Assignment-Data Analysis

Data-LMS data set.

Tables

SELECT \* from LMS\_BOOK\_DETAILS;

SELECT \* from LMS\_BOOK\_ISSUE;

SELECT \* from LMS\_FINE\_DETAILS;

SELECT \* from LMS\_SUPPLIERS\_DETAILS;

SELECT \* from LMS\_MEMBERS;

Q1) Find the percentage contribution of all books , if the book is not issued mark 0.

Ans.

SELECT BD.BOOK\_CODE, BD.BOOK\_TITLE,

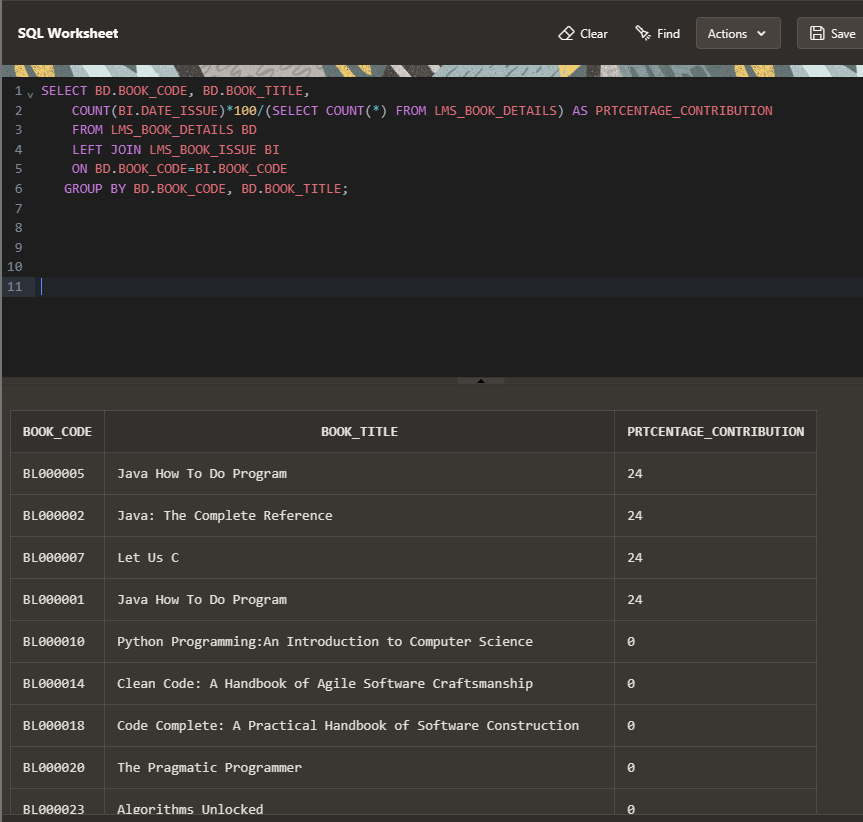
COUNT(BI.DATE\_ISSUE)\*100/(SELECT COUNT(\*) FROM LMS\_BOOK\_DETAILS) AS PRTCENTAGE\_CONTRIBUTION

FROM LMS\_BOOK\_DETAILS BD

LEFT JOIN LMS\_BOOK\_ISSUE BI

ON BD.BOOK\_CODE=BI.BOOK\_CODE

GROUP BY BD.BOOK\_CODE, BD.BOOK\_TITLE;



Explanation:

count the number of times each book has been issued and divide it by the total number of books

subquery is used to get the total number of books.

Q2) Find book name, member name, fine range for those members who paid fine (remove duplicates)

Ans.

SELECT BD.BOOK\_TITLE, M.MEMBER\_NAME, FD.FINE\_RANGE

FROM LMS\_BOOK\_DETAILS BD

LEFT JOIN LMS\_BOOK\_ISSUE BI

ON BD.BOOK\_CODE=BI.BOOK\_CODE

LEFT JOIN LMS\_MEMBERS M

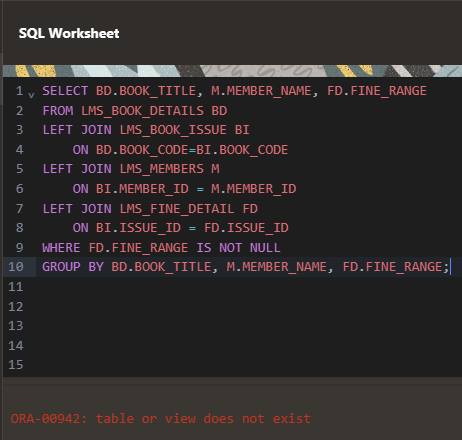
ON BI.MEMBER\_ID = M.MEMBER\_ID

LEFT JOIN LMS\_FINE\_DETAIL FD

ON BI.ISSUE\_ID = FD.ISSUE\_ID

WHERE FD.FINE\_RANGE IS NOT NULL

GROUP BY BD.BOOK\_TITLE, M.MEMBER\_NAME, FD.FINE\_RANGE;



error

Attempt 2.

will join tables

SELECT \* from LMS\_BOOK\_DETAILS; (book code)

SELECT \* from LMS\_BOOK\_ISSUE; (book code, member id)

SELECT \* from LMS\_MEMBERS; (member id)

will use where and not having since 1.Before group by 2.need to filter rows

Ans.

SELECT BD.BOOK\_TITLE, M.MEMBER\_NAME, BI.FINE\_RANGE

FROM lms\_BOOK\_DETAILS BD

LEFT JOIN LMS\_BOOK\_ISSUE BI

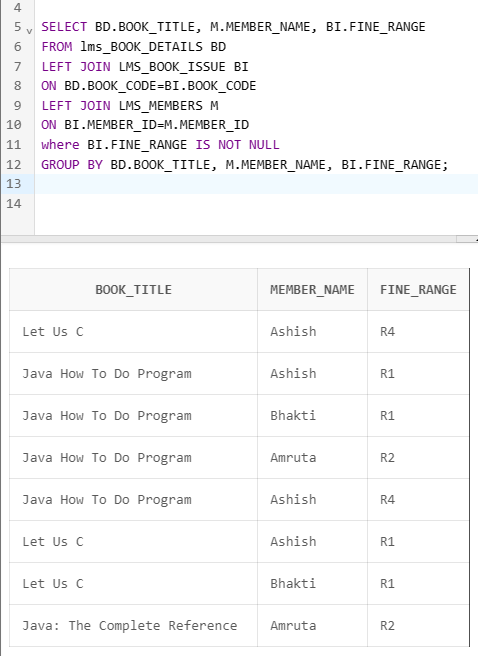
ON BD.BOOK\_CODE=BI.BOOK\_CODE

LEFT JOIN LMS\_MEMBERS M

ON BI.MEMBER\_ID=M.MEMBER\_ID

WHERE BI.FINE\_RANGE IS NOT NULL

GROUP BY BD.BOOK\_TITLE, M.MEMBER\_NAME, BI.FINE\_RANGE;



Q3) Create bucket based on fine paid or not.

Ans.

Method 1.

SELECT MEMBER\_ID, DATE\_RETURN,

CASE WHEN FINE\_RANGE IS NULL THEN 'NOT PAID'

ELSE 'PAID'

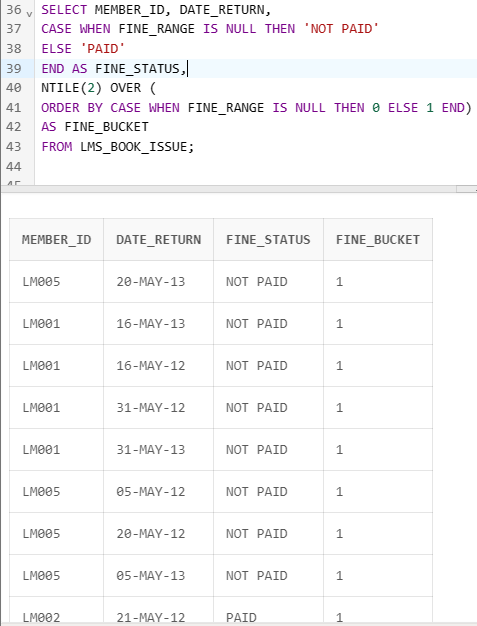
END AS FINE\_STATUS,

NTILE(2) OVER (

ORDER BY CASE WHEN FINE\_RANGE IS NULL THEN 0 ELSE 1 END)

AS FINE\_BUCKET

FROM LMS\_BOOK\_ISSUE;



Method 2.

COALESCE function to replace NULL values with "NO".

SELECT MEMBER\_ID, DATE\_RETURN,

COALESCE(CASE WHEN FINE\_RANGE IS NOT NULL THEN 'YES' END, 'NO') AS FINE\_PAID

FROM LMS\_BOOK\_ISSUE

ORDER BY FINE\_RANGE;



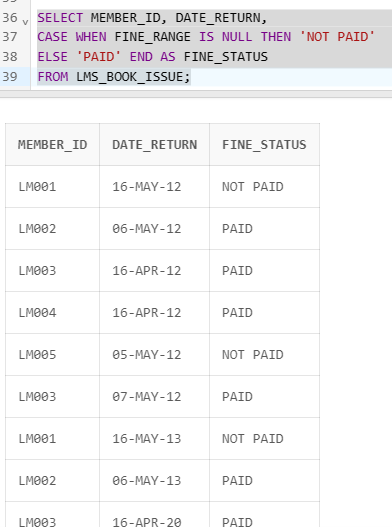
Method3.

SELECT MEMBER\_ID, DATE\_RETURN,

CASE WHEN FINE\_RANGE IS NULL THEN 'NOT PAID'

ELSE 'PAID' END AS FINE\_STATUS

FROM LMS\_BOOK\_ISSUE;



Q4) Find running total based on category of book issued

Ans-

Attempt 1.

SELECT BD.CATEGORY, BD.BOOK\_TITLE, COUNT(BI.DATE\_ISSUE) AS CATEGORY\_ISSUE\_COUNT

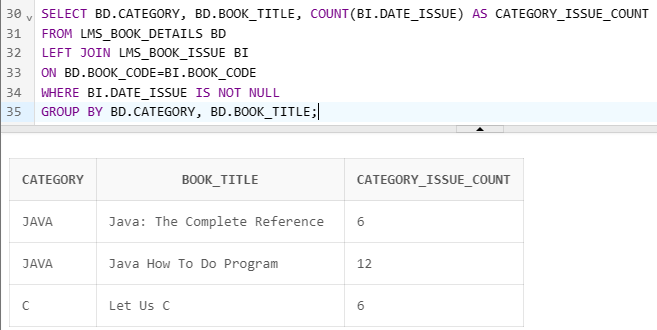
FROM LMS\_BOOK\_DETAILS BD

LEFT JOIN LMS\_BOOK\_ISSUE BI

ON BD.BOOK\_CODE=BI.BOOK\_CODE

WHERE BI.DATE\_ISSUE IS NOT NULL

GROUP BY BD.CATEGORY, BD.BOOK\_TITLE;



Attempt 2.

SELECT

CATEGORY,

BOOK\_TITLE,

COUNT(DATE\_ISSUE) AS CATEGORY\_ISSUE\_COUNT,

SUM(COUNT(DATE\_ISSUE)) OVER (PARTITION BY CATEGORY ORDER BY BOOK\_TITLE) AS RUNNING\_TOTAL

FROM

LMS\_BOOK\_DETAILS

LEFT JOIN LMS\_BOOK\_ISSUE ON LMS\_BOOK\_DETAILS.BOOK\_CODE = LMS\_BOOK\_ISSUE.BOOK\_CODE

WHERE

DATE\_ISSUE IS NOT NULL

GROUP BY

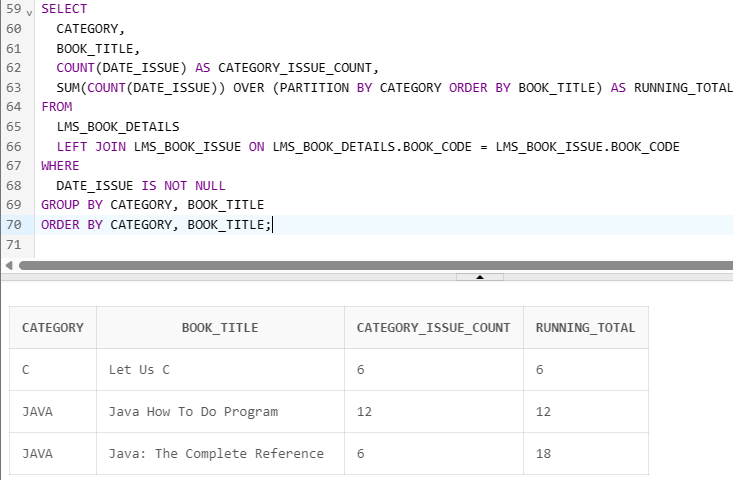
CATEGORY,

BOOK\_TITLE

ORDER BY

CATEGORY,

BOOK\_TITLE;



Q5) Find running total for fine paid

Attempt1.

SELECT

bi.BOOK\_CODE,

fd.FINE\_RANGE,

COUNT(fd.FINE\_RANGE) as FINE\_COUNT,

SUM(COUNT(fd.FINE\_RANGE)) OVER (PARTITION BY bi.BOOK\_CODE ORDER BY fd.FINE\_AMOUNT) AS RUNNING\_AMOUNT

FROM

LMS\_BOOK\_ISSUE BI

LEFT JOIN LMS\_FINE\_DETAILS FD ON BI.FINE\_RANGE=FD.FINE\_RANGE

WHERE BI.FINE\_RANGE IS NOT NULL

GROUP BY bi.BOOK\_CODE, fd.FINE\_RANGE;

