LIBRARY MANAGEMENT SYSTEM

(Computer Science & Engineering)

Registration ID	Name
122010326026	*Mehak Agarwal*
122010326025	*I. Sushanth*
122010326027	*M. Naga Pranay*
122010326024	*G.V. Supriya*
122010326028	*E. Manikanta*

Under the Guidance

of

Prof. Srinivas Prasad

GITAM School of Technology, Visakhapatnam

GITAM (Deemed to be University)

TABLE OF CONTENTS:

INTRODUCTION.....3

ER DIAGRAM.....4

ER Diagram to Table.....5

TABLE CREATION.....6

AGGREGATE FUNCTION QUERIES.....9

QUERIES.....12

TRIGGER.....18

CURSOR.....18

FUNCTION.....19

PROCEDURE.....19

FUTURE ADVANCEMENTS.....20

CONCLUSION.....20

REFERENCES.....21

INTRODUCTION:

AIM:

To keep track of all the information about the books of the library using SQL queries.

Description:

A library is a place where books and sources of information are stored. They make it easier for people to get access to them for various purposes. Libraries are very helpful and economical too. They include books, magazines, newspapers, DVDs, manuscripts, and more. The bulk collection means tough to handle and tedious to use all this stuff. With the growth of an advanced technological world, Library management must have to be digitized; this system will incorporate all competencies of the library for a librarian.

A **Library Management System** is software that is designed to manage all the functions of a library. It helps the librarian to maintain the reports of the books issued and returned by the members with their due dates.

Keeps track of all the books by recording the data of the issue date, due date, book_ID, Member name, and respective contact info, also recording the fine charges if not returned within the due date.

Library management systems can help libraries with their data management process. It automates many of the tasks that are usually done by a librarian. The most important part is that it not only improves data accuracy but also optimizes workflows for staff members.

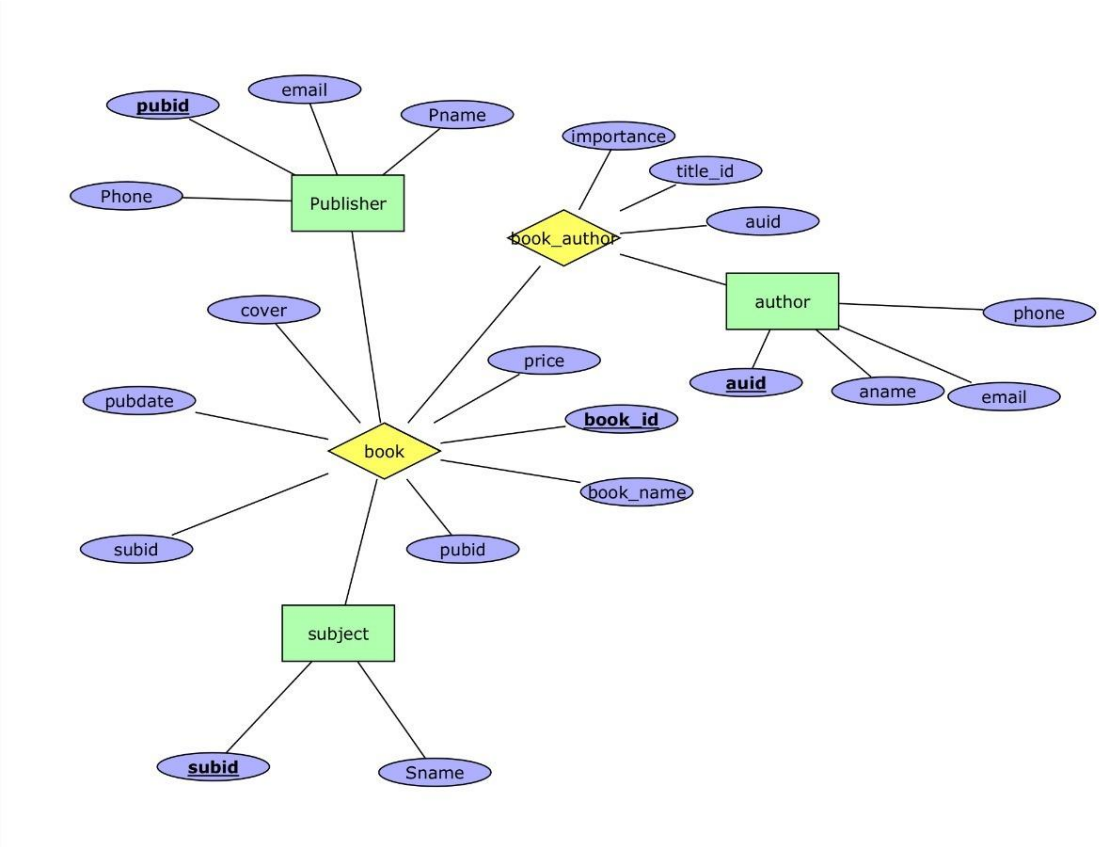
ENTIY RELATION DIAGRAM:

Entity Relationship Diagram – ER Diagram in DBMS. An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database.

There are three basic elements in an ER Diagram:

1. Entity
2. Attribute
3. Relationship.

ER Diagram of Library Management System:



E.R DIAGRAM to TABLE:

- Entity type becomes a table.
- All single-valued attribute becomes a column for the table.
- A key attribute of the entity type represented by the primary key.
- The multivalued attribute is represented by a separate table.
- Composite attribute represented by components.
- Derived attributes are not considered in the table.

CREATION OF TABLE:

Create PUBLISHER Table:

```
CREATE TABLE PUBLISHERS
(
    PUBID      NUMBER(3)  CONSTRAINT  PUBLISHERS_PK PRIMARY KEY,
    PNAME      VARCHAR2(30),
    EMAIL      VARCHAR2(50) CONSTRAINT PUBLISHERS_EMAIL_U UNIQUE,
    PHONE      VARCHAR2(30)
);
```

```
INSERT INTO PUBLISHERS VALUES (1,'WILLEY','WDT@VSNL.NET','91-11-23260877');
INSERT INTO PUBLISHERS VALUES (2,'WROX','INFO@WROX.COM',NULL);
INSERT INTO PUBLISHERS VALUES (3,'TATA MCGRAW-HILL','FEEDBACK@TATAMCGRAWHILL.COM','91-11-33333322');
INSERT INTO PUBLISHERS VALUES (4,'TECHMEDIA','BOOKS@TECHMEDIA.COM','91-11-33257660');
```

PUBID	PNAME	EMAIL	PHONE
1	WILLEY	WDT@VSNL.NET	91-11-23260877
2	WROX	INFO@WROX.COM	-
3	TATA MCGRAW-HILL	FEEDBACK@TATAMCGRAWHILL.COM	91-11-33333322
4	TECHMEDIA	BOOKS@TECHMEDIA.COM	91-11-33257660

Download CSV

4 rows selected.

Create AUTHORS Table:

```
CREATE TABLE  AUTHORS
(
    AUID      NUMBER(5)  CONSTRAINT  AUTHORS_PK PRIMARY KEY,
    ANAME       VARCHAR2(30),
    EMAIL       VARCHAR2(50) CONSTRAINT AUTHORS_EMAIL_U UNIQUE,
    PHONE       VARCHAR2(30)
);
```

```
INSERT INTO AUTHORS VALUES (101, 'HERBERT SCHILD','HERBERT@YAHOO.COM',NULL);
INSERT INTO AUTHORS VALUES (102, 'JAMES GOODWILL','GOODWILL@HOTMAIL.COM',NULL);
INSERT INTO AUTHORS VALUES (103, 'DAVAID HUNTER','HUNTER@HOTMAIL.COM',NULL);
INSERT INTO AUTHORS VALUES (104, 'STEPHEN WALTHER','WALTHER@GMAIL.COM',NULL);
INSERT INTO AUTHORS VALUES (105, 'KEVIN LONEY','LONEY@ORACLE.COM',NULL);
INSERT INTO AUTHORS VALUES (106, 'ED. ROMANS','ROMANS@THESERVERSIDE.COM',NULL);
```

AUID	ANAME	EMAIL	PHONE
101	HERBERT SCHILD	HERBERT@YAHOO.COM	-
102	JAMES GOODWILL	GOODWILL@HOTMAIL.COM	-
103	DAVAID HUNTER	HUNTER@HOTMAIL.COM	-
104	STEPHEN WALTHER	WALTHER@GMAIL.COM	-
105	KEVIN LONEY	LONEY@ORACLE.COM	-
106	ED. ROMANS	ROMANS@THESERVERSIDE.COM	-

Download CSV

6 rows selected.

Create SUBJECTS Table:

```
17  CREATE TABLE  SUBJECTS
18  (
19      SUBID       VARCHAR2(5)  CONSTRAINT  SUBJECTS_PK PRIMARY KEY,
20      SNAME       VARCHAR2(30)
21  );
22
```

```
46  INSERT INTO SUBJECTS VALUES ('ORA','ORACLE DATABASE 10g');
47  INSERT INTO SUBJECTS VALUES ('JAVA','JAVA LANGUAGE');
48  INSERT INTO SUBJECTS VALUES ('JEE','JAVA ENTEPRISE EDITION');
49  INSERT INTO SUBJECTS VALUES ('VB','VISUAL BASIC.NET');
50  INSERT INTO SUBJECTS VALUES ('ASP','ASP.NET');
51
```

SUBID	SNAME
ORA	ORACLE DATABASE 10g
JAVA	JAVA LANGUAGE
JEE	JAVA ENTEPRISE EDITION
VB	VISUAL BASIC.NET
ASP	ASP.NET

Download CSV

5 rows selected.

Create BOOK Table:

```
23  CREATE TABLE  BOOK
24  (
25      BOOK_ID     NUMBER(5)     PRIMARY KEY,
26      BOOK_NAME VARCHAR2(30),
27      PUBID       NUMBER(3),
28      SUBID       VARCHAR2(5),
29      PUBDATE     DATE ,
30      COVER       CHAR(1),
31      PRICE       NUMBER(4),
32      foreign key(PUBID) REFERENCES PUBLISHERS(PUBID),
33      foreign key(SUBID) REFERENCES SUBJECTS(SUBID)
34  );
35
```

```
64  INSERT INTO BOOK VALUES (1001,'ASP.NET UNLEASHED',4,'ASP','12-APR-02','P',540);
65  INSERT INTO BOOK VALUES (1002,'ORACLE10G COMP. REF.',3,'ORA','1-MAY-05','P',575);
66  INSERT INTO BOOK VALUES (1003,'MASTERING EJB',1,'JEE','3-FEB-05','P',475);
67  INSERT INTO BOOK VALUES (1004,'JAVA COMP. REF',3,'JAVA','3-APR-05','P',499);
68  INSERT INTO BOOK VALUES (1005,'PRO. VB.NET',2,'VB','15-JUN-05','P',450);
69
```

BOOK_ID	BOOK_NAME	PUBID	SUBID	PUBDATE	COVER	PRICE
1001	ASP.NET UNLEASHED	4	ASP	12-APR-02	P	540
1002	ORACLE10G COMP. REF.	3	ORA	01-MAY-05	P	575
1003	MASTERING EJB	1	JEE	03-FEB-05	P	475
1004	JAVA COMP. REF	3	JAVA	03-APR-05	P	499
1005	PRO. VB.NET	2	VB	15-JUN-05	P	450

Download CSV

5 rows selected.

Create BOOK_AUTHORS Table:

```
36  CREATE TABLE  BOOK_AUTHORS
37  (
38      BOOK_ID    NUMBER(5),
39      AUID       NUMBER(5),
40      IMPORTANCE NUMBER(2),
41      CONSTRAINT  BOOK_AUTHORS_PK  PRIMARY KEY(BOOK_ID,AUID),
42      foreign key(BOOK_ID) REFERENCES BOOK(BOOK_ID),
43      foreign key(AUID) REFERENCES AUTHORS(AUID)
44  );
45
```

```
70  INSERT INTO BOOK_AUTHORS VALUES (1001,104,1);
71  INSERT INTO BOOK_AUTHORS VALUES (1002,105,1);
72  INSERT INTO BOOK_AUTHORS VALUES (1003,106,1);
73  INSERT INTO BOOK_AUTHORS VALUES (1004,101,1);
74  INSERT INTO BOOK_AUTHORS VALUES (1005,103,1);
75  INSERT INTO BOOK_AUTHORS VALUES (1005,102,2);
76
```

BOOK_ID	AUID	IMPORTANCE
1001	104	1
1002	105	1
1003	106	1
1004	101	1
1005	103	1
1005	102	2

Download CSV

6 rows selected.

AGGREGATE FUNCTION QUERIES:

1). Display publisher ID and maximum price for books on the subject ORACLE.

```
93  SELECT PUBID, max(price)
94  FROM BOOK
95  WHERE SUBID = 'ORA'
96  GROUP BY PUBID;
97
98
```

PUBID	MAX(PRICE)
3	575

Download CSV

2). Display the Subject ID and min price.

```
99  SELECT subid, min(price)
100 FROM BOOK
101 GROUP BY subid;
102
103
```

SUBID	MIN(PRICE)
ORA	575
JAVA	499
ASP	540
JEE	475
VB	450

Download CSV

5 rows selected.

3). Display Subject ID, Publisher ID and average price.

```
104  SELECT subid,pubid, AVG(price)
105  FROM BOOK
106  GROUP BY subid,pubid;
107
108
```

SUBID	PUBID	AVG(PRICE)
JEE	1	575
JAVA	3	499
VB	2	450
ASP	4	540
ORA	3	575

Download CSV

5 rows selected.

4). Display subname and average price of the books.

```
109  SELECT sname,AVG(price)
110  FROM SUBJECTS s, BOOK t
111  WHERE s.subid = t.subid
112  GROUP BY sname;
113
```

SNAME	AVG(PRICE)
JAVA LANGUAGE	499
ASP.NET	540
ORACLE DATABASE 10g	575
JAVA ENTEPRISE EDITION	475
VISUAL BASIC.NET	450

Download CSV

5 rows selected.

5). Display the maximum price of all books.

```
114  SELECT BOOK_NAME, max(price)
115  FROM BOOK
116  GROUP BY (BOOK_NAME);
117
```

BOOK_NAME	MAX(PRICE)
ORACLE10G COMP. REF.	575
PRO. VB.NET	450
ASP.NET UNLEASHED	540
MASTERING EJB	475
JAVA COMP. REF	499

Download CSV

5 rows selected.

QUERIES:

1). Display names of the authors who wrote a book with a price<500

```
118  SELECT aname FROM AUTHORS
119  WHERE auid in (SELECT auid FROM BOOK_AUTHORS WHERE BOOK_ID in
120                      (SELECT BOOK_ID FROM BOOK WHERE price < 500) );
121
122
```

ANAME
HERBERT SCHILD
JAMES GOODWILL
DAVAID HUNTER
ED. ROMANS

Download CSV

4 rows selected.

2). Display Book name published by the publisher whose name starts with 'w'.

```
123  SELECT BOOK_NAME FROM BOOK
124  WHERE pubid in
125      ( SELECT pubid FROM PUBLISHERS WHERE pname like 'W%');
126
```

BOOK_NAME
MASTERING EJB
PRO. VB.NET

Download CSV

2 rows selected.

3). Display Book published by the publisher whose name contains the letter 'w'.

```
127  SELECT BOOK_NAME FROM BOOK
128  WHERE pubid in
129      ( SELECT pubid FROM PUBLISHERS WHERE pname like '%W%');
130
```

BOOK_NAME
ORACLE10G COMP. REF.
MASTERING EJB
JAVA COMP. REF
PRO. VB.NET

Download CSV

4 rows selected.

4). Change price of the Book id 1003 to average price of the Book.

```
131  UPDATE BOOK set price = ( SELECT AVG(price) FROM BOOK)
132  WHERE BOOK_ID = 1003;
133  SELECT * FROM BOOK
134
```

1 row(s) updated.

BOOK_ID	BOOK_NAME	PUBID	SUBID	PUBDATE	COVER	PRICE
1001	ASP.NET UNLEASHED	4	ASP	12-APR-02	P	540
1002	ORACLE10G COMP. REF.	3	ORA	01-MAY-05	P	575
1003	MASTERING EJB	1	JEE	03-FEB-05	P	508
1004	JAVA COMP. REF	3	JAVA	03-APR-05	P	499
1005	PRO. VB.NET	2	VB	15-JUN-05	P	450

Download CSV

5 rows selected.

5). Delete authors who have not written any Book.

```
135  DELETE FROM AUTHORS
136  WHERE auid not in ( SELECT auid FROM BOOK_AUTHORS);
137  SELECT * FROM AUTHORS
138
```

0 row(s) deleted.

AUID	ANAME	EMAIL	PHONE
101	HERBERT SCHILD	HERBERT@YAHOO.COM	-
102	JAMES GOODWILL	GOODWILL@HOTMAIL.COM	-
103	DAVAID HUNTER	HUNTER@HOTMAIL.COM	-
104	STEPHEN WALTHER	WALTHER@GMAIL.COM	-
105	KEVIN LONEY	LONEY@ORACLE.COM	-
106	ED. ROMANS	ROMANS@THESERVERSIDE.COM	-

Download CSV

6 rows selected.

6). Display pubname, and Book name of all Book written by author 103.

```
140  SELECT pname,BOOK_NAME FROM BOOK t, PUBLISHERS p
141  WHERE t.pubid = p.pubid and  BOOK_ID in
142     ( SELECT BOOK_ID FROM BOOK_AUTHORS WHERE auid = 103);
143
144
```

PNAME	BOOK_NAME
WROX	PRO. VB.NET

Download CSV

7). Display the publishers who published a book on ORACLE but no book on JAVA.

```
145  SELECT  * FROM PUBLISHERS
146  WHERE pubid in
147    ( SELECT pubid
148      FROM  BOOK
149      WHERE subid = 'ORA' and subid != 'JAVA');
150
151
```

PUBID	PNAME	EMAIL	PHONE
3	TATA MCGRAW-HILL	FEEDBACK@TATAMCGRAWHILL.COM	91-11-33333322

Download CSV

8). Display the year the highest price by taking into account books that are written by more than one author.

```
152  SELECT to_char(pubdate,'yyyy') , max(price)
153  FROM BOOK
154  WHERE  BOOK_ID in
155       (SELECT BOOK_ID
156        FROM BOOK_AUTHORS
157        GROUP BY BOOK_ID
158        having count(*) > 1)
159  GROUP BY to_char(pubdate,'yyyy');
160
161
```

TO_CHAR(PUBDATE,'YYYY')	MAX(PRICE)
2005	450

Download CSV

9). Change the Publisher date of Book id 1003 to the least publishing date of any book of that subject in the same year.

```
164
165  UPDATE BOOK t set pubdate =
166     ( SELECT min(pubdate)
167       FROM BOOK WHERE  subid = t.subid
168       and  to_char(pubdate,'yyyy') = to_char(t.pubdate,'yyyy')
169     )
170  WHERE BOOK_ID = 1003;
171
172  SELECT * FROM BOOK
173
```

```
1 row(s) updated.
```

BOOK_ID	BOOK_NAME	PUBID	SUBID	PUBDATE	COVER	PRICE
1001	ASP.NET UNLEASHED	4	ASP	12-APR-02	P	540
1002	ORACLE10G COMP. REF.	3	ORA	01-MAY-05	P	575
1003	MASTERING EJB	1	JEE	03-FEB-05	P	508
1004	JAVA COMP. REF	3	JAVA	03-APR-05	P	499
1005	PRO. VB.NET	2	VB	15-JUN-05	P	450

Download CSV

5 rows selected.

10). Display author's names who has written any book with the author 'HERBERT SCHILD'.

```
175
176  SELECT aname FROM AUTHORS WHERE auid in
177  (SELECT auid FROM BOOK_AUTHORS WHERE BOOK_ID in
178    (SELECT BOOK_ID FROM BOOK_AUTHORS WHERE auid=
179          (SELECT auid FROM AUTHORS WHERE aname='HERBERT SCHILD')
180    )
181  );
182
```

ANAME
HERBERT SCHILD

Download CSV

11). Change the price of book-1003 to the average price of "oracle books".

```
183
184  UPDATE BOOK set price=(SELECT  AVG(price)FROM SUBJECTS s,BOOK B
185  WHERE s.subid=B.subid and sname like '%ORA%')
186  WHERE BOOK_ID=1003;
187  SELECT * FROM BOOK
188
```

```
1 row(s) updated.
```

BOOK_ID	BOOK_NAME	PUBID	SUBID	PUBDATE	COVER	PRICE
1001	ASP.NET UNLEASHED	4	ASP	12-APR-02	P	540
1002	ORACLE10G COMP. REF.	3	ORA	01-MAY-05	P	575
1003	MASTERING EJB	1	JEE	03-FEB-05	P	575
1004	JAVA COMP. REF	3	JAVA	03-APR-05	P	499
1005	PRO. VB.NET	2	VB	15-JUN-05	P	450

Download CSV

5 rows selected.

12). Display pubid,no.of books published where the book is written by more than one author.

```
190
191  SELECT pubid,count(*) FROM BOOK WHERE
192  BOOK_ID in(SELECT BOOK_ID FROM BOOK_AUTHORS GROUP BY BOOK_ID
193  having count(*)>1)
194  GROUP BY(pubid);
195
```

PUBID	COUNT(*)
2	1

Download CSV

13). Display the author who has written the costliest book.

```
196
197 SELECT * FROM AUTHORS
198 WHERE auid  =
199   ( SELECT auid FROM BOOK_AUTHORS
200     WHERE BOOK_ID =
201           (SELECT BOOK_ID FROM BOOK
202            WHERE price =
203                  (SELECT max(price) FROM BOOK)
204           )
205   );
206
```

AUID	ANAME	EMAIL	PHONE
105	KEVIN LONEY	LONEY@ORACLE.COM	-
106	ED. ROMANS	ROMANS@THESERVERSIDE.COM	-

Download CSV

2 rows selected.

14). Display the Publisher's name a and the number of books published.

```
201
202 SELECT pname, count(*)
203 FROM   PUBLISHERS p, BOOK t
204 WHERE  p.pubid = t.pubid
205 group  by pname;
206
207
```

PNAME	COUNT(*)
TECHMEDIA	1
WROX	1
WILLEY	1
TATA MCGRAW-HILL	2

Download CSV

4 rows selected.

TRIGGERS:

```
207  CREATE OR REPLACE TRIGGER DISPLAY_PUBLISHER_CHANGES
208  BEFORE INSERT ON PUBLISHERS
209  FOR EACH ROW
210  WHEN (NEW.PUBID > 0)
211  DECLARE
212  BEGIN
213  DBMS_OUTPUT.PUT_LINE('NEW ROW HAS BEEN INSERTED');
214  END;
215
216  INSERT INTO PUBLISHERS VALUES(5,'SUSHANTH','SINAMPUD@GITAM.IN','+91-7675003287');
217
```

```
Trigger created.


1 row(s) inserted.
NEW ROW HAS BEEN INSERTED
```

CURSORS:

```
220  --CURSOR
221
222  DECLARE
223  C_PUBID PUBLISHERS.PUBID%type;
224  C_PNAME PUBLISHERS.PNAME%type;
225  C_EMAIL PUBLISHERS.EMAIL%type;
226  C_PHONE PUBLISHERS.PHONE%type;
227  CURSOR PRANAY is
228  SELECT PUBID,PNAME,EMAIL,PHONE FROM PUBLISHERS;
229  BEGIN
230  OPEN PRANAY;
231  LOOP
232  FETCH PRANAY into C_PUBID, C_PNAME, C_EMAIL,C_PHONE;
233  EXIT WHEN PRANAY%notfound;
234  DBMS_OUTPUT.PUT_LINE
235  (' PUBID IS'||C_PUBID||'WHOSE NAME IS '||C_PNAME||' AND EMAIL IS '||C_EMAIL||' AND PHONE NUMBER IS '||C_PHONE);
236  END LOOP;
237  CLOSE PRANAY;
238  END;
239
```

```
Statement processed.
 PUBID IS1WHOSE NAME IS WILLEY AND EMAIL IS WDT@VSNL.NET AND PHONE NUMBER IS 91-11-23260877
 PUBID IS2WHOSE NAME IS WROX AND EMAIL IS INFO@WROX.COM AND PHONE NUMBER IS
 PUBID IS3WHOSE NAME IS TATA MCGRAW-HILL AND EMAIL IS FEEDBACK@TATAMCGRAWHILL.COM AND PHONE NUMBER IS 91-11-33333322
 PUBID IS4WHOSE NAME IS TECHMEDIA AND EMAIL IS BOOKS@TECHMEDIA.COM AND PHONE NUMBER IS 91-11-33257660
 PUBID IS5WHOSE NAME IS SUSHANTH AND EMAIL IS SINAMPUD@GITAM.IN AND PHONE NUMBER IS +91-7675003287
```

FUNCTIONS:

```
DECLARE
X NUMBER(2);
FUNCTION TOTAL_PUBLISHERS
RETURN NUMBER IS
TOTAL NUMBER(2):=0;
BEGIN
SELECT COUNT(*) INTO TOTAL
FROM PUBLISHERS;
RETURN TOTAL;
END;
BEGIN
X := TOTAL_PUBLISHERS();
DBMS_OUTPUT.PUT_LINE
('TOTAL NUMBER OF PUBLISHERS = '||X);
END;
```

```
Statement processed.
TOTAL NUMBER OF PUBLISHERS = 5
```

PROCEDURE:

```
DECLARE
P_PUBID PUBLISHERS.PUBID%TYPE;
P_PNAME PUBLISHERS.PNAME%TYPE;
P_EMAIL PUBLISHERS.EMAIL%TYPE;
P_PHONE PUBLISHERS.PHONE%TYPE;
PROCEDURE GET_DETAILS (X IN INT) IS
BEGIN
SELECT PUBID,PNAME,EMAIL,PHONE INTO P_PUBID,P_PNAME,P_EMAIL,P_PHONE
FROM PUBLISHERS
WHERE PUBID=X;
DBMS_OUTPUT.PUT_LINE
(' PUBID = '||P_PUBID||' PNAME = '||P_PNAME||' EMAIL = '||P_EMAIL||' PHONE = '||P_PHONE);
END;
BEGIN
GET_DETAILS(4);
END;
```

```
Statement processed.
 PUBID = 4 PNAME = TECHMEDIA EMAIL = BOOKS@TECHMEDIA.COM PHONE = 91-11-33257660
```

FUTURE ADVANCEMENTS:

- This concept will be furthermore explored such that there doesn't occur any false entries.
- Should be safe from cyber hacking.
- Will reduce the manual work of librarians and staff.
- The automated library software is user-friendly, powerful and developed for easy entry of data, makes library operations free from errors.

CONCLUSION:

The library management system is essential for colleges, schools, and many more places these days. A lot of manual work can be reduced with this library management system. And also, a lot of glitches like wrong borrow date and miscalculation of fine amount are avoided. As it is a computer-managed system and so these are all avoided. It is also efficient and cost-effective. The Library management system stores the details of books and also details of persons. So overall, we have seen-

- To build a database to maintain all the related information
- We built tables separately to store data.
- Learned the purpose of the library management system.
- What features are required for students and librarians to use LMS?
- We have seen all the implementations using MYSQL
- And how the software allows storing all the details related to the library.
- Finally, we tested the final database.

REFERENCES:

- ✓ https://studentprojectguide.com/project-report/library-management-system-case-study/
- ✓ http://srikanthtechnologies.com/oracle/catalog/catalog.html
- ✓ https://www.javatpoint.com/pl-sql-procedure
- ✓ https://www.javatpoint.com/cursor-in-sql
- ✓ https://www.javatpoint.com/triggers-in-sql-server