WEEK 1

13/07/2025

GitHub

Problem Scenario: The Indian Government wants to develop a Library Management System (LMS) to store information of the members, books, status of books issue, book availability and suppliers details

List of Tables:

Table #1: LMS_MEMBERS
Table #2: LMS_SUPPLIERS_DETAILS
Table #3: LMS_FINE_DETAILS
Table #4: LMS_BOOK_DETAILS
Table #5: LMS_BOOK_ISSUE

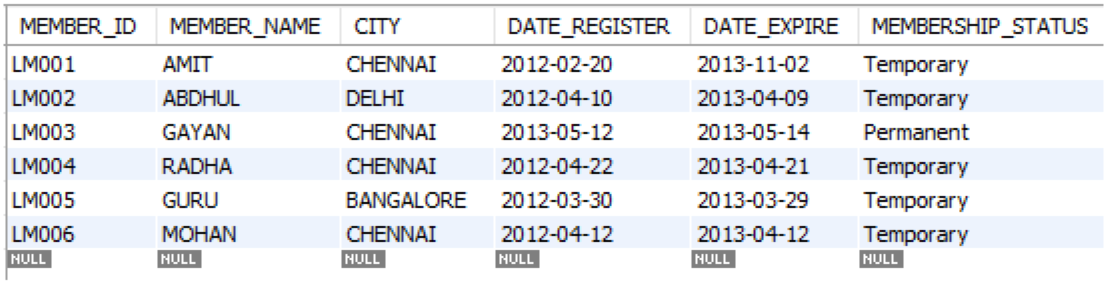
```
create database lms;
use lms;
```

-- LMS_MEMBERS table

```
create table LMS_MEMBERS (
        MEMBER_ID varchar(10) primary key,
   MEMBER_NAME varchar(30) not null,
   CITY varchar(20) not null,
   DATE_REGISTER date not null,
   DATE_EXPIRE date not null,
   MEMBERSHIP_STATUS varchar(15)
);
```

```
insert into LMS_MEMBERS (MEMBER_ID, MEMBER_NAME, CITY, DATE_REGISTER,
DATE_EXPIRE, MEMBERSHIP_STATUS) values
('LM001', 'AMIT', 'CHENNAI', ('2012-02-20'), ('2013-11-02'),'Temporary'),
('LM002', 'ABDHUL', 'DELHI', ('2012-04-10'),('2013-04-09'),'Temporary'),
('LM003', 'GAYAN', 'CHENNAI', ('2013-05-12'),('2013-05-14'), 'Permanent'),
('LM004', 'RADHA', 'CHENNAI', ('2012-04-22'), ('2013-04-21'), 'Temporary'),
('LM005', 'GURU', 'BANGALORE', ('2012-03-30'), ('2013-03-29'),'Temporary'),
('LM006', 'MOHAN', 'CHENNAI', ('2012-04-12'), ('2013-04-12'),'Temporary');
```

```
select * from LMS_MEMBERS;
```



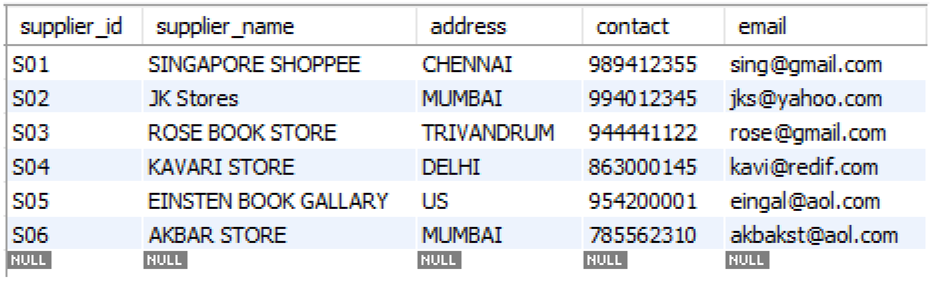
MEMBER_ID	MEMBER_NAME	CITY	DATE_REGISTER	DATE_EXPIRE	MEMBERSHIP_STATUS
LM001	AMIT	CHENNAI	2012-02-20	2013-11-02	Temporary
LM002	ABDHUL	DELHI	2012-04-10	2013-04-09	Temporary
LM003	GAYAN	CHENNAI	2013-05-12	2013-05-14	Permanent
LM004	RADHA	CHENNAI	2012-04-22	2013-04-21	Temporary
LM005	GURU	BANGALORE	2012-03-30	2013-03-29	Temporary
LM006	MOHAN	CHENNAI	2012-04-12	2013-04-12	Temporary
NULL	NULL	NULL	NULL	NULL	NULL

-- LMS_SUPPLIERS_DETAILS table

```
create table LMS_SUPPLIERS_DETAILS (
   supplier_id VARCHAR(3) PRIMARY KEY,
   supplier_name VARCHAR(30),
   address VARCHAR(50),
   contact BIGINT,
   email VARCHAR(50)
);

insert into LMS_SUPPLIERS_DETAILS (supplier_id, supplier_name, address, contact, email)
values
('S01', 'SINGAPORE SHOPPEE', 'CHENNAI', 989412355, 'sing@gmail.com'),
('S02', 'JK Stores', 'MUMBAI', 994012345, 'jks@yahoo.com'),
('S03', 'ROSE BOOK STORE', 'TRIVANDRUM', 944441122, 'rose@gmail.com'),
('S04', 'KAVARI STORE', 'DELHI', 863000145, 'kavi@redif.com'),
('S05', 'EINSTEN BOOK GALLARY', 'US', 954200001, 'eingal@aol.com'),
('S06', 'AKBAR STORE', 'MUMBAI', 785562310, 'akbakst@aol.com');

select * from LMS_SUPPLIERS_DETAILS;
```



supplier_id	supplier_name	address	contact	email
S01	SINGAPORE SHOPPEE	CHENNAI	989412355	sing@gmail.com
S02	JK Stores	MUMBAI	994012345	jks@yahoo.com
S03	ROSE BOOK STORE	TRIVANDRUM	944441122	rose@gmail.com
S04	KAVARI STORE	DELHI	863000145	kavi@redif.com
S05	EINSTEN BOOK GALLARY	US	954200001	eingal@aol.com
S06	AKBAR STORE	MUMBAI	785562310	akbakst@aol.com
NULL	NULL	NULL	NULL	NULL

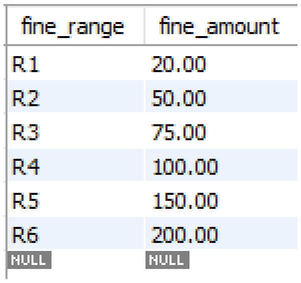
-- LMS_FINE_DETAILS table

```
create table LMS_FINE_DETAILS (
   fine_range VARCHAR(3) PRIMARY KEY,
   fine_amount DECIMAL(10,2)
);

insert into LMS_FINE_DETAILS (fine_range, fine_amount) values
```

```
('R1', 20),
('R2', 50),
('R3', 75),
('R4', 100),
('R5', 150),
('R6', 200);

select * from LMS_FINE_DETAILS;
```



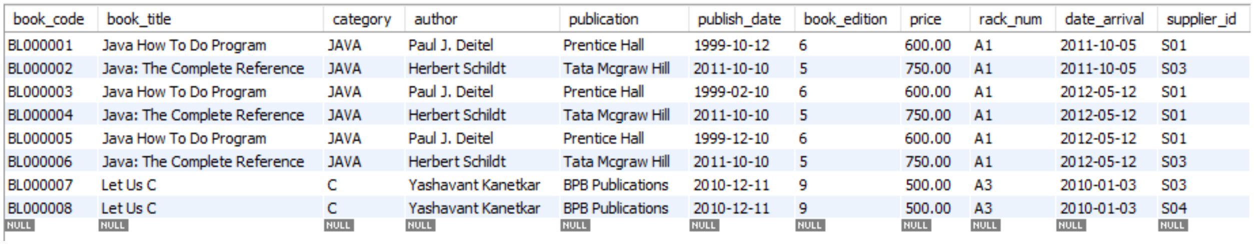
fine_range	fine_amount
R1	20.00
R2	50.00
R3	75.00
R4	100.00
R5	150.00
R6	200.00
NULL	NULL

-- LMS_BOOK_DETAILS table

```
create table LMS_BOOK_DETAILS (
   book_code VARCHAR(10) PRIMARY KEY,
   book_title VARCHAR(50),
   category VARCHAR(15),
   author VARCHAR(30),
   publication VARCHAR(30),
   publish_date DATE,
   book_edition INT,
   price DECIMAL(10,2),
   rack_num VARCHAR(3),
   date_arrival DATE,
   supplier_id VARCHAR(3),
   FOREIGN KEY (supplier_id) REFERENCES LMS_SUPPLIERS_DETAILS(supplier_id)
);

insert into LMS_BOOK_DETAILS (book_code, book_title, category, author, publication,
publish_date, book_edition, price, rack_num, date_arrival, supplier_id) values
('BL000001', 'Java How To Do Program', 'JAVA', 'Paul J. Deitel', 'Prentice Hall', '1999-10-12', 6,
600.00, 'A1', '2011-10-05', 'S01'),
('BL000002', 'Java: The Complete Reference', 'JAVA', 'Herbert Schildt', 'Tata Mcgraw Hill',
'2011-10-10', 5, 750.00, 'A1', '2011-10-05', 'S03'),
('BL000003', 'Java How To Do Program', 'JAVA', 'Paul J. Deitel', 'Prentice Hall', '1999-02-10', 6,
600.00, 'A1', '2012-05-12', 'S01'),
('BL000004', 'Java: The Complete Reference', 'JAVA', 'Herbert Schildt', 'Tata Mcgraw Hill',
'2011-10-10', 5, 750.00, 'A1', '2012-05-12', 'S01'),
('BL000005', 'Java How To Do Program', 'JAVA', 'Paul J. Deitel', 'Prentice Hall', '1999-12-10', 6,
600.00, 'A1', '2012-05-12', 'S01'),
('BL000006', 'Java: The Complete Reference', 'JAVA', 'Herbert Schildt', 'Tata Mcgraw Hill',
'2011-10-10', 5, 750.00, 'A1', '2012-05-12', 'S03'),
('BL000007', 'Let Us C', 'C', 'Yashavant Kanetkar', 'BPB Publications', '2010-12-11', 9, 500.00,
'A3', '2010-01-03', 'S03'),
('BL000008', 'Let Us C', 'C', 'Yashavant Kanetkar', 'BPB Publications', '2010-12-11', 9, 500.00,
'A3', '2010-01-03', 'S04');

select * from LMS_BOOK_DETAILS;
```

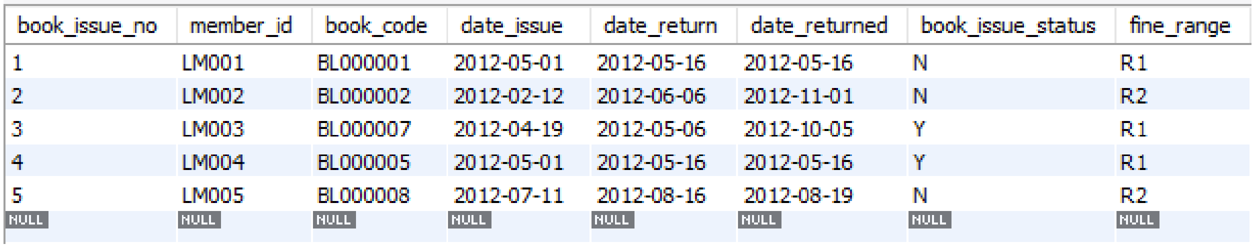


book_code	book_title	category	author	publication	publish_date	book_edition	price	rack_num	date_arrival	supplier_id
BL000001	Java How To Do Program	JAVA	Paul J. Deitel	Prentice Hall	1999-10-12	6	600.00	A1	2011-10-05	S01
BL000002	Java: The Complete Reference	JAVA	Herbert Schildt	Tata Mcgraw Hill	2011-10-10	5	750.00	A1	2011-10-05	S03
BL000003	Java How To Do Program	JAVA	Paul J. Deitel	Prentice Hall	1999-02-10	6	600.00	A1	2012-05-12	S01
BL000004	Java: The Complete Reference	JAVA	Herbert Schildt	Tata Mcgraw Hill	2011-10-10	5	750.00	A1	2012-05-12	S01
BL000005	Java How To Do Program	JAVA	Paul J. Deitel	Prentice Hall	1999-12-10	6	600.00	A1	2012-05-12	S01
BL000006	Java: The Complete Reference	JAVA	Herbert Schildt	Tata Mcgraw Hill	2011-10-10	5	750.00	A1	2012-05-12	S03
BL000007	Let Us C	C	Yashavant Kanetkar	BPB Publications	2010-12-11	9	500.00	A3	2010-01-03	S03
BL000008	Let Us C	C	Yashavant Kanetkar	BPB Publications	2010-12-11	9	500.00	A3	2010-01-03	S04
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
-- LMS_BOOK_ISSUE table
create table LMS_BOOK_ISSUE (
   book_issue_no INT PRIMARY KEY,
   member_id VARCHAR(10),
   book_code VARCHAR(10),
   date_issue DATE,
   date_return DATE,
   date_returned DATE,
   book_issue_status CHAR(1),
   fine_range VARCHAR(3),
   FOREIGN KEY (member_id) REFERENCES LMS_MEMBERS(member_id),
   FOREIGN KEY (book_code) REFERENCES LMS_BOOK_DETAILS(book_code),
   FOREIGN KEY (fine_range) REFERENCES LMS_FINE_DETAILS(fine_range)
);

insert into LMS_BOOK_ISSUE (book_issue_no, member_id, book_code, date_issue,
date_return, date_returned, book_issue_status, fine_range) values
(001, 'LM001', 'BL000001', '2012-05-01', '2012-05-16', '2012-05-16', 'N', 'R1'),
(002, 'LM002', 'BL000002', '2012-02-12', '2012-06-06', '2012-11-01', 'N', 'R2'),
(003, 'LM003', 'BL000007', '2012-04-19', '2012-05-06', '2012-10-05', 'Y', 'R1'),
(004, 'LM004', 'BL000005', '2012-05-01', '2012-05-16', '2012-05-16', 'Y', 'R1'),
(005, 'LM005', 'BL000008', '2012-07-11', '2012-08-16', '2012-08-19', 'N', 'R2');

select * from LMS_BOOK_ISSUE;
```



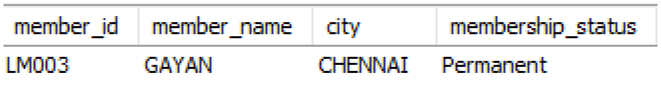
book_issue_no	member_id	book_code	date_issue	date_return	date_returned	book_issue_status	fine_range
1	LM001	BL000001	2012-05-01	2012-05-16	2012-05-16	N	R1
2	LM002	BL000002	2012-02-12	2012-06-06	2012-11-01	N	R2
3	LM003	BL000007	2012-04-19	2012-05-06	2012-10-05	Y	R1
4	LM004	BL000005	2012-05-01	2012-05-16	2012-05-16	Y	R1
5	LM005	BL000008	2012-07-11	2012-08-16	2012-08-19	N	R2
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

use lms;

-- 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”.

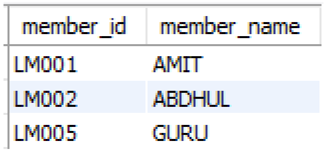
```
SELECT member_id, member_name, city, membership_status
FROM LMS_MEMBERS
WHERE membership_status = 'Permanent';
```



member_id	member_name	city	membership_status
LM003	GAYAN	CHENNAI	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'.

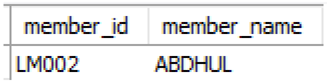
```
SELECT DISTINCT m.member_id, m.member_name
FROM LMS_MEMBERS m
JOIN LMS_BOOK_ISSUE b ON m.member_id = b.member_id
WHERE b.book_issue_status = 'N';
```



member_id	member_name
LM001	AMIT
LM002	ABDHUL
LM005	GURU

-- Problem # 3:
-- Write a query to display the member id, member name who have taken the book with book code 'BL000002'.

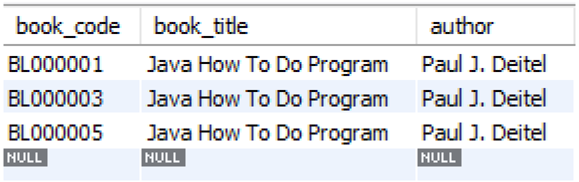
```
SELECT DISTINCT m.member_id, m.member_name
FROM LMS_MEMBERS m
JOIN LMS_BOOK_ISSUE b ON m.member_id = b.member_id
WHERE b.book_code = 'BL000002';
```



member_id	member_name
LM002	ABDHUL

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

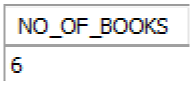
```
SELECT book_code, book_title, author
FROM LMS_BOOK_DETAILS
WHERE author LIKE 'P%';
```



book_code	book_title	author
BL000001	Java How To Do Program	Paul J. Deitel
BL000003	Java How To Do Program	Paul J. Deitel
BL000005	Java How To Do Program	Paul J. Deitel
NULL	NULL	NULL

-- Problem # 5:
-- Write a query to display the total number of Java books available in library with alias name ‘NO_OF_BOOKS’.

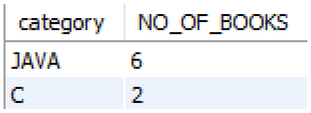
```
SELECT COUNT(*) AS NO_OF_BOOKS
FROM LMS_BOOK_DETAILS
WHERE category = 'JAVA';
```



NO_OF_BOOKS
6

-- Problem # 6:
-- Write a query to list the category and number of books in each category with alias name ‘NO_OF_BOOKS’.

```
SELECT category, COUNT(*) AS NO_OF_BOOKS
FROM LMS_BOOK_DETAILS
GROUP BY category;
```

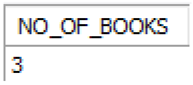


category	NO_OF_BOOKS
JAVA	6
C	2

-- Problem # 7:
-- Write a query to display the number of books published by "Prentice Hall” with the alias name “NO_OF_BOOKS”.

```
SELECT COUNT(*) AS NO_OF_BOOKS
FROM LMS_BOOK_DETAILS
```

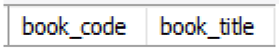
```
WHERE publication = 'Prentice Hall';
```



NO_OF_BOOKS
3

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

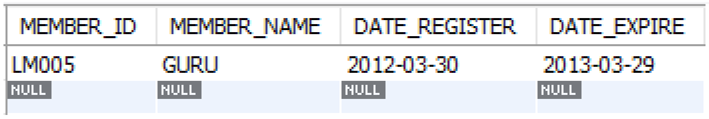
```
SELECT bd.book_code, bd.book_title
FROM LMS_BOOK_ISSUE bi
JOIN LMS_BOOK_DETAILS bd ON bi.book_code = bd.book_code
WHERE bi.date_issue = '2012-04-01';
```



book_code	book_title

-- 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.

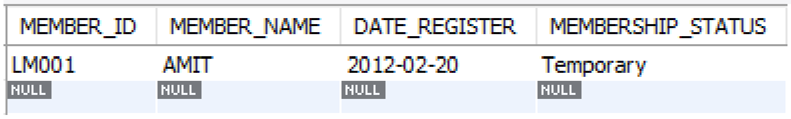
```
SELECT MEMBER_ID, MEMBER_NAME, DATE_REGISTER, DATE_EXPIRE
FROM LMS_MEMBERS
WHERE DATE_EXPIRE < '2013-04-01';
```



MEMBER_ID	MEMBER_NAME	DATE_REGISTER	DATE_EXPIRE
LM005	GURU	2012-03-30	2013-03-29
NULL	NULL	NULL	NULL

-- 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"

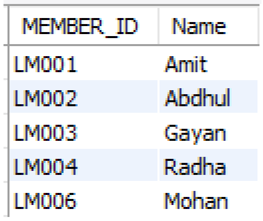
```
select MEMBER_ID, MEMBER_NAME, DATE_REGISTER, MEMBERSHIP_STATUS
from LMS_MEMBERS
where DATE_REGISTER < '2012-03-01' and MEMBERSHIP_STATUS = 'Temporary';
```



MEMBER_ID	MEMBER_NAME	DATE_REGISTER	MEMBERSHIP_STATUS
LM001	AMIT	2012-02-20	Temporary
NULL	NULL	NULL	NULL

-- 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'.

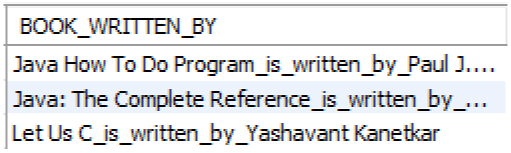
```
select MEMBER_ID, concat(ucase(LEFT(MEMBER_NAME, 1)),
lcase(substring(MEMBER_NAME, 2))) AS Name
from LMS_MEMBERS
where CITY in ('CHENNAI', 'DELHI');
```



MEMBER_ID	Name
LM001	Amit
LM002	Abdhul
LM003	Gayan
LM004	Radha
LM006	Mohan

-- 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.

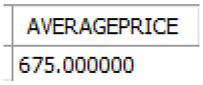
```
select distinct concat(BOOK_TITLE, '_is_written_by_', AUTHOR) AS BOOK_WRITTEN_BY
from LMS_BOOK_DETAILS;
```



BOOK_WRITTEN_BY
Java How To Do Program_is_written_by_Paul J....
Java: The Complete Reference_is_written_by_...
Let Us C_is_written_by_Yashavant Kanetkar

-- Problem #13:
-- Write a query to display the average price of books which is belonging to ‘JAVA’ category with alias name “AVERAGEPRICE”.

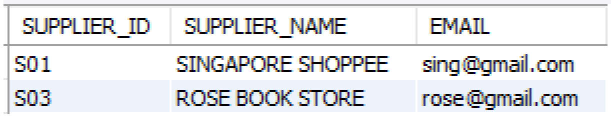
```
select avg(PRICE) AS AVERAGEPRICE
from LMS_BOOK_DETAILS
where CATEGORY = 'JAVA';
```



AVERAGEPRICE
675.000000

-- Problem #14:
-- Write a query to display the supplier id, supplier name and email of the suppliers who are all having gmail account.

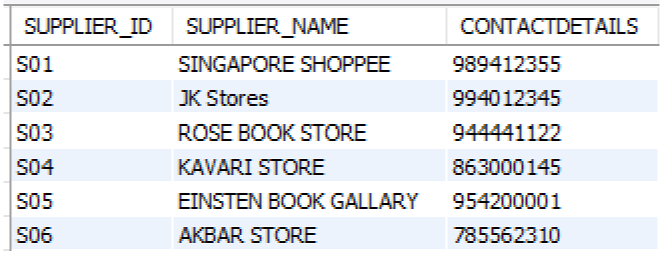
```
select SUPPLIER_ID, SUPPLIER_NAME, EMAIL
from LMS_SUPPLIERS_DETAILS
where EMAIL like '%@gmail.com';
```



SUPPLIER_ID	SUPPLIER_NAME	EMAIL
S01	SINGAPORE SHOPPEE	sing@gmail.com
S03	ROSE BOOK STORE	rose@gmail.com

-- 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.

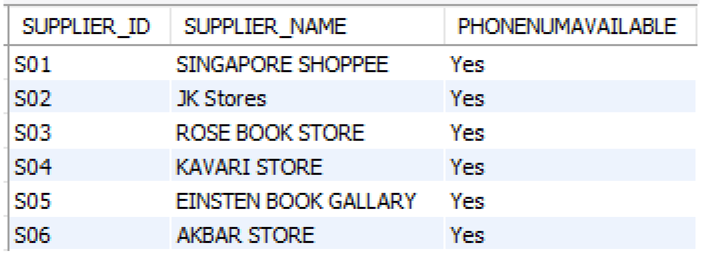
```
select SUPPLIER_ID, SUPPLIER_NAME,
coalesce(cast(CONTACT AS CHAR), EMAIL, ADDRESS) AS CONTACTDETAILS
from LMS_SUPPLIERS_DETAILS;
```



SUPPLIER_ID	SUPPLIER_NAME	CONTACTDETAILS
S01	SINGAPORE SHOPPEE	989412355
S02	JK Stores	994012345
S03	ROSE BOOK STORE	944441122
S04	KAVARI STORE	863000145
S05	EINSTEN BOOK GALLARY	954200001
S06	AKBAR STORE	785562310

-- 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.

```
select SUPPLIER_ID, SUPPLIER_NAME,
if(CONTACT is NOT NULL, 'Yes', 'No') AS PHONENUMAVAILABLE
from LMS_SUPPLIERS_DETAILS;
```

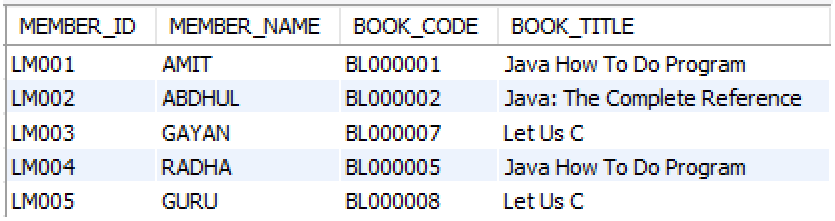


SUPPLIER_ID	SUPPLIER_NAME	PHONENUMAVAILABLE
S01	SINGAPORE SHOPPEE	Yes
S02	JK Stores	Yes
S03	ROSE BOOK STORE	Yes
S04	KAVARI STORE	Yes
S05	EINSTEN BOOK GALLARY	Yes
S06	AKBAR STORE	Yes

-- 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.

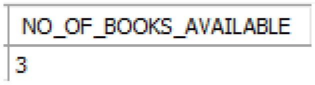
```
select M.MEMBER_ID, M.MEMBER_NAME, B.BOOK_CODE, D.BOOK_TITLE
from LMS_MEMBERS M
join LMS_BOOK_ISSUE B on M.MEMBER_ID = B.MEMBER_ID
join LMS_BOOK_DETAILS D on B.BOOK_CODE = D.BOOK_CODE;
```



MEMBER_ID	MEMBER_NAME	BOOK_CODE	BOOK_TITLE
LM001	AMIT	BL000001	Java How To Do Program
LM002	ABDHUL	BL000002	Java: The Complete Reference
LM003	GAYAN	BL000007	Let Us C
LM004	RADHA	BL000005	Java How To Do Program
LM005	GURU	BL000008	Let Us C

-- 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.

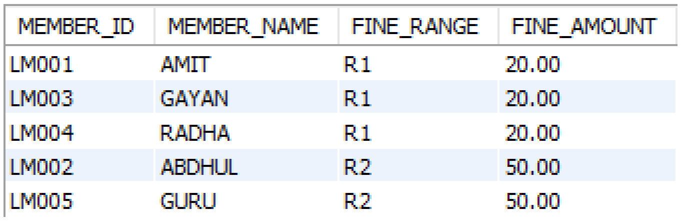
```
select count(*) AS NO_OF_BOOKS_AVAILABLE
from LMS_BOOK_DETAILS
where BOOK_CODE not in (
   select distinct BOOK_CODE from LMS_BOOK_ISSUE
);
```



NO_OF_BOOKS_AVAILABLE
3

-- 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.

```
select M.MEMBER_ID, M.MEMBER_NAME, F.FINE_RANGE, F.FINE_AMOUNT
from LMS_MEMBERS M
join LMS_BOOK_ISSUE B ON M.MEMBER_ID = B.MEMBER_ID
join LMS_FINE_DETAILS F ON B.FINE_RANGE = F.FINE_RANGE
where F.FINE_AMOUNT < 100;
```



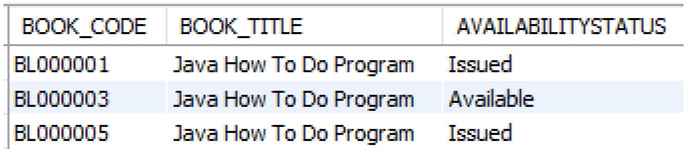
MEMBER_ID	MEMBER_NAME	FINE_RANGE	FINE_AMOUNT
LM001	AMIT	R1	20.00
LM003	GAYAN	R1	20.00
LM004	RADHA	R1	20.00
LM002	ABDHUL	R2	50.00
LM005	GURU	R2	50.00

-- 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.

```
select D.BOOK_CODE, D.BOOK_TITLE,
     if(B.BOOK_ISSUE_STATUS IS NULL, 'Available', 'Issued') AS AVAILABILITYSTATUS
from LMS_BOOK_DETAILS D
left join LMS_BOOK_ISSUE B ON D.BOOK_CODE = B.BOOK_CODE
where D.CATEGORY = 'JAVA' AND D.BOOK_EDITION = 6;
```

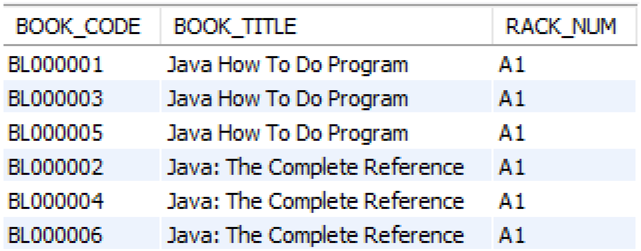


BOOK_CODE	BOOK_TITLE	AVAILABILITYSTATUS
BL000001	Java How To Do Program	Issued
BL000003	Java How To Do Program	Available
BL000005	Java How To Do Program	Issued

-- 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 BOOK_CODE, BOOK_TITLE, RACK_NUM
from LMS_BOOK_DETAILS
where RACK_NUM = 'A1'
order by BOOK_TITLE ASC;
```



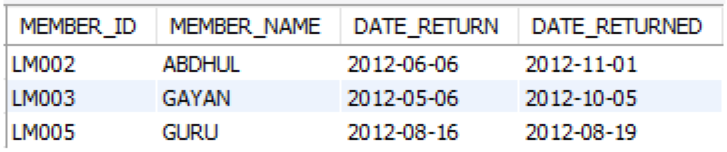
BOOK_CODE	BOOK_TITLE	RACK_NUM
BL000001	Java How To Do Program	A1
BL000003	Java How To Do Program	A1
BL000005	Java How To Do Program	A1
BL000002	Java: The Complete Reference	A1
BL000004	Java: The Complete Reference	A1
BL000006	Java: The Complete Reference	A1

-- 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.

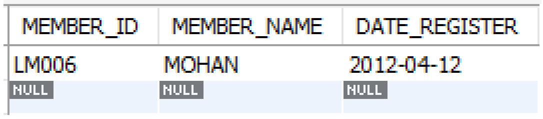
```
select M.MEMBER_ID, M.MEMBER_NAME, B.DATE_RETURN, B.DATE_RETURNED
from LMS_BOOK_ISSUE B
join LMS_MEMBERS M ON B.MEMBER_ID = M.MEMBER_ID
where B.DATE_RETURNED > B.DATE_RETURN;
```



MEMBER_ID	MEMBER_NAME	DATE_RETURN	DATE_RETURNED
LM002	ABDHUL	2012-06-06	2012-11-01
LM003	GAYAN	2012-05-06	2012-10-05
LM005	GURU	2012-08-16	2012-08-19

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

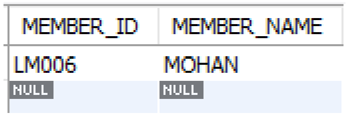
```
select M.MEMBER_ID, M.MEMBER_NAME, M.DATE_REGISTER
from LMS_MEMBERS M
where M.MEMBER_ID NOT IN (
   select distinct MEMBER_ID from LMS_BOOK_ISSUE
);
```



MEMBER_ID	MEMBER_NAME	DATE_REGISTER
LM006	MOHAN	2012-04-12
NULL	NULL	NULL

-- 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.

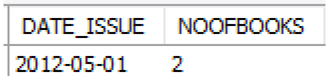
```
select distinct M.MEMBER_ID, M.MEMBER_NAME
from LMS_MEMBERS M
where M.MEMBER_ID NOT IN (
   select B.MEMBER_ID
   from LMS_BOOK_ISSUE B
   where year(B.DATE_RETURNED) = 2012
);
```



MEMBER_ID	MEMBER_NAME
LM006	MOHAN
NULL	NULL

-- 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”.

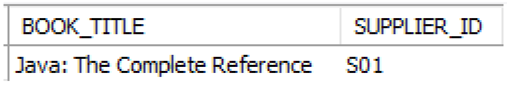
```
select DATE_ISSUE, count(*) AS NOOFBOOKS
from LMS_BOOK_ISSUE
group by DATE_ISSUE
order by NOOFBOOKS DESC
limit 1;
```



DATE_ISSUE	NOOFBOOKS
2012-05-01	2

-- 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’.

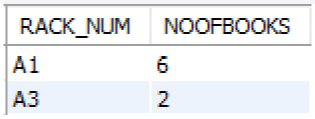
```
select BOOK_TITLE, SUPPLIER_ID
from LMS_BOOK_DETAILS
where AUTHOR = 'Herbert Schildt' and BOOK_EDITION = 5 and SUPPLIER_ID = 'S01';
```



BOOK_TITLE	SUPPLIER_ID
Java: The Complete Reference	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.

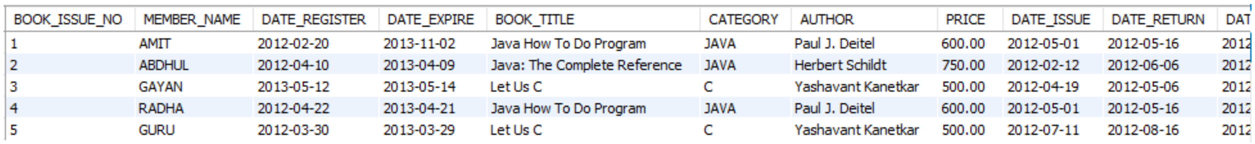
```
select RACK_NUM, COUNT(*) AS NOOFBOOKS
from LMS_BOOK_DETAILS
group by RACK_NUM
order by RACK_NUM ASC;
```



RACK_NUM	NOOFBOOKS
A1	6
A3	2

-- 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.

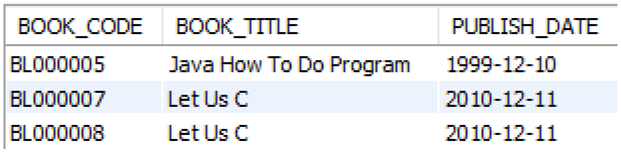
```
select B.BOOK_ISSUE_NO, M.MEMBER_NAME, M.DATE_REGISTER, M.DATE_EXPIRE,
    D.BOOK_TITLE, D.CATEGORY, D.AUTHOR, D.PRICE,
    B.DATE_ISSUE, B.DATE_RETURN, B.DATE_RETURNED,
    B.BOOK_ISSUE_STATUS, F.FINE_AMOUNT
from LMS_BOOK_ISSUE B
join LMS_MEMBERS M ON B.MEMBER_ID = M.MEMBER_ID
join LMS_BOOK_DETAILS D ON B.BOOK_CODE = D.BOOK_CODE
join LMS_FINE_DETAILS F ON B.FINE_RANGE = F.FINE_RANGE;
```



BOOK_ISSUE_NO	MEMBER_NAME	DATE_REGISTER	DATE_EXPIRE	BOOK_TITLE	CATEGORY	AUTHOR	PRICE	DATE_ISSUE	DATE_RETURN	DAT
1	AMIT	2012-02-20	2013-11-02	Java How To Do Program	JAVA	Paul J. Deitel	600.00	2012-05-01	2012-05-16	2012
2	ABDHUL	2012-04-10	2013-04-09	Java: The Complete Reference	JAVA	Herbert Schildt	750.00	2012-02-12	2012-06-06	2012
3	GAYAN	2013-05-12	2013-05-14	Let Us C	C	Yashavant Kanetkar	500.00	2012-04-19	2012-05-06	2012
4	RADHA	2012-04-22	2013-04-21	Java How To Do Program	JAVA	Paul J. Deitel	600.00	2012-05-01	2012-05-16	2012
5	GURU	2012-03-30	2013-03-29	Let Us C	C	Yashavant Kanetkar	500.00	2012-07-11	2012-08-16	2012

-- 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.

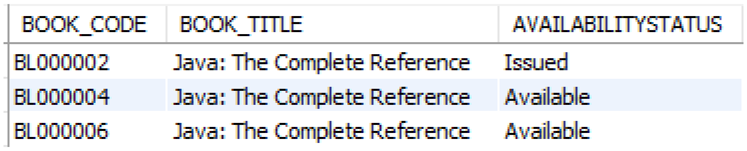
```
select BOOK_CODE, BOOK_TITLE, PUBLISH_DATE
from LMS_BOOK_DETAILS
where month(PUBLISH_DATE) = 12;
```



BOOK_CODE	BOOK_TITLE	PUBLISH_DATE
BL000005	Java How To Do Program	1999-12-10
BL000007	Let Us C	2010-12-11
BL000008	Let Us C	2010-12-11

-- 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.

```
select D.BOOK_CODE, D.BOOK_TITLE,
if(B.BOOK_ISSUE_STATUS IS NULL, 'Available', 'Issued') AS AVAILABILITYSTATUS
from LMS_BOOK_DETAILS D
left join LMS_BOOK_ISSUE B ON D.BOOK_CODE = B.BOOK_CODE
where D.CATEGORY = 'JAVA' AND D.BOOK_EDITION = 5;
```



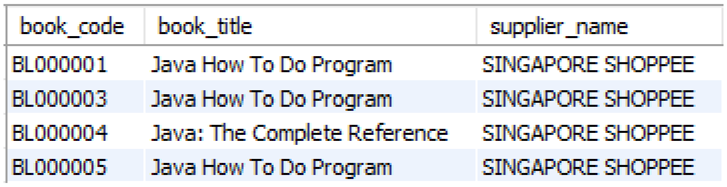
BOOK_CODE	BOOK_TITLE	AVAILABILITYSTATUS
BL000002	Java: The Complete Reference	Issued
BL000004	Java: The Complete Reference	Available
BL000006	Java: The Complete Reference	Available

-- 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

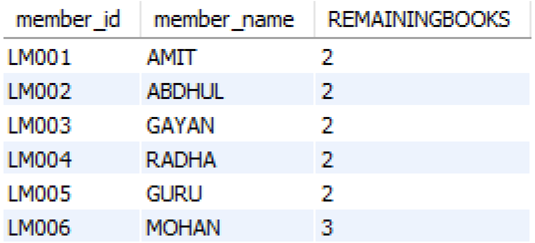
```
SELECT bd.book_code, bd.book_title, sd.supplier_name
FROM LMS_BOOK_DETAILS bd
JOIN LMS_SUPPLIERS_DETAILS sd ON bd.supplier_id = sd.supplier_id
WHERE bd.supplier_id = (
   SELECT supplier_id
   FROM LMS_BOOK_DETAILS
   GROUP BY supplier_id
   ORDER BY COUNT(*) DESC
   LIMIT 1
);
```



book_code	book_title	supplier_name
BL000001	Java How To Do Program	SINGAPORE SHOPPEE
BL000003	Java How To Do Program	SINGAPORE SHOPPEE
BL000004	Java: The Complete Reference	SINGAPORE SHOPPEE
BL000005	Java How To Do Program	SINGAPORE SHOPPEE

-- 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

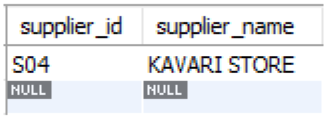
```
select m.member_id, m.member_name,
      (3 - ifnull(count(b.book_code), 0)) AS REMAININGBOOKS
from LMS_MEMBERS m
left join LMS_BOOK_ISSUE b ON m.member_id = b.member_id
group by m.member_id, m.member_name;
```



member_id	member_name	REMAININGBOOKS
LM001	AMIT	2
LM002	ABDHUL	2
LM003	GAYAN	2
LM004	RADHA	2
LM005	GURU	2
LM006	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
-- First, find the supplier(s) who supplied the least books

select sd.supplier_id, sd.supplier_name
from LMS_SUPPLIERS_DETAILS sd
where sd.supplier_id = (
   select supplier_id
   from LMS_BOOK_DETAILS
   group by supplier_id
   order by COUNT(*) ASC
   limit 1
);
```



supplier_id	supplier_name
S04	KAVARI STORE
NULL	NULL