**Adam Winders**

**CS-300-ON**

**20 September 2022**

**Writing Queries Using SQL**

**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:

**lastName firstName**

**Morrison Toni**

**Solotaroff Paul**

**Vintage Vernor**

**Francis Dick**

**Straub Peter**

**King Stephen**

**Pratt Philip**

**Chase Truddi**

**Collins Bradley**

**Heller Joseph**

**Wills Gary**

**Hofstadter Douglas R.**

**Lee Harper**

**Ambrose Stephen E.**

**Rowling J.K.**

**Salinger J.D.**

**Heaney Seamus**

**Camus Albert**

**Collins, Jr. Bradley**

**Steinbeck John**

**Castelman Riva**

**Owen Barbara**

**O'Rourke Randy**

**Kidder Tracy**

**Schleining Lon**

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

SQL Statement:

**SELECT title, TYPE**

**FROM book;**

Output:

**title 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**

**Jazz FIC**

**Band of Brothers HIS**

**A Guide to SQL CMP**

**Franny and Zooey FIC**

**East of Eden FIC**

**Harry Potter and the Goblet of Fire SFI**

**The Fall FIC**

**Godel, Escher, Bach PHI**

**When Rabbit Howls PSY**

**Black House HOR**

**Song of Solomon FIC**

**The Grapes of Wrath FIC**

**Slay Ride MYS**

**The Catcher in the Rye FIC**

**To Kill a Mockingbird 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 (publisherCode)**

**FROM book;**

Output:

**publisherCode**

**BA**

**BP**

**BY**

**CT**

**FA**

**FS**

**HC**

**JP**

**LB**

**PE**

**PL**

**PU**

**RH**

**SC**

**SS**

**ST**

**TA**

**TB**

**TO**

**VB**

**WP**

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

SQL Statement:

**SELECT title, price, (price-(price \* 0.25)) AS discount**

**FROM book;**

Output:

**title 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 18.3750**

**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 title, price**

**FROM book**

**WHERE price > 20;**

Output:

**title price**

**Venice 24.50**

**Second Wind 24.95**

**Treasure Chests 24.46**

**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 publisherName**

**FROM publisher**

**WHERE city = ‘New York’;**

Output:

**publisherName**

**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 publisherName**

**FROM publisher**

**WHERE city != 'New York';**

Output:

**publisherName**

**Arkham House**

**Basic books**

**Berkley Publishing**

**Course Technology**

**Jeremy 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 b.bookCode, onHand**

**FROM book b, inventory i**

**WHERE onHand BETWEEN 2 AND 4 AND b.bookCode = i.bookCode;**

Output:

**bookCode onHand**

**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**

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 title, TYPE**

**FROM book**

**WHERE TYPE IN ('SFI','HOR','ART','PSY')**

**ORDER BY TITLE ASC;**

Output:

**title 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 title**

**FROM book**

**WHERE title LIKE 'The%';**

Output:

**title**

**The Stranger**

**The Edge**

**The Soul of a New Machine**

**The Fall**

**The Grapes of Wrath**

**The Catcher in the Rye**

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 title**

**FROM book**

**WHERE TYPE IS NULL**

**ORDER BY title ASC;**

Output:

**title**

1. 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:

**authorNum lastName firstName**

**11 Wills Gary**

**3 Vintage Vernor**

**5 Straub Peter**

**20 Steinbeck John**

**2 Solotaroff Paul**

**25 Schleining Lon**

**16 Salinger J.D.**

**15 Rowling J.K.**

**7 Pratt Philip**

**22 Owen Barbara**

**23 O'Rourke Randy**

**1 Morrison Toni**

**13 Lee Harper**

**6 King Stephen**

**24 Kidder Tracy**

**12 Hofstadter Douglas R.**

**10 Heller Joseph**

**17 Heaney Seamus**

**4 Francis Dick**

**19 Collins, Jr. Bradley**

**9 Collins Bradley**

**8 Chase Truddi**

**21 Castelman Riva**

**18 Camus Albert**

**14 Ambrose Stephen E.**

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 title, TYPE**

**FROM book**

**ORDER BY TYPE, title;**

Output:

**title TYPE**

**Treasure Chests ART**

**Van Gogh and Gauguin ART**

**Venice ART**

**A Guide to SQL CMP**

**Beloved FIC**

**Azkaban SFI**

**Travels with Charley TRA**

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(\*) AS 'Penguin Books'**

**FROM book**

**WHERE publisherCode = 'PE';**

Output:

**Penguin Books**

**4**

1. Retrieve the publisherCode and the number of books 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(\*)**

**FROM book**

**GROUP BY publisherCode**

**ORDER BY COUNT(\*) DESC;**

Output:

**publisherCode COUNT(\*)**

**PE 4**

**JP 3**

**LB 3**

**PL 3**

**SC 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(\*)**

**FROM book**

**GROUP BY publisherCode**

**HAVING COUNT(\*) >= 3**

**ORDER BY publisherCode;**

Output:

**publisherCode 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 COUNT(\*)**

**FROM book**

**WHERE price <= 20;**

Output:

**COUNT(\*)**

**28**

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**

**FROM book**

**WHERE price = (SELECT MAX(price) FROM book);**

Output:

**title**

**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**

**FROM book**

**WHERE price = (SELECT MIN(price) FROM book);**

Output:

**title**

**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, city**

**FROM book, publisher**

**WHERE city IN (**

**SELECT city**

**FROM publisher**

**WHERE city != 'New York')**

**ORDER BY city;**

Output:

**This question did not specify what row(s) of the result to show.**

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 b.\*, p.\***

**FROM book b, publisher p**

**WHERE b.publisherCode = p.publisherCode;**

Output:

**This question did not specify what row(s) of the result to show.**

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

SQL Statement:

**SELECT b.\*, p.\***

**FROM book b**

**JOIN publisher p ON b.publisherCode = p.publisherCode;**

Output:

**This question did not specify what row(s) of the result to show.**

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 b.title, p.city**

**FROM book b**

**CROSS JOIN publisher p**

**ORDER BY b.title;**

Output:

**This question did not specify what row(s) of the result to show.**

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, lastName, a.authorNum, b.bookCode**

**FROM book b, author a, wrote w**

**WHERE a.authorNum = w.authorNum AND b.bookCode = w.bookCode**

**ORDER BY lastName;**

Output:

**This question did not specify what row(s) of the result to show.**

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, b.bookCode**

**FROM book b, inventory i**

**WHERE b.bookCode = i.bookCode**

**ORDER BY title;**

Output:

**This question did not specify what row(s) of the result to show.**

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, branchName, onHand, br.branchNum, b.bookCode**

**FROM book b, branch br, inventory i**

**WHERE br.branchNum = i.branchNum AND b.bookCode = i.bookCode**

**ORDER BY title;**

Output:

**This question did not specify what row(s) of the result to show.**

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:

**SELECT title, b.bookCode, br.branchNum AS Inventory**

**FROM book b, inventory i, branch br**

**WHERE b.bookCode = i.bookCode AND br.branchNum = i.branchNum**

**ORDER BY Inventory DESC;**

Output:

**title bookCode Inventory**

**Second Wind 079X 4**

**Harry Potter and the Prisoner of Azkaban 2226 4**

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 firstName, lastName, title, paperback**

**FROM author, book**

**WHERE paperback = 'Y'**

**ORDER BY lastName AND title;**

Output:  
  
**This question did not specify what row(s) of the result to show.**

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:

**This question did not specify what row(s) of the result to show.**

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 branchName = 'Henry Downtown' AND branchNum = 1;**

Output:

**This question did not specify what row(s) of the result to show.**