Output:

'A Deepness in the Sky', 'New York'
'A Guide to SQL', 'Boston'
'Band of Brothers', 'Westport CT'

24. 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 b.title, a.lastName AS authorlastName
FROM book b
JOIN wrote w ON b.bookCode = w.bookCode
JOIN author a ON w.authorNum = a.authorNum
ORDER BY a.lastName

Output:

'Band of Brothers', 'Ambrose'
'The Stranger', 'Camus'
'The Fall', 'Camus'
'When Rabbit Howls', 'Chase'

25. 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 b.title, i.branchNum, i.onHand
FROM book b
JOIN inventory i ON b.bookCode = i.bookCode
ORDER BY b.title;

Output:

'A Deepness in the Sky', '1', '2'
'A Guide to SQL', '1', '1'
'Band of Brothers', '2', '2'
'Beloved', '2', '3'

26. 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 b.title, br.branchName, i.onHand
FROM book b
JOIN inventory i ON b.bookCode = i.bookCode
JOIN branch br ON i.branchNum = br.branchNum
ORDER BY b.title ASC;

Output:

'A Deepness in the Sky', 'Henry Downtown', '2'
 'A Guide to SQL', 'Henry Downtown', '1'
 'Band of Brothers', 'Henry On The Hill', '2'

27. 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: SELECT b.title, SUM(i.onHand) AS Inventory
 FROM book b
 JOIN inventory i ON b.bookCode = i.bookCode
 GROUP BY b.title
 ORDER BY Inventory DESC;

Output:

The Grapes of Wrath 8
 Jazz 7

28. 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 a.firstName, a.lastName, b.title
 FROM author a
 JOIN wrote w ON a.authorNum = w.authorNum
 JOIN book b ON w.bookCode = b.bookCode
 WHERE b.paperback = "Y"
 ORDER BY a.lastName, b.title;

Output:

Stephen E. Ambrose Band of Brothers

29. 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 (branchNum, branchName, branchLocation, numEmployees)
 VALUES (5, "Henry Lexington Green", "127 South Road", 7);

Output:

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

30. 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 = 5 AND branchName = "Henry Lexington Green";

SELECT * FROM branch;

Output:

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	184 St. John's Way	7