1) List all the data in the books database:

authors(7)

SELECT \* FROM authors

publishers(4)

SELECT \* FROM publishers

titles(13)

SELECT \* FROM titles

title\_authors(17)

SELECT \* FROM title\_authors

royalties(13)

SELECT \* FROM royalties

2) Find all the info on authors whose last name starts with 'H'. (3)

SELECT \* from authors

WHERE au\_lname LIKE “H%”;

3) Find the title, type, salesprice, published date for all titles published between July 15, 2014 and August 15, 2014. (2)

SELECT title, name, type, price, pubdate FROM titles

WHERE pubdate BETWEEN ‘2014-07-15’ AND ‘2014-08-15’

4) Find all the information for titles T01, T04, and T07. (3)

Hint: this can be done two ways. Try both of them. Which is easier if you have a large number of selection criteria?

SELECT \* FROM titles

WHERE title\_id = 'T01' OR title\_id = 'T04' OR title\_id = 'T07';

5) Find the last name, first name, address, city, state for all authors.

Display them in alphabetical order by city within state. (7)

SELECT au\_fname, au\_lname, address, city, state FROM authors

ORDER BY state ASC, city ASC;

6) Find the title and sales price for all books that sell for more than $20. (4)

SELECT title\_name, price FROM titles

WHERE price > 20;

7) Find the title and number of pages for all books that have sold < 5000 copies. (2)

SELECT title\_name, pages FROM titles

WHERE sales < 5000;

8) Find the title of all books that have been published by 'Core Dump Books'. (1)

SELECT \* FROM titles WHERE pub\_ID = 'P02';

9) Find the title of all books that have not been publsihed by 'Core Dump Books' (12)

SELECT \* FROM titles WHERE pub\_ID != 'P02';

10) List all the different types of books. (5)

SELECT DISTINCT type FROM titles;

11) Find the publisher name of all books that were published in August 2014. (1)

SELECT DISTINCT pub\_name from publishers natural join titles where pubdate >='2014-08-01' AND pubdate <='2014-08-31';

OR

SELECT DISTINCT pub\_name FROM titles right OUTER JOIN publishers on titles.PUB\_ID = publishers.PUB\_ID

where pubdate Between '2014-08-01' AND '2014-08-31';

12) Find the authors that live in the same state as their publisher. (4)

This is a complicated join. Prove to yourself that your answer is correct.

SELECT DISTINCT au\_fname, au\_lname, authors.state FROM authors INNER JOIN publishers USING ( state );

OR

SELECT DISTINCT au\_fname, au\_lname FROM ( publishers RIGHT OUTER JOIN titles USING (pub\_id)

RIGHT OUTER JOIN title\_authors USING (title\_id) RIGHT OUTER JOIN authors USING (au\_id) )

WHERE authors.state = publishers.state;

Find the authors that live in a different state than their publisher.

(2)

SELECT DISTINCT au\_fname, au\_lname FROM ( publishers RIGHT OUTER JOIN titles USING (pub\_id)

RIGHT OUTER JOIN title\_authors USING (title\_id) RIGHT OUTER JOIN authors USING (au\_id) )

WHERE authors.state != publishers.state;

13) Find the list of authors that write history books. (1)

SELECT DISTINCT au\_fname, au\_lname FROM authors RIGHT OUTER JOIN title\_authors USING (au\_id)

RIGHT OUTER JOIN titles ON titles.TITLE\_ID = title\_authors.TITLE\_ID

WHERE type = 'history';

14) Find the publisher name of all books whose title starts with 'E'. (1)

SELECT DISTINCT pub\_name FROM publishers NATURAL JOIN titles WHERE title\_name LIKE 'E%'

15) List all the books and their authors of all books that have been published by 'Schedenfrude Press'. (4)

SELECT \* FROM ( publishers RIGHT OUTER JOIN titles using (pub\_id)

RIGHT OUTER JOIN title\_authors USING (title\_id) RIGHT OUTER JOIN authors USING (au\_id) ) WHERE PUB\_NAME = 'Schadenfreude Press';

16) List the authors and titles and all the advances paid for all books. Assume that the author receives the

percentage of the advance indicated by their share of the royalties.

Include books that have not had any advances paid. (17)

17) How many books were published by each publisher. (4)

SELECT pub\_name, COUNT (title\_name) AS booksPub FROM

publishers RIGHT OUTER JOIN titles USING (pub\_id)

GROUP BY ( pub\_name );

18) How many books were published by each publisher where the number of books published is greater than 2. (3)

19) What is the highest advance paid for a book.

SELECT MAX (advance) FROM royalties;

20) What is the average royalty rate paid.

SELECT AVG ( royalty\_rate ) FROM royalties;

21) List the total royalties paid out for each book.

SELECT title\_name, ( royalty\_rate \* sales ) AS totalRoyalties FROM titles RIGHT OUTER JOIN royalties USING (title\_id);

22) What is the total money paid out to each author to date (include advances).

23) How many authors live in each state.

SELECT authors.state, COUNT (au\_fname) AS authorsBYstate FROM

publishers RIGHT OUTER JOIN authors ON (publishers.state = authors.state)

GROUP BY ( authors.state )

ORDER BY authorsBYstate DESC;

24) How many books were written by each author. Display the list with the highest number of books first.

SELECT au\_fname, au\_lname, COUNT (title\_name) AS bookSold FROM ( titles RIGHT OUTER JOIN

title\_authors USING ( title\_id) RIGHT OUTER JOIN authors USING (au\_id) )

GROUP BY au\_fname,au\_lname

ORDER BY bookSold DESC;

25) Find the first name and last name of all authors who live in the same state as Sarah Buchman. (2)

26) Find the title of all books that have less pages than "I Blame my Mother". (5)

SELECT title\_name FROM titles WHERE pages < ( SELECT pages FROM titles WHERE title\_name = 'I Blame My Mother') ;

27) Find the books that pay out a lower royalty rate than "Ask Your System Administrator". (5)

SELECT DISTINCT title\_name FROM ( titles NATURAL JOIN royalties ) WHERE royalty\_rate < ( SELECT royalty\_rate FROM titles NATURAL JOIN royalties WHERE title\_name = 'Ask Your System Administrator' )

28) What is the title and price of the lowest priced book?

SELECT title\_name, price FROM titles WHERE PRICE = ( SELECT MIN(price) FROM titles);

29) Find the publishers of all books that are the same type as "How About Never?" (2)

SELECT DISTINCT pub\_name FROM ( publishers NATURAL JOIN titles ) WHERE titles.type = ( SELECT titles.type FROM titles WHERE title\_name = 'How About Never?' )

30) List all city and states where we do business -- where we have authors or publishers. Identify each row as either an author or a publisher. (10)

SELECT city, state, 'Authors' AS "Business" FROM authors UNION select city, state, 'Publishers' AS "Business" FROM publishers;

31) List the states (in alphabetical order) in which we have both authors and publishers. (2)

SELECT DISTINCT state FROM authors INNER JOIN publishers USING (state) ORDER BY state ASC

32) Find the list of all authors who have not been published. (1)

SELECT au\_fname, au\_lname FROM authors LEFT OUTER JOIN title\_authors ON authors.au\_id = title\_authors.au\_id WHERE title\_id IS NULL;

33) List the publishers (in alphabetical order) that did not publish a book during 2014. (1)

SELECT pub\_name FROM publishers EXCEPT SELECT pub\_name FROM titles NATURAL JOIN publishers WHERE pubdate BETWEEN ‘01/01/2014’ AND ‘12/31/2014’ ORDER BY pub\_name;

33) List the titles (in alphabetical order) of all books that were not published during 2014. (8)

SELECT title\_name FROM titles WHERE pubdate < '2014-01-01' OR pubdate > '2014-12-31' OR pubdate IS NULL ORDER BY title\_name ;

34) List the states that (in alphabetical order) in which we have authors, publishers, or both. List the states in alphabetical order, and indicate which it is (‘Author’, ‘Publisher’, or ‘Both’) in the output. If a given state is in both Authors and Publishers, list it only once, designated as ‘Both’. (5)

SELECT DISTINCT state, 'Both' AS Business FROM publishers NATURAL JOIN authors WHERE publishers.state = authors.state

    UNION (SELECT DISTINCT state, 'Publisher' AS Business FROM publishers)

    UNION (SELECT DISTINCT state, 'Author' AS Business FROM authors)

    EXCEPT (SELECT DISTINCT state, 'Author' AS Business FROM authors NATURAL JOIN publishers WHERE authors.state = publishers.state)

    EXCEPT (SELECT DISTINCT state, 'Publisher' AS Business FROM publishers NATURAL JOIN authors WHERE publishers.state = authors.state);

35) List author name, and book title in which the author is the last author listed (they are lowest on the au\_order). In other words, for a given title, if there are only three authors listed, list the third author and them alone. (13)

SELECT au\_fname, au\_lname, title\_name FROM titles

    NATURAL JOIN authors

    NATURAL JOIN title\_authors

    WHERE au\_order = (SELECT MIN(au\_order) FROM title\_authors);

36) Display the title name, and the number of authors for that title. Order the rows from the book with the highest number of authors to the lowest. (13)

SELECT title\_name, count(title\_name) AS “Number of Authors” FROM titles NATURAL JOIN title\_authors GROUP BY title\_name ORDER BY count(title\_name) DESC, title\_name;

37) List the author(s) of the book with the highest sales. (2)

SELECT au\_fname, au\_lname FROM authors RIGHT OUTER JOIN title\_authors USING (au\_id) RIGHT OUTER JOIN titles USING (title\_id) WHERE sales = ( SELECT MAX(sales) FROM titles);