**MySQL Case Study**

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 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');

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');

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);

Insert into LMS\_BOOK\_DETAILS

Values('BL000001', 'Java How To Do Program', 'JAVA', 'Paul J. Deitel', 'Prentice Hall', '1999-12-10', 6, 600.00, '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\_ISSUE

Values (001, 'LM001', 'BL000001', '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\_MEMBERS;

-- select \* from LMS\_SUPPLIERS\_DETAILS;

-- select \* from LMS\_FINE\_DETAILS;

-- select \* from LMS\_BOOK\_DETAILS;

-- select \* from LMS\_BOOK\_ISSUE;

Simple Questions:

-- 1

SELECT MEMBER\_ID, MEMBER\_NAME, CITY, MEMBERSHIP\_STATUS

FROM LMS\_MEMBERS WHERE MEMBERSHIP\_STATUS='PERMANENT';

-- 2

SELECT b.BOOK\_CODE, b.PUBLICATION, b.PRICE, s.SUPPLIER\_NAME

FROM LMS\_BOOK\_DETAILS b

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

JOIN (

SELECT BOOK\_CODE

FROM LMS\_BOOK\_ISSUE

GROUP BY BOOK\_CODE

ORDER BY COUNT(\*) DESC

LIMIT 1

) AS most\_issued\_book ON b.BOOK\_CODE = most\_issued\_book.BOOK\_CODE;

-- 3

SELECT LMS\_MEMBERS.MEMBER\_ID, LMS\_MEMBERS.MEMBER\_NAME, LMS\_FINE\_DETAILS.FINE\_RANGE, LMS\_FINE\_DETAILS.FINE\_AMOUNT

FROM LMS\_MEMBERS

JOIN LMS\_BOOK\_ISSUE ON LMS\_MEMBERS.MEMBER\_ID = LMS\_BOOK\_ISSUE.MEMBER\_ID

JOIN LMS\_FINE\_DETAILS ON LMS\_BOOK\_ISSUE.FINE\_RANGE = LMS\_FINE\_DETAILS.FINE\_RANGE

WHERE LMS\_FINE\_DETAILS.FINE\_AMOUNT < 100;

-- 4

SELECT BOOK\_CODE, BOOK\_TITLE, PUBLICATION, BOOK\_EDITION, PRICE, PUBLISH\_DATE

FROM LMS\_BOOK\_DETAILS

ORDER BY PUBLISH\_DATE, PUBLICATION, BOOK\_EDITION;

-- 5

SELECT COUNT(\*) AS NO\_OF\_BOOKS

FROM LMS\_BOOK\_DETAILS

WHERE CATEGORY = 'JAVA';

-- 6

SELECT CATEGORY, COUNT(\*) AS NO\_OF\_BOOKS

FROM LMS\_BOOK\_DETAILS

GROUP BY CATEGORY;

-- 7

SELECT COUNT(\*) AS NO\_OF\_BOOKS

FROM LMS\_BOOK\_DETAILS

WHERE PUBLICATION = 'Prentice Hall';

-- 8

SELECT BOOK\_CODE, BOOK\_TITLE

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 < '2013-04-01';

-- 10

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

FROM LMS\_MEMBERS

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

-- 11

SELECT MEMBER\_ID, INITCAP(MEMBER\_NAME) AS Name

FROM LMS\_MEMBERS

WHERE CITY IN ('CHENNAI', 'DELHI');

-- 12

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

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.com';

-- 15

SELECT SUPPLIER\_ID, SUPPLIER\_NAME, COALESCE(CONTACT, EMAIL, ADDRESS) AS CONTACTDETAILS

FROM LMS\_SUPPLIERS\_DETAILS;

-- 16

SELECT SUPPLIER\_ID, SUPPLIER\_NAME,

CASE WHEN CONTACT IS NULL THEN 'No' ELSE 'Yes' END AS PHONENUMAVAILABLE

FROM LMS\_SUPPLIERS\_DETAILS;

-- 17

SELECT LMS\_MEMBERS.MEMBER\_ID, LMS\_MEMBERS.MEMBER\_NAME, LMS\_MEMBERS.CITY, LMS\_MEMBERS.MEMBERSHIP\_STATUS,

SUM(CASE WHEN LMS\_BOOK\_ISSUE.FINE\_RANGE IS NOT NULL THEN LMS\_FINE\_DETAILS.FINE\_AMOUNT ELSE 0 END) AS Fine

FROM LMS\_MEMBERS

JOIN LMS\_BOOK\_ISSUE ON LMS\_MEMBERS.MEMBER\_ID = LMS\_BOOK\_ISSUE.MEMBER\_ID

JOIN LMS\_FINE\_DETAILS ON LMS\_BOOK\_ISSUE.FINE\_RANGE = LMS\_FINE\_DETAILS.FINE\_RANGE

GROUP BY LMS\_MEMBERS.MEMBER\_ID;

Average Questions:

-- 1

SELECT LMS\_MEMBERS.MEMBER\_ID, LMS\_MEMBERS.MEMBER\_NAME, LMS\_BOOK\_ISSUE.BOOK\_CODE, LMS\_BOOK\_DETAILS.BOOK\_TITLE

FROM LMS\_MEMBERS

JOIN LMS\_BOOK\_ISSUE ON LMS\_MEMBERS.MEMBER\_ID = LMS\_BOOK\_ISSUE.MEMBER\_ID

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

-- 2

SELECT COUNT(\*) AS NO\_OF\_BOOKS\_AVAILABLE

FROM LMS\_BOOK\_DETAILS

WHERE BOOK\_CODE NOT IN (SELECT BOOK\_CODE FROM LMS\_BOOK\_ISSUE);

-- 3

SELECT LMS\_MEMBERS.MEMBER\_ID, LMS\_MEMBERS.MEMBER\_NAME, LMS\_FINE\_DETAILS.FINE\_RANGE, LMS\_FINE\_DETAILS.FINE\_AMOUNT

FROM LMS\_MEMBERS

JOIN LMS\_BOOK\_ISSUE ON LMS\_MEMBERS.MEMBER\_ID = LMS\_BOOK\_ISSUE.MEMBER\_ID

JOIN LMS\_FINE\_DETAILS ON LMS\_BOOK\_ISSUE.FINE\_RANGE = LMS\_FINE\_DETAILS.FINE\_RANGE

WHERE LMS\_FINE\_DETAILS.FINE\_AMOUNT < 100;

-- 4

SELECT BOOK\_CODE, BOOK\_TITLE, PUBLICATION, BOOK\_EDITION, PRICE, PUBLISH\_DATE

FROM LMS\_BOOK\_DETAILS

ORDER BY PUBLISH\_DATE, PUBLICATION, BOOK\_EDITION;

-- 5

SELECT BOOK\_CODE, BOOK\_TITLE, RACK\_NUM

FROM LMS\_BOOK\_DETAILS

WHERE RACK\_NUM = 'A1'

ORDER BY BOOK\_TITLE ASC;

-- 6

SELECT LMS\_MEMBERS.MEMBER\_ID, LMS\_MEMBERS.MEMBER\_NAME, LMS\_BOOK\_ISSUE.DATE\_RETURN AS DUE\_DATE, LMS\_BOOK\_ISSUE.DATE\_RETURNED

FROM LMS\_BOOK\_ISSUE

JOIN LMS\_MEMBERS ON LMS\_BOOK\_ISSUE.MEMBER\_ID = LMS\_MEMBERS.MEMBER\_ID

WHERE LMS\_BOOK\_ISSUE.DATE\_RETURNED > LMS\_BOOK\_ISSUE.DATE\_RETURN;

-- 7

SELECT MEMBER\_ID, MEMBER\_NAME

FROM LMS\_MEMBERS

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 DISTINCT MEMBER\_ID

FROM LMS\_BOOK\_ISSUE

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

);

-- 9

SELECT DATE\_ISSUE, COUNT(\*) AS NOOFBOOKS

FROM LMS\_BOOK\_ISSUE

GROUP BY DATE\_ISSUE

ORDER BY COUNT(\*) DESC LIMIT 1;

-- 10

SELECT BOOK\_TITLE, SUPPLIER\_ID

FROM LMS\_BOOK\_DETAILS

WHERE AUTHOR = 'Herbert Schildt'

AND BOOK\_EDITION = 5

AND SUPPLIER\_ID = 'S01';

-- 11

SELECT RACK\_NUM, COUNT(\*) AS NOOFBOOKS

FROM LMS\_BOOK\_DETAILS

GROUP BY RACK\_NUM

ORDER BY RACK\_NUM ASC;

-- 12

SELECT BOOK\_ISSUE\_NO, LMS\_MEMBERS.MEMBER\_NAME, LMS\_MEMBERS.DATE\_REGISTER, LMS\_MEMBERS.DATE\_EXPIRE,

LMS\_BOOK\_DETAILS.BOOK\_TITLE, LMS\_BOOK\_DETAILS.CATEGORY, LMS\_BOOK\_DETAILS.AUTHOR, LMS\_BOOK\_DETAILS.PRICE,

LMS\_BOOK\_ISSUE.DATE\_ISSUE, LMS\_BOOK\_ISSUE.DATE\_RETURN, LMS\_BOOK\_ISSUE.DATE\_RETURNED,

LMS\_BOOK\_ISSUE.FINE\_RANGE, LMS\_FINE\_DETAILS.FINE\_AMOUNT

FROM LMS\_BOOK\_ISSUE

JOIN LMS\_MEMBERS ON LMS\_BOOK\_ISSUE.MEMBER\_ID = LMS\_MEMBERS.MEMBER\_ID

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

JOIN LMS\_FINE\_DETAILS ON LMS\_BOOK\_ISSUE.FINE\_RANGE = LMS\_FINE\_DETAILS.FINE\_RANGE;

-- 13

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

FROM LMS\_BOOK\_DETAILS

WHERE MONTH(PUBLISH\_DATE) = 12;

-- 14

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

);

-- 15

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 questions:

-- 1

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 SUPPLIER\_ID

FROM LMS\_BOOK\_DETAILS

GROUP BY SUPPLIER\_ID

ORDER BY COUNT(\*) DESC

LIMIT 1

);

-- 2

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 b.SUPPLIER\_ID, 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 SUPPLIER\_ID

FROM LMS\_BOOK\_DETAILS

GROUP BY SUPPLIER\_ID

ORDER BY COUNT(\*) ASC

LIMIT 1

);