‘PATEL GROUP’

Library Management System Database

Monroe College

**INDEX**

1. ABSTRACT 3
2. ER DIAGRAM
3. SQL QUERIES

INSERT

ALTER

1. NESTED QUERIES 21
2. QUERIES FOR JOINS 26
3. PROCEDURE AND VIEWS

**Abstract**

Library Managing System is a web based application. The fundamental subject of the database is to store the details about books in a library and keep track of books the books in the library and out the library. Using this database we can easily find the book as per our input source. This will diminish the weight on clients who will keep up all records physically. General library administration is the most complex thing as it includes a great deal of database with in that. Overseeing Books in classification wise dispensing books to users is finished by manual procedure; computerization of this procedure will enhance the execution and decrease the mistakes.

Purpose of the Project

The Library Management System helps to improve efficiency of admin to keep track of all books easily. In addition, just searching with any one criterion they can find the book status.

Tables for Library Management System

1. L\_STUDENT
2. L\_STAFF
3. L\_DEPARTMENT
4. L\_BOOK
5. L\_AUTHOR
6. L\_ISSUEDETAILS
7. L\_PUBLISHER

We have created seven tables and the beginning letter with underscore represents tables for Library Management System (**L**). At the beginning, we created table with primary key constraint but the foreign key constraint was added after values added into each tables. Additionally, we added genuine values to the each tables to make the project error resist. Furthermore, Constraints provides one methods of implementing business rules in the database. Constraint restricts the data that stored in the relation. Many constraints like NOT NULL, FOREING KEY, UNIQUE, CHECK are used in this project. These help the users and admin both.

**Normalized each tables for minimize data redundancy**

We started with first normalized form of each tables in which we defined the primary key and identified dependencies for each tables. In addition, in the first normalization form, we eliminated repeating groups. Secondly, it is required for the new tables to remove partial dependencies. Afterwards, we had assigned corresponding attribute for dependent. At last, in third normalization, it is necessary to remove transitive dependency. So we removed it and again reassigned the corresponding attribute for dependant.

**SQL (Structural Query Language)**

Here, it used the query to confirm the users’ requirements. First of all, DML (Data Manipulation Language) which was used here to insert new rows and removing rows from the existing tables. Secondly, DDL (Data Definition Language) which helped us to create the tables, drop the tables. These are CREATE, ALTER, DROP and RENAME. Also, this DDL helps us to insert the appropriate data in to each table.

Queries used are listed below

/\*Student Table Created\*/

CREATE TABLE L\_STUDENT

(STUDENTID VARCHAR(10) CONSTRAINT L\_STUDENT\_STUDENTID\_PK PRIMARY KEY,

FNAME VARCHAR(20) CONSTRAINT L\_STUDENT\_FNAME\_NN NOT NULL,

LNAME VARCHAR(20) CONSTRAINT L\_STUDENT\_LNAME\_NN NOT NULL,

BIRTHDATE DATE ,

ADDRESS VARCHAR(200) CONSTRAINT L\_STUDENT\_STATE\_UK UNIQUE,

EMAILID VARCHAR(30) CONSTRAINT L\_STUDENT\_EMAILID\_UK UNIQUE,

MOBILENUM CHAR(15) CONSTRAINT L\_STUDENT\_MOBILENUM\_NN NOT NULL,

DEPTID VARCHAR(10),

CURRENTYEAR VARCHAR(30) CONSTRAINT L\_STUDENT\_CURRENTYEAR\_NN NOT NULL,

SECURITY\_Q VARCHAR(30) CONSTRAINT L\_STUDENT\_SECURITY\_Q\_NN NOT NULL,

SECURITY\_A VARCHAR(30) CONSTRAINT L\_STUDENT\_SECURITY\_A\_NN NOT NULL,

REGI\_DATE DATE DEFAULT SYSDATE,

USER\_TYPE CHAR(2) CONSTRAINT L\_STUDENT\_USER\_TYPE\_CK CHECK(USER\_TYPE IN('AD','SM','ST'))

);

/\*Create Department Table\*/

CREATE TABLE L\_DEPARTMENT

(DEPTID VARCHAR(10) CONSTRAINT L\_DEPARTMENT\_DEPTID\_PK PRIMARY KEY,

DEPTNAME VARCHAR(30) CONSTRAINT L\_DEPARTMENT\_DEPTNAME\_NN NOT NULL,

STUDENTID VARCHAR(10),

STAFFID VARCHAR(10)

);

/\*Create BOOK TABLE\*/

CREATE TABLE L\_BOOK

(BOOKNUM VARCHAR(10) CONSTRAINT L\_BOOK\_BOOKNUM\_PK PRIMARY KEY,

BOOKTITLE VARCHAR(30) CONSTRAINT L\_BOOK\_BOOKTITLE\_NN NOT NULL,

ISBNNUM VARCHAR(30) CONSTRAINT L\_BOOK\_ISBNNUMBER\_UK UNIQUE,

AUTHOR\_ID VARCHAR(10),

PUBLISHER\_ID VARCHAR(10),

DEPTID VARCHAR(10),

COPIESLEFT CHAR(1) CONSTRAINT L\_BOOK\_COPIESLEFT\_CK CHECK (COPIESLEFT IN ('Y','N'))

);

/\*Create Staff-Member Table\*/

CREATE TABLE L\_STAFF

(STAFFID VARCHAR(10) CONSTRAINT L\_STAFF\_STAFFID\_PK PRIMARY KEY,

FNAME VARCHAR(20) CONSTRAINT L\_STAFF\_FNAME\_NN NOT NULL,

LNAME VARCHAR(20) CONSTRAINT L\_STAFF\_LNAME\_NN NOT NULL,

DEPTID VARCHAR(10),

BIRTHDATE DATE ,

ADDRESS VARCHAR(200) CONSTRAINT L\_STAFF\_STATE\_UK UNIQUE,

EMAILID VARCHAR(30) CONSTRAINT L\_STAFF\_EMAILID\_UK UNIQUE,

MOBILENUM CHAR(15) CONSTRAINT L\_STAFF\_MOBILENUM\_NN NOT NULL,

SECURITY\_Q VARCHAR(30) CONSTRAINT L\_STAFF\_SECURITY\_Q\_NN NOT NULL,

SECURITY\_A VARCHAR(30) CONSTRAINT L\_STAFF\_SECURITY\_A\_NN NOT NULL,

REGI\_DATE DATE DEFAULT SYSDATE,

USER\_TYPE CHAR(2) CONSTRAINT L\_STAFF\_USER\_TYPE\_CK CHECK(USER\_TYPE IN('AD','SM','ST'))

);

/\*Create Book Issue Details Table\*/

CREATE TABLE L\_ISSUEDETAILS

(ISSUENUM VARCHAR(30) CONSTRAINT L\_ISSUEDETAILS\_ISSUENAME\_PK PRIMARY KEY,

ISSUE\_DATE DATE CONSTRAINT L\_ISSUEDETAILS\_ISSUEDATE\_NN NOT NULL,

BOOKNUM VARCHAR(10),

DUE\_DATE DATE CONSTRAINT L\_ISSUEDETAILS\_DUEDATE\_NN NOT NULL,

RETURN\_DATE DATE CONSTRAINT L\_ISSUEDETAILS\_RETURN\_DATE\_NN NOT NULL,

STUDENTID VARCHAR(10),

STAFFID VARCHAR(10),

DEPTID VARCHAR(10)

);

/\*Create Publisher Details Table\*/

CREATE TABLE L\_PUBLISHER

(PUBLISHER\_ID VARCHAR(10) CONSTRAINT L\_PUBLISHER\_PUBLISHER\_ID\_PK PRIMARY KEY,

PUB\_NAME VARCHAR(20) CONSTRAINT L\_PUBLISHER\_PUB\_NAME\_NN NOT NULL,

PUB\_YEAR VARCHAR(30) CONSTRAINT L\_PUBLISHER\_PUB\_YEAR\_NN NOT NULL,

EDITION VARCHAR(10) CONSTRAINT L\_PUBLISHER\_EDITION\_NN NOT NULL,

ADDRESS VARCHAR(200) CONSTRAINT L\_PUBLISHER\_ADDRESS\_NN NOT NULL,

PHONE\_NUM VARCHAR(13) CONSTRAINT L\_PUBLISHER\_PHONE\_NUM\_NN NOT NULL,

EMAILID VARCHAR(30) CONSTRAINT L\_PUBLISHER\_EMAILID\_UK UNIQUE

);

/\*Create Author Details Table\*/

CREATE TABLE L\_AUTHOR

(AUTHOR\_ID VARCHAR(10) CONSTRAINT L\_AUTHOR\_AUTHOR\_ID\_PK PRIMARY KEY,

AUTHOR\_NAME VARCHAR(20) CONSTRAINT L\_AUTHOR\_AUTHOR\_NAME\_NN NOT NULL,

ADDRESS VARCHAR(200) CONSTRAINT L\_AUTHOR\_STATE\_UK UNIQUE,

EMAILID VARCHAR(30) CONSTRAINT L\_AUTHOR\_EMAILID\_UK UNIQUE,

PHONE VARCHAR(13) CONSTRAINT L\_AUTHOR\_PHONE\_NN NOT NULL

);

**INSERT, UPDATE,**

/\*Insert Row into Author-Table\*/

INSERT INTO L\_AUTHOR(AUTHOR\_ID,AUTHOR\_NAME,ADDRESS,EMAILID,PHONE)

VALUES('A101','Krystal ADCOCK ','12-1260 ROOSERVELT NY-113352','krystal23@gmail.com',800-399-6716);

INSERT INTO L\_AUTHOR(AUTHOR\_ID,AUTHOR\_NAME,ADDRESS,EMAILID,PHONE)

VALUES('A102','Krish PATEL','7-260 Collegepoint NY-113351','krish@gmail.com',203-395-6719);

INSERT INTO L\_AUTHOR(AUTHOR\_ID,AUTHOR\_NAME,ADDRESS,EMAILID,PHONE)

VALUES('A103','Elisha BRAMSON','137-54 ROOSERVELT NY-113353','Elisha87@hotmail.com',403-366-3082);

INSERT INTO L\_AUTHOR(AUTHOR\_ID,AUTHOR\_NAME,ADDRESS,EMAILID,PHONE)

VALUES('A104','Herry COLBERT','72-901 MORRISSON NY-113357','herry1992@gmail.com',203-711-0028);

INSERT INTO L\_AUTHOR(AUTHOR\_ID,AUTHOR\_NAME,ADDRESS,EMAILID,PHONE)

VALUES('A105','Ruhi VILLIUMS','APT#12 - 200 MORRISSON NY-103256','ruhv9010@gmail.com',347-546-7476);

INSERT INTO L\_AUTHOR(AUTHOR\_ID,AUTHOR\_NAME,ADDRESS,EMAILID,PHONE)

VALUES('A106','Shreya Agrwal','78-272 Kashtoori IN-231546','sagrwal0909@gmail.com',882-541-0095);

commit;

SAVEPOINT;

/\*Insert into Department Table\*/

INSERT INTO L\_DEPARTMENT(DEPTID,deptname,studentid,staffid)

VALUES('D101','COMPUTER SCIENCE','S001','11');

INSERT INTO L\_DEPARTMENT(DEPTID,deptname,studentid,staffid)

VALUES('D102','ELECTRICAL','S004','55');

INSERT INTO L\_DEPARTMENT(DEPTID,deptname,studentid,staffid)

VALUES('D103','AUTOMOBILE','S002','33');

INSERT INTO L\_DEPARTMENT(DEPTID,deptname,studentid,staffid)

VALUES('D104','CIVIL','S003','22');

INSERT INTO L\_DEPARTMENT(DEPTID,deptname,studentid,staffid)

VALUES('D105','INFORMATION TECHNOLOGY','S005','44');

INSERT INTO L\_DEPARTMENT(DEPTID,deptname,studentid,staffid)

VALUES('D106','COMPUTER SCIENCE','S002','66');

INSERT INTO L\_DEPARTMENT(DEPTID,deptname,studentid,staffid)

VALUES('D107','AUTOMOBILE','S008','55');

/\*Insert values into Student Table\*/

INSERT INTO L\_STUDENT(STUDENTID,FNAME,LNAME,BIRTHDATE,ADDRESS,EMAILID,MOBILENUM,DEPTID,CURRENTYEAR,SECURITY\_Q,SECURITY\_A,REGI\_DATE,USER\_TYPE,STAFFID)

VALUES('S001','Devika','Soni','09-March-1990','8 jagatnathnagar G-234756','devika99@gmail.com',987-825-6378,'D104','3rd Year','What is your favorite movie?','Dhadkan','08-JAN-2014','ST','22');

INSERT INTO L\_STUDENT(STUDENTID,FNAME,LNAME,BIRTHDATE,ADDRESS,EMAILID,MOBILENUM,DEPTID,CURRENTYEAR,SECURITY\_Q,SECURITY\_A,REGI\_DATE,USER\_TYPE,STAFFID)

VALUES('S002','Vivan','Vangdu','09-APRIL-1992','12-098 Moneroe NJ-234809','vivan09@hotmail.com',302-567-7867,'D101','2nd Year','What is your favorite movie?','HARRY POTTER','08-JAN-2015','ST','66');

INSERT INTO L\_STUDENT(STUDENTID,FNAME,LNAME,BIRTHDATE,ADDRESS,EMAILID,MOBILENUM,DEPTID,CURRENTYEAR,SECURITY\_Q,SECURITY\_A,REGI\_DATE,USER\_TYPE,STAFFID)

VALUES('S003','Stafeny','Frendosi','10-JUNE-1991','81-235 Flushing NY-11353','Stafeny7878@gmail.com',890-638-0090,'D102','4th Year','What is your favorite movie?','Iron Man','08-JAN-2013','ST','44');

INSERT INTO L\_STUDENT(STUDENTID,FNAME,LNAME,BIRTHDATE,ADDRESS,EMAILID,MOBILENUM,DEPTID,CURRENTYEAR,SECURITY\_Q,SECURITY\_A,REGI\_DATE,USER\_TYPE,STAFFID)

VALUES('S004','Susan','Akhtar','05-MAY-1994','90-278 Heliot CT-89024','Susan05@hotmail.com',203-098-8976,'D105','2nd Year','What is your favorite movie?','The Hero','09-JAN-2015','ST','55');

INSERT INTO L\_STUDENT(STUDENTID,FNAME,LNAME,BIRTHDATE,ADDRESS,EMAILID,MOBILENUM,DEPTID,CURRENTYEAR,SECURITY\_Q,SECURITY\_A,REGI\_DATE,USER\_TYPE,STAFFID)

VALUES('S005','Greek','Joseph','18-FEB-1990','13-293 Lettin CI-902637','mesook90@gmail.com',809-030-0040,'D103','2nd Year','What is your favorite movie?','Sixteen','08-MARCH-2013','ST','22');

INSERT INTO L\_STUDENT(STUDENTID,FNAME,LNAME,BIRTHDATE,ADDRESS,EMAILID,MOBILENUM,DEPTID,CURRENTYEAR,SECURITY\_Q,SECURITY\_A,REGI\_DATE,USER\_TYPE,STAFFID)

VALUES('S006','Avani','Bhatt','20-OCT-1993','A-78 Maninagar GJ-23489','avanip20@gmail.com',347-648-0070,'D101','3rd Year','What is your favorite movie?','Aevengers','08-JAN-2014','ST','88');

INSERT INTO L\_STUDENT(STUDENTID,FNAME,LNAME,BIRTHDATE,ADDRESS,EMAILID,MOBILENUM,DEPTID,CURRENTYEAR,SECURITY\_Q,SECURITY\_A,REGI\_DATE,USER\_TYPE,STAFFID)

VALUES('S007','Neel','Patel','14-NOV-1990','9-56 Franklin NY-80354','neel90@hotmail.com',375-798-2040,'D104','2nd Year','What is your favorite movie?','Kick','20-MARCH-2015','ST','77');

INSERT INTO L\_STUDENT(STUDENTID,FNAME,LNAME,BIRTHDATE,ADDRESS,EMAILID,MOBILENUM,DEPTID,CURRENTYEAR,SECURITY\_Q,SECURITY\_A,REGI\_DATE,USER\_TYPE,STAFFID)

VALUES('S008','Janki','Patel','24-MAY-1995','38-345 Huston street-783261','janki.patel@gmail.com',203-040-0070,'D103','2nd Year','What is your favorite movie?','Jannat','24-MARCH-2014','ST','33');

INSERT INTO L\_STUDENT(STUDENTID,FNAME,LNAME,BIRTHDATE,ADDRESS,EMAILID,MOBILENUM,DEPTID,CURRENTYEAR,SECURITY\_Q,SECURITY\_A,REGI\_DATE,USER\_TYPE,STAFFID)

VALUES('S009','Bhumit','Patel','12-JAN-1994','20-4567 STORY AVE.-783261','bhumit.patel@gmail.com',347-980-0643,'D103','3rd Year','What is your favorite movie?','Jack-port','20-MARCH-2013','ST','11');

commit;

savepoint;

/\*Insert into Staff-Member Table\*/

INSERT INTO L\_STAFF(STAFFID,FNAME,LNAME,BIRTHDATE,ADDRESS,EMAILID,MOBILENUM,DEPTID,SECURITY\_Q,SECURITY\_A,REGI\_DATE,USER\_TYPE)

VALUES('88','Avani','Bhatt','20-OCT-1993','A-78 Maninagar GJ-23489','avanip20@gmail.com',347-648-0070,'D101','What is your favorite colour?','Yellow','18-JAN-2015','SM');

COMMIT;

INSERT INTO L\_STAFF(STAFFID,FNAME,LNAME,BIRTHDATE,ADDRESS,EMAILID,MOBILENUM,DEPTID,SECURITY\_Q,SECURITY\_A,REGI\_DATE,USER\_TYPE)

VALUES('11','Jeet','Patel','12-JAN-1985','16-384 Georejiya-11633','jeet89@gmail.com',901-567-8690,'D103','What is your favorite colour?','Green','08-JAN-2013','SM');

INSERT INTO L\_STAFF(STAFFID,FNAME,LNAME,BIRTHDATE,ADDRESS,EMAILID,MOBILENUM,DEPTID,SECURITY\_Q,SECURITY\_A,REGI\_DATE,USER\_TYPE)

VALUES('22','Nilakshi','Patel','01-Dec-1986','23-485 Hudson-23894','Nilakshi86@hotmail.com',981-754-0090,'D101','What is your favorite colour?','PINK','08-JAN-2011','SM');

INSERT INTO L\_STAFF(STAFFID,FNAME,LNAME,BIRTHDATE,ADDRESS,EMAILID,MOBILENUM,DEPTID,SECURITY\_Q,SECURITY\_A,REGI\_DATE,USER\_TYPE)

VALUES('77','Neel','Patel','14-NOV-1990','9-56 Franklin NY-80354','neel90@hotmail.com',375-798-2040,'D104','What is your favorite colour?','Blue','12-MARCH-2014','SM');

INSERT INTO L\_STAFF(STAFFID,FNAME,LNAME,BIRTHDATE,ADDRESS,EMAILID,MOBILENUM,DEPTID,SECURITY\_Q,SECURITY\_A,REGI\_DATE,USER\_TYPE)

VALUES('33','Riya','Shah','14-Feb-1988','16-432 Texas-89046','riya@gmail.com',201-426-9060,'D105','What is your favorite colour?','Red','08-JAN-2012','SM');

INSERT INTO L\_STAFF(STAFFID,FNAME,LNAME,BIRTHDATE,ADDRESS,EMAILID,MOBILENUM,DEPTID,SECURITY\_Q,SECURITY\_A,REGI\_DATE,USER\_TYPE)

VALUES('44','Kenia','Freck','14-JULY-1985','7-290 NJ-097835','kenia85@hotmail.com',203-402-0030,'D102','What is your favorite colour?','Broun','20-JAN-2013','SM');

INSERT INTO L\_STAFF(STAFFID,FNAME,LNAME,BIRTHDATE,ADDRESS,EMAILID,MOBILENUM,DEPTID,SECURITY\_Q,SECURITY\_A,REGI\_DATE,USER\_TYPE)

VALUES('55','Dhwani','Bhatt','21-NOV-1989','88 Columbia-087688','dhanno89@gmail.com',374-678-5409,'D104','What is your favorite colour?','Purpal','24-JUN-2012','SM');

INSERT INTO L\_STAFF(STAFFID,FNAME,LNAME,BIRTHDATE,ADDRESS,EMAILID,MOBILENUM,DEPTID,SECURITY\_Q,SECURITY\_A,REGI\_DATE,USER\_TYPE)

VALUES('66','Payal','Shah','28-NOV-1989','790-43 Verginia - 324670','payu88@hotmail.com',204-986-8950,'D101','What is your favorite colour?','Yellow','24-JUN-2012','SM');

commit;

/\*Enter the values into Publisher Table\*/

INSERT INTO L\_PUBLISHER(publisher\_id,pub\_name,pub\_year,edition,address,phone\_num,emailid)

VALUES('P201','PEARSON',2010,'10th','NewYork',347-546-0503,'scott.overland@pearson.com');

INSERT INTO L\_PUBLISHER(publisher\_id,pub\_name,pub\_year,edition,address,phone\_num,emailid)

VALUES('P202','CambridgeUniversity',2011,'5th','United Kingdom',422-335-8331,'information@cambridge.org');

INSERT INTO L\_PUBLISHER(publisher\_id,pub\_name,pub\_year,edition,address,phone\_num,emailid)

VALUES('P203','Heinle',2008,'2nd','Singapore 556741',302-643-7350,' asia.info@cengage.com');

INSERT INTO L\_PUBLISHER(publisher\_id,pub\_name,pub\_year,edition,address,phone\_num,emailid)

VALUES('P204','Compass Publishing',2010,'3rd','Korea',303-926-9354,' info@compasspub.com');

INSERT INTO L\_PUBLISHER(publisher\_id,pub\_name,pub\_year,edition,address,phone\_num,emailid)

VALUES('P205','Delta Publishing',2007,'1st','Surrey GU5 9SW UK',204-809-2001,' Not Available');

commit;

/\* Inser into Issue-details Table \*/

INSERT INTO L\_ISSUEDETAILS(ISSUENUM,ISSUE\_DATE,BOOKNUM,DUE\_DATE,RETURN\_DATE,STUDENTID,STAFFID,DEPTID)

VALUES('I1001','01-APRIL-2016','901','23-APRIL-2016','02-APRIL-2016','S001','22','D104');

INSERT INTO L\_ISSUEDETAILS(ISSUENUM,ISSUE\_DATE,BOOKNUM,DUE\_DATE,RETURN\_DATE,STUDENTID,STAFFID,DEPTID)

VALUES('I1002','11-MARCH-2016','904','02-APRIL-2016','22-MARCH-2016','S004','33','D101');

INSERT INTO L\_ISSUEDETAILS(ISSUENUM,ISSUE\_DATE,BOOKNUM,DUE\_DATE,RETURN\_DATE,STUDENTID,STAFFID,DEPTID)

VALUES('I1003','01-FEB-2016','901','23-FEB-2016','20-FEB-2016','S002','11','D102');

INSERT INTO L\_ISSUEDETAILS(ISSUENUM,ISSUE\_DATE,BOOKNUM,DUE\_DATE,RETURN\_DATE,STUDENTID,STAFFID,DEPTID)

VALUES('I1004','19-FEB-2016','905','10-MARCH-2016','05-MARCH-2016','S007','55','D105');

INSERT INTO L\_ISSUEDETAILS(ISSUENUM,ISSUE\_DATE,BOOKNUM,DUE\_DATE,RETURN\_DATE,STUDENTID,STAFFID,DEPTID)

VALUES('I1005','01-APRIL-2016','902','23-APRIL-2016','02-APRIL-2016','S002','44','D103');

INSERT INTO L\_ISSUEDETAILS(ISSUENUM,ISSUE\_DATE,BOOKNUM,DUE\_DATE,RETURN\_DATE,STUDENTID,STAFFID,DEPTID)

VALUES('I1006','01-APRIL-2016','905','23-APRIL-2016','02-APRIL-2016','S001','22','D104');

INSERT INTO L\_ISSUEDETAILS(ISSUENUM,ISSUE\_DATE,BOOKNUM,DUE\_DATE,RETURN\_DATE,STUDENTID,STAFFID,DEPTID)

VALUES('I1007','01-APRIL-2016','903','23-APRIL-2016','24-APRIL-2016','S003','33','D103');

INSERT INTO L\_ISSUEDETAILS(ISSUENUM,ISSUE\_DATE,BOOKNUM,DUE\_DATE,RETURN\_DATE,STUDENTID,STAFFID,DEPTID)

VALUES('I1008','20-FEB-2016','901','12-MARCH-2016','07-MARCH-2016','S006','66','D107');

COMMIT;

/\* Insert into book Table \*/

INSERT INTO MPATEL6771.L\_BOOK (BOOKNUM,BOOKTITLE,ISBNNUM,AUTHOR\_ID,PUBLISHER\_ID,DEPTID,COPIESLEFT,ISSUENUM)

VALUES ('B901','COMPUTER ARCHITECHTURE','ISBN5001','A105','P201','D105','N','I1007');

INSERT INTO MPATEL6771.L\_BOOK (BOOKNUM,BOOKTITLE,ISBNNUM,AUTHOR\_ID,PUBLISHER\_ID,DEPTID,COPIESLEFT,ISSUENUM)

VALUES ('B905','SOFTWARE ENGINEERING','ISBN5005','A101','P205','D105','Y','I1001');

INSERT INTO MPATEL6771.L\_BOOK (BOOKNUM,BOOKTITLE,ISBNNUM,AUTHOR\_ID,PUBLISHER\_ID,DEPTID,COPIESLEFT,ISSUENUM)

VALUES ('B902','DATABASE SYSTEMS','ISBN5004','A102','P202','D105','N','1002');

INSERT INTO MPATEL6771.L\_BOOK (BOOKNUM,BOOKTITLE,ISBNNUM,AUTHOR\_ID,PUBLISHER\_ID,DEPTID,COPIESLEFT,ISSUENUM)

VALUES ('B903','ECONOMICS','ISBN5002','A103','P203','D102','Y','1005');

INSERT INTO MPATEL6771.L\_BOOK (BOOKNUM,BOOKTITLE,ISBNNUM,AUTHOR\_ID,PUBLISHER\_ID,DEPTID,COPIESLEFT,ISSUENUM)

VALUES ('B906','ETHICS AND LEADERSHIP','ISBN5003','A104','P204','D104','Y','1004');

INSERT INTO MPATEL6771.L\_BOOK (BOOKNUM,BOOKTITLE,ISBNNUM,AUTHOR\_ID,PUBLISHER\_ID,DEPTID,COPIESLEFT,ISSUENUM)

VALUES ('B904','STATISTIC','ISBN5006','A105','P203','D102','N','1003');

COMMIT MPATEL6771.L\_BOOK;

**Queries with ALTER**

/\*Alter Queries for Student Table\*/

ALTER TABLE L\_STUDENT

ADD FOREIGN KEY (DEPTID)

REFERENCES L\_DEPARTMENT(DEPTID);

/\*Alter Queries for STAFF Table\*/

ALTER TABLE L\_STAFF

ADD FOREIGN KEY (DEPTID)

REFERENCES L\_DEPARTMENT(DEPTID);

/\*Alter Queries for BOOK Table\*/

ALTER TABLE L\_BOOK

ADD FOREIGN KEY (DEPTID)

REFERENCES L\_DEPARTMENT(DEPTID);

ALTER TABLE L\_BOOK

ADD FOREIGN KEY (AUTHOR\_ID)

REFERENCES L\_AUTHOR(AUTHOR\_ID);

ALTER TABLE L\_BOOK

ADD FOREIGN KEY (PUBLISHER\_ID)

REFERENCES L\_PUBLISHER(PUBLISHER\_ID);

/\*Alter Queries for ISSUEDETAILS Table\*/

ALTER TABLE L\_ISSUEDETAILS

ADD FOREIGN KEY (STUDENTID)

REFERENCES L\_STUDENT(STUDENTID);

ALTER TABLE L\_ISSUEDETAILS

ADD FOREIGN KEY (DEPTID)

REFERENCES L\_DEPARTMENT(DEPTID);

ALTER TABLE L\_ISSUEDETAILS

ADD FOREIGN KEY (STAFFID)

REFERENCES L\_STAFF(STAFFID);

ALTER TABLE L\_BOOK

ADD ISSUENUM VARCHAR(30);

ALTER TABLE L\_BOOK

ADD FOREIGN KEY (ISSUENUM)

REFERENCES L\_ISSUEDETAILS(ISSUENUM);

ALTER TABLE L\_STUDENT

ADD STAFFID VARCHAR2(10);

**NESTED QUERIES**

/\*Queries for 'Single-row Subqueries' using two-tables\*/

SELECT LNAME,FNAME,DEPTID,BIRTHDATE,emailid,regi\_date

FROM L\_STUDENT

WHERE DEPTID = (SELECT DEPTID

FROM L\_DEPARTMENT

WHERE UPPER(DEPTNAME) = 'COMPUTER SCIENCE'

);

SELECT LNAME,FNAME,DEPTID,BIRTHDATE,emailid,regi\_date

FROM L\_STUDENT

WHERE DEPTID = (SELECT DEPTID

FROM L\_DEPARTMENT

WHERE UPPER(DEPTNAME) = 'AUTOMOBILE'

);

/\*Queries for 'Single-row Subqueries' using one table\*/

SELECT LNAME,FNAME,DEPTID,BIRTHDATE,emailid,regi\_date,mobilenum

FROM L\_STUDENT

WHERE CURRENTYEAR >= (SELECT CURRENTYEAR

FROM L\_STUDENT

WHERE CURRENTYEAR = '4th Year'

);

SELECT LNAME,FNAME,DEPTID,BIRTHDATE,emailid,regi\_date,mobilenum

FROM L\_STUDENT

WHERE REGI\_DATE >= (SELECT REGI\_DATE

FROM L\_STUDENT

WHERE REGI\_DATE = '24-MAR-14'

);

/\* Subquery with != operator \*/

SELECT LNAME,FNAME,DEPTID,BIRTHDATE,emailid,regi\_date,USER\_TYPE

FROM L\_STUDENT

WHERE DEPTID != (SELECT DEPTID

FROM L\_STUDENT

WHERE LNAME = 'Soni'

);

SELECT LNAME,FNAME,DEPTID,BIRTHDATE,emailid,regi\_date,USER\_TYPE

FROM L\_STUDENT

WHERE DEPTID != (SELECT DEPTID

FROM L\_STUDENT

WHERE FNAME = 'Bhumit'

);

/\* Subquery to three levels \*/

SELECT ISSUENUM , ISSUE\_DATE , BOOKNUM , DEPTID

FROM L\_ISSUEDETAILS

WHERE DEPTID = (SELECT DEPTID

FROM L\_STAFF

WHERE STAFFID =

(SELECT STAFFID

FROM L\_STUDENT

WHERE FNAME = 'Bhumit'

)

);

/\* Creating table using Subquery\*/

CREATE TABLE TEMP\_S

AS

SELECT STUDENTID,LNAME,FNAME,DEPTID

FROM L\_STUDENT

WHERE DEPTID = 'D101';

DESC TEMP\_S;

SELECT \* FROM TEMP\_S;

/\* INSERT USING A SUBQUERY \*/

INSERT INTO TEMP\_S(STUDENTID,LNAME,FNAME)

SELECT STUDENTID,LNAME,FNAME

FROM L\_STUDENT

WHERE DEPTID = 'D104';

SELECT \* FROM TEMP\_S;

/\* Update using a SubQuery \*/

UPDATE L\_AUTHOR

SET PHONE = '902-348-6297'

WHERE AUTHOR\_ID = (SELECT AUTHOR\_ID

FROM L\_BOOK

WHERE BOOKTITLE = 'ECONOMICS');

UPDATE TEMP\_S

SET ( LNAME , FNAME , DEPTID) =

(SELECT LNAME , FNAME , DEPTID

FROM L\_ISSUEDETAILS

WHERE BOOKNUM = '904');

SELECT \* FROM TEMP\_S;

/\* DELETE using a subquery \*/

DELETE FROM TEMP\_S

WHERE DEPTID = (SELECT DEPTID

FROM TEMP\_S

WHERE LNAME = 'Soni');

/\* Subquery with IN operator \*/

SELECT STUDENTID, LNAME , FNAME , STAFFID

FROM L\_STUDENT

WHERE STAFFID IN (SELECT STAFFID

FROM L\_STAFF

WHERE DEPTID = 'D101');

/\* Subquery with >ANY operator \*/

SELECT STUDENTID , LNAME , FNAME , CURRENTYEAR

FROM L\_STUDENT

WHERE CURRENTYEAR >ANY

(SELECT CURRENTYEAR FROM L\_STUDENT WHERE STAFFID = '22')

AND STAFFID <> 22;

SELECT STUDENTID , LNAME , FNAME , CURRENTYEAR

FROM L\_STUDENT

WHERE CURRENTYEAR >ANY

(SELECT CURRENTYEAR FROM L\_STUDENT WHERE STAFFID = '22')

AND STAFFID <> 33;

/\* Subquery with <ALL operator \*/

SELECT STUDENTID , LNAME , FNAME , STAFFID

FROM L\_STUDENT

WHERE STAFFID <ALL

(SELECT AVG(STAFFID) FROM L\_STUDENT GROUP BY DEPTID);

/\* Query for Top-4 DEPTID \*/

SELECT ROWNUM , BOOKNUM , BOOKTITLE , DEPTID

FROM (SELECT BOOKNUM , BOOKTITLE , DEPTID

FROM L\_BOOK

ORDER BY DEPTID DESC)

WHERE ROWNUM <= 4;

**Queries for JOINS**

/\*Select query for Equijoin\*/

SELECT L\_STUDENT.FNAME || ',' || L\_STUDENT.LNAME STUDENT ,

L\_STAFF.FNAME || ',' || L\_STAFF.LNAME STAFF,

L\_STUDENT.deptid,L\_STAFF.DEPTID

FROM L\_STUDENT , L\_STAFF

WHERE l\_student.deptid = l\_staff.deptid;

/\*Queries for Multiple Joins \*/

SELECT \* FROM L\_DEPARTMENT;

Queries for VIEWS

/\* CREATING A SIMPLE VIEW FOR L\_STUDENT \*/

CREATE VIEW MITAL\_ST

(S\_ID , S\_LAST , S\_FIRST , ADVISOR\_ID , MAJOR\_ID)

AS SELECT STUDENTID , LNAME , FNAME , STAFFID , DEPTID

FROM L\_STUDENT

WHERE DEPTID = 'D102'

WITH CHECK OPTION;

DESC mital\_st;

SELECT \* FROM mital\_st;

/\* CREATING A SIMPLE VIEW FOR L\_STUDENT \*/

CREATE VIEW MITAL\_SF

(ADVISOR\_ID, F\_LAST , F\_FIRST , MAJOR\_ID)

AS SELECT STAFFID , LNAME , FNAME , DEPTID

FROM L\_STAFF

WHERE DEPTID = 'D101'

WITH CHECK OPTION;

SELECT \* FROM mital\_sf;

/\* CREATING A SIMPLE VIEW FOR L\_AUTHOR \*/

CREATE VIEW MITAL\_AR

(A\_ID , A\_NAME , CONTACT\_N0)

AS SELECT AUTHOR\_ID , AUTHOR\_NAME ,PHONE

FROM L\_AUTHOR

WHERE AUTHOR\_NAME = 'Shreya Agrwal'

WITH CHECK OPTION;

/\*

SET LINESIZE 50;

DESC mital\_ar;

SELECT \* FROM mital\_ar;

SET LINESIZE 100;

SELECT \* FROM mital\_ar;

\*/

SELECT \* FROM MPATEL6771.L\_DEPARTMENT;

SELECT \* FROM MPATEL6771.L\_BOOK;

CREATE VIEW DEPARTMENT AS

SELECT DEPTNAME,DEPTID,STUDENTID

FROM MPATEL6771.L\_DEPARTMENT

WHERE DEPTNAME='INFORMATION TECHNOLOGY';

CREATE VIEW BOOK\_DEPT AS

SELECT BOOKNUM,BOOKTITLE,DEPTID

FROM MPATEL6771.L\_BOOK

WHERE DEPTID='D105';

COMMIT MPATEL6771.DEPARTMENT;

COMMIT MPATEL6771.BOOK\_DEPT;

Procedure

A procedure is a named PL/SQL program that can perform one or more tasks. A procedure is the building block of modular programming.

The procedure body contains declaration, executable, and exception-handling sections. The declaration and exception-handling sections are optional. The executable section contains action statements, and it must contain at least one action statement. The procedure body starts after the IS reserved word. If there is no local declaration, the IS reserved word is followed by the reserved word BEGIN. The body ends with the reserved word END. There can be more than one END statement in the program, so it is a good idea to use the procedure name as the optional label after reserved word END.

Trigger

A database trigger, known simply as a trigger, is a PL/SQL block. It is stored in the database and is called automatically when a triggering event occurs. A user cannot call a trigger explicitly. The triggering event is based on a Data Manipulation Language statement such as INSERT, UPDATE, or DELETE. A trigger can be created to fire before or after the triggering event. For example, if you design a trigger to execute after you INSERT a new employee in the EMPLOYEE table, the trigger executes after the INSERT statement. The execution of a trigger is also known as firing the trigger.

A trigger is very useful in generating values for derived columns, for keeping track of table access, for preventing invalid entries, for performing validity checks, or for maintaining security. There are some restrictions involving creation of a trigger:

• A trigger cannot use a Transaction Control Language statement such as COMMIT, ROLLBACK, or SAVEPOINT. All operations performed by a trigger become part of the transaction. The trigger operations get committed or rolled back with the transaction.

• A procedure or function called by a trigger cannot perform Transaction Control Language statements.

• A variable in a trigger cannot be declared with LONG or LONG RAW data types.

PL/SQL blocks:

**Procedure/function:**

A named block that can be stored in the Oracle server, can be called by its name, and can take arguments. Functions can be called from procedures whereas procedures cannot be called from function.

**Description:**

A PL/SQL function is also known as subroutine or a subprogram. You specify the function name function \_name after the FUNCTION keyword. A function may have zero or more than one parameter.

A PL/SQL procedure is a named block that does a specific task.PL/SQL allows to encapsulate complex business logic and reuse it in both database layer and application layer.

**Package:**

A named PL/SQL module that is a group of functions, procedures, and identifiers.

**Description:**

PL/SQL package is like a library once written stored in the Oracle database and can be used by many applications.

A PL/SQL package has two parts: Package specification and Package body

* A package specification is the public interface of your applications. The public means the stored function, procedures, types are accessible from other applications.
* A package body contains the code that implements the package specification.

/\* TRIGGER FOR STUDENT TABLE \*/

CREATE OR REPLACE TRIGGER STU\_TRIGGER

BEFORE INSERT ON L\_STUDENT

FOR EACH ROW

DECLARE

V\_STID L\_STUDENT.STUDENTID%TYPE;

BEGIN

SELECT l\_student\_s\_id\_seq.nextval

INTO V\_STID FROM DUAL;

:NEW.STUDENTID := V\_STID;

:NEW.REGI\_DATE := SYSDATE;

END;

/\* TRIGGER FOR STAFF TABLE \*/

CREATE OR REPLACE TRIGGER SF\_TRIGGER

BEFORE INSERT ON L\_STAFF

FOR EACH ROW

DECLARE

V\_SFID L\_STAFF.STAFFID%TYPE;

BEGIN

SELECT L\_STAFF\_FID\_SEQ.nextval

INTO V\_SFID FROM DUAL;

:NEW.STAFFID := V\_SFID;

:NEW.REGI\_DATE := SYSDATE;

END;

/\* CREATE TRIGGER FOR DEPARTMENT \*/

CREATE OR REPLACE TRIGGER DE\_TRIGGER

BEFORE INSERT ON L\_DEPARTMENT

FOR EACH ROW

DECLARE

V\_DID L\_DEPARTMENT.DEPTID%TYPE;

BEGIN

SELECT L\_DEPARTMENT\_DEPTID\_SEQ.nextval

INTO V\_DID FROM DUAL;

:NEW.DEPTID := V\_DID;

END;

/\* CREATE TRIGGER FOR AUTHOR \*/

CREATE OR REPLACE TRIGGER AU\_TRIGGER

BEFORE INSERT ON L\_author

FOR EACH ROW

DECLARE

V\_AID L\_AUTHOR.AUTHOR\_ID%TYPE;

BEGIN

SELECT L\_AUTHOR\_A\_ID\_SEQ.nextval

INTO V\_AID FROM DUAL;

:NEW.AUTHOR\_ID := V\_AID;

END;

CREATE OR REPLACE TRIGGER BK\_TRIGGER

BEFORE INSERT ON L\_BOOK

FOR EACH ROW

DECLARE

V\_BID L\_BOOK.BOOKNUM%TYPE;

BEGIN

SELECT SEQ\_L\_BOOK .nextval

INTO V\_BID FROM DUAL;

:NEW.BOOKNUM := V\_BID;

END;

CREATE OR REPLACE TRIGGER BID\_TRIGGER

BEFORE INSERT ON L\_ISSUEDETAILS

FOR EACH ROW

DECLARE

V\_IID L\_ISSUEDETAILS.ISSUENUM%TYPE;

BEGIN

SELECT L\_ISSNUM\_SEQ.nextval

INTO V\_IID FROM DUAL;

:NEW.ISSUENUM := V\_IID;

END;

CREATE OR REPLACE TRIGGER P\_TRIGGER

BEFORE INSERT ON L\_PUBLISHER

FOR EACH ROW

DECLARE

V\_PID L\_PUBLISHER.PUBLISHER\_ID%TYPE;

BEGIN

SELECT PUB\_SEQ.nextval

INTO V\_PID FROM DUAL;

:NEW.PUBLISHER\_ID := V\_PID;

END;