**Task 2: Creating queries**

|  |  |  |
| --- | --- | --- |
| Q1 | SELECT publisher\_name  FROM publisher  WHERE publisher\_city <> 'New York'; |  |
| Q2 | SELECT book\_title AS 'title'  FROM book  WHERE publisher\_code =  (SELECT publisher\_code  FROM publisher  WHERE publisher\_name = 'Penguin USA'); |  |
| Q3 | SELECT book\_title AS 'title'  FROM book  WHERE book\_type = 'SFI'  AND book\_paperback = 'TRUE'; |  |
| Q4 | SELECT book\_title AS 'title'  FROM book  WHERE book\_type IN ('CMP', 'HIS', 'SCI'); |  |
| Q5 | SELECT count(\*) AS 'book count'  FROM author JOIN wrote USING (author\_num)  WHERE (author\_first = 'Dick' AND author\_last = 'Francis')  OR (author\_first = 'Vernor' AND author\_last = 'Vintage'); |  |
| Q6 | SELECT book\_title  FROM book JOIN wrote USING (book\_code)  JOIN author USING (author\_num)  WHERE book\_type = 'FIC'  AND author\_first = 'John'  AND author\_last = 'Steinbeck'; |  |
| Q7 | SELECT count(\*) AS 'book count'  FROM copy JOIN branch USING (branch\_num)  WHERE branch\_name = 'JM Downtown'  AND copy\_price > 10  AND copy\_price < 20; |  |
| Q8 | SELECT branch\_name,  copy\_num,  copy\_quality AS 'quality',  copy\_price AS 'price'  FROM copy JOIN branch USING (branch\_num)  WHERE book\_code =  (SELECT book\_code  FROM book  WHERE book\_title = 'The Stranger'); |  |
| Q9 | SELECT book\_title as 'title',  count(\*) AS 'count',  concat('$', round(avg(copy\_price), 2)) AS 'average price'  FROM book JOIN copy USING (book\_code)  GROUP BY book\_title  HAVING count(\*) > 4  ORDER BY book\_title; |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Q10 | | SELECT book\_title AS 'title',  author\_first,  author\_last  FROM branch JOIN copy USING (branch\_num)  JOIN book USING (book\_code)  JOIN wrote USING (book\_code)  JOIN author USING (author\_num)  WHERE branch\_name = 'JM on the Hill'  AND copy\_quality = 'Excellent'  ORDER BY wrote\_sequence; | |  |
| Q11 | | CREATE TABLE FictionCopies (  book\_code VARCHAR(5),  book\_title VARCHAR(100),  branch\_num INT(2),  copy\_num INT(2),  copy\_quality VARCHAR(10),  copy\_price DECIMAL(5,2)  );  INSERT INTO FictionCopies  SELECT book\_code, book\_title, branch\_num, copy\_num, copy\_quality,  copy\_price  FROM book JOIN copy USING (book\_code)  WHERE book\_type = 'FIC';  SELECT \*  FROM fictioncopies; | |  |
| Q12 | SELECT book\_title,  branch\_num,  copy\_num,  copy\_quality,  copy\_price AS 'old price',  if(copy\_price < 10.00,  round(copy\_price\*1.1, 2),  ' ') AS 'increased price'  FROM fictioncopies  WHERE copy\_quality = 'Excellent'; | |  | |

|  |  |  |
| --- | --- | --- |
| Q13 | SELECT distinct book\_code,  book\_title AS 'title'  FROM book JOIN copy USING (book\_code)  WHERE publisher\_code =  (SELECT publisher\_code  FROM publisher  WHERE publisher\_name = 'Vintage Books')  OR branch\_num =  (SELECT branch\_num  FROM branch  WHERE branch\_name = 'JM Brentwood')); |  |
| Q14 | SELECT book\_title AS 'title',  publisher\_name,  author\_last,  author\_first  FROM book JOIN publisher USING (publisher\_code)  JOIN wrote USING (book\_code)  JOIN author USING (author\_num)  WHERE (SELECT book\_code  FROM wrote AS wrote2  WHERE wrote2.book\_code = book.book\_code  GROUP BY wrote2.book\_code  HAVING count(\*) = 2); |  |

|  |  |  |
| --- | --- | --- |
| Q15 | If we assume that a book can exist without a corresponding copy existing:  SELECT distinct publisher\_name  FROM publisher JOIN book USING (publisher\_code)  WHERE publisher\_code NOT IN  (SELECT publisher\_code  FROM copy JOIN book USING (book\_code));  If we follow the ERD’s mandatory relationship between *book* and *copy*, then the query becomes the following and the output matches the table supplied in the assignment specifications:  SELECT publisher\_name  FROM publisher  WHERE publisher\_code NOT IN  (SELECT publisher\_code  FROM copy JOIN book USING (book\_code)); |  |
| Q16 | SELECT branch\_name,  concat('$', cast(sum(comp\_cost) AS CHAR))  AS 'total cost of computers'  FROM branch JOIN employee USING (branch\_num)  JOIN hire USING (emp\_num)  JOIN computer USING (comp\_num)  WHERE hire\_end IS NULL  GROUP BY branch\_name; |  |

|  |  |  |
| --- | --- | --- |
| Q17 | SELECT comp\_num,  year(comp\_purchase\_date) AS 'Year Purchased'  FROM computer  WHERE year(comp\_purchase\_date) <= 2008  AND comp\_num NOT IN  (SELECT comp\_num  FROM hire JOIN employee USING (emp\_num)  WHERE (emp\_last = 'Chopping' AND emp\_first = 'Marie')  OR (emp\_last = 'Smith' AND emp\_first = 'Ronald')); |  |
| Q18 | SELECT branch\_name AS 'branch',  count(emp\_num) AS 'num of employees'  FROM branch JOIN employee USING (branch\_num)  GROUP BY branch\_name  ORDER BY 2 desc; |  |
| Q19 | SELECT distinct book\_code, book\_title  FROM book JOIN copy USING (book\_code)  JOIN branch USING (branch\_num)  WHERE branch\_name = 'JM Brentwood'  AND book\_code IN  (SELECT book\_code  FROM copy JOIN branch USING (branch\_num)  WHERE branch\_name = 'JM On the Hill'); |  |

|  |  |  |
| --- | --- | --- |
| Q20 | SELECT distinct publisher\_name  FROM publisher JOIN book USING (publisher\_code)  JOIN copy USING (book\_code)  WHERE branch\_num =  (SELECT max(2)  FROM (SELECT branch\_num, count(distinct book\_code)  FROM copy JOIN branch USING (branch\_num)  GROUP BY branch\_num) AS sub); |  |