



Gopalan College
of Engineering and Management
ISO 9001 : 2008

APPROVED BY AICTE NEW DELHI, AFFILIATED TO VTU BELGAUM



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DATABASE APPLICATIONS LAB MANUAL-10CSL57

2016-2017

SEMESTER-VII

Prepared by:

**J.Somasekar,
Assc. Professor
Dept. of CSE,
GCEM**

Reviewed by:

**N S Saradha Devi
Head of the Department
Dept. of CSE
GCEM**

Approved by:

**Dr. A A Powly Thomas
Principal
GCEM**

181/1, 182/1, Hoodi Village, Sonnenahalli, K.R. Puram, Bengaluru,
Karnataka 560048

COURSE OBJECTIVE

Upon successful completion of this Lab the student will be able to:

- Creating database objects
- Modifying database objects
- Manipulating the data
- Retrieving the data from the database server
- Performing database operations in a procedural manner using pl/sql
- Performing database operations (create, update, modify, retrieve, etc.,) .
- Design and Develop applications like banking, reservation system, etc.,

COURSE OUTCOME

- Gaining the Knowledge on SQL.
- Able to create tables with and without constraints
- Able to make relation between two or more tables.
- Able to retrieve data based on various conditions.

COMPUTER LAB DO'S AND DON'T

DO'S

1. Know the location of the fire extinguisher and the first aid box and how to use them in case of an emergency.
2. Read and understand how to carry out an activity thoroughly before coming to the laboratory.
3. Report fires or accidents to your lecturer/laboratory technician immediately.
4. Report any broken plugs or exposed electrical wires to your lecturer/laboratory technician immediately.

DON'TS

1. Do not eat or drink in the laboratory.
2. Avoid stepping on electrical wires or any other computer cables.
3. Do not open the system unit casing or monitor casing particularly when the power is turned on. Some internal components hold electric voltages of up to 30000 volts, which can be fatal.
4. Do not insert metal objects such as clips, pins and needles into the computer casings. They may cause fire.
5. Do not remove anything from the computer laboratory without permission.
6. Do not touch, connect or disconnect any plug or cable without your lecturer/laboratory technician's permission.
7. Do not misbehave in the computer laboratory.

SYLLABUS

Database Applications Laboratory

1. Consider the following relations:

STUDENT (snum: integer, sname: string, major: string, level: string, age: integer)

CLASS (name: string, meets at: string, room: string, d: integer)

ENROLLED (snum: integer, cname: string)

FACULTY (fid: integer, fname: string, deptid: integer)

The meaning of these relations is straightforward; for example, Enrolled has one record per student-class pair such that the student is enrolled in the class. Level is a two character code with 4 different values (example: Junior: JR etc) Write the following queries in SQL. No duplicates should be printed in any of the answers.

- i. Find the names of all Juniors (level = JR) who are enrolled in a class taught by Prof. Harshith
- ii. Find the names of all classes that either meet in room R128 or have five or more Students enrolled.
- iii. Find the names of all students who are enrolled in two classes that meet at the same time.
- iv. Find the names of faculty members who teach in every room in which some class is taught.
- v. Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.

2. The following relations keep track of airline flight information:

FLIGHTS (no: integer, from: string, to: string, distance: integer, Departs: time, arrives: time, price: real)

AIRCRAFT (aid: integer, aname: string, cruisingrange: integer)

CERTIFIED (eid: integer, aid: integer)

EMPLOYEES (eid: integer, ename: string, salary: integer)

Note that the Employees relation describes pilots and other kinds of employees as well; Every pilot is certified for some aircraft, and only pilots are certified to fly. Write each of the following queries in SQL.

- i. Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80, 000.
- ii. For each pilot who is certified for more than three aircrafts, find the eid and the maximum cruisingrange of the aircraft for which she or he is certified.
- iii. Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Frankfurt.
- iv. For all aircraft with cruisingrange over 1000 Kms, .find the name of the aircraft and the average salary of all pilots certified for this aircraft.
- v. Find the names of pilots certified for some Boeing aircraft. vi. Find the aids of all aircraft that can be used on routes from Bengaluru to New Delhi.

3. Consider the following database of student enrollment in courses & books adopted for each course.

STUDENT (regno: string, name: string, major: string, bdate:date)

COURSE (course #:int, cname:string, dept:string)

ENROLL (regno:string, course#:int, sem:int, marks:int)

BOOK _ ADOPTION (course# :int, sem:int, book-ISBN:int)

TEXT (book-ISBN:int, book-title:string, publisher:string, author:string)

- i. Create the above tables by properly specifying the primary keys and the foreign keys.
- ii. Enter at least five tuples for each relation.
- iii. Demonstrate how you add a new text book to the database and make this book be adopted by some department.
- iv. Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses offered by the 'CS' department that use more than two books.
- v. List any department that has all its adopted books published by a specific publisher.
- vi. Generate suitable reports.
- vii. Create suitable front end for querying and displaying the results.

4. The following tables are maintained by a book dealer.

AUTHOR (author-id:int, name:string, city:string, country:string)

PUBLISHER (publisher-id:int, name:string, city:string, country:string)

CATALOG (book-id:int, title:string, author-id:int, publisher-id:int, category-id:int, year:int, price:int)

CATEGORY (category-id:int, description:string)

ORDER-DETAILS (order-no:int, book-id:int, quantity:int)

- i. Create the above tables by properly specifying the primary keys and the foreign keys.
- ii. Enter at least five tuples for each relation.
- iii. Give the details of the authors who have 2 or more books in the catalog and the price of the books is greater than the average price of the books in the catalog and the year of publication is after 2000.
- iv. Find the author of the book which has maximum sales.
- v. Demonstrate how you increase the price of books published by a specific publisher by 10%.
- vi. Generate suitable reports.
- vii. Create suitable front end for querying and displaying the results.

5. Consider the following database for a banking enterprise

BRANCH(branch-name:string, branch-city:string, assets:real)

ACCOUNT(accno:int, branch-name:string, balance:real)

DEPOSITOR(customer-name:string, accno:int)

CUSTOMER(customer-name:string, customer-street:string, customer-city:string)

LOAN(loan-number:int, branch-name:string, amount:real)

BORROWER(customer-name:string, loan-number:int)

- i. Create the above tables by properly specifying the primary keys and the foreign keys
- ii. Enter at least five tuples for each relation
- iii. Find all the customers who have at least two accounts at the Main branch.
- iv. Find all the customers who have an account at all the branches located in a specific city.
- v. Demonstrate how you delete all account tuples at every branch located in a specific city.
- vi. Generate suitable reports.
- vii. Create suitable front end for querying and displaying the results.

INSTRUCTIONS:

1. The exercises are to be solved in an RDBMS environment like Oracle or DB2.
2. Suitable tuples have to be entered so that queries are executed correctly.
3. Front end may be created using either VB or VAJ or any other similar tool.
4. The student need not create the front end in the examination. The results of the queries may be displayed directly.
5. Relevant queries other than the ones listed along with the exercises may also be asked in the examination.
6. Questions must be asked based on lots.

TABLE OF CONTENTS

SL NO	DATE	NAME OF THE EXPERIMENT	Page No.
1		STUDENT FACULTY DATABASE – Solving Queries and Report Generation	12-14
2		AIRCRAFT DATABASE – Solving Queries and Report Generation	15-17
3		STUDENT BOOK DATABASE – Solving Queries and Report Generation	18-20
4		BOOK DEALER DATABASE – Solving Queries and Report Generation	21-23
5		BANK DATABASE – Creation & Insertion of Values	24-26

BACKGROUND THEORY OF SQL

Oracle workgroup or server is the largest selling RDBMS product. It is estimated that the combined sales of both these Oracle database products account for around 80% of the RDBMS systems sold worldwide. These products are constantly undergoing change and evolving. The natural language of this RDBMS product is ANSI SQL, PL/SQL, a superset of ANSI SQL. Oracle 8i and 9i also understand SQLJ. Oracle Corp has also incorporated a full-fledged Java virtual machine into its database engine. Since both executables share the same memory space, the JVM can communicate with the database engine with ease and has direct access to Oracle tables and their data.

SQL is a structured query language. SQL contains different data types, those are

1. char(size)
2. varchar2(size)
3. date
4. number(p,s)
5. long
6. raw/long raw

Different types of commands in SQL:

- A). **DDL commands:** - To create a database object
- B). **DML commands:** - To manipulate data of a database object
- C). **DQL command:** - To retrieve the data from a database.
- D). **DCL/DTL commands:** - To control the data of a database...

DDL commands:

1. The Create Table Command: - It defines each column of the table uniquely. Each column has a minimum of three attributes, a name, data type, and size.

Syntax:

Create table <table name> (<col1> <datatype><size>,<col2> <datatype><size>);

Ex:

```
create table emp(empno number(4) primary key, ename char(10));
```

2. Modifying the structure of tables.

a) add new columns

Syntax:

Alter table <tablename> add(<new col><datatype>(size),<new col><datatype>(size));

Ex:

```
alter table emp add(sal number(7,2));
```

3. Dropping a column from a table.**Syntax:**

Alter table <tablename> drop column <col>;

Ex:

```
alter table emp drop column sal;
```

4. Modifying existing columns.**Syntax:**

```
Alter table <tablename> modify(<col><newdatatype>(<newsized>));
```

Ex:

```
alter table emp modify(ename varchar2(15));
```

5. Renaming the tables**Syntax:**

```
Rename <oldtable> to <new table>;
```

Ex:

```
rename emp to emp1;
```

6. truncating the tables.**Syntax:**

```
Truncate table <tablename>;
```

Ex:

```
trunc table emp1;
```

7. Destroying tables.**Syntax:**

```
Drop table <tablename>;
```

Ex:

```
drop table emp;
```

DML commands:

8. Inserting Data into Tables: - once a table is created the most natural thing to do is load this table with data to be manipulated later.

Syntax:

```
insert into <tablename> (<col1>,<col2>) values(<exp>,<exp>);
```

9. Delete operations.

a) remove all rows

Syntax:

```
delete from <tablename>;
```

- b) removal of a specified row/s

Syntax:

delete from <tablename> where <condition>;

10. Updating the contents of a table.

- a) updating all rows

Syntax:

Update <tablename> set <col>=<exp>,<col>=<exp>;

- b) updating seleted records.

Syntax:

Update<tablename>set<col>=<exp>,<col>=<exp> where <condition>;

11. Types of data constrains.

- a) not null constraint at column level.

Syntax:

<col><datatype>(size)not null

- b) unique constraint

Syntax:

Unique constraint at column level.

<col><datatype>(size)unique;

- c) unique constraint at table level:

Syntax:

Create table tablename(col=format,col=format,unique(<col1>,<col2>);

- d) primary key constraint at column level

Syntax:

<col><datatype>(size)primary key;

- e) primary key constraint at table level.

Syntax:

Create table tablename(col=format,col=format
primary key(col1>,<col2>);

- f) foreign key constraint at column level.

Syntax:

<col><datatype>(size) references <tablename>[<col>];

- g) foreign key constraint at table level

Syntax:

foreign key(<col>[,<col>])references <tablename>[(<col>,<col>)

- h) check constraint

check constraint constraint at column level.

Syntax: <col><datatype>(size) check(<logical expression>)

i) check constraint constraint at table level.

Syntax: check(<logical expression>)

DQL Commands:

12. Viewing data in the tables: - once data has been inserted into a table, the next most logical operation would be to view what has been inserted.

a) all rows and all columns

Syntax:

Select <col> to <col n> from tablename;

Select * from tablename;

13. Filtering table data: - while viewing data from a table, it is rare that all the data from table will be required each time. Hence, sql must give us a method of filtering out data that is not required data.

a) Selected columns and all rows:

Syntax:

select <col1>,<col2> from <tablename>;

b) selected rows and all columns:

Syntax:

select * from <tablename> where <condition>;

c) selected columns and selected rows

Syntax:

select <col1>,<col2> from <tablename> where<condition>;

14. Sorting data in a table.

Syntax:

Select * from <tablename> order by <col1>,<col2> <[sortorder]>;

DCL commands:

Oracle provides extensive feature in order to safeguard information stored in its tables from unauthorised viewing and damage. The rights that allow the user of some or all oracle resources on the server are called privileges.

a) Grant privileges using the GRANT statement

The grant statement provides various types of access to database objects such as tables, views and sequences and so on.

Syntax:

GRANT <object privileges> ON <objectname> TO<username> [WITH GRANT OPTION];

b) Revoke permissions using the REVOKE statement

The REVOKE statement is used to deny the Grant given on an object.

Syntax:

REVOKE<object privilege> ON FROM<user name>;

Experiment No.1**STUDENT_FACULTY DATABASE**

1. Consider the following relations:

STUDENT (snum: integer, sname: string, major: string, level: string, age: integer)

CLASS (name: string, meets at: string, room: string, d: integer)

ENROLLED (snum: integer, cname: string)

FACULTY (fid: integer, fname: string, deptid: integer)

The meaning of these relations is straightforward; for example, Enrolled has one record per student-class pair such that the student is enrolled in the class. Level is a two character code with 4 different values (example: Junior: JR etc) Write the following queries in SQL. No duplicates should be printed in any of the answers.

- i. Find the names of all Juniors (level = JR) who are enrolled in a class taught by Prof. Harshith
- ii. Find the names of all classes that either meet in room R128 or have five or more Students enrolled.
- iii. Find the names of all students who are enrolled in two classes that meet at the same time.
- iv. Find the names of faculty members who teach in every room in which some class is taught.
- v. Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.

SQL QUERIES:

- i) Create tables specifying primary key and foreign key.

Create table STUDENT (

```
snum int not null,
sname varchar(15),
major varchar(15),
slevel varchar(15),
age int,
constraint STUDENTPK primary key (snum));
```

Create table DEPARTMENT(

```
deptid int not null,
dname varchar(15),
constraint DEPARTMENTPK primary key (deptid));
```

Create table CLASS (

```
name varchar(15) not null,
meetsat varchar(5),
room varchar(4),
```

d int,
 constraint CLASSPK primary key (name),
 constraint CLASSFK foreign key (d) references department (deptid) on
 delete set null);

Create table FACULTY (

fid int not null,
 fname varchar(15),
 deptid int,
 constraint FACULTYPK primary key (fid),
 constraint FACULTYFK foreign key (deptid) references department

(deptid) on delete set null);

Create table ENROLLED(

snum int,
 cname varchar(15),
 constraint ENROLLEDPK primary key (snum, cname),
 constraint ENROLLEDFK1 foreign key (snum) references student
 (snum) on delete set null,
 constraint ENROLLEDFK2 foreign key (cname) references class (name)

on delete set null);

(ii) Enter atleast five tuples for each relation

Insert into STUDENT values(' &snum', '&sname', '&major', '&slevel', '&age');

Similarly insert into other tables with different tuple values.

(iii) (a) Find the names of all Juniors (level = JR) who are enrolled in a class taught by Prof. Harshith

Select sname from STUDENT

where slevel = 'JR' and snum in (

Select snum from ENROLLED E, CLASS C, DEPARTMENT D, FACULTY F

where E.cname = C.name and C.d= D.deptid and D.deptid = F.deptid and F.fname=
 'Harshith');

(b) Find the names of all classes that either meet in room R128 or have five or more Students
 enrolled.

Select C.name from CLASS C, ENROLLED E

where C.room = 'R128' Union (Select C.name from CLASS C , ENROLLED E

where C.name= E.cname

group by C.name

having count(E.snum) >= 5);

(c) Find the names of all students who are enrolled in two classes that meet at the same time.

Select distinct sname from STUDENT

where snum in(Select snum from ENROLLED

where cname in(

Select C1.cname from CLASS C1, CLASS C2

where C1.meetsat= C2.meetsat)

group by snum

having count (cname)>1);

(d) Find the names of faculty members who teach in every room in which some class is taught.

Select distinct F.fname from FACULTY F, CLASS C, DEPARTMENT D

where F.deptid = D.deptid and D.deptid= C.d

group by F.fname

having count (distinct C.room) = (Select Count (distinct room) from CLASS);

(e) Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.

Select distinct fname from FACULTY F, CALSS C, DEPARTMENT D

where F.deptid =D.deptid and D.deptid =C.d group by fname

having Count(distinct C.name) <5;

Experiment No: 2**AIRCRAFT DATABASE**

The following relations keep track of airline flight information:

FLIGHTS (no: integer, from: string, to: string, distance: integer, Departs: time, arrives: time, price: real)

AIRCRAFT (aid: integer, aname: string, cruisingrange: integer)

CERTIFIED (eid: integer, aid: integer)

EMPLOYEES (eid: integer, ename: string, salary: integer)

Note that the Employees relation describes pilots and other kinds of employees as well; Every pilot is certified for some aircraft, and only pilots are certified to fly. Write each of the following queries in SQL.

- i. Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80, 000.
- ii. For each pilot who is certified for more than three aircrafts, find the eid and the maximum cruisingrange of the aircraft for which she or he is certified.
- iii. Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Frankfurt.
- iv. For all aircraft with cruisingrange over 1000 Kms, .find the name of the aircraft and the average salary of all pilots certified for this aircraft.
- v. Find the names of pilots certified for some Boeing aircraft. vi. Find the aids of all aircraft that can be used on routes from Bengaluru to New Delhi.

SQL QUERIES:

- II. Create the above tables by properly specifying the primary keys and foreign keys.

Create table AIRCRAFT (

AID int not null,

Aname varchar(15),

Cruisingrange int,

Constraint AIRCRAFTPK primary key (AID));

Create table FLIGHTS (

No int,

From varchar(15),

To varchar(15),

Distance int,

Departs varchar(05),

Arrives varchar(05),

Prices int,

Constraint FLIGHTSFK foreign key (No) references AIRCRAFT(AID) on delete set null);

Create table EMPLOYEES(

EID int not null,

Ename varchar(15),

Salary int,

Constraint EMPLOYEEESPk primary key (EID));

Create table CERTIFIED(

EID int,

AID int,

Constraint CERTIFIEDPK primary key(EID,AID),

Constraint CERTIFIEDFK1 foreign key (EID) references employees(EID) on delete set null,

Constraint CERTIFIEDFK2 foreign key (AID) references aircraft(AID) on delete set null);

II. Enter atleast five tuples for each relation.

Insert into AIRCRAFT values(' &AID', '&Aname', '&cruisingrange');

Similarly enter different values into each relation.

III.

- i. Find the names of aircraft such that all pilots certified to operate them have salaries more than Rs.80,000

Select Aname from AIRCRAFT A, CERTIFIED C, EMPLOYEES E

Where A.AID=C.AID and C.EID=E.EID and E.Salary >80000;

- ii. For each pilot who is certified for more than three aircrafts, find the eid and the maximum cruisingrange of the aircraft for which she or he is certified

Select C.EID, MAX(A.cruisingrange) from AIRCRAFT A, CERTIFIED C

Where A.AID=C.AID

Group by EID

Having count(AID)>3;

- iii. Find the names of pilots whose salary is less than the price of the cheapest route from Bengaluru to Frankfurt

Select distinct Ename from EMPLOYEES E, CERTIFIED C, AIRCRAFT A, FLIGHT F

Where E.EID=C.EID and C.AID=A.AID and A.AID=F.No and E.salary<(

Select MIN(Prices) from FLIGHT

Where from='Bangalore' and to='Frankfurt');

- iv. For all aircraft with cruisingrange over 1000 Kms, .find the name of the aircraft and the average salary of all pilots certified for this aircraft

Select A.Aname, AVG(E.salary) from AIRCRAFT A, EMPLOYEE E, CERTIFIED C

Where A.AID=C.AID and C.EID=E.EID and A.cruisingrange> 1000

Group by A.Aname;

- v. Find the names of pilots certified for some Boeing aircraft

Select E.Ename from EMPLOYEE E, CERTIFIED C, AIRCRAFT A

Where E.EID= C.EID and C.AID = A.AID and A.Aname='Boeing';

- vi. Find the aids of all aircraft that can be used on routes from Bengaluru to New Delhi

Select A. AID from AIRCRAFT A , FLIGHT F

Where A.AID=F.No and F.from='Bengaluru' and F.to='New Delhi';

Experiment No.3**STUDENT DATABASE**

Consider the following database of student enrollment in courses & books adopted for each course.

STUDENT (regno: string, name: string, major: string, bdate:date)

COURSE (course #:int, cname:string, dept:string)

ENROLL (regno:string, course#:int, sem:int, marks:int)

BOOK _ ADOPTION (course# :int, sem:int, book-ISBN:int)

TEXT (book-ISBN:int, book-title:string, publisher:string, author:string)

- i. Create the above tables by properly specifying the primary keys and the foreign keys.
- ii. Enter at least five tuples for each relation.
- iii. Demonstrate how you add a new text book to the database and make this book be adopted by some department.
- iv. Produce a list of text books (include Course #, Book-ISBN, Book-title) in the alphabetical order for courses offered by the 'CS' department that use more than two books.
- v. List any department that has all its adopted books published by a specific publisher.
- vi. Generate suitable reports.
- vii. Create suitable front end for querying and displaying the results.

SQL Queries:

- (i) Create the above tables by properly specifying the primary keys and foreign keys

```

Create table Student (
    regno varchar(15) not null,
    name varchar(15),
    Major varchar(15),
    bdate date,
    constraint Studentpk primary key (regno));

```

```

Create table course (
    Courseno int not null,
    Cname varchar(15),
    Dept varchar(15),
    Constraint Coursepk primarykey (courseno));

```

```

Create table Text (
    book_Isbn int not null,

```

```
book_Title varchar(15),
publisher  varchar(15),
author    varchar(15),
constraint Textpk primary key (book_Isbn));
```

Create table Enroll (

```
    Regno varchar(15),
    Courseno int,
    Sem int,
    Marks int,
    Constraint Enrollpk primary key( Regno, Courseno,sem),
    constraint Enrollfk1 foreign key (regno1) references student (regno) on delete set null,
    constraint Enrollfk2 foreign key (course1no ) references course(courseno) on delete set
```

null);

Create table Book_Adoption(

```
    Courseno int,
    Sem int,
    Book_Isbn int,
    Constraint Book_Adoptionpk primary key (courseno, Book_Isbn, sem)
    constraint Book_Adoptionfk1 foreign key courseno references course (courseno) on delete
set null,
    constraint Book_Adoptionfk2 foreign key bookIsbn references Text(book-Isbn) on delete
```

set null);

ii) Enter atleast five tuples for each relation.

Insert into student values ('&Regno', '&name', '&major', '&Bdate');

iii) Demonstrate how you add a new text book to the database and make this book be adopted by some department.

Insert into text values ('B1', 'Complete Reference', 'Tata Mac', 'Robert');

Insert into Book-Adoption values('1','v','B1');

iv) Produce a list of textbooks with courseno , Book_Isbn , Book_title in the alphabetical order for courses offered by the CSE department that use more than two books .

Select B.courseno ,T.Book_Isbn,T.Book_title from course C , text T, Book_Adoption B
where C.Department like 'CS' and T.BookISBN =B.BookISBN and C.courseno=B.courseno and B.courseno in(

Select courseno from Book_Adoption

Group by Courseno having count(courseno) > 2)

Order by T.BookTitle ASC ;

v) List any dept that has all its adopted books published by a specific publisher.

Select all dept from course

Where courseno in (select courseno from book adoption

where book_Isbn in(select book_Isbn from text

where publisher = 'Tata McGrawHill')));

vi) Generation of suitable reports.

Select Distinct C.Dept,T.Publisher From Course As C,Text As T,Book_Adoption As Ba Where C.C_Id=Ba.C_Id And T.Bookisbn=Ba.Bookisbn And T.Publisher="Pearson";

Experiment No.4**BOOK DATABASE**

The following tables are maintained by a book dealer.

AUTHOR (author-id:int, name:string, city:string, country:string)

PUBLISHER (publisher-id:int, name:string, city:string, country:string)

CATALOG (book-id:int, title:string, author-id:int, publisher-id:int, category-id:int, year:int, price:int)

CATEGORY (category-id:int, description:string)

ORDER-DETAILS (order-no:int, book-id:int, quantity:int)

- i. Create the above tables by properly specifying the primary keys and the foreign keys.
- ii. Enter at least five tuples for each relation.
- iii. Give the details of the authors who have 2 or more books in the catalog and the price of the books is greater than the average price of the books in the catalog and the year of publication is after 2000.
- iv. Find the author of the book which has maximum sales.
- v. Demonstrate how you increase the price of books published by a specific publisher by 10%.
- vi. Generate suitable reports.
- vii. Create suitable front end for querying and displaying the results.

SQL Queries:

(i) Create table Author (

author_id int not null,

Name varchar(15),

City varchar(15),

Country varchar(15),

Constraint Authorpk primary key (author_id));

Create table publisher (

publisher_id int not null,

name varchar(15),

city varchar(15),

country varchar(15),

constraint publisherpk primary key(publisher_id));

Create table category (

Category_id int not null,

description varchar(15),

```
constraint categorypk primary key ( category_id));
```

```
Create table catalog (
```

```
    book_id int not null,
```

```
    title varchar(15),
```

```
    author_id int,
```

```
    publisher_id int,
```

```
    category_id int,
```

```
    year int.
```

```
    price int,
```

```
    constraint catalogpk primary key ( book_id) ,
```

```
    constraint catalogfk1 foreign key (author_id) references author (author_id) on delete set null ,
```

```
    constraint catalogfk2 foreign key (publisher_id) references publisher(publisher_id) on delete set null,
```

```
    constraint catalogfk3 foreign key (category_id) references category (category_id) on delete set null);
```

```
Create table order_details (
```

```
    Order_no int,
```

```
    Book_id int,
```

```
    qty int,
```

```
    constraint order_detailspk primary key (order_no, book_id),
```

```
    constraint order_detailsfk foreign key (book_id) references catalog (book_id) on delete set null) ;
```

(ii) Insert into Author values (101,'Ullman','Bangalore','India');

Similarly insert values into other tables.

- (iii) Give the details of the authors who have two or more books in the catalog and the price of the books is greater than the average price of the books in the catalog and the year of the publication is after 2000.

Select distinct (*) from author

Where author_id in (

Select author_id from catalog where price > (

Select avg(price) from catalog)and year >2000 group by author_id having count (*)>=2);

- (iv) Find the author of the books who has maximum sales.

Select name , author_id from author

Where author_id in (Select author_id, Book_id from catalog

Where book_id in (Select book_id from order_details

Where qty in (Select max(qty) from order details));

- (v) Demonstrate how you increase the price of books published by specific publisher by 10%.

Update catalog

set price = price*1.1

where publisher_id in (Select publisher_id from publisher

where name= 'TMH');

- (vi) Generation of suitable Reports.

Select A.Aname As 'Author_Name',C.Bookid As 'Book_Id',C.Title As 'Book_Title' From Author A,Catalog C Where A.Authorid=C.Authorid And A.Aname='Tennenbaum';

Experiment No.5**BANK DATABASE**

Consider the following database for a banking enterprise

BRANCH(branch-name:string, branch-city:string, assets:real)

ACCOUNT(accno:int, branch-name:string, balance:real)

DEPOSITOR(customer-name:string, accno:int)

CUSTOMER(customer-name:string, customer-street:string, customer-city:string)

LOAN(loan-number:int, branch-name:string, amount:real)

BORROWER(customer-name:string, loan-number:int)

- i. Create the above tables by properly specifying the primary keys and the foreign keys
- ii. Enter at least five tuples for each relation
- iii. Find all the customers who have at least two accounts at the Main branch.
- iv. Find all the customers who have an account at all the branches located in a specific city.
- v. Demonstrate how you delete all account tuples at every branch located in a specific city.
- vi. Generate suitable reports.
- vii. Create suitable front end for querying and displaying the results.

SQL Queries:

(i) Create table Branch (

Branch_name varchar(15) not null,

Branch_city varchar(15),

assets real,

constraint Branchpk primary key (Branch_name));

Create table Account (

Acc_no int not null,

branch_name varchar(15),

balance real ,

constraint Accountpk primary key (accno),

constraint Accountfk foreign key branch_name references Branch (Branch_name) on delete set null);

Create table Customer (

Customer_name varchar(15) not null,

Customer_street varchar(15),

Customer_city varchar(15),

Constraint Customerpk primary key customer_name);

Create table Depositor (

Customer_name varchar(15),

acc_no int,

constraint Depositorpk primary key(Customer_name, acc_no),

constraint Depositorfk1 foreign key (customer_name) references customer (customer_name)

on delete set null,

constraint Depositorfk2 foreign key (acc_no) references Account (acc_no) on delete set

null);

Create table Loan (

Loan_number int,

Branch_name varchar(15),

Amount real,

constraint Loanpk primary key (Loan_number),

constraint Loanfk foreign key (branch_name) references Branch (Branch_name) on delete set

null);

Create table Borrower (

Customer_name varchar(15),

Loan_number varchar(15),

Constraint Borrowerpk primary key(customer_name, Loan_number),

Constraint Borrowerfk1 foreign key (customer_name) references customer (customer_name)

on delete set null,

Constraint Borrowerfk2 foreign key (Loan_number) references Loan (Loan_number) on delete set null);

(ii) Insert into Branch values ('MVJCE', 'Bangalore', '3650.50');

Similarly enter values for all other tables

(iii) Find the entire customers who have at least two accounts at the main branch.

Select customer_name from Depositor

Where acc_no in (Select acc_no from Account branch_name = 'main')

Group by acc_no having count(accno)>2 ;

iv) Find the entire customers who have an account at all the branches located in specific city.

Select Customer_Name from Customer

Where acc_no in(Select acc_no from Account

where branch_name in(Select Branch_name from Branch

where Branch_city like 'Bangalore');

v) Demonstrate how you delete all account tuples at every branch located in a specific city.

Delete Acc_no from Account where Branch_name in(Select branch_name , branch_city from Branch

Where branch_city like 'Bangalore'

Group by Branch _city);

(vi)Generation of of suitable reports Report:

List customer names and address who have taken loan in Bangalore Branches.

Select Ba.Cname, C.Cstreet,C.City From Customer As C,Loan As L,Borrower As Ba,Branch As B Where Ba.Loanno=L.Loanno And Ba.Cname=C.Cname And B.Bname=L.Bname And B.Bcity="Bangalore";

VIVA VOCE

1. What is database?

A database is a collection of data in an organized manner.

2. What is DBMS?

DBMS is a software that is used to perform operations on a database. These operations may include reading, writing, modifying of data and even provide control over accessing of data when multiple users were accessing the data at the same time or even at different times.

3. Advantages of DBMS?

- ✓ Redundancy is controlled.
- ✓ Unauthorized access is restricted.
- ✓ Providing multiple user interfaces.
- ✓ Enforcing integrity constraints.
- ✓ Providing backup and recovery.

4. Disadvantage in File Processing System?

- ✓ Data redundancy & inconsistency.
- ✓ Difficult in accessing data.
- ✓ Data isolation.
- ✓ Data integrity.
- ✓ Concurrent access is not possible.
- ✓ Security Problems.

5. Describe the three levels of data abstraction?

Three levels of abstraction:

- Physical level: The lowest level of abstraction describes how data are stored.
- Logical level: The next higher level of abstraction, describes what data are stored in database and what relationship among those data.
- View level: The highest level of abstraction describes only part of entire database.

6. Define the "integrity rules"

There are two Integrity rules.

- Entity Integrity: States that? Primary key cannot have NULL value?
- Referential Integrity: States that? Foreign Key can be either a NULL value or should be Primary Key value of other relation.

7. What is Data Independence?

Data independence means, the ability to modify the schema definition in one level should not affect the schema definition in the next higher level.

Two types of Data Independence:

- Physical Data Independence: Modification in physical level should not affect the logical level.
- Logical Data Independence: Modification in logical level should affect the view level.

NOTE: Logical Data Independence is more difficult to achieve

8. What is a view? How it is related to data independence?

A view may be thought of as a virtual table, that is, a table that does not really exist in its own right but is instead derived from one or more underlying base tables. Growth and restructuring of base tables is not reflected in views. Thus the view can insulate users from the effects of restructuring and growth in the database. Hence accounts for logical data independence.

9. What is Data Model?

A collection of conceptual tools for describing data, data relationships, data semantics and constraints is called Data Model.

10. What is E-R model?

This data model is based on real world that consists of basic objects called entities and of relationship among these objects. Entities are described in a database by a set of attributes.

11. What is Object Oriented model?

This model is based on collection of objects. An object contains values stored in instance variables within the object. An object also contains bodies of code that operate on the object. These bodies of code are called methods. Objects that contain same types of values and the same methods are grouped together into classes.

12. What is an Entity?

It is a 'thing' in the real world with an independent existence.

13. What is an Entity type?

It is a collection (set) of entities that have same attributes.

14. What is an Entity set?

It is a collection of all entities of particular entity type in the database.

15. What is Weak Entity set?

An entity set may not have sufficient attributes to form a primary key, and its primary key compromises of its partial key and primary key of its parent entity, then it is said to be Weak Entity set.

16. What is an attribute?

It is a particular property, which describes the entity.

17. What is a Relation Schema and a Relation?

A relation Schema denoted by $R (A_1, A_2, \dots, A_n)$ is made up of the relation name R and the list of attributes A_i that it contains. A relation is defined as a set of tuples. Let r be the relation which contains set tuples $(t_1, t_2, t_3 \dots t_n)$. Each tuple is an ordered list of n -values $t = (v_1, v_2 \dots v_n)$.

18. What is degree of a Relation?

It is the number of attributes of its relation schema.

19. What is Relationship?

It is an association among two or more entities.

20. What is Relationship set?

The collection (or set) of similar relationships.

21. What is Relationship type?

Relationship type defines a set of associations or a relationship set among a given set of entity types.

22. What is DML Compiler?

It translates DML statements in a query language into low-level instruction that the query evaluation engine can understand.

23. What is Query evaluation engine?

It executes low-level instruction generated by compiler.

24. What is DDL Interpreter?

It interprets DDL statements and record them in tables containing metadata.

25. What is Record-at-a-time?

The Low level or Procedural DML can specify and retrieve each record from a set of records. This retrieve of a record is said to be Record-at-a-time.

26. What is Relational Algebra?

It is procedural query language. It consists of a set of operations that take one or two relations as input and produce a new relation.

27. What is normalization?

It is a process of analysing the given relation schemas based on their Functional Dependencies (FDs) and primary key to achieve the properties

- Minimizing redundancy
- Minimizing insertion, deletion and update anomalies.

28. What is Functional Dependency?

A Functional dependency is denoted by $X \rightarrow Y$ between two sets of attributes X and Y that are subsets of R specifies a constraint on the possible tuple that can form a relation state r of R . The constraint is for any two tuples t_1 and t_2 in r if $t_1[X] = t_2[X]$ then they have $t_1[Y] = t_2[Y]$. This means the value of X component of a tuple uniquely determines the value of component Y .

29. When is a functional dependency F said to be minimal?

- Every dependency in F has a single attribute for its right hand side.

- We cannot replace any dependency $X \rightarrow A$ in F with a dependency $Y \rightarrow A$ where Y is a proper subset of X and still have a set of dependency that is equivalent to F .
- We cannot remove any dependency from F and still have set of dependency that is equivalent to F .

30. What is Lossless join property?

It guarantees that the spurious tuples generation does not occur with respect to relation schemas after decomposition.

31. What is Fully Functional dependency?

It is based on concept of full functional dependency. A functional dependency $X \rightarrow Y$ is full functional dependency if removal of any attribute A from X means that the dependency does not hold any more.

32. What is 1 NF (Normal Form)?

The domain of attribute must include only atomic (simple, indivisible) values.

33. What is 2NF?

A relation schema R is in 2NF if it is in 1NF and every non-prime attribute A in R is fully functionally dependent on primary key.

34. What is 3NF?

A relation schema R is in 3NF if it is in 2NF and for every FD $X \rightarrow A$ either of the following is true

- X is a Super-key of R .
- A is a prime attribute of R .

In other words, if every non prime attribute is non-transitively dependent on primary key.

35. What is BCNF (Boyce-Codd Normal Form)?

A relation schema R is in BCNF if it is in 3NF and satisfies an additional constraint that for every FD $X \rightarrow A$, X must be a candidate key.