Assignment-Hierarchical Queries (15-5-23)

LMS Dataset SQL File

Tables

SELECT \* FROM LMS\_MEMBERS;

SELECT \* FROM LMS\_BOOK\_details;

SELECT \* FROM LMS\_BOOK\_ISSUE;

SELECT \* FROM LMS\_FINE\_DETAILS;

SELECT \* FROM LMS\_SUPPLIERS\_DETAILS;

Q1) Fetch last 5 rows from LMS\_Members Table, order by their

Registration.

Ans.

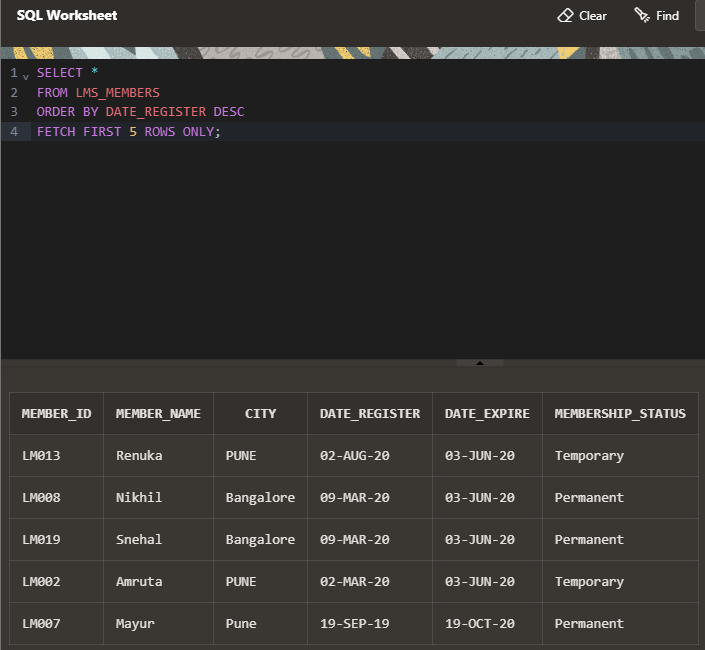
METHOD 1.

SELECT \*

FROM LMS\_MEMBERS

ORDER BY DATE\_REGISTER DESC

FETCH FIRST 5 ROWS ONLY;



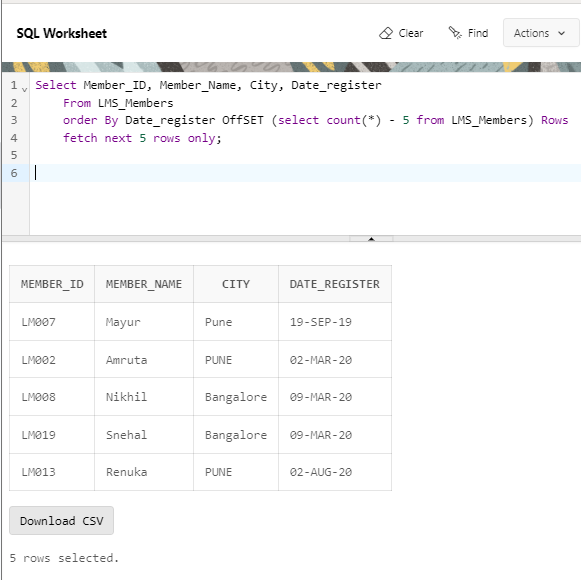
METHOD 2.

Select Member\_ID, Member\_Name, City, Date\_register

From LMS\_Members

order By Date\_register OffSET (select count(\*) - 5 from LMS\_Members) Rows

fetch next 5 rows only;



Q2) Write a query to display Member name, Book Title, Issue and

return date then show the hierarchy of tables (Grandparent -

parent-child relationship)

Ans.

Tables used

LMS members - member name (Member\_id) = CHILD

LMS\_BOOK\_ISSUE - issue & return date (Member\_id)(BOOK\_CODE) = PARENT

LMS\_BOOK\_details - BOOK TITLE (BOOK\_CODE) =GRAND PARENT

ATTEMPT 1:

SELECT M. MEMBER\_NAME, BD.BOOK\_TITLE, BI.DATE\_ISSUE, BI.DATE\_RETURN

FROM LMS\_MEMBERS M

LEFT JOIN LMS\_BOOK\_ISSUE BI

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

LEFT JOIN LMS\_BOOK\_DETAILS BD

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

NOTE-

JUST JOINED NO RELATION

ATTEMPT 2:

METHOD 1.

SELECT M.MEMBER\_NAME, BD.BOOK\_TITLE, BI.DATE\_ISSUE, BI.DATE\_RETURN, 'LMS\_BOOK\_DETAILS' AS Hierarchy

FROM LMS\_BOOK\_DETAILS BD

JOIN LMS\_BOOK\_ISSUE BI ON BD.BOOK\_CODE = BI.BOOK\_CODE

JOIN LMS\_MEMBERS M ON BI.MEMBER\_ID = M.MEMBER\_ID

UNION ALL

SELECT M.MEMBER\_NAME, BD.BOOK\_TITLE, BI.DATE\_ISSUE, BI.DATE\_RETURN, 'LMS\_BOOK\_ISSUE' AS Hierarchy

FROM LMS\_BOOK\_ISSUE BI

JOIN LMS\_MEMBERS M ON BI.MEMBER\_ID = M.MEMBER\_ID

JOIN LMS\_BOOK\_DETAILS BD ON BI.BOOK\_CODE = BD.BOOK\_CODE

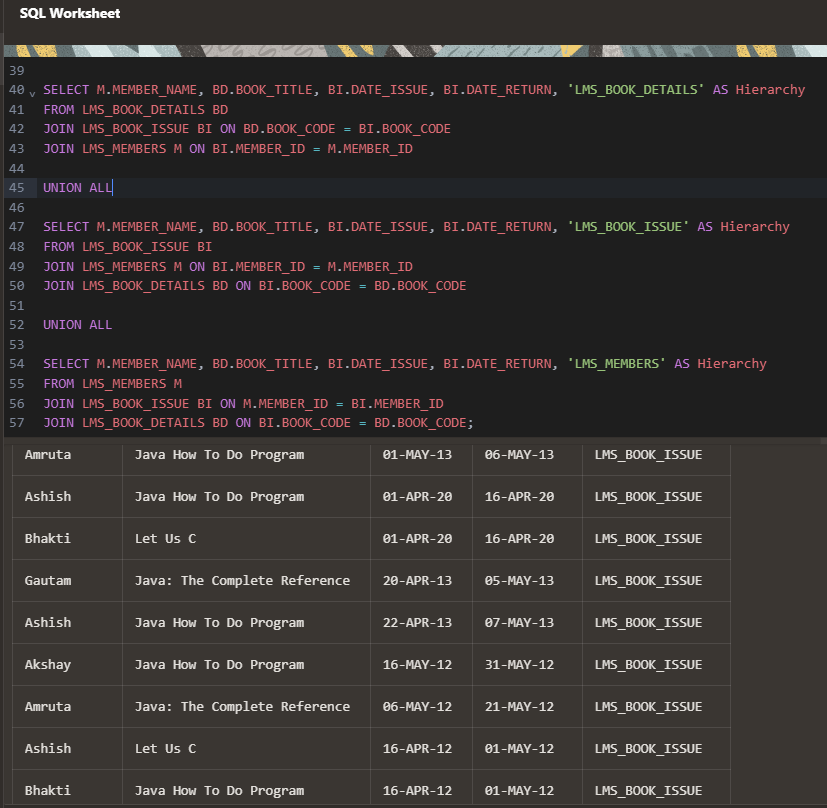
UNION ALL

SELECT M.MEMBER\_NAME, BD.BOOK\_TITLE, BI.DATE\_ISSUE, BI.DATE\_RETURN, 'LMS\_MEMBERS' AS Hierarchy

FROM LMS\_MEMBERS M

JOIN LMS\_BOOK\_ISSUE BI ON M.MEMBER\_ID = BI.MEMBER\_ID

JOIN LMS\_BOOK\_DETAILS BD ON BI.BOOK\_CODE = BD.BOOK\_CODE;



METHOD 2.

SELECT lms\_members.MEMBER\_NAME,

lms\_Book\_Details.BOOK\_TITLE,

lms\_book\_issue.DATE\_ISSUE,

lms\_book\_issue.DATE\_RETURN

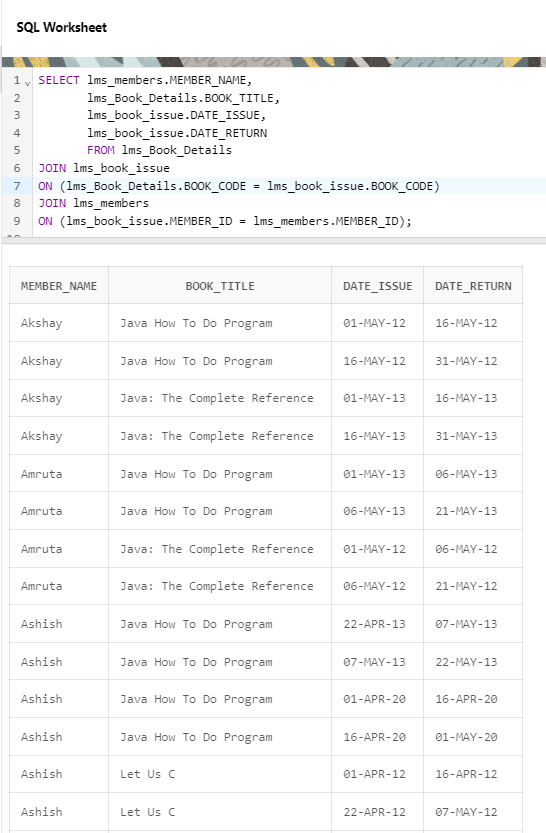
FROM lms\_Book\_Details

JOIN lms\_book\_issue

ON (lms\_Book\_Details.BOOK\_CODE = lms\_book\_issue.BOOK\_CODE)

JOIN lms\_members

ON (lms\_book\_issue.MEMBER\_ID = lms\_members.MEMBER\_ID);



Q3) Write a query to compare the price of different books

according to their book editions. (using Lead/Lag function)

Ans.

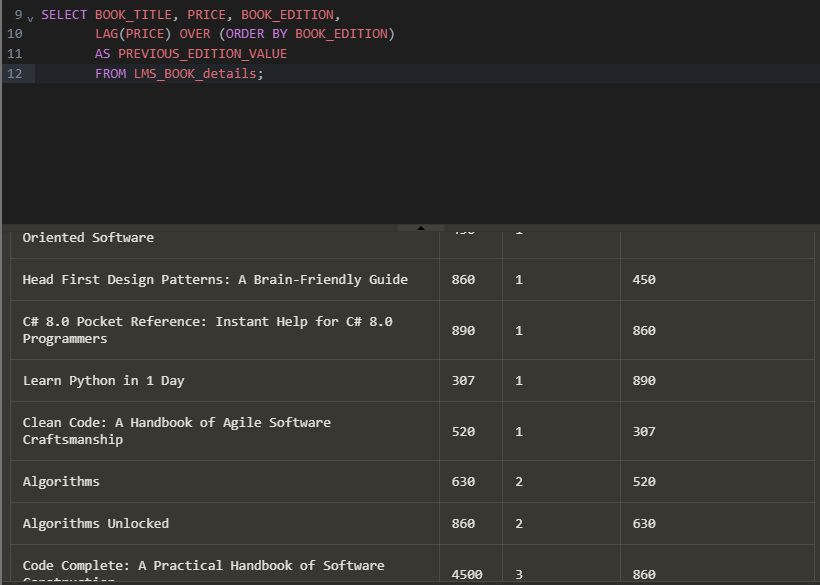
Attempt 1.

SELECT BOOK\_TITLE, PRICE, BOOK\_EDITION,

LAG(PRICE) OVER (ORDER BY BOOK\_EDITION)

AS PREVIOUS\_EDITION\_VALUE

FROM LMS\_BOOK\_details;



Attempt 2.

SELECT

BOOK\_TITLE,

PRICE,

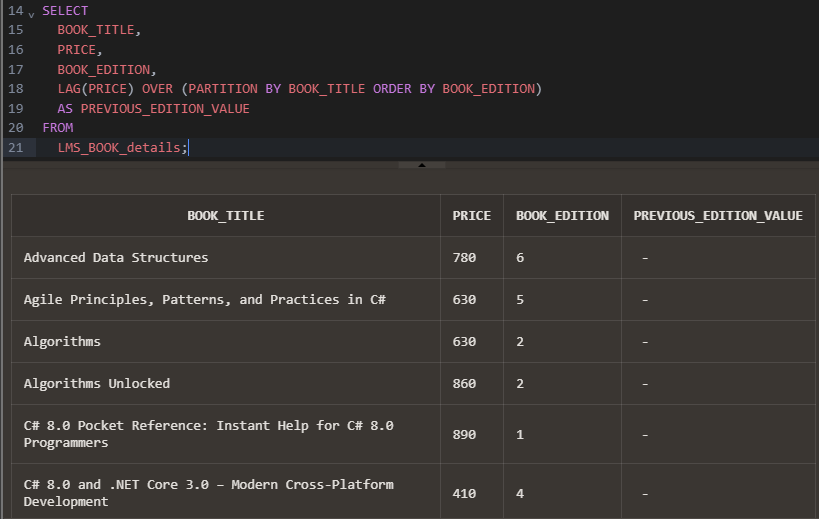
BOOK\_EDITION,

LAG(PRICE) OVER (PARTITION BY BOOK\_TITLE ORDER BY BOOK\_EDITION)

AS PREVIOUS\_EDITION\_VALUE

FROM

LMS\_BOOK\_details;



Attempt 3.

SELECT

BOOK\_TITLE,

BOOK\_EDITION,

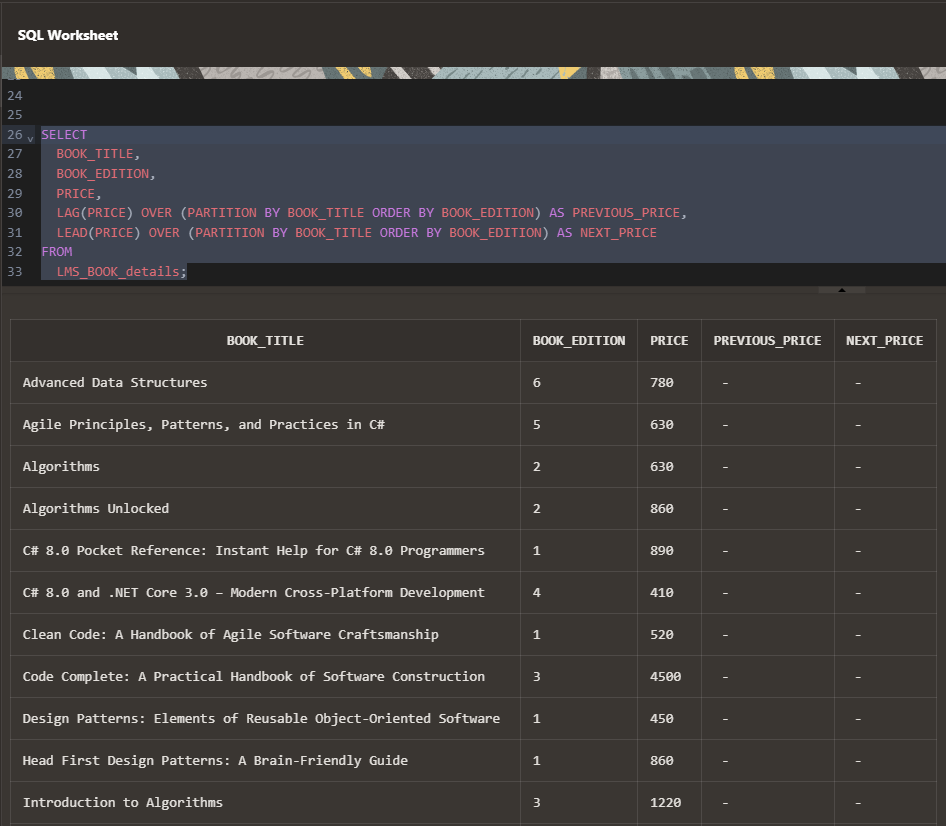
PRICE,

LAG(PRICE) OVER (PARTITION BY BOOK\_TITLE ORDER BY BOOK\_EDITION) AS PREVIOUS\_PRICE,

LEAD(PRICE) OVER (PARTITION BY BOOK\_TITLE ORDER BY BOOK\_EDITION) AS NEXT\_PRICE

FROM

LMS\_BOOK\_details;



LAG-to fetch the price of the previous book edition within the same book title group

lEAD-fetches the price of the next book edition within the same book title group

PARTITION BY BOOK\_TITLE-ensures that the functions operate within each book title group

ORDER BY BOOK\_EDITION-specifies the order

Actual Ans-

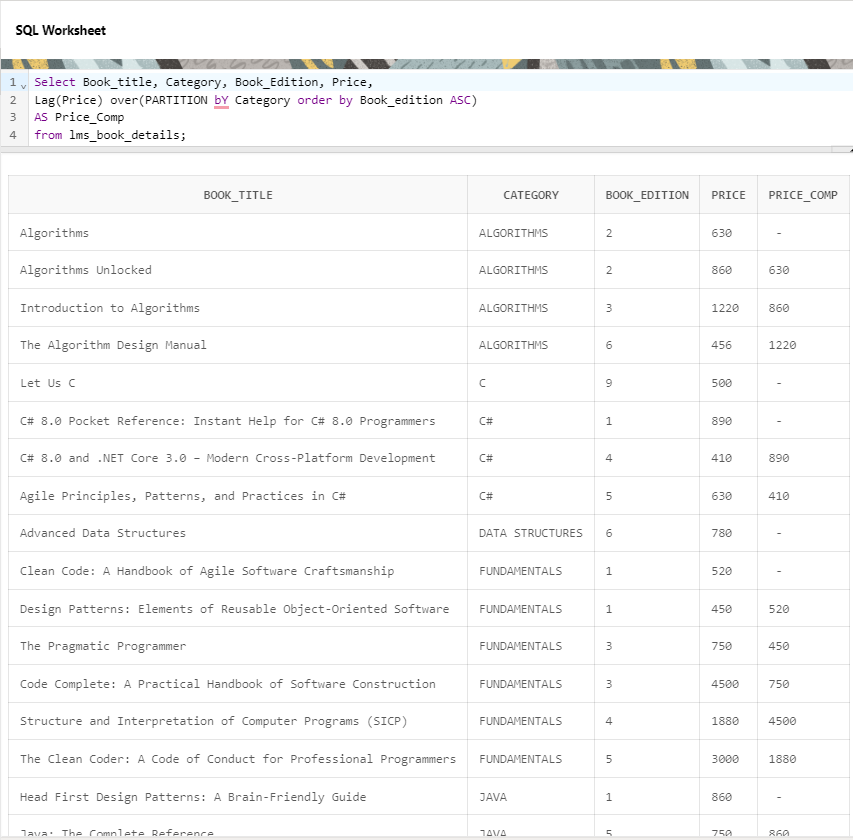
SELECT \* FROM LMS\_BOOK\_DETAILS;

Select Book\_title, Category, Book\_Edition, Price,

Lag(Price) over(PARTITION bY Category order by Book\_edition ASC)

AS Price\_Comp

from lms\_book\_details;



Q4) List down the books according to Rack number except A1 and

A2.

Ans.

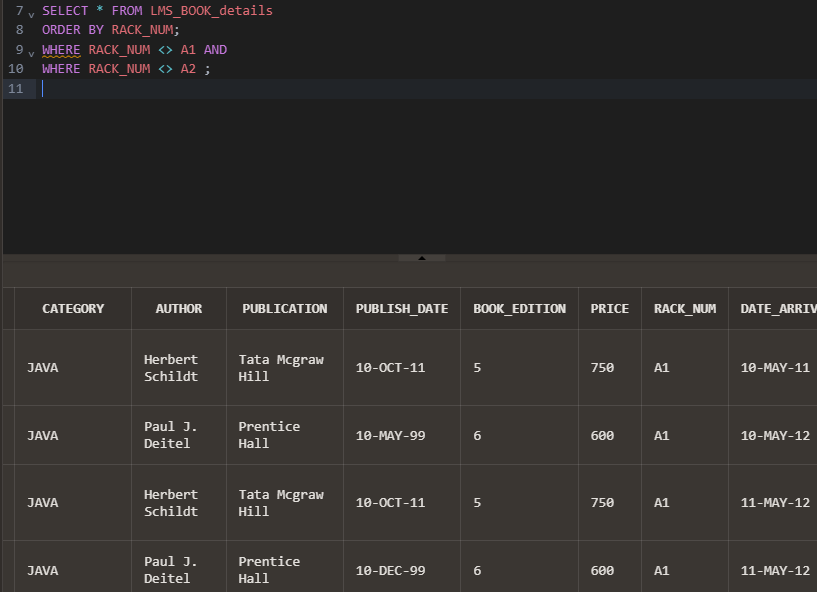
Attempt 1.

SELECT \* FROM LMS\_BOOK\_details

ORDER BY RACK\_NUM;

WHERE RACK\_NUM <> A1 AND

WHERE RACK\_NUM <> A2 ;



Attempt 2.

THE ORDER BY WILL COME AFTER WHERE SINCE FIRST DATA IS FILTERED (; RACK\_NO <> A1, A2)AND THEN ITS PUT IN ORDER.

SELECT \*

FROM LMS\_BOOK\_details

WHERE RACK\_NUM <> 'A1' AND

WHERE RACK\_NUM <> 'A2'

ORDER BY RACK\_NUM;

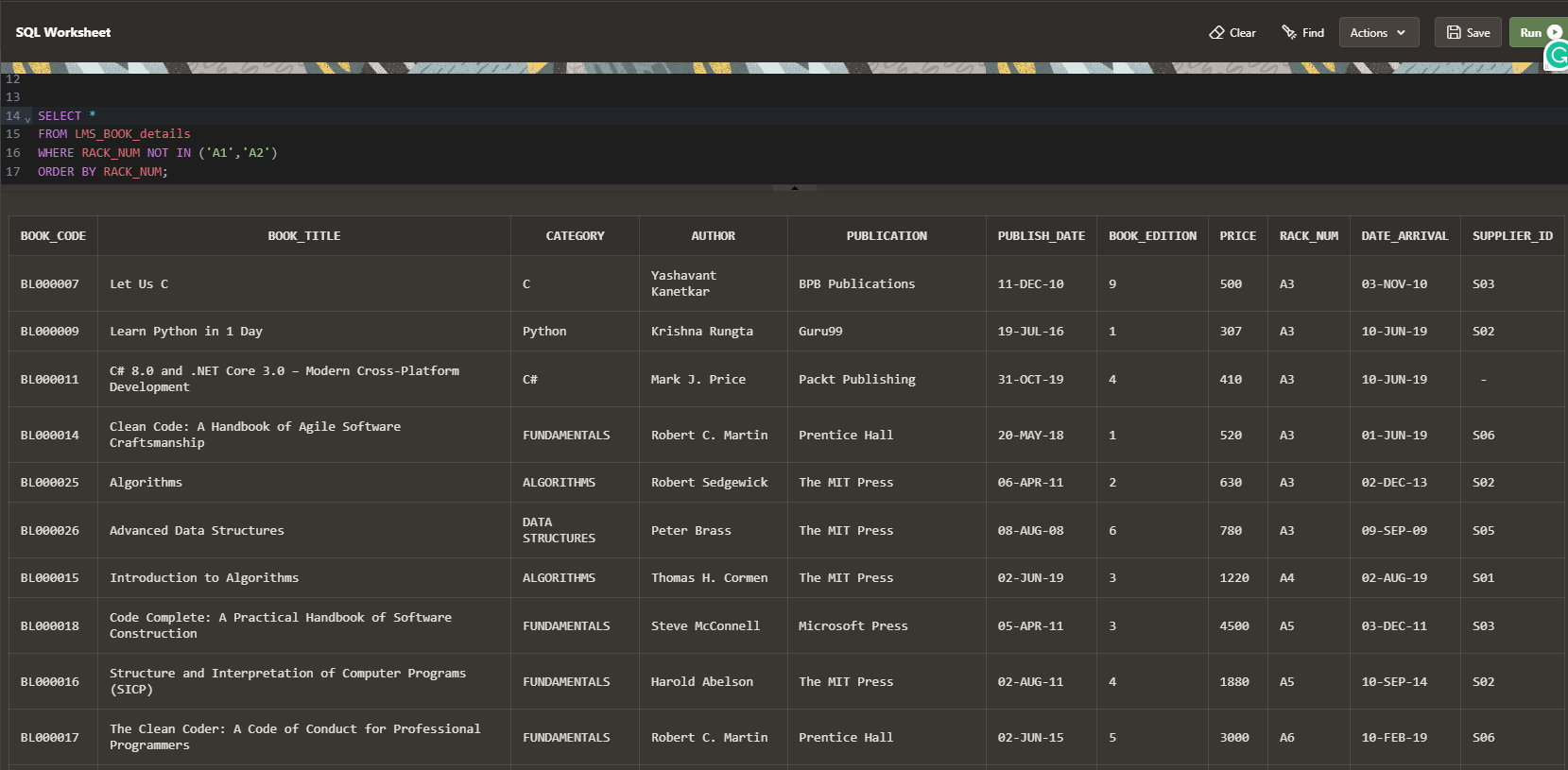
OR

SELECT \*

FROM LMS\_BOOK\_details

WHERE RACK\_NUM NOT IN ('A1','A2')

ORDER BY RACK\_NUM;



ACTUAL ANS-

Select Book\_Title, Rack\_Num from LMS\_Book\_Details

Order By Rack\_Num

Offset (Select Count(Rack\_Num)

from LMS\_Book\_details

where Rack\_Num = 'A1' or Rack\_Num = 'A2') Rows;

