**Simple Questions:**

**Problem # 1:**

Write a query to display the member id, member name, city and membership status who are all having life time membership. Hint: Life time membership status is “Permanent”.

**Problem # 2:**

Write a query to display the member id, member name who have not returned the books. Hint: Book return status is book\_issue\_status ='Y' or 'N'.

**Problem # 3:**

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

**Problem # 4:**

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

**Problem # 5:**

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

**Problem # 6:**

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

**Problem # 7:**

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

**Problem # 8:**

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

**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.

**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"

**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'.

**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 name.

**Problem #13:**

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

**Problem #14:**

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

**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.

**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 NVL2.

**Average Questions:**

**Problem # 1:**

Write a query to display the member id, member name of the members, book code and book title of the books taken by them.

**Problem # 2:**

Write a query to display the total number of books available in the library with alias name “NO\_OF\_BOOKS\_AVAILABLE” (Which is not issued). Hint: The issued books details are available in the LMS\_BOOK\_ISSUE table.

**Problem # 3:**

Write a query to display the member id, member name, fine range and fine amount of the members whose fine amount is less than 100.

**Problem # 4:**

Write a query to display the book code, book title and availability status of the ‘JAVA’ books whose edition is "6”. Show the availability status with alias name “AVAILABILITYSTATUS”. Hint: Book availability status can be fetched from “BOOK\_ISSUE\_STATUS” column of LMS\_BOOK\_ISSUE table.

**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.

**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.

**Problem # 7:**

Write a query to display the member id, member name and date of registration who have not taken any book.

**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.

**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”.

**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’.

**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.

**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.

**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.

**Problem # 14:**

Write a query to display the book code, book title and availability status of the ‘JAVA’ books whose edition is "5”. Show the availability status with alias name “AVAILABILITYSTATUS”. Hint: Book availability status can be fetched from “BOOK\_ISSUE\_STATUS” column of LMS\_BOOK\_ISSUE table.

**Complex Questions:**

**Problem # 1:**

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

**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

Problem # 3

Write a query to display the supplier id and supplier name of the supplier who has supplied minimum 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 “XYZ Store” has supplied minimum number of books, hence display the details as mentioned below.

Example:

SUPPLIER\_ID SUPPLIER\_NAME

S04 XYZ STORE

**Creating Database and tables:**

CREATE DATABASE lms\_db;

USE lms\_db;

Create table LMS\_MEMBERS

(

MEMBER\_ID Varchar(10),

MEMBER\_NAME Varchar(30) NOT NULL,

CITY Varchar(20),

DATE\_REGISTER Date NOT NULL,

DATE\_EXPIRE Date ,

MEMBERSHIP\_STATUS Varchar(15)NOT NULL,

Constraint LMS\_cts1 PRIMARY KEY(MEMBER\_ID)

);

Create table LMS\_SUPPLIERS\_DETAILS

(

SUPPLIER\_ID Varchar(3),

SUPPLIER\_NAME Varchar(30) NOT NULL,

ADDRESS Varchar(50),

CONTACT bigint(10) NOT NULL,

EMAIL Varchar(15) NOT NULL,

Constraint LMS\_cts2 PRIMARY KEY(SUPPLIER\_ID)

);

Create table LMS\_FINE\_DETAILS

(

FINE\_RANGE Varchar(3),

FINE\_AMOUNT decimal(10,2) NOT NULL,

Constraint LMS\_cts3 PRIMARY KEY(FINE\_RANGE)

);

Create table LMS\_BOOK\_DETAILS

(

BOOK\_CODE Varchar(10),

BOOK\_TITLE Varchar(50) NOT NULL,

CATEGORY Varchar(15) NOT NULL,

AUTHOR Varchar(30) NOT NULL,

PUBLICATION Varchar(30),

PUBLISH\_DATE Date,

BOOK\_EDITION int(2),

PRICE decimal(8,2) NOT NULL,

RACK\_NUM Varchar(3),

DATE\_ARRIVAL Date NOT NULL,

SUPPLIER\_ID Varchar(3) NOT NULL,

Constraint LMS\_cts4 PRIMARY KEY(BOOK\_CODE),

Constraint LMS\_cts41 FOREIGN KEY(SUPPLIER\_ID) References LMS\_SUPPLIERS\_DETAILS(SUPPLIER\_ID)

);

Create table LMS\_BOOK\_ISSUE

(

BOOK\_ISSUE\_NO int,

MEMBER\_ID Varchar(10) NOT NULL,

BOOK\_CODE Varchar(10) NOT NULL,

DATE\_ISSUE Date NOT NULL,

DATE\_RETURN Date NOT NULL,

DATE\_RETURNED Date NULL,

FINE\_RANGE Varchar(3),

Constraint LMS\_cts5 PRIMARY KEY(BOOK\_ISSUE\_NO),

Constraint LMS\_Mem FOREIGN KEY(MEMBER\_ID) References LMS\_MEMBERS(MEMBER\_ID),

Constraint LMS\_BookDetail FOREIGN KEY(BOOK\_CODE) References LMS\_BOOK\_DETAILS(BOOK\_CODE),

Constraint LMS\_FineDetail FOREIGN KEY(FINE\_RANGE) References LMS\_FINE\_DETAILS(FINE\_RANGE)

);

Insert into LMS\_MEMBERS

Values('LM001', 'AMIT', 'CHENNAI', '2012-02-12', '2013-02-11','Temporary');

Insert into LMS\_MEMBERS

Values('LM002', 'ABDHUL', 'DELHI', '2012-04-10', '2013-04-09','Temporary');

Insert into LMS\_MEMBERS

Values('LM003', 'GAYAN', 'CHENNAI', '2012-05-13','2013-05-12', 'Permanent');

Insert into LMS\_MEMBERS

Values('LM004', 'RADHA', 'CHENNAI', '2012-04-22', '2013-04-21', 'Temporary');

Insert into LMS\_MEMBERS

Values('LM005', 'GURU', 'BANGALORE', '2012-03-30', '2013-05-16','Temporary');

Insert into LMS\_MEMBERS

Values('LM006', 'MOHAN', 'CHENNAI', '2012-04-12', '2013-05-16','Temporary');

select \* from lms\_members;

Insert into LMS\_SUPPLIERS\_DETAILS

Values ('S01','SINGAPORE SHOPPEE', 'CHENNAI', 9894123555,'sing@gmail.com');

Insert into LMS\_SUPPLIERS\_DETAILS

Values ('S02','JK Stores', 'MUMBAI', 9940123450 ,'jks@yahoo.com');

Insert into LMS\_SUPPLIERS\_DETAILS

Values ('S03','ROSE BOOK STORE', 'TRIVANDRUM', 9444411222,'rose@gmail.com');

Insert into LMS\_SUPPLIERS\_DETAILS

Values ('S04','KAVARI STORE', 'DELHI', 8630001452,'kavi@redif.com');

Insert into LMS\_SUPPLIERS\_DETAILS

Values ('S05','EINSTEN BOOK GALLARY', 'US', 9542000001,'eingal@aol.com');

Insert into LMS\_SUPPLIERS\_DETAILS

Values ('S06','AKBAR STORE', 'MUMBAI',7855623100 ,'akbakst@aol.com');

select \* from LMS\_SUPPLIERS\_DETAILS;

Insert into LMS\_FINE\_DETAILS Values('R0', 0);

Insert into LMS\_FINE\_DETAILS Values('R1', 20);

insert into LMS\_FINE\_DETAILS Values('R2', 50);

Insert into LMS\_FINE\_DETAILS Values('R3', 75);

Insert into LMS\_FINE\_DETAILS Values('R4', 100);

Insert into LMS\_FINE\_DETAILS Values('R5', 150);

Insert into LMS\_FINE\_DETAILS Values('R6', 200);

select \* from LMS\_FINE\_DETAILS;

Insert into LMS\_BOOK\_DETAILS

Values('BL000010', 'Java ForvDummies', 'JAVA', 'Paul J. Deitel', 'Prentice Hall', '1999-12-10', 6, 575.00, 'A1', '2011-05-10', 'S01');

Insert into LMS\_BOOK\_DETAILS

Values('BL000002', 'Java: The Complete Reference ', 'JAVA', 'Herbert Schildt', 'Tata Mcgraw Hill ', '2011-10-10', 5, 750.00, 'A1', '2011-05-10', 'S03');

Insert into LMS\_BOOK\_DETAILS

Values('BL000003', 'Java How To Do Program', 'JAVA', 'Paul J. Deitel', 'Prentice Hall', '1999-05-10', 6, 600.00, 'A1', '2012-05-10', 'S01');

Insert into LMS\_BOOK\_DETAILS

Values('BL000004', 'Java: The Complete Reference ', 'JAVA', 'Herbert Schildt', 'Tata Mcgraw Hill ', '2011-10-10', 5, 750.00, 'A1', '2012-05-11', 'S01');

Insert into LMS\_BOOK\_DETAILS

Values('BL000005', 'Java How To Do Program', 'JAVA', 'Paul J. Deitel', 'Prentice Hall', '1999-12-10', 6, 600.00, 'A1', '2012-05-11', 'S01');

Insert into LMS\_BOOK\_DETAILS

Values('BL000006', 'Java: The Complete Reference ', 'JAVA', 'Herbert Schildt', 'Tata Mcgraw Hill ', '2011-10-10', 5, 750.00, 'A1', '2012-05-12', 'S03');

Insert into LMS\_BOOK\_DETAILS

Values('BL000007', 'Let Us C', 'C', 'Yashavant Kanetkar ', 'BPB Publications', '2010-12-11', 9, 500.00 , 'A3', '2010-11-03', 'S03');

Insert into LMS\_BOOK\_DETAILS

Values('BL000008', 'Let Us C', 'C', 'Yashavant Kanetkar ','BPB Publications', '2010-05-12', 9, 500.00 , 'A3', '2011-08-09', 'S04');

Insert into LMS\_BOOK\_DETAILS

Values('BL000009', 'Let Us C#', 'C', 'Yashavant Kanetkar ','BPB Publications', '2010-05-12', 9, 550.00 , 'A3', '2011-08-09', 'S04');

Insert into LMS\_BOOK\_DETAILS

Values('BL000011', 'Let Us C++', 'C', 'Yashavant Kanetkar ','BPB Publications', '2010-05-12', 9, 650.00 , 'A3', '2011-08-09', 'S04');

select \* from LMS\_BOOK\_DETAILS;

Insert into LMS\_BOOK\_ISSUE

Values (001, 'LM001', 'BL000010', '2012-05-01', '2012-05-16', '2012-05-16', 'R0');

Insert into LMS\_BOOK\_ISSUE

Values (002, 'LM002', 'BL000002', '2012-05-01', '2012-05-06','2012-05-16', 'R2');

Insert into LMS\_BOOK\_ISSUE

Values (003, 'LM003', 'BL000007', '2012-04-01', '2012-04-16', '2012-04-20','R1');

Insert into LMS\_BOOK\_ISSUE

Values (004, 'LM004', 'BL000005', '2012-04-01', '2012-04-16','2012-04-20', 'R1');

Insert into LMS\_BOOK\_ISSUE

Values (005, 'LM005', 'BL000008', '2012-03-30', '2012-04-15','2012-04-20' , 'R1');

Insert into LMS\_BOOK\_ISSUE

Values (006, 'LM005', 'BL000008', '2012-04-20', '2012-05-05','2012-05-05' , 'R0');

Insert into LMS\_BOOK\_ISSUE

Values (007, 'LM003', 'BL000007', '2012-04-22', '2012-05-07','2012-05-25' , 'R4');

select \* from LMS\_BOOK\_ISSUE ;

**Solutions**

\\1\\

select member\_id,member\_name,city,membership\_status

from lms\_members

where membership\_status='permanent';

select\*from lms\_book\_issue;

\\2\\

select bd.book\_code,bd.publication,bd.price,sd.SUPPLIER\_NAME,

count(bi.book\_code)

From lms\_book\_details bd

join lms\_suppliers\_details sd

join lms\_book\_issue bi

on sd.SUPPLIER\_ID= bd.SUPPLIER\_ID

and bd.BOOK\_CODE=bi.BOOK\_CODE

group by bd.BOOK\_CODE

Having count(bi.book\_code)=(select count(Book\_code)

from lms\_book\_issue

group by book\_code

order by count(Book\_code)desc

limit 1);

\\3\\

select member\_id,member\_name

from lms\_members

where member\_id in

(select member\_id

from lms\_book\_issue

where book\_code like'%02');

\\4\\

select book\_code,author,book\_title

from lms\_book\_details

where author like 'p%';

\\5\\

select count(category) no\_of\_books

from lms\_book\_details

where category ='java';

\\6\\

select category,count(category) no\_of\_book

from lms\_book\_details

where category ='java' or category='C'

group by category;

\7\

select count(author) no\_of\_books

from lms\_book\_details

where publication like 'p%';

\\8\\

select book\_title,book\_code

from lms\_book\_details

where book\_code in

(select book\_code

from lms\_book\_issue

where date\_issue ='2012-04-01');

\\9\\

select member\_id,member\_name,date\_register,date\_expire

from lms\_members

where date\_expire<'2013-04-00';

\\10\\

select member\_name,member\_id,date\_register,membership\_status

from lms\_members

where membership\_status='temporary' and date\_register<'2012-03-00';

\\11\\

select member\_id,member\_name name,city

from lms\_members

where city='delhi' or city='chennai';

\\12\\

select concat(book\_title,'\_Is\_Written\_by\_',author)

from lms\_book\_details;

\\13\\

select avg(price) as averageprice from lms\_book\_details

where category='java'

\\14\\

select supplier\_id,supplier\_name,email

from lms\_suppliers\_details

where email like '%gmail%';

\\15\\

SELECT supplier\_Name,supplier\_id,

COALESCE(contact,email,address) contactaddress

FROM lms\_suppliers\_details;

select supplier\_name,supplier\_id,contact,

if (contact is null,'no','yes')

from lms\_suppliers\_details;

\\17\\

SELECT m.MEMBER\_ID, m.MEMBER\_NAME, m.CITY,m.membership\_status,

SUM(f.FINE\_AMOUNT) AS Fine

FROM LMS\_MEMBERS m JOIN

LMS\_BOOK\_ISSUE i ON m.MEMBER\_ID = i.MEMBER\_ID

JOIN

LMS\_FINE\_DETAILS f ON i.FINE\_RANGE = f.FINE\_RANGE

GROUP BY m.MEMBER\_ID, m.MEMBER\_NAME, m.CITY,m.membership\_status

\\avg problems\\

\\1\\

select member\_id,member\_name,book\_title,book\_code

from lms\_members

join

lms\_fine\_details;

\\2\\

select count(book\_code)no\_books\_available

from lms\_book\_details

where book\_code not in (select book\_code from lms\_book\_issue);

\\3\\

select m.member\_id,m.member\_name,f.fine\_range,sum(f.fine\_amount)

from lms\_members m

join

lms\_fine\_details f

join

lms\_book\_issue bi

on m.member\_id=bi.member\_id and

f.fine\_range=bi.fine\_range

group by m.member\_id

having sum(f.fine\_amount)>100;

\\4\\

select book\_code,book\_title,

publication,price,book\_edition,publish\_date

from lms\_book\_details

order by book\_edition,publication,publish\_date ;

\\5\\

select book\_code,book\_title,rack\_num

from lms\_book\_details

where rack\_num='A1'

ORDER BY BOOK\_title;

\\6\\

select m.member\_id,m.member\_name,bi.date\_return,bi.date\_returned

from lms\_members m

join

lms\_book\_issue bi

on m.member\_id=bi.member\_id

where date\_return != date\_returned;

\\7\\

select m.member\_id,m.member\_name,m.date\_register

from lms\_members m

where member\_id not in(select member\_id from lms\_book\_issue);

\\8\\

select member\_id,member\_name

from

lms\_members

where

member\_id not in(select bi.member\_id

from lms\_book\_issue bi

join lms\_fine\_details f

on f.fine\_range=bi.fine\_range

where year(date\_returned)=2012

group by bi.member\_id

having sum(f.fine\_amount)>0);

\\9\\

select count(date\_issue)no\_of\_books,date\_issue from lms\_book\_issue

group by book\_code

having count(date\_issue)=

(select count(date\_issue)

from lms\_book\_issue

group by date\_issue

order by count(date\_issue)desc

limit 1);

\\10\\

select book\_title,supplier\_id

from lms\_book\_details

where author like 'h%'

and supplier\_id='S01'

and book\_edition=5;

\\11\\

select count(rack\_num)no\_of\_books,rack\_num

from lms\_book\_details

group by rack\_num

order by count(rack\_num);

\\12\\

select member\_name,book\_title,category,author,price,

date\_issue,date\_return,date\_returned,book\_issue\_no

fine\_amount

from lms\_book\_details

join lms\_members

join lms\_suppliers\_details

join lms\_book\_issue

join lms\_fine\_details;

\\13\\

select book\_code,book\_title,publish\_date

from lms\_book\_details

where month(publish\_date)=12;

\\14\\

select book\_code,book\_title,bd.supplier\_id,max(price)

,supplier\_name

from lms\_book\_details bd

join lms\_suppliers\_details sd

on sd.supplier\_id=bd.supplier\_id

group by supplier\_id;

\\15\\

select book\_code,book\_title,

publication,datediff(current\_date,publish\_date)no\_of\_days

from lms\_book\_details

order by datediff(current\_date,publish\_date) ;

select \* from LMS\_BOOK\_ISSUE ;

select \* from lms\_members;

select \* from LMS\_BOOK\_DETAILS;

select \* from LMS\_FINE\_DETAILS;

select \* from LMS\_SUPPLIERS\_DETAILS;

**Complex**

\\2\\

select m.member\_id,member\_name,3-count( date\_issue)remaining

from lms\_members m

left join lms\_book\_issue bi

on m.member\_id=bi.member\_id

group by member\_id;

\\3\\

select sd.supplier\_id,sd.supplier\_name,count(rack\_num)

from lms\_suppliers\_details sd

join lms\_book\_details bd

on sd.supplier\_id=bd.supplier\_id

group by sd.supplier\_id

having count(rack\_num)=(select count(rack\_num)

from lms\_book\_details bd join lms\_suppliers\_details sd

on sd.supplier\_id=bd.supplier\_id

group by sd.supplier\_id

order by count(rack\_num)

limit 1);