

# **CSP362: Database Management System Assignment 1**

**Pratik Parmar 2016UCS0049**

# **Problem 1**

## **Entities & their Relationships**

1. User
  - (a) Borrows Book
  - (b) Borrows Periodical
  - (c) Has Messages
2. Author
  - (a) Has Paper
  - (b) Has Book
3. Book
  - (a) Written By Authors
  - (b) Has Tag
  - (c) Borrowed by User
  - (d) Has Publisher
4. Paper
  - (a) Has Author
  - (b) Part of Periodical
5. Periodical
  - (a) Borrowed by User
  - (b) Has Paper
  - (c) Has Publisher
  - (d) Has Tag
6. Tag
  - (a) Relates Book

- (b) Relates Periodical
7. Message

- (a) For User

8. Publisher

- (a) Has Books

- (b) Has Periodicals

## Problem 2

Database has been Normalized to 2nd Normalization Form.

## Problem 3

### Implementation of Various Tables in Database

```

1      Create Database LMS_0049;
2
3      USE LMS_0049;
4
5      CREATE TABLE `User` (
6          `Id` varchar(20) NOT NULL,
7          `Name` varchar(50) NOT NULL,
8          `Username` varchar(50) NOT NULL,
9          `Password` varchar(20) NOT NULL,
10         `Email` varchar(50) NOT NULL,
11         PRIMARY KEY (`Id`),
12         UNIQUE KEY `Username` (`Username`)
13     );
14
15     CREATE TABLE `Message` (
16         `Id` varchar(20) NOT NULL,
17         `Text` varchar(200) NOT NULL,
18         `UserId` varchar(20) NOT NULL,
19         PRIMARY KEY (`Id`),
20         KEY `UserId` (`UserId`),
21         CONSTRAINT `Message_ibfk_1`
22             FOREIGN KEY (`UserId`) REFERENCES `User` (`Id`)
23     );
24
25     CREATE TABLE `Publisher` (
26         `Id` varchar(30) NOT NULL,
27         `Name` varchar(50) NOT NULL,
28         PRIMARY KEY (`Id`)
29     );
30
31     CREATE TABLE `Book` (
32         `Id` varchar(20) NOT NULL,
33         `Title` varchar(50) NOT NULL,
34         `Year` varchar(10) DEFAULT NULL,
35         `Isbn` varchar(30) DEFAULT NULL,
36         `Pages` int(11) DEFAULT '0',
37         `UserId` varchar(20) DEFAULT NULL,
38         `PublisherId` varchar(20) DEFAULT NULL,

```

```

39      'Issue_date' date DEFAULT NULL,
40      PRIMARY KEY ('Id'),
41      KEY 'UserId' ('UserId'),
42      KEY 'PublisherId' ('PublisherId'),
43      CONSTRAINT 'Book_ibfk_1'
44      FOREIGN KEY ('UserId') REFERENCES 'User' ('Id'),
45      CONSTRAINT 'Book_ibfk_2'
46      FOREIGN KEY ('PublisherId') REFERENCES 'Publisher' ('Id')
47 );
48
49 CREATE TABLE 'Author' (
50     'Id' varchar(20) NOT NULL,
51     'Name' varchar(50) NOT NULL,
52     'PaperId' varchar(20) DEFAULT NULL,
53     'BookId' varchar(20) DEFAULT NULL,
54     PRIMARY KEY ('Id'),
55     KEY 'BookId' ('BookId'),
56     KEY 'PaperId' ('PaperId'),
57     CONSTRAINT 'Author_ibfk_1'
58     FOREIGN KEY ('BookId') REFERENCES 'Book' ('Id'),
59     CONSTRAINT 'Author_ibfk_2'
60     FOREIGN KEY ('PaperId') REFERENCES 'Paper' ('Id')
61 );
62
63 CREATE TABLE 'Periodical' (
64     'Id' varchar(20) NOT NULL,
65     'Title' varchar(50) NOT NULL,
66     'Year' varchar(10) DEFAULT NULL,
67     'Volume' int(11) NOT NULL,
68     'Isbn' varchar(30) NOT NULL,
69     'UserId' varchar(20) DEFAULT NULL,
70     'PublisherId' varchar(20) DEFAULT NULL,
71     PRIMARY KEY ('Id'),
72     KEY 'UserId' ('UserId'),
73     KEY 'PublisherId' ('PublisherId'),
74     CONSTRAINT 'Periodical_ibfk_1'
75     FOREIGN KEY ('UserId') REFERENCES 'User' ('Id'),
76     CONSTRAINT 'Periodical_ibfk_2'
77     FOREIGN KEY ('PublisherId') REFERENCES 'Publisher' ('Id')
78 );
79
80 CREATE TABLE 'Paper' (
81     'Id' varchar(20) NOT NULL,
82     'Name' varchar(50) NOT NULL,
83     'PeriodicalId' varchar(20) DEFAULT NULL,
84     PRIMARY KEY ('Id'),
85     KEY 'PeriodicalId' ('PeriodicalId'),
86     CONSTRAINT 'Paper_ibfk_1'
87     FOREIGN KEY ('PeriodicalId') REFERENCES 'Periodical' ('Id')
88 );
89
90 CREATE TABLE 'Tag' (
91     'Id' varchar(20) NOT NULL,
92     'Value' varchar(50) NOT NULL,
93     PRIMARY KEY ('Id')
94 );
95
96 CREATE TABLE 'BookTag' (
97     'BookId' varchar(20) NOT NULL,
98     'TagId' varchar(20) NOT NULL,
99     PRIMARY KEY ('BookId', 'TagId'),
100    KEY 'TagId' ('TagId'),
101    CONSTRAINT 'BookTag_ibfk_1'
102    FOREIGN KEY ('BookId') REFERENCES 'Book' ('Id'),
103    CONSTRAINT 'BookTag_ibfk_2'
104    FOREIGN KEY ('TagId') REFERENCES 'Tag' ('Id')
105 );
106
107 CREATE TABLE 'PeriodicalTag' (
108     'PeriodicalId' varchar(20) NOT NULL,
109     'TagId' varchar(20) NOT NULL,
110     PRIMARY KEY ('PeriodicalId', 'TagId'),
111     KEY 'TagId' ('TagId'),
112     CONSTRAINT 'PeriodicalTag_ibfk_1'
113     FOREIGN KEY ('PeriodicalId') REFERENCES 'Periodical' ('Id'),
114     CONSTRAINT 'PeriodicalTag_ibfk_2'
115     FOREIGN KEY ('TagId') REFERENCES 'Tag' ('Id')

```

```

116 );
117
118 Desc User;
119
120 +-----+-----+-----+-----+-----+
121 | Field | Type | Null | Key | Default |
122 +-----+-----+-----+-----+-----+
123 | Id | varchar(20) | NO | PRI | NULL |
124 | Name | varchar(50) | NO | NULL | NULL |
125 | Username | varchar(50) | NO | UNI | NULL |
126 | Password | varchar(20) | NO | NULL | NULL |
127 | Email | varchar(50) | NO | NULL | NULL |
128
129 Desc Publisher;
130
131 +-----+-----+-----+-----+-----+
132 | Field | Type | Null | Key | Default | Extra |
133 +-----+-----+-----+-----+-----+
134 | Id | varchar(30) | NO | PRI | NULL |
135 | Name | varchar(50) | NO | NULL | |
136
137 Desc Book;
138
139 +-----+-----+-----+-----+-----+-----+
140 | Field | Type | Null | Key | Default | Extra |
141 +-----+-----+-----+-----+-----+
142 | Id | varchar(20) | NO | PRI | NULL |
143 | Title | varchar(50) | NO | NULL | NULL |
144 | Year | varchar(10) | YES | NULL | NULL |
145 | Isbn | varchar(30) | YES | NULL | NULL |
146 | Pages | int(11) | YES | 0 | NULL |
147 | UserId | varchar(20) | YES | MUL | NULL |
148 | PublisherId | varchar(20) | YES | MUL | NULL |
149 | Issue_date | date | YES | NULL | NULL |
150
151 Desc Tag;
152
153 +-----+-----+-----+-----+-----+
154 | Field | Type | Null | Key | Default | Extra |
155 +-----+-----+-----+-----+
156 | Id | varchar(20) | NO | PRI | NULL |
157 | Value | varchar(50) | NO | NULL | |
158
159 Desc Periodical;
160
161 +-----+-----+-----+-----+-----+-----+
162 | Field | Type | Null | Key | Default | Extra |
163 +-----+-----+-----+-----+
164 | Id | varchar(20) | NO | PRI | NULL |
165 | Title | varchar(50) | NO | NULL | NULL |
166 | Year | varchar(10) | YES | NULL | NULL |
167 | Volume | int(11) | NO | NULL | NULL |
168 | Isbn | varchar(30) | NO | NULL | NULL |
169 | UserId | varchar(20) | YES | MUL | NULL |
170 | PublisherId | varchar(20) | YES | MUL | NULL |
171
172 Desc Paper;
173
174 +-----+-----+-----+-----+-----+
175 | Field | Type | Null | Key | Default | Extra |
176 +-----+-----+-----+-----+
177 | Id | varchar(20) | NO | PRI | NULL |
178 | Name | varchar(50) | NO | NULL | NULL |
179 | PeriodicalId | varchar(20) | YES | MUL | NULL |
180
181 Desc Author;
182
183 +-----+-----+-----+-----+-----+
184 | Field | Type | Null | Key | Default | Extra |
185 +-----+-----+-----+-----+
186 | Id | varchar(20) | NO | PRI | NULL |
187 | Name | varchar(50) | NO | NULL | NULL |
188 | PaperId | varchar(20) | YES | MUL | NULL |
189 | BookId | varchar(20) | YES | MUL | NULL |
190
191 Desc Message;
192

```

```

193      | Field   | Type          | Null | Key | Default | Extra |
194      | Id      | varchar(20)  | NO   | PRI  | NULL    |       |
195      | Text    | varchar(200) | NO   | PRI  | NULL    |       |
196      | UserId  | varchar(20)  | NO   | MUL  | NULL    |       |
197
198
199
200      Desc BookTag
201      +-----+-----+-----+-----+-----+-----+
202      | Field   | Type          | Null | Key | Default | Extra |
203      |         |               |       |     |          |       |
204      | BookId  | varchar(20)  | NO   | PRI  | NULL    |       |
205      | TagId   | varchar(20)  | NO   | PRI  | NULL    |       |
206
207
208      Desc PeriodicalTag
209      +-----+-----+-----+-----+-----+-----+
210      | Field   | Type          | Null | Key | Default | Extra |
211      |         |               |       |     |          |       |
212      | PeriodicalId | varchar(20) | NO   | PRI  | NULL    |       |
213      | TagId   | varchar(20)  | NO   | PRI  | NULL    |       |
214

```

## 100 Records for Book and User Table

Data has been inserted using loop from frontend.

### Add User\_Type Attribute in User Table

```

1      Alter table User
2      Add Column User-Type enum("Student","Faculty","Staff","Guest") not null;

```

The students, faculties, staff members and guests are allowed a maximum of three, six, four and two books for 15, 30, 30 and 7 days respectively.

This Feature has been incorporated in frontend on LMS Application.

## Problem 4

### List All tables and their attributes

```

1      Show tables;
2      +-----+
3      | Tables_in_LMS_0049 |
4      +-----+
5      | Author
6      | Book
7      | BookTag
8      | Message
9      | Paper
10     | Periodical
11     | PeriodicalTag
12     | Publisher
13     | Tag
14     | User
15

```

Attributes of this table are already been shown.

Display the total inventory in the LMS.

```

1      mysql> Select count(*) from User;
2      +-----+
3      | count(*) |
4      +-----+
5      |      101 |
6      +-----+
7      1 row in set (0.00 sec)
8
9      mysql> Select count(*) from Book;
10     +-----+
11     | count(*) |
12     +-----+
13     |      201 |
14     +-----+
15     1 row in set (0.00 sec)
16
17     mysql> Select count(*) from Author;
18     +-----+
19     | count(*) |
20     +-----+
21     |      20 |
22     +-----+
23     1 row in set (0.00 sec)
24
25     mysql> Select count(*) from Periodical;
26     +-----+
27     | count(*) |
28     +-----+
29     |      1 |
30     +-----+
31     1 row in set (0.00 sec)

```

### **List the number of available books requested by a user.**

In this program, If Issue\_date field is null that means a book has not been issued to anyone.

```

1      mysql> Select count(*) from Book where Issue_date is null;
2      +-----+
3      | count(*) |
4      +-----+
5      |      199 |
6      +-----+
7      1 row in set (0.00 sec)

```

### **List the author(s) of a given (queried) book.**

Let the name of queried book is 'Book1'.

```

1      mysql> Select Author.Id , Author.Name from Book , Author
2      where Author.BookId=Book.Id and Book.Title="Book1";
3      +-----+
4      | Id | Name   |
5      +-----+
6      | A1 | Author1 |
7      | A2 | Author2 |
8      | A4 | Author4 |
9      | A5 | Author5 |
10     +-----+
11     4 rows in set (0.03 sec)

```

### **List the total number of books issued for a user.**

Let the name of queried user is 'User2'.

```

1 mysql> Select Book.* from Book,User
2      where Book.UserId=User.Id and User.Name="User2";
3
4 +----+-----+-----+-----+-----+-----+-----+-----+
5 | Id | Title | Year | Isbn | Pages | UserId | PublisherId | Issue_date | Discipline |
6 +----+-----+-----+-----+-----+-----+-----+-----+
7 | B1 | Book1 | 2018 | 12324 | 204 | U2 | P2 | 2019-02-20 | CSE
8 | B2 | Book2 | 2018 | 12324 | 407 | U2 | P3 | 2019-02-20 | CSE
9
10 2 rows in set (0.01 sec)
11
12 mysql> Select count(*) "#_of_Issued_Books" from Book,User where Book.UserId=User.Id and User.Name="User2";
13
14 +-----+
15 | # of Issued Books |
16 +-----+
17 | 2 |
18
19 1 row in set (0.00 sec)

```

**List the number of issued and returned books on daily basis (for a given day/period).**

```

1 mysql> Select Count(*) from Book where Issue_date=curdate();
2
3 +-----+
4 | Count(*) |
5 +-----+
6 | 1 |
7
8 1 row in set (0.00 sec)
9
10 mysql> Select Count(*) from Book where Return_date=curdate();
11
12 +-----+
13 | Count(*) |
14 +-----+
15 | 1 |
16
17 1 row in set (0.00 sec)
18
19 mysql> Select Count(*) from Book where Issue_date between curdate() and "20190101";
20
21 +-----+
22 | Count(*) |
23 +-----+
24 | 0 |
25
26 1 row in set (0.00 sec)

```

**List the users with book details if there are any dues.**

This Feature has been implemented in Software.

**Check whether a user is allowed to borrow a book or not.**

This Feature has been implemented in Software.

**List the newly added book records for a given period.**

To query this, we need to add a new column to determine the date of adding Book. Let the name of Column be EntryDate.

```

1 mysql> Select Book.* from Book
2      where EntryDate=curdate();

```

**Display the user name and the books issued to them..**

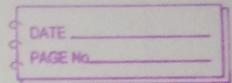
To query this, we need to add a new column to determine the date of adding Book. Let the name of Column be EntryDate.

```
1      mysql> Select Name,Book.Title from User,Book where Book.UserId=User.Id ;
2      +-----+-----+
3      | Name   | Title  |
4      +-----+-----+
5      | Pratik | Book10 |
6      +-----+-----+
7      1 row in set (0.00 sec)
```

**Display all the books with all details issued to a user.**

```
1      mysql> Select Book.* from Book,User where Book.UserId=User.Id and User.Name="Pratik_Parmar";
2      +-----+-----+-----+-----+-----+-----+-----+-----+
3      | Id   | Title | Year | Isbn | Pages | UserId | PublisherId | Issue_date | Discipline |
4      +-----+-----+-----+-----+-----+-----+-----+-----+
5      | B10  | Book10 | 2017 | 1324 | 1011 | U101  | P1          | 2019-02-24 | CSE        |
6      +-----+-----+-----+-----+-----+-----+-----+-----+
7      1 row in set (0.00 sec)
```

## Problem 5



→ Display the total inventory in LMS.

# Number of User

$\text{Count}(\star) \text{ (User)}$

# Number of Books

$\text{Count}(\star) \text{ (Book)}$

# Number of Author

$\text{Count}(\star) \text{ (Author)}$

# Number of Periodical

$\text{Count}(\star) \text{ (Periodical)}$

→ List the number of available books required by a user.

$\text{Count}(\star) \left( \sigma_{\text{Issue.Date} = \emptyset} \text{ (Book)} \right)$

→ List the authors of a given book.

$\Pi_{\text{Author.Id}, \text{Author.Name}} \left( \sigma_{\text{Author.BookId} = \text{Book.Id} \wedge \text{Book.Title} = \text{'Books'}} \text{ (Book X Author)} \right)$

→ List the total number of books issued for a user.

$\text{Count}(\star) \left( \sigma_{\text{User.Name} = \text{'User2'}} \text{ (Book} \bowtie \text{User)} \right)$

DATE \_\_\_\_\_  
PAGE NO. \_\_\_\_\_

- list the number of issued & returned books on the daily basis.

$\text{Gcount}(*)(\sigma_{\text{return\_date} = \text{getdate}()} "20190223" (\text{BOOK}))$

$\text{Gcount}(*)(\sigma_{\text{return\_date} = "20190223"} (\text{BOOK}))$

- list the newly added book records for a given period.

$\sigma_{\text{EntryDate} = "20190202"} (\text{BOOK})$

- Display the username ~~the~~ and the books issued to them.

$\pi_{\text{Name}, \text{Book.T.title}} (\sigma_{\text{Book.UserId} = \text{User.Id}} (\text{User} \times \text{BOOK}))$

- Display all the books with all details issued to a user.

$\sigma_{(\text{Book.UserId} = \text{User.Id}) \wedge (\text{User.Name} = "Pratik")} (\text{BOOK} \times \text{User})$

## Problem 6

Calculation of Fine is done in Software. A function has been created in SQL to calculate number of days from a given date excluding weekends.

```
1      CREATE FUNCTION TOTAL_WEEKDAYS(date1 DATE, date2 DATE)
2          RETURNS INT
3          RETURN ABS(DATEDIFF(date2, date1)) + 1
4          - ABS(DATEDIFF(ADDDATE(date2, INTERVAL 1 - DAYOFWEEK(date2) DAY),
5                          ADDDATE(date1, INTERVAL 1 - DAYOFWEEK(date1) DAY))) / 7 * 2
6          - (DAYOFWEEK(IF(date1 < date2, date1, date2)) = 1)
7          - (DAYOFWEEK(IF(date1 > date2, date1, date2)) = 7);
```

## Problem 7

```
1      mysql> Alter table Book
2          -> Add Column Discipline enum("CS", "EE", "ME", "CE", "Math", "Phy", "Hindi");
```

## Problem 8

Check Software.

## Problem 9

A Separate table Reserve has been created for this problem with 3 attributes 'UserId', 'BookId' and 'Date'.

```
1      mysql> desc Reserve;
2
3      +-----+-----+-----+-----+-----+-----+
4      | Field | Type  | Null | Key  | Default | Extra |
5      +-----+-----+-----+-----+-----+-----+
6      | UserId | varchar(20) | NO   | PRI  | NULL    |        |
7      | BookId | varchar(20) | NO   | PRI  | NULL    |        |
8      | Date   | date       | NO   |        | NULL    |        |
9
10     3 rows in set (0.00 sec)
```

Book automatically gets issued to person who reserved it based on first come first serve basis when that book is returned.