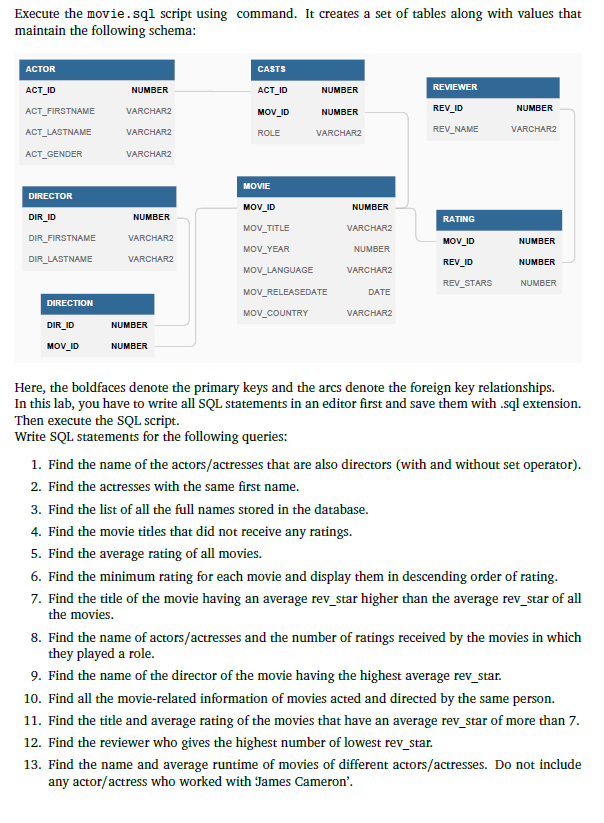
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | |
|  | Report on Lab-04  DATABASE MANAGEMENT SYSTEMS LAB | | | | | |  | |
|  |  | | | | | | |  | | |
| Submitted by  Adid-Al-Mahamud Shazid  Student Id: 210042172  Department: CSE  Programme: SWE  Course Title: CSE 4308  Submitted to  Zannatun Naim Sristy  Lecturer, Department of CSE  September 07, 2023 | |  | | | | |
|  |
|  | | |  |
|  | |  | | | |  | | | |

**Introduction**

In this lab class, we were given tasks based on advanced data manipulation techniques to solve using SQL command line to understand the basics of data definition and data manipulation.

**Task**



**Solution**

1. --TASK 1
2. SELECT ACT\_FIRSTNAME|| ' ' || ACT\_LASTNAME AS NAME FROM ACTOR, DIRECTOR WHERE ACT\_FIRSTNAME=DIR\_FIRSTNAME AND ACT\_LASTNAME=DIR\_LASTNAME;
3. SELECT ACT\_FIRSTNAME || ' ' || ACT\_LASTNAME AS NAME FROM ACTOR INTERSECT SELECT DIR\_FIRSTNAME || ' ' || DIR\_LASTNAME AS NAME FROM DIRECTOR;
4. --TASK 2
5. SELECT ACT\_FIRSTNAME FROM ACTOR WHERE ACT\_GENDER = 'F' GROUP BY ACT\_FIRSTNAME HAVING COUNT(\*)>1;
6. --TASK 3
7. SELECT ACT\_LASTNAME|| ' ' || ACT\_LASTNAME AS NAME FROM ACTOR UNION SELECT DIR\_FIRSTNAME|| ' ' || DIR\_LASTNAME FROM DIRECTOR;
8. --TASK 4
9. SELECT MOV\_TITLE FROM MOVIE WHERE MOV\_ID NOT IN (SELECT MOV\_ID FROM RATING);
10. --TASK 5
11. SELECT AVG(REV\_STARS) FROM RATING;
12. --TASK 6
13. SELECT MOV\_TITLE, MIN(REV\_STARS) AS MIN\_RATE FROM RATING NATURAL JOIN MOVIE GROUP BY MOV\_TITLE ORDER BY MIN\_RATE DESC;
14. --TASK 7
15. SELECT MOV\_TITLE FROM MOVIE WHERE MOV\_ID IN(SELECT MOV\_ID FROM RATING GROUP BY MOV\_ID HAVING AVG(REV\_STARS) > (SELECT AVG(REV\_STARS) FROM RATING));
16. --TASK 8
17. SELECT ACT\_FIRSTNAME, ACT\_LASTNAME, COUNT(\*) AS number\_of\_ratings FROM ACTOR, CASTS, RATING WHERE ACTOR.ACT\_ID = CASTS.ACT\_ID AND CASTS.MOV\_ID = RATING.MOV\_ID
18. GROUP BY ACT\_FIRSTNAME , ACT\_LASTNAME ORDER BY number\_of\_ratings DESC;
19. --TASK 9
20. SELECT DIR\_FIRSTNAME,DIR\_LASTNAME FROM DIRECTOR WHERE DIR\_ID IN (SELECT DIR\_ID FROM MOVIE JOIN RATING USING (MOV\_ID) GROUP BY DIR\_ID
21. HAVING AVG(REV\_STARS) = (SELECT MAX(AVG(REV\_STARS)) FROM MOVIE JOIN RATING USING (MOV\_ID) GROUP BY DIR\_ID));
22. --TASK 10
23. SELECT \*
24. FROM MOVIE
25. WHERE MOV\_ID
26. IN
27. (
28. SELECT DN.MOV\_ID
29. FROM DIRECTION DN
30. WHERE DN.DIR\_ID
31. IN
32. (
33. SELECT D.DIR\_ID
34. FROM DIRECTOR D
35. WHERE D.DIR\_FIRSTNAME || ' ' || D.DIR\_LASTNAME
36. IN
37. (
38. SELECT D1.DIR\_FIRSTNAME || ' ' || D1.DIR\_LASTNAME AS DIR\_NAME1
39. FROM DIRECTOR D1
40. INTERSECT
41. SELECT A.ACT\_FIRSTNAME || ' ' || A.ACT\_LASTNAME AS ACT\_NAME
42. FROM ACTOR A
43. )
44. )
45. );
46. --TASK 11
47. SELECT M.MOV\_TITLE,
48. (
49. SELECT avg(R.REV\_STARS)
50. FROM RATING R
51. WHERE R.MOV\_ID = M.MOV\_ID
52. AND R.REV\_STARS IS NOT NULL
53. ) AS AVG\_RATE
54. FROM MOVIE M
55. WHERE M.MOV\_ID
56. IN
57. (
58. SELECT R1.MOV\_ID
59. FROM RATING R1
60. WHERE R1.REV\_STARS IS NOT NULL
61. GROUP BY R1.MOV\_ID
62. HAVING avg(R1.REV\_STARS) > 7
63. )
64. ORDER BY AVG\_RATE DESC;
65. --TASK 12
66. SELECT R.REV\_NAME
67. FROM REVIEWER R
68. WHERE R.REV\_ID
69. IN
70. (
71. SELECT RT.REV\_ID
72. FROM RATING RT
73. WHERE RT.REV\_STARS =
74. (
75. SELECT min(REV\_STARS)
76. FROM RATING
77. )
78. );
79. --TASK 13
80. SELECT A.ACT\_FIRSTNAME || ' ' ||A.ACT\_LASTNAME AS ACT\_NAME,
81. (
82. SELECT avg(M.MOV\_TIME)
83. FROM MOVIE M
84. WHERE M.MOV\_ID
85. IN
86. (
87. SELECT C.MOV\_ID
88. FROM CASTS C
89. WHERE C.ACT\_ID = A.ACT\_ID
90. )
91. AND M.MOV\_TIME IS NOT NULL
92. ) AS AVG\_RUNTIME
93. FROM ACTOR A
94. WHERE A.ACT\_ID
95. IN
96. (
97. SELECT ACT\_ID FROM CASTS
98. )
99. AND A.ACT\_ID
100. NOT IN
101. (
102. SELECT C2.ACT\_ID
103. FROM CASTS C2
104. WHERE C2.MOV\_ID
105. IN
106. (
107. SELECT DN.MOV\_ID
108. FROM DIRECTION DN
109. WHERE DN.DIR\_ID =
110. (
111. SELECT DR.DIR\_ID
112. FROM DIRECTOR DR
113. WHERE DR.DIR\_FIRSTNAME = 'James' AND DR.DIR\_LASTNAME = 'Cameron'
114. )
115. )
116. )
117. ORDER BY ACT\_NAME;

**Analysis and Explanation**

From this task I learnt to use sub queries properly. Also got to know some new functionalities as well.

**Difficulties**

I faced difficulties while I was trying to use nested queries. It was bit confusing. Thus, it took much time and tries for me to complete the tasks.