**#PROBLEM 1**

select B.MEMBER\_ID,M.MEMBER\_NAME,M.CITY,M.MEMBERSHIP\_STATUS from LMS\_BOOK\_ISSUE B

JOIN LMS\_MEMBERS M ON

B.MEMBER\_ID=M.MEMBER\_ID

WHERE

M.MEMBERSHIP\_STATUS="Permanent";

**#PROBLEM 2**

**#Write a query to display the book code, publication, price and supplier name of the book witch is taken frequently.**

SELECT BD.BOOK\_CODE,BD.PUBLICATION,BD.PRICE,SD.SUPPLIER\_NAME FROM LMS\_BOOK\_ISSUE BI

JOIN

LMS\_BOOK\_DETAILS BD

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

JOIN LMS\_SUPPLIERS\_DETAILS SD ON BD.SUPPLIER\_ID=SD.SUPPLIER\_ID

GROUP BY

BD.BOOK\_CODE,BD.PUBLICATION,BD.PRICE,SD.SUPPLIER\_NAME

ORDER BY COUNT(BI.BOOK\_ISSUE\_NO) DESC

LIMIT 1

**#Problem 3**

**#Write a query to display the member id, member name who have taken the book with book code 'BL000002**

SELECT M.MEMBER\_ID, M.MEMBER\_NAME

FROM LMS\_MEMBERS M

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

WHERE BI.BOOK\_CODE = 'BL000002';

**#Problem 4**

**# 4: Write a query to display the book code, book title and author of the books whose author name begins with 'P'**

select BOOK\_CODE,BOOK\_TITLE,AUTHOR FROM LMS\_BOOK\_DETAILS

WHERE AUTHOR LIKE 'P%';

**#Problem 5**

**#Write a query to display the total number of Java books available in library with alias name ‘NO\_OF\_BOOKS’.**

select count( Distinct BOOK\_CODE) as no\_of\_books

from LMS\_BOOK\_ISSUE

**#PROBLEM 6**

**#Problem # 6: Write a query to list the category and number of books in each category with alias name ‘NO\_OF\_BOOKS’.**

SELECT BD.CATEGORY, COUNT(BD.BOOK\_CODE) AS NO\_OF\_BOOKS

FROM LMS\_BOOK\_DETAILS BD

GROUP BY BD.CATEGORY;

**#Problem 7**

**# Write a query to display the number of books published by "Prentice Hall” with the alias name “NO\_OF\_BOOKS”.**

select PUBLICATION,count(BOOK\_CODE) FROM LMS\_BOOK\_DETAILS

WHERE PUBLICATION="Prentice Hall"

**#Problem 8**

**# Problem # 8: Write a query to display the book code, book title of the books which are issued on the date "1st April 2012".**

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

FROM LMS\_BOOK\_ISSUE BI

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

WHERE BI.DATE\_ISSUE = '2012-04-01';

**#Problem 9**

**#Problem # 9: Write a query to display the member id, member name, date of registration and expiry date of the members whose membership expiry date is before APR 2013.**

select M.MEMBER\_ID,M.MEMBER\_NAME,M.DATE\_REGISTER,M.DATE\_EXPIRE FROM LMS\_MEMBERS M

WHERE M.DATE\_EXPIRE< '2013-04-01';

**#Problem 10**

**#Problem # 10: write a query to display the member id, member name, date of registration, membership status of the members who registered before "March 2012" and membership status is "Temporary"**

SELECT M.MEMBER\_ID, M.MEMBER\_NAME, M.DATE\_REGISTER, M.MEMBERSHIP\_STATUS

FROM LMS\_MEMBERS M

WHERE M.DATE\_REGISTER < '2012-03-01' AND M.MEMBERSHIP\_STATUS = 'Temporary';

**#Problem 11**

**#Problem #11: Write a query to display the member id, member name who’s City is CHENNAI or DELHI. Hint: Display the member name in title case with alias name 'Name'.**

select M.CITY,M.MEMBER\_ID,M.MEMBER\_NAME FROM LMS\_MEMBERS M

WHERE M.CITY='CHENNAI' OR M.CITY='DELHI';

**#Problem 12**

**Write a query to concatenate book title, author and display in the following format. Book\_Title\_is\_written\_by\_Author Example: Let Us C\_is\_written\_by\_Yashavant Kanetkar Hint: display unique books. Use “BOOK\_WRITTEN\_BY” as alias nam**

SELECT DISTINCT CONCAT(BD.BOOK\_TITLE, '\_is\_written\_by\_', BD.AUTHOR) AS BOOK\_WRITTEN\_BY

FROM LMS\_BOOK\_DETAILS BD;

**#Program13**

**#Write a query to display the average price of books which is belonging to ‘JAVA’ category with alias name “AVERAGEPRICE”**

select avg(BD.PRICE) AS AVERAGEPRICE from LMS\_BOOK\_DETAILS BD

WHERE BD.CATEGORY='JAVA';

**#Program14**

**#Write a query to display the supplier id, supplier name and email of the suppliers who are all having gmail account.**

select SD.SUPPLIER\_ID,SD.SUPPLIER\_NAME,SD.EMAIL FROM LMS\_SUPPLIERS\_DETAILS SD

where SD.EMAIL LIKE '%@gmail.com';

**#Program15**

**# Problem#15: Write a query to display the supplier id, supplier name and contact details. Contact details can be either phone number or email or address with alias name “CONTACTDETAILS”. If phone number is null then display email, even if email also null then display the address of the supplier. Hint: Use Coalesce function**.

SELECT SD.SUPPLIER\_ID,

SD.SUPPLIER\_NAME,

COALESCE(SD.CONTACT, SD.EMAIL, SD.ADDRESS) AS CONTACTDETAILS

from (SELECT SUPPLIER\_ID, SUPPLIER\_NAME,

CONTACT, EMAIL, ADDRESS

FROM LMS\_SUPPLIERS\_DETAILS

) AS SD;

**#Program16**

**#Problem#16: Write a query to display the supplier id, supplier name and contact. If phone number is null then display ‘No’ else display ‘Yes’ with alias name “PHONENUMAVAILABLE”. Hint: Use ISNULL.**

SELECT SD.SUPPLIER\_ID,

SD.SUPPLIER\_NAME,

ISNULL(CASE WHEN SD.CONTACT IS NULL THEN 'No' ELSE 'Yes' END, 'No') AS PHONENUMAVAILABLE

FROM LMS\_SUPPLIERS\_DETAILS SD;

**#Program 17**

**#Problem#17: Write a query to display the member id, member name, city and member status of members with the total fine paid by them with alias name “Fine”.**

select M.MEMBER\_ID,M.MEMBER\_NAME,M.CITY,M.MEMBERSHIP\_STATUS,SUM(FD.FINE\_AMOUNT) AS FINE FROM

LMS\_MEMBERS M

JOIN

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

JOIN

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

GROUP BY M.MEMBER\_ID ,M.MEMBER\_NAME,M.CITY,M.MEMBERSHIP\_STATUS

**#Program 1 avg**

**#Write a query to display the member id, member name of the members, book code and book title of the books**

**#taken by them.**

SELECT M.MEMBER\_ID,

M.MEMBER\_NAME,

BD.BOOK\_CODE,

BD.BOOK\_TITLE

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;

**#Program 2**

SELECT M.MEMBER\_ID,

M.MEMBER\_NAME,

BD.BOOK\_CODE,

BD.BOOK\_TITLE

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;

**#Program3**

#Write a query to display the member id, member name, fine range and fine amount of the members whose fine amount is less than 10

SELECT M.MEMBER\_ID,

M.MEMBER\_NAME,

FD.FINE\_RANGE,

FD.FINE\_AMOUNT

FROM LMS\_MEMBERS M

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

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

WHERE FD.FINE\_AMOUNT < 100;

**Problem 4**

**Write a query to display the book code, book title, publisher, edition, price and year of publication and sort based on year of publication, publisher and edition.**

SELECT

BOOK\_CODE,

BOOK\_TITLE,

PUBLICATION AS PUBLISHER,

BOOK\_EDITION AS EDITION,

PRICE,

EXTRACT(YEAR FROM PUBLISH\_DATE) AS YEAR\_OF\_PUBLICATION

FROM

LMS\_BOOK\_DETAILS

ORDER BY

YEAR\_OF\_PUBLICATION ASC,

PUBLICATION ASC,

BOOK\_EDITION ASC;

**#Problem 5**

**Write a query to display the book code, book title and rack number of the books which are placed in rack 'A1' and sort by book title in ascending order.**

SELECT

BD.BOOK\_CODE,

BD.BOOK\_TITLE,

BD.RACK\_NUM

FROM

LMS\_BOOK\_DETAILS BD

WHERE

BD.RACK\_NUM = 'A1'

ORDER BY BD.BOOK\_TITLE ASC;

**Problem # 6:**

**Write a query to display the member id, member name, due date and date returned of the members who has returned the books after the due date. Hint: Date\_return is due date and Date\_returned is actual book return date.**

SELECT

M.MEMBER\_ID,

M.MEMBER\_NAME,

BI.DATE\_RETURN AS DUE\_DATE,

BI.DATE\_RETURNED AS ACTUAL\_RETURN\_DATE

FROM

LMS\_BOOK\_ISSUE BI

JOIN

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

WHERE

BI.DATE\_RETURNED > BI.DATE\_RETURN;

**Problem # 7: Write a query to display the member id, member name and date of registration who have not taken any book**

select M.MEMBER\_ID,MEMBER\_NAME,M.DATE\_REGISTER FROM LMS\_MEMBERS M

WHERE M.MEMBER\_ID NOT IN (SELECT MEMBER\_ID FROM LMS\_BOOK\_ISSUE);

**Problem # 8: Write a Query to display the member id and member name of the members who has not paid any fine in the year 2012.**

SELECT MEMBER\_ID, MEMBER\_NAME

FROM LMS\_MEMBERS

WHERE MEMBER\_ID NOT IN (

SELECT DISTINCT MEMBER\_ID

FROM LMS\_BOOK\_ISSUE

WHERE DATE\_ISSUE BETWEEN '2012-01-01' AND '2012-12-31'

);

**Problem # 9: Write a query to display the date on which the maximum numbers of books were issued and the number of books issued with alias name “NOOFBOOKS**

SELECT

DATE\_ISSUE AS ISSUE\_DATE,

COUNT(\*) AS NOOFBOOKS

FROM

LMS\_BOOK\_ISSUE

GROUP BY

DATE\_ISSUE

ORDER BY

NOOFBOOKS DESC

FETCH FIRST 1 ROW ONLY;

**Problem # 10: Write a query to list the book title and supplier id for the books authored by “Herbert Schildt" and the book edition is 5 and supplied by supplier ‘S01’.**

SELECT

BD.BOOK\_TITLE,

BD.SUPPLIER\_ID

FROM

LMS\_BOOK\_DETAILS BD

WHERE

BD.AUTHOR = 'Herbert Schildt'

AND BD.BOOK\_EDITION = 5

AND BD.SUPPLIER\_ID = 'S01';

**Problem # 11: Write a query to display the rack number and the number of books in each rack with alias name “NOOFBOOKS” and sort by rack number in ascending order.**

SELECT

RACK\_NUM,

COUNT(\*) AS NOOFBOOKS

FROM

LMS\_BOOK\_DETAILS

GROUP BY

RACK\_NUM

ORDER BY

RACK\_NUM ASC;

**Problem # 12: Write a query to display book issue number, member name, date or registration, date of expiry, book title, category author, price, date of issue, date of return, actual returned date, issue status, fine amount.**

SELECT

BI.BOOK\_ISSUE\_NO,

M.MEMBER\_NAME,

M.DATE\_REGISTER,

M.DATE\_EXPIRE,

BD.BOOK\_TITLE,

BD.CATEGORY,

BD.AUTHOR,

BD.PRICE,

BI.DATE\_ISSUE,

BI.DATE\_RETURN,

BI.DATE\_RETURNED,

CASE

WHEN BI.DATE\_RETURNED > BI.DATE\_RETURN THEN 'Late'

ELSE 'On Time'

END AS ISSUE\_STATUS,

FD.FINE\_AMOUNT

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

LEFT JOIN

LMS\_FINE\_DETAILS FD ON BI.FINE\_RANGE = FD.FINE\_RANGE;

**Problem # 13: Write a query to display the book code, title, publish date of the books which is been published in the month of December.**

SELECT BOOK\_CODE, BOOK\_TITLE, PUBLISH\_DATE

FROM LMS\_BOOK\_DETAILS

WHERE MONTH(PUBLISH\_DATE) = 12;

**Problem # 14: Write a query to display the book code, book title ,supplier name and price of the book witch takes maximum price based on each supplier.**

SELECT BOOK\_CODE, BOOK\_TITLE, SUPPLIER\_NAME, PRICE

FROM LMS\_BOOK\_DETAILS

JOIN LMS\_SUPPLIERS\_DETAILS ON LMS\_BOOK\_DETAILS.SUPPLIER\_ID = LMS\_SUPPLIERS\_DETAILS.SUPPLIER\_ID

WHERE PRICE = (

SELECT MAX(PRICE)

FROM LMS\_BOOK\_DETAILS

WHERE SUPPLIER\_ID = LMS\_BOOK\_DETAILS.SUPPLIER\_ID

**);**

**Problem # 15: Write a query to display book code, book name, and publisher, how old the book is. Sorted as older to newer**

SELECT BOOK\_CODE, BOOK\_TITLE, PUBLICATION, YEAR(CURRENT\_DATE) - YEAR(PUBLISH\_DATE) AS BOOK\_AGE

FROM LMS\_BOOK\_DETAILS

ORDER BY BOOK\_AGE DESC;

**COMPLEX**

**Write a query to display the book code, book title and supplier name of the supplier who has supplied**

**maximum number of books. For example, if “ABC Store” supplied 3 books, “LM Store” has supplied 2 books and “XYZ Store” has supplied 1 book. So “ABC Store” has supplied maximum number of books, hence display the details as mentioned below. Example: BOOK\_CODE BOOK\_TITLE SUPPLIER\_NAME BL000008 Easy Reference for Java ABC STORE BL000001 Easy Reference for C ABC STORE BL000003 Easy Reference for VB ABC STORE**

SELECT b.BOOK\_CODE, b.BOOK\_TITLE, s.SUPPLIER\_NAME

FROM LMS\_BOOK\_DETAILS b

JOIN LMS\_SUPPLIERS\_DETAILS s ON b.SUPPLIER\_ID = s.SUPPLIER\_ID

WHERE b.SUPPLIER\_ID = (

SELECT TOP 1 SUPPLIER\_ID

FROM LMS\_BOOK\_DETAILS

GROUP BY SUPPLIER\_ID

ORDER BY COUNT(\*) DESC

);

**Problem # 2: Write a query to display the member id, member name and number of remaining books he/she can take with “REMAININGBOOKS” as alias name. Hint: Assuming a member can take maximum 3 books. For example, Ramesh has already taken 2 books; he can take only one book now. Hence display the remaining books as 1 in below format. Example: MEMBER\_ID MEMBER\_NAME REMAININGBOOKS LM001 RAMESH 1 LM002 MOHAN 3**

SELECT m.MEMBER\_ID, m.MEMBER\_NAME,

3 - COUNT(bi.BOOK\_ISSUE\_NO) AS REMAININGBOOKS

FROM LMS\_MEMBERS m

LEFT JOIN LMS\_BOOK\_ISSUE bi ON m.MEMBER\_ID = bi.MEMBER\_ID

GROUP BY m.MEMBER\_ID

HAVING REMAININGBOOKS >= 0;

3

SELECT

SUPPLIER\_ID,

SUPPLIER\_NAME

FROM

LMS\_SUPPLIERS\_DETAILS

WHERE

SUPPLIER\_ID = (

SELECT SUPPLIER\_ID

FROM LMS\_BOOK\_DETAILS

GROUP BY SUPPLIER\_ID

ORDER BY COUNT(\*) ASC

FETCH FIRST 1 ROWS ONLY

);