MALNAD COLLEGE OF ENGINEERING

*(An Autonomous Institute Affiliated to VTU, Belagavi)*

Under the auspices of the MTES®, Hassan

## Department of Computer Science & Engineering

## INTRODUCTION

**Database is a collection of related data. Database management system is software designed to assist the maintenance and utilization of large scale collection of data. DBMS came into existence in 1960 by Charles. Integrated data store which is also called as the first general purpose DBMS. Again in 1960 IBM brought IMS-Information management system. In 1970 Edgor Codd at IBM came with new database called RDBMS. In 1980 then came SQL Architecture- Structure Query Language. In 1980 to 1990 there were advances in DBMS e.g. DB2, ORACLE.**

## Relational Model

**Domain**: A (usually named) set/universe of atomic values, where by "atomic" we mean simply that, from the point of view of the database, each value in the domain is indivisible (i.e., cannot be broken down into component parts).

Examples of domains:

- USA_phone_number: string of digits of length ten
- SSN: string of digits of length nine
- Name: string of characters beginning with an upper case letter
- GPA: a real number between 0.0 and 4.0
- Sex: a member of the set { female, male }
- Dept_Code: a member of the set { CMPS, MATH, ENGL, PHYS, PSYC, ... }

These are all logical descriptions of domains. For implementation purposes, it is necessary to provide descriptions of domains in terms of concrete **data types** (or **formats**) that are provided by the DBMS (such as String, int, boolean), in a manner analogous to how programming languages have intrinsic data types.

**Attribute**: the name of the role played by some value (coming from some domain) in the context of a **relational schema**. The domain of attribute A is denoted dom(A).

**Tuple**: A tuple is a mapping from attributes to values drawn from the respective domains of those attributes. A tuple is intended to describe some entity (or relationship between entities) in the mini world.

As an example, a tuple for a PERSON entity might be

{Name --> "Uday", Sex --> “Male”, Id--> 153 }

**Relation**: A (named) set of tuples, all of the same form (i.e., having the same set of attributes). Each table resembles a table of values, to some extent. Each row in the table represents a collection of related values.

**Relational Schema**: used for describing (the structure of) a relation. E.g., R(A1, A2, ..., An) says that R is a relation with attributes A1, ... An. The **degree** of a relation is the number of attributes it has, here n. for the below example the degree of STUDENT relation is 3.

Example: STUDENT (Name, SSN, Address)

**Relational Database**: A collection of **relations**, each one consistent with its specified relational schema.

**STRUCTURED QUERY LANGUAGE**

- SQL stands for Structured Query Language
- SQL lets you access and manipulate databases
- SQL is an ANSI (American National Standards Institute) standard
- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database 2i 39 h 0
- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views

SQL having different language categories as follows:

1. **Data Definition Language** (DDL) statements are used to define the database structure or schema. Some examples:

- **CREATE** - to create objects in the database

  Syntax: Create table ( (), ));

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

- **ALTER** - alters the structure of the database

  Add new columns Syntax: Alter table add(<datatype(size),datatype(size));

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

- **DROP** - delete objects from the database

  Syntax: Drop table ;
  Ex:drop table emp;

- **TRUNCATE** - remove all records from a table, including all spaces allocated for the records are removed

  Syntax: Truncate table <table_name>;
  Ex: truncate table emp1;

- **COMMENT** - add comments to the data dictionary
- **RENAME** - rename an object
  Syntax: Rename old _name to new_name;
  Ex: Rename emp to emp1;

- **VIEW**- Views can be defined in MySQL using a statement similar to Oracle’s
  CREATE [OR REPLACE] VIEW view_name[(col-list)]AS *SELECT-statement*;

**2. Data Manipulation Language** (DML) statements are used for managing data within schema objects. Some examples:

- SELECT - retrieve data from the a database
- INSERT - insert data into a table

**Syntax**
INSERT [TRANSACTION transaction] INTO object [(col [, col …])] {VALUES (val [, val …]) | select_expr};
OR
Insert into Relation_Name values(Value1, Value2,……….., Valuen);

- UPDATE - updates existing data within a table
  UPDATE table_name SET column1=value, column2=value2,...;

- DELETE - deletes all records from a table, the space for the records remain
  DELETE FROM table_name WHERE some_column=some_value;

- MERGE - UPSERT operation (insert or update)
  MERGE **INTO** tablename **USING** table_reference **ON** (condition)
  **WHEN** MATCHED **THEN**
  **UPDATE SET** column1 = value1 [, column2 = value2 ...]
  **WHEN NOT** MATCHED **THEN**
  **INSERT** (column1 [, column2 ...]) **VALUES** (value1 [, value2 ...]);

- REPLACE- MySQL also has a REPLACE command that can either update an old record or insert a new one, depending on whether the record exists already. The form is similar to the INSERT form.

  REPLACE INTO *tablename*(*colnames*)VALUES(*column_values*);

**3. Data Control Language** (DCL) statements. Some examples:

- GRANT - gives user's access privileges to database
- REVOKE - withdraw access privileges given with the GRANT command

**4. Transaction Control** (TCL) statements are used to manage the changes made by DML statements. It allows statements to be grouped together into logical transactions.

- COMMIT - save work done

  - SAVEPOINT - identify a point in a transaction to which you can later roll back
  - ROLLBACK - restore database to original since the last COMMIT

SET TRANSACTION - Change transaction options like isolation level and what rollback
DML statements can be roll backed where DDL are auto commit.

**5. Database Design:**

- To create a database in which your tables will reside, you can use the command
  CREATE DATABASE [IF NOT EXISTS] *database_name*;
- You can optionally use the form CREATE SCHEMA instead. You can then connect to the database either by
  USE *database_name*;
  or
  CONNECT *database_name*;
- To create a table, MySQL uses a form very similar to Oracle's
  CREATE TABLE [IF NOT EXISTS] *table_name*(*column_list*)[*table options*][*partition options*];
- To see all the tables you have created in a database, enter at the command prompt
  Show tables;
- To see the structure of a particular table, enter
  Describe *table_name*;
- To see a list of all databases that exist on the server, enter Show databases;

**1. Design an Employee database and answer following queries:**

**a) List all the employees who are above 40 years of age**
**b) List the employees who work in a particular department**
**c) List the female employees who are 30 years of age and drawing salary >8000**
**d) List the employee number, employee name and his department name of all employees**

**Solution:**

```
create table department(Deptno int primary key,Dname varchar(20) not null,Location varchar(20));
create table Employee(Empid int primary key,Empn_name varchar(20) not null,Gender varchar(2),age
int,
Salary int,Dno int,foreign key(Dno) references department(Deptno));
insert into department values(1,'CSE','CS-block');
insert into department values(2,'ISE','IS-block');
insert into department values(3,'EEE','EE-block');
insert into department values(4,'ECE','ECE-block');
insert into department values(5,'CV','Civil-block');
select * from department;
+--------+-------+-------------+
| Deptno | Dname | Location |
+--------+-------+-------------+
| 1 | CSE | CS-block |
| 2 | ISE | IS-block |
| 3 | EEE | EE-block |
| 4 | ECE | ECE-block |
| 5 | CV | Civil-block |
+--------+-------+-------------+
insert into Employee values(101,'Yashvanth','M',21,30000,1);
insert into Employee values(102,'Yashas','M',21,30000,1);
insert into Employee values(103,'Yashaswini','F',20,30000,4);
insert into Employee values(104,'Ashwini','F',30,8000,4);
insert into Employee values(105,'Asha','F',30,90000,3);
insert into Employee values(106,'Arun','F',50,90000,3);
select * from Employee;
+-------+------------+--------+------+--------+------+
| Empid | Empn_name | Gender | age | Salary | Dno |
+-------+------------+--------+------+--------+------+
| 101 | Yashvanth | M | 21 | 30000 | 1 |
| 102 | Yashas | M | 21 | 30000 | 1 |
| 103 | Yashaswini | F | 20 | 30000 | 4 |
| 104 | Ashwini | F | 30 | 8000 | 4 |
| 105 | Asha | F | 30 | 90000 | 3 |
+-------+------------+--------+------+--------+------+
a) select * from Employee where age>40;
+-------+-----------+--------+------+--------+------+
| Empid | Empn_name | Gender | age | Salary | Dno |
+-------+-----------+--------+------+--------+------+
| 106 | Arun | F | 50 | 90000 | 3 |
+-------+-----------+--------+------+--------+------+
b) select * from Employee where age>40;
+-------+-----------+--------+------+--------+------+
| Empid | Empn_name | Gender | age | Salary | Dno |
+-------+-----------+--------+------+--------+------+
| 106 | Arun | F | 50 | 90000 | 3 |
+-------+-----------+--------+------+--------+------+
c) select *from Employee where age=30 and Gender='F' and Salary>8000;
+-------+-----------+--------+------+--------+------+
| Empid | Empn_name | Gender | age | Salary | Dno |
+-------+-----------+--------+------+--------+------+
| 105 | Asha | F | 30 | 90000 | 3 |
+-------+-----------+--------+------+--------+------+
d) select Empid,Empn_name,Dname
from Employee,department
where Deptno=Dno;
+-------+------------+-------+
| Empid | Empn_name | Dname |
+-------+------------+-------+
| 101 | Yashvanth | CSE |
| 102 | Yashas | CSE |
| 105 | Asha | EEE |
| 106 | Arun | EEE |
| 103 | Yashaswini | ECE |
| 104 | Ashwini | ECE |
+-------+------------+-------+
```

**2. Design a video parlor database and solve the following queries as outlined below using SQL.**

**a) List only videos in the Children category with a daily rental rate of less than 100 and sorted according to video title.**

**b) List the catalogNo, title and category of the Video table, ordered by video title**

**c) List all videos with a certification of “U” or “B” in the Video table.**

```
CREATE TABLE MEMBERDETAILS (MID INT PRIMARY KEY, NAME VARCHAR (20),
YOJ INT);
CREATE TABLE VIDEODETAILS (VID INT PRIMARY KEY, CATEGORY
VARCHAR(20),
TITLE VARCHAR(25),CERTIFICATE VARCHAR(10));
CREATE TABLE VEDRENT (VID INT, COST INT, FOREIGN KEY(VID) REFERENCES
VIDEODETAILS(VID));
CREATE TABLE RENT_BY_MEM(MID INT,VID INT,FOREIGN KEY(VID)
REFERENCES
VIDEODETAILS(VID),FOREIGN KEY(MID) REFERENCES MEMBERDETAILS(MID));

INSERT INTO MEMBERDETAILS VALUES(101,&#39;UJWAL&#39;,2023);
INSERT INTO MEMBERDETAILS VALUES(102,&#39;USHA&#39;,2023);
INSERT INTO MEMBERDETAILS VALUES(103,&#39;UTTAM&#39;,2023);
INSERT INTO MEMBERDETAILS VALUES(104,&#39;ULLAS&#39;,2022);
INSERT INTO MEMBERDETAILS VALUES(105,&#39;UMA&#39;,2022);
SELECT * FROM MEMBERDETAILS;
+-----+-------+------+
| MID | NAME | YOJ |
+-----+-------+------+
| 101 | UJWAL | 2023 |
| 102 | USHA | 2023 |
| 103 | UTTAM | 2023 |
| 104 | ULLAS | 2022 |
| 105 | UMA | 2022 |
+-----+-------+------+
INSERT INTO VIDEODETAILS VALUES (11, CHILDREN,CHOTA BHEEM,U);
INSERT INTO VIDEODETAILS VALUES(12, CHILDREN,MASHA,U;);
INSERT INTO VIDEODETAILS VALUES(13,CHILDREN,SPIDER,V);
INSERT INTO VIDEODETAILS VALUES(14,ADULT,SPIDER,B);
INSERT INTO VIDEODETAILS VALUES(15,ADULT,DRAGON,B);
SELECT * FROM VIDEODETAILS;
+-----+----------+-------------+-------------+
| VID | CATEGORY | TITLE | CERTIFICATE |
+-----+----------+-------------+-------------+
| 11 | CHILDREN | CHOTA BHEEM | U |
| 12 | CHILDREN | MASHA | U |
| 13 | CHILDREN | SPIDER | V |
| 14 | ADULT | SPIDER | B |
| 15 | ADULT | DRAGON | B |
+-----+----------+-------------+-------------+

INSERT INTO VEDRENT VALUES(12,100);
INSERT INTO VEDRENT VALUES(11,90);
INSERT INTO VEDRENT VALUES(13,110);
INSERT INTO VEDRENT VALUES(14,800);
INSERT INTO VEDRENT VALUES(15,80);
SELECT * FROM VEDRENT;
+------+------+
| VID | COST |
+------+------+
| 12 | 100 |
| 11 | 90 |
| 13 | 110 |
| 14 | 800 |
| 15 | 80 |
+------+------+
INSERT INTO RENT_BY_MEM VALUES(101,12);
INSERT INTO RENT_BY_MEM VALUES(101,14);
INSERT INTO RENT_BY_MEM VALUES(102,13);
INSERT INTO RENT_BY_MEM VALUES(103,11);
INSERT INTO RENT_BY_MEM VALUES(104,15);
INSERT INTO RENT_BY_MEM VALUES(105,14);
SELECT * FROM RENT_BY_MEM;
+------+------+
| MID | VID |
+------+------+
| 101 | 12 |
| 101 | 14 |
| 102 | 13 |
| 103 | 11 |
| 104 | 15 |
| 105 | 14 |
+-----+------+
```

a) List only videos in the Children category with a daily rental rate of less than 100 and sorted according to video title.

```
SELECT VD.TITLE, VR.COST, VD.VID FROM VIDEODETAILS VD, VEDRENT VR
WHERE VD.CATEGORY=’CHILDREN’ AND VR.COST<100 AND VD.VID=VR.VID
ORDER BY VD.TITLE;
+-------------+------+-----+
| TITLE | COST | VID |
+-------------+------+-----+
| CHOTA BHEEM | 90 | 11 |
+-------------+------+-----+
```

b) List the catalogNo, title and category of the Video table, ordered by video title

```
SELECT VID AS CATALOGNO, TITLE, CATEGORY FROM VIDEODETAILS ORDER BY TITLE;
```

```
+-----------+-------------+----------+
| CATALOGNO | TITLE | CATEGORY |
+-----------+-------------+----------+
| 11 | CHOTA BHEEM | CHILDREN |
| 15 | DRAGON | ADULT |
| 12 | MASHA | CHILDREN |
| 13 | SPIDER | CHILDREN |
| 14 | SPIDER | ADULT |
+-----------+-------------+----------+
```

c) List all videos with a certification of “U” or “B” in the Video table.

```
SELECT VID, TITLE, CERTIFICATE FROM VIDEODETAILS WHERE CERTIFICATE=’U’ OR CERTIFICATE=’B’;
```

```
+-----+-------------+-------------+
| VID | TITLE | CERTIFICATE |
+-----+-------------+-------------+
| 11 | CHOTA BHEEM | U |
| 12 | MASHA | U |
| 14 | SPIDER | B |
| 15 | DRAGON | B |
+-----+-------------+-------------+
```

**3. Consider the Insurance database. Create the tables by properly specifying the primary keys and the foreign keys, Enter at least five tuples for each relation.**

**a) Demonstrate how you**

**i. Update the damage amount for the car with a specific regno in accident with report number 12 to 25000**

**ii. Add a new accident to the database**

**b) Find the total number of people who owned cars that were involved in accidents in 2006.**

**c) Find the number of accidents in which cars belonging to a specific model were involved.**

```
create table person(driverid varchar(10) primary key,name varchar(20),address varchar(10));
create table car(regno varchar(10) primary key,model varchar(10),year int);
create table accident(reportno int primary key,date_of_accident int,location varchar(20));
create table owns(driverid varchar(10),regno varchar(10),foreign key(driverid) references
person(driverid),foreign key(regno) references car(regno));
create table participated(driverid varchar(10),regno varchar(10),reportno int,damage_amount
int,foreign key(driverid) references person(driverid),foreign key(regno) references car(regno),foreign
key (reportno) references accident (reportno));

insert into person values('HSN1','SUHAS','HASSAN');
 insert into person values('HSN2','SUHANA','HASSAN');
 insert into person values('HSN3','SUMA','MYSURU');
insert into person values('HSN4','SUMANTH','BELUR');
insert into person values
('HSN5','SUMATHI','BELLURU');
SELECT * FROM person;
+          +          +          +
| driverid | name    | address |
+          +          +          +
| HSN1     | SUHAS | HASSAN |
| HSN2     | SUHANA | HASSAN |
| HSN3     | SUMA    | MYSURU |
| HSN4     | SUMANTH | BELUR |
| HSN5     | SUMATHI | BELLURU |
+          +          +          +
insert into car values('KA13','ALTO-
800',2006); insert into car
values('KA14','BENZ',2007);
 insert into car values('KA15','BENZ',2007);
insert into car
values('KA16','SANTRO',2007);
insert into car values('KA17','SWIFT',2008);
```

```
 insert into car
values('KA18','BALENO',2009);

SELECT * FROM car;
+ + + +
| regno | model    | year |
+ + + +
| KA13 | ALTO-800 | 2006 |
| KA14 | BENZ     | 2007 |
| KA15 | BENZ     | 2007 |
| KA16 | SANTRO | 2007 |
| KA17 | SWIFT    | 2008 |
| KA18 | BALENO | 2009 |
+ + + +
insert into accident values(11,2006,'RC ROAD');
 insert into accident values(12,2007,'MG
ROAD'); insert into accident
values(13,2007,'URS ROAD'); insert into
accident values(14,2005,'JP ROAD');
insert into accident values(15,2007,'BM
ROAD');

SELECT * FROM accident;
+ + + +
| reportno | date_of_accident | location |
+ + + +
|      11 |           2006 | RC ROAD |
       12 |           2007 | MG ROAD |

|      13 |           2007 | URS ROAD |

|      14 |           2005 | JP ROAD |

|      15 |           2007 | BM ROAD |

insert into owns
values('HSN1','KA13');

insert into owns
values('HSN1','KA14'); insert into
owns values('HSN2','KA15'); insert
into owns values('HSN3','KA16');
insert into owns
values('HSN4','KA17'); insert into
```

```
owns values('HSN5','KA18');

SELECT * FROM owns;
| driverid | regno |
| HSN1     | KA13 |
| HSN1     | KA14 |
| HSN2   | KA15    |

| HSN4   | KA17    |

insert                    into                    participated
values('HSN1','KA17',11,30000);
insert                    into                    participated
values('HSN1','KA16',12,30000);
insert                    into                    participated
values('HSN2','KA15',12,35000);
insert                    into                    participated
values('HSN3','KA16',12,45000);
insert                    into                    participated
values('HSN4','KA13',13,45000);
insert                    into                    participated
values('HSN5','KA14',14,5000);
 insert                    into                   participated
values('HSN5','KA15',12,5000);

SELECT * FROM participated;
+          +        +          +                +
| driverid | regno | reportno | damage_amount |
+          +        +          +                +
| HSN1   | KA17    |      11 |    30000 |

| HSN1   | KA16    |      12 |    30000 |

| HSN2   | KA15    |      12 |    35000 |

| HSN3   | KA16    |      12 |    45000 |

| HSN4   | KA13    |      13 |    45000 |

| HSN5   | KA14    |      14 |    5000 |

| HSN5   | KA15    |      12 |    5000 |
+          +        +          +                +
update participated SET damage_amount=40000 where reportno=12;

mysql> SELECT * FROM participated;

+          +        +          +                +
| driverid | regno | reportno | damage_amount |
+            +         +            +                  +
| HSN1   | KA17     |       11 |    30000 |
| HSN1   | KA16     |       12 |    40000 |
| HSN2   | KA15     |       12 |    40000 |
| HSN3   | KA16     |       12 |    40000 |
| HSN4   | KA13     |       13 |    45000 |
| HSN5   | KA14     |       14 |    5000 |
| HSN5   | KA15     |       12 |    40000 |
```

```
insert into accident values(20,2020,'KP ROAD');

SELECT * FROM accident;
+            +                    +            +
| reportno | date_of_accident | location |
+            +                    +            +
|       11 |            2006 | RC ROAD |
|       12 |            2007 | MG ROAD |
|       13 |            2007 | URS ROAD |
|       14 |            2005 | JP ROAD |
|       15 |            2007 | BM ROAD |
|       20 |            2020 | KP ROAD |
+            +                    +            +
insert into participated values('HSN5','KA15',20,2000);
```

```
SELECT * FROM participated;
+          +        +          +              +
| driverid | regno | reportno | damage_amount |
+          +        +          +              +
| HSN1   | KA17    |      11 |    30000 |
| HSN1   | KA16    |      12 |    40000 |
| HSN2   | KA15    |      12 |    40000 |
| HSN3   | KA16    |      12 |    40000 |
| HSN4   | KA13    |      13 |    45000 |
| HSN5   | KA14    |      14 |    5000 |
| HSN5   | KA15    |      12 |    40000 |
| HSN5   | KA15    |      20 |    2000 |
+          +        +          +              +
```

2) select MAX(name) from person p,accident a,participated pa where date_of_accident like '2007' and a.reportno=pa.reportno and p.driverid=pa.driverid;

```
+           +
| MAX(name) |
+           +
| SUMATHI |
+           +
```

3) select count(*) from car c,participated p
-> where model='BENZ' and c.regno=p.regno;

```
+          +
| count(*) |
+          +
|      4    |
+          +
```

**4. Consider the database of student enrollment in courses and books adopted for each course.**

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

**b) 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 2 books.**

**c) List the department that has adopted books published by specific publisher.**

Solution:

```
create database STUDENT;
use STUDENT;
create table student(regno char(15) primary key,name char(15),major char(10),bdate
date);
 create table course(courseno int primary key,cname char(15),dept char(10));
 create table enroll(regno char(15),courseno int,sem int,marks int,foreign key(regno)
references student(regno),foreign key(courseno) references course(courseno));
create table text(book_isbn int primary key,book_title char(20),publisher char(15),author char(15));
create table book_adoption(courseno int,sem int,book_isbn int,foreign key(courseno)
references course(courseno),foreign key(book_isbn) references text(book_isbn));

insert into student values('CS001','Krishna','CSE','2000-09-
01');
insert into student values('CS002','Seema','CSE','2000-05-06');
 insert into student values('CS005','Chaithra','CSE','2000-02-
13'); insert into student values('IS003','Smitha','ISE','2000-10-
06');
insert into student values('IS004','Hemanth','ISE','2000-08-12');
insert into student values('AU006','Chethana','AU','2000-08-
13'); insert into student values('EE006','Chethan','EE','2000-08-
25');
+       +          +       +           +
| regno | name      | major | bdate      |
+       +          +       +           +
| AU006 | Chethana | AU    | 2000-08-13 |
| CS001 | Krishna | CSE | 2000-09-01 |
| CS002 | Seema     | CSE | 2000-05-06 |
| CS005 | Chaithra | CSE | 2000-02-13 |
| EE006 | Chethan | EE      | 2000-08-25 |
| IS003 | Smitha | ISE | 2000-10-06 |
| IS004 | Hemanth | ISE | 2000-08-12 |
+       +          +       +           +
insert into course
values(1,'AJAVA','CSE'); insert into
course values(2,'OS','CSE');
```

```
 insert                into               course
 values(3,'DBMS','CSE');    insert    into
 course values(4,'WP','ISE');
 insert into course values(5,'CN','ISE');
insert       into        course        values(6,'Machine
Design','Auto');         insert         into         course
values(7,'Electronics','EEE');
 select * from course;
 +          +                  +       +
 | courseno | cname           | dept |
 +          +                  +       +
 |        1 | AJAVA        | CSE |
 |        2 | OS              | CSE |
 |        3 | DBMS          | CSE  |
 |        4 | WP             | ISE |
 |        5 | CN             | ISE |
 |        6 | Machine Design | Auto |
 |        7 | Electronics    | EEE |
 +          +                  +       +

 insert                into                enroll
 values('CS002',2,5,90);    insert    into
 enroll    values('CS005',3,5,92);    insert
 into    enroll    values('AU006',6,3,92);
 insert into enroll values('IS003',5,5,82);
 insert into enroll values('IS004',4,6,82);
 select * from enroll;
 +       +           +       +        +
 | regno | courseno | sem | marks |
 +       +           +       +        +
  | CS001 |    1 |     5 |    95 |
  | CS002 |    2 |     5 |    90 |
  | CS005 |    3 |     5 |    92 |
  | AU006 |   6 |     3 |    92 |
  | IS003 |        5 |    5 |    82 |
  | IS004 |        4 |    6 |    82 |
 +       +           +       +        +
 insert into text values(11,'ADVANCED JAVA','Pearson','Herbert');
 insert into text values(12,'OPERATING SYSTEM','Tata','Peter');
 insert          into          text          values(13,'DATABASE
 SYSTEM','Pearson','Elmasri');
 insert into text values(14,'WEB PROGRAMMING','McGraw','Robert');
  insert          into          text          values(15,'COMPUTER
 NETWORKS','Pearson','Randy');
  select * from text;
 +        +              +         +       +
 | book_isbn | book_title      | publisher | author |
 +        +              +         +       +
 |      11 | ADVANCED JAVA    | Pearson | Herbert |
 |      12 | OPERATING SYSTEM | Tata      | Peter   |
 |      13 | DATABASE SYSTEM | Pearson | Elmasri |
 |      14 | WEB PROGRAMMING | McGraw      | Robert |
 |      15 | COMPUTER NETWORKS | Pearson | Randy |
 +        +              +         +
          +
 insert into book_adoption values(1,3,11);
 insert          into          book_adoption
 values(2,4,11);          insert          into
 book_adoption   values(3,5,12);   insert
 into   book_adoption    values(4,6,13);
 insert          into          book_adoption
 values(1,3,12);          insert          into
 book_adoption values(1,3,13);
 select * from book_adoption;
 +        +     +         +
 | courseno | sem | book_isbn |
 +        +     +         +
  |   1 |      3 |    11 |
  |   2 |      4 |    11 |
  |   3 |      5 |    12 |
  |   4 |      6 |    13 |
  |   1 |      3 |    12 |
  |   1 |      3 |    13 |
 +        +     +         +
```

a) insert into text values(16,'MANAGEMENT','Pearson','Ramez'); insert into book_adoption values(6,5,16);

```
select * from text;
+          +                +          +        +
| book_isbn | book_title       | publisher | author |
+          +                +          +        +
|        11 | ADVANCED JAVA   | Pearson | Herbert |
|        12 | OPERATING SYSTEM | Tata      | Peter   |
|        13 | DATABASE SYSTEM | Pearson | Elmasri |
|        14 | WEB PROGRAMMING | McGraw      | Robert |
|        15 | COMPUTER NETWORKS | Pearson | Randy |
|        16 | MANAGEMENT       | Pearson | Ramez |
+          +                +          +        +

select * from book_adoption;
+         +     +          +
| courseno | sem | book_isbn |
+         +     +          +
|    1 |       3 |    11 |
|    2 |       4 |    11 |
|    3 |       5 |    12 |
|    4 |       6 |    13 |
|    1 |       3 |    12 |
|    1 |       3 |    13 |
|    6 |       5 |    16 |
```

b)

```
select
c.courseno,t.book_title,b.book_isbn
from course c,book_adoption b,text t
where    c.courseno=b.courseno    and    b.book_isbn=t.book_isbn    and
c.dept='CSE' and b.courseno IN(select b.courseno
                    from book_adoption b,course
                    c                       where
                    b.courseno=c.courseno
                    GROUP    BY    b.courseno
                    HAVING          count(*)>2)
                    ORDER BY t.book_title;
```

```
+         +                +          +
| courseno | book_title       | book_isbn |
+         +                +          +
|       1 | ADVANCED JAVA  |       11 |
|       1 | DATABASE SYSTEM |       13 |
|       1 | OPERATING SYSTEM |       12 |
+      +                           +  +
```

c)

```
select c.dept,t.publisher
from course c,book_adoption b,text t
where c.courseno=b.courseno and t.book_isbn=b.book_isbn and publisher='Pearson';
```

```
+      +          +
| dept | publisher |
+      +          +
| CSE  | Pearson   |
| CSE  | Pearson   |
| ISE | Pearson |
| CSE  | Pearson   |
| Auto | Pearson |
+      +          +
```

**5. Consider an order processing database application in a company. Create the tables by properly specifying the primary keys and the foreign keys. Enter at least five tuples for each relation.**

- **a) Produce a listing: CUSTNAME, NO_OF_ORDERS, and AVG_ORDER_AMT, where the middle column is the total number of orders by the customer and the last column is the average order amount for that customer.**
- **b) List the Order# for the orders that were shipped from all the warehouses that the company has in a specific city.**
- **c) Demonstrate how you delete a customer from the CUSTOMER table and make that field null in the ORDER table.**

Solution:

```
create database COMPANY1;
use COMPANY1;
create table customer (cno int primary key,cname varchar(10),city varchar(15));
create table orders(ono int primary key,odate date,cno int ,oamt int,foreign key(cno) references
customer(cno));
create table item(ino int primary key,price int);
create table order_item(ono int,ino int,qty int,foreign key(ono) references orders(ono),foreign
key(ino) references item(ino));
create table warehouse(wno int primary key,city varchar(15));
create table shipment(ono int,wno int,sdate date not null,foreign key(ono) references
orders(ono),foreign key(wno) references warehouse(wno));
```

```
insert into customer values(1,'Rachana','Raipur');
insert into customer
values(2,'Rachitha','Chikmagaluru'); insert into
customer values(3,'Rakshitha','Bellary'); insert into
customer values(4,'Rajatha','Bengaluru'); insert into
customer values(5,'Ranjitha','Mysuru');
```

```
select * from customer;
+    +           +              +
| cno | cname     | city         |
+    +           +              +
| 1 | Rachana | Raipur          |
| 2 | Rachitha | Chikmagaluru |
| 3 | Rakshitha | Bellary       |
| 4 | Rajatha   | Bengaluru     |
| 5 | Ranjitha | Mysuru         |
+    +           +              +
```

```
insert into orders values(11,'2020-12-
12',1,20000); insert into orders values(12,'2020-
12-12',1,25000); insert into orders
values(13,'2021-02-12',1,25000); insert into
orders values(14,'2021-05-16',2,35000); insert
into orders values(15,'2021-05-16',3,35000);
insert into orders values(16,'2021-05-
23',3,45000);
 insert into orders values(17,'2021-11-23',4,45000);
select * from orders;
+     +            +     +      +
| ono | odate      | cno | oamt |
+     +            +     +      +
| 11 | 2020-12-12 |    1 | 20000 |
| 12 | 2020-12-12 |    1 | 25000 |
| 13 | 2021-02-12 |    1 | 25000 |
| 14 | 2021-05-16 |    2 | 35000 |
| 15 | 2021-05-16 |    3 | 35000 |
| 16 | 2021-05-23 |    3 | 45000 |
| 17 | 2021-11-23 |    4 | 45000 |
+     +            +   +   +
 insert into item
values(21,500); insert into
item values(22,500); insert
into item values(23,200);
insert into item
values(24,400); insert into
item values(25,450);
select * from item;
+     +       +
| ino | price |
+     +       +
| 21 | 500 |
| 22 | 500 |
| 23 | 200 |
| 24 | 400 |
| 25 | 450 |
+     +       +
insert into order_item
values(11,21,40); insert into
order_item values(11,21,45); insert
into order_item values(12,22,35);
insert into order_item
values(13,23,45); insert into
order_item values(14,24,55); insert
```

```
  into     order_item     values(15,25,65);
  insert              into             order_item
  values(15,25,75);
  select * from order_item;
  +       +       +       +
  | ono | ino | qty |
  +       +       +       +
|      11 |  21 |   40 |
|      11 |  21 |   45 |
|      12 |  22 |   35 |
|      13 |  23 |   45 |
|      14 |  24 |   55 |
|      15 |  25 |   65 |
|      15 |  25 |   75 |
  +       +       +       +
```

```
insert              into              warehouse
values(31,'Hassan');        insert         into
warehouse values(32,'Haveri'); insert into
warehouse values(33,'Hubli');
 insert             into              warehouse
values(34,'Hirisave');       insert        into
warehouse  values(35,'Hassan');  select  *
from warehouse;
+      +            +
  | wno | city      |
  +      +            +
  | 31 | Hassan   |
  | 32 | Haveri   |
  | 33 | Hubli      |
  | 34 | Hirisave |
  | 35 | Hassan   |
  +      +            +
```

```
insert into shipment values(11,31,'2021-11-02');
insert into shipment values(11,31,'2020-11-02');
insert into shipment values(12,32,'2019-10-12');
insert into shipment values(13,32,'2019-10-12');
insert into shipment values(14,33,'2019-12-05');
insert into shipment values(15,34,'2017-04-05');
select * from shipment;
  +       +       +              +
  | ono | wno | sdate         |
 +      +      +            +
 | 11 | 31 | 2021-11-02 |
 | 11 | 31 | 2020-11-02 |
|     12     32 | 2019-10-12 |
|     13     32 | 2019-10-12 |
|     14     33 | 2019-12-05 |
|     15     34 | 2017-04-05 |
 +      +      +            +
```

```
1.select      c.cno,      count(*),avg
  (o.oamt)  from  customer  c,orders
  o
 where
c.cno=o.cno
group by c.cno;
+      +          +              +
| cno | count(*) | avg (o.oamt) |
+      +          +              +
| 1 |         3 | 23333.3333 |
| 2 |         1 | 35000.0000 |
| 3 |         2 | 40000.0000 |
| 4 |         1 | 45000.0000 |
+      +          +              +
```

2) create view view1 as(select avg(i.price*o.qty) as avg_item_price,sum(o.qty) as no_of_items fromitem i,order_item o where o.ino=i.ino group by o.ono);

```
select * from view1;
+                +             +
| avg_item_price | no_of_items |
+                +             +
|      21250.0000 |       85 |
|      17500.0000 |       35 |
|       9000.0000 |      45 |
|      22000.0000 |       55 |
|      31500.0000 |       140 |
+                +             +
```

**6. Design a relational database for a real estate agency. It should store information about houses for sale, seller information, Buyer information, agents information (who can act on behalf of either the buyer or the seller), and the sale of houses.**

**Answer the following queries in SQL.**

**1) What are the ids, addresses, asking_price, and selling_price of all houses that sold for less than the asking price?**

**2) What are names of all of (prospective) buyers who have not bought a house? Each name should appear only once.**

**3) Find the addresses and asking prices of all houses that have at least 3 bedrooms and two bathrooms that have not sold. Each address, asking price pair should appear only once.**

```
CREATE TABLE HOUSE(HID INT PRIMARY KEY,BEDROOM INT,BATHROOM
INT,ADDRESS CHAR(20),POSTALCODE INT,PRICE INT);
CREATE TABLE SELLERINFO(SID INT PRIMARY KEY,SNAME VARCHAR(20),HID
INT,ADDRESS VARCHAR(20),FOREIGN KEY(HID) REFERENCES HOUSE(HID) ON
DELETE CASCADE);
CREATE TABLE BUYERINFO(BID INT PRIMARY KEY,BNAME VARCHAR(20),ADDRESS
VARCHAR(20),HID INT,FOREIGN KEY(HID) REFERENCES HOUSE(HID));
CREATE TABLE AGENTINFO(AID INT PRIMARY KEY,ANAME VARCHAR(20));
CREATE TABLE SALESINFO(SALENO INT PRIMARY KEY,HID INT,SID INT,BID INT,AID
INT,SELPRICE INT, FOREIGN KEY(HID) REFERENCES HOUSE(HID),FOREIGN KEY(SID)
REFERENCES SELLERINFO(SID), FOREIGN KEY (BID) REFERENCES
BUYERINFO(BID),FOREIGN KEY(AID) REFERENCES AGENTINFO(AID));
INSERT INTO HOUSE VALUES(101,4,3,'HASSAN',573202,500000);
INSERT INTO HOUSE VALUES(102,4,4,'BENGALURU',560001,600000);
INSERT INTO HOUSE VALUES(103,3,1,'BENGALURU',560001,400000);
INSERT INTO HOUSE VALUES(104,2,1,'HASSAN',573201,350000);
INSERT INTO HOUSE VALUES(105,5,5,'MYSURU',571602,450000);
SELECT * FROM HOUSE;
+-----+---------+----------+-----------+------------+--------+
| HID | BEDROOM | BATHROOM | ADDRESS | POSTALCODE | PRICE |
+-----+---------+----------+-----------+------------+--------+
| 101 | 4 | 3 | HASSAN | 573202 | 500000 |
| 102 | 4 | 4 | BENGALURU | 560001 | 600000 |
| 103 | 3 | 1 | BENGALURU | 560001 | 400000 |
| 104 | 2 | 1 | HASSAN | 573201 | 350000 |
| 105 | 5 | 5 | MYSURU | 571602 | 450000 |
+-----+---------+----------+-----------+------------+--------+
INSERT INTO SELLERINFO VALUES(201,'ARPITHA',101,'HASSAN');
INSERT INTO SELLERINFO VALUES(202,'ARYAN',102,'HASSAN');
INSERT INTO SELLERINFO VALUES(203,'ANJAN',103,'MYSURU');
INSERT INTO SELLERINFO VALUES(204,'ARJUN',104,'MYSURU');
INSERT INTO SELLERINFO VALUES(205,'ANJALI',105,'BENGALURU');
SELECT * FROM SELLERINFO;
```

```
+-----+---------+------+-----------+
| SID | SNAME | HID | ADDRESS |
+-----+---------+------+-----------+
| 201 | ARPITHA | 101 | HASSAN |
| 202 | ARYAN | 102 | HASSAN |
| 203 | ANJAN | 103 | MYSURU |

| 204 | ARJUN | 104 | MYSURU |
| 205 | ANJALI | 105 | BENGALURU |
+-----+---------+------+-----------+
INSERT INTO BUYERINFO VALUES(301,'BHARATH','BENGALURU',105);
INSERT INTO BUYERINFO VALUES(302,'BHARATHI','BELAGAVI',104);
INSERT INTO BUYERINFO VALUES(303,'BHAVANA','HASSAN',103);
INSERT INTO BUYERINFO VALUES(304,'BHARANI','BELUR',102); INSERT
INTO BUYERINFO VALUES(305,'BHAVESH','HASSAN',101);
SELECT * FROM BUYERINFO;
+-----+----------+-----------+------+
| BID | BNAME | ADDRESS | HID |
+-----+----------+-----------+------+
| 301 | BHARATH | BENGALURU | 105 |
| 302 | BHARATHI | BELAGAVI | 104 |
| 303 | BHAVANA | HASSAN | 103 |
| 304 | BHARANI | BELUR | 102 |
| 305 | BHAVESH | HASSAN | 101 |
+-----+----------+-----------+------+
INSERT INTO AGENTINFO VALUES (401,'CHIRANTH');
INSERT INTO AGENTINFO VALUES (402,'DARSHAN');
INSERT INTO AGENTINFO VALUES (403,'RAKESH');
INSERT INTO AGENTINFO VALUES (404,'RAMYA');
INSERT INTO AGENTINFO VALUES (405,'VIKAS');
SELECT* FROM AGENTINFO;
+-----+----------+
| AID | ANAME |
+-----+----------+
| 401 | CHIRANTH |
| 402 | DARSHAN |
| 403 | RAKESH |
| 404 | RAMYA |
| 405 | VIKAS |
+-----+----------+
INSERT INTO SALESINFO VALUES(501,101,201,301,401,40000);
INSERT INTO SALESINFO VALUES(502,102,202,302,402,350000);
INSERT INTO SALESINFO VALUES(504,104,204,303,404,350000);
INSERT INTO SALESINFO VALUES(505,103,203,305,405,450000);
```

```
SELECT* FROM SALESINFO;
+--------+------+------+------+------+----------+
| SALENO | HID | SID | BID | AID | SELPRICE |
+--------+------+------+------+------+----------+
| 501 | 101 | 201 | 301 | 401 | 40000 |
| 502 | 102 | 202 | 302 | 402 | 350000 |
| 504 | 104 | 204 | 303 | 404 | 350000 |
| 505 | 103 | 203 | 305 | 405 | 450000 |
+--------+------+------+------+------+----------+
```

```
1) SELECT H.HID,H.ADDRESS,H.PRICE,S.SELPRICE FROM HOUSE H,SALESINFO
SWHERE H.HID=S.HID AND S.SELPRICE<H.PRICE;
+-----+-----------+--------+----------+
| HID | ADDRESS | PRICE | SELPRICE |
+-----+-----------+--------+----------+
| 101 | HASSAN | 500000 | 40000 |
| 102 | BENGALURU | 600000 | 350000 |
+-----+-----------+--------+----------+
```

```
2) SELECT DISTINCT B.BNAME FROM BUYERINFO B, SELLERINFO S
-> WHERE B.BID NOT IN(SELECT S.BID FROM SALESINFO S);
+---------+

| BNAME |
+---------+
| BHARANI |
+---------+
```

```
3) SELECT DISTINCT H.HID,H.ADDRESS,H.PRICE FROM HOUSE H,SALESINFO S
-> WHERE (H.BEDROOM>=3 AND H.BATHROOM>=2) AND H.HID NOT IN(SELECT
S.HID FROM SALESINFO S);
+-----+---------+--------+
| HID | ADDRESS | PRICE |
+-----+---------+--------+
| 105 | MYSURU | 450000 |
+-----+---------+--------+
```

**7. Consider the details maintained by a book dealer. Create the tables by properly specifying the primary keys and the foreign keys. Enter at least five tuples for each relation.**

**a) 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 .**

**b) Find the author of the book which has maximum sales.**

**c) Demonstrate how you increase the price of books published by a specific publisher by 10%.**

```
create table Author(authorid int primary key,name varchar(15),city char(15),country char(15));
create table Publisher(publisherid int primary key,pname char(10),city char(15),country char(15));
create table Catalog1(book_id int primary key,title char(15),authorid int,publisherid int,year int,price
int,foreign key (authorid) references Author(authorid),foreign key (publisherid)
references Publisher(publisherid));
create table Order_details1(orderno int primary key,book_id int,quantity int,foreign key(book_id)

references Catalog1(book_id));
insert into Author values(111,'Ramesh','Delhi','India');
insert into Author values(112,'Suresh','Goa','India');
insert into Author values(113,'John','Sydney','Australia');
insert into Author values(114,'Jenny','Texas','America');
insert into Author values(115,'Ramesh','Delhi','India');
select * from Author;
+----------+--------+--------+-----------+
| authorid | name | city | country |
+----------+--------+--------+-----------+
| 111 | Ramesh | Delhi | India |
| 112 | Suresh | Goa | India |
| 113 | John | Sydney | Australia |
| 114 | Jenny | Texas | America |
| 115 | Ramesh | Delhi | India |
+----------+--------+--------+-----------+
insert into Publisher values(11,'Smrithi','Hassan','India');
insert into Publisher values(12,'Pearson','Kerala','India');
insert into Publisher values(13,'IPE','Hyderabad','India');
insert into Publisher values(14,'RCB','Bengaluru','India');
insert into Publisher values(15,'CSK','Chennai','India');
select * from Publisher;
+-------------+---------+-----------+---------+
| publisherid | pname | city | country |
+-------------+---------+-----------+---------+
| 11 | Smrithi | Hassan | India |
| 12 | Pearson | Kerala | India |
| 13 | IPE | Hyderabad | India |
| 14 | RCB | Bengaluru | India |
| 15 | CSK | Chennai | India |
+-------------+---------+-----------+---------+
insert into Catalog1 values(1,'Comedy day',111,11,101,2008,150);
insert into Catalog1 values(2,'Horror Night',111,12,102,2009,250);
insert into Catalog1 values(3,'clash of titans',112,13,103,2010,250);
insert into Catalog1 values(4,'Innovator',114,14,105,2011,450);

insert into Catalog1 values(5,'ABCD',111,11,101,2011,2000);
insert into Catalog1 values(6,'DBMS',111,11,101,2011,2000);
select * from Catalog1;
+---------+-----------------+----------+-------------+------------+------+-------+
| book_id | title | authorid | publisherid | categoryid | year | price |
+---------+-----------------+----------+-------------+------------+------+-------+
| 1 | Comedy day | 111 | 11 | 101 | 2008 | 150 |
| 2 | Horror Night | 111 | 12 | 102 | 2009 | 250 |
| 3 | clash of titans | 112 | 13 | 103 | 2010 | 250 |
| 4 | Innovator | 114 | 14 | 105 | 2011 | 450 |
| 5 | ABCD | 111 | 11 | 101 | 2011 | 2000 |
| 6 | DBMS | 111 | 11 | 101 | 2011 | 2000 |
+---------+-----------------+----------+-------------+------------+------+-------+
insert into Order_details1 values(121,1,100);
insert into Order_details1 values(122,2,100);
insert into Order_details1 values(123,3,100);
insert into Order_details1 values(124,4,150);
insert into Order_details1 values(125,5,250);
select * from Order_details1;
+---------+---------+----------+
| orderno | book_id | quantity |
+---------+---------+----------+
| 121 | 1 | 100 |
| 122 | 2 | 100 |
| 123 | 3 | 100 |
| 124 | 4 | 150 |
| 125 | 5 | 250 |
+---------+---------+----------+
```

1. select * from Author a where a.authorid IN (select c.authorid from Catalog1 c where year>2000 and c.price>(select AVG(price) from Catalog1) group by c.authorid having count(authorid)>=2);

```
+----------+--------+-------+---------+
| authorid | name | city | country |
+----------+--------+-------+---------+
| 111 | Ramesh | Delhi | India |
+----------+--------+-------+---------+
```

2. select name from Author a,Catalog1 c where a.authorid=c.authorid and book_id IN(select book_id

from Order_details1 where quantity =(select MAX(quantity) from Order_details1));

```
+--------+
| name |
+--------+
| Ramesh |
+--------+
```

3. update Catalog1 set price=price+ price*0.10 where publisherid IN(select p.publisherid from Publisher p where p.pname='RCB');
select * from Catalog1;

```
+---------+-----------------+----------+-------------+------------+------+-------+
| book_id | title | authorid | publisherid | categoryid | year | price |
+---------+-----------------+----------+-------------+------------+------+-------+
| 1 | Comedy day | 111 | 11 | 101 | 2008 | 150 |
| 2 | Horror Night | 111 | 12 | 102 | 2009 | 250 |
| 3 | clash of titans | 112 | 13 | 103 | 2010 | 250 |
| 4 | Innovator | 114 | 14 | 105 | 2011 | 495 |
| 5 | ABCD | 111 | 11 | 101 | 2011 | 2000 |
| 6 | DBMS | 111 | 11 | 101 | 2011 | 2000 |
```

**8. Consider the following schema:**

**SUPPLIERS (Sid: integer, sname:string, address: string)**
**PARTS (PID: integer, pname: string, color:string)**
**CATALOG (Sid: integer, PID: integer, cost:real)**
**Design a database to satisfy the above requirements and answer the following queries:**

**a. Find the names of parts for which there is some supplier.**
**b. Find the names of suppliers who supply every part.**
**c. Find the id’s of suppliers who supply only red parts.**

```
CREATE TABLE SUPPLIERS(SID INT PRIMARY KEY,SNAME VARCHAR(20),ADDRESS
CHAR(20));
CREATE TABLE PARTS(PID INT PRIMARY KEY,PNAME VARCHAR(20),COLOR
VARCHAR(10));
CREATE TABLE CATALOG(SID INT,PID INT,COST FLOAT(15),FOREIGN KEY(SID)
REFERENCES SUPPLIERS(SID),FOREIGN KEY(PID) REFERENCES PARTS(PID));
INSERT INTO SUPPLIERS VALUES(1,&#39;SANDEEP&#39;,&#39;MYSURU&#39;);
INSERT INTO SUPPLIERS VALUES(2,&#39;MADHU&#39;,&#39;MYSURU&#39;);
INSERT INTO SUPPLIERS VALUES(3,&#39;&#39;,&#39;MYSURU&#39;);
INSERT INTO SUPPLIERS VALUES(4,&#39;MOHAN&#39;,&#39;MANDYA&#39;);
INSERT INTO SUPPLIERS VALUES(5,&#39;CHETHAN&#39;,&#39;MANDYA&#39;);
SELECT * FROM SUPPLIERS;
+-----+---------+ +
| SID | SNAME | ADDRESS |
+-----+---------+ +
| 1 | SANDEEP | MYSURU |
| 2 | MADHU | MYSURU |
| 3 | | MYSURU |
| 4 | MOHAN | MANDYA |
| 5 | CHETHAN | MANDYA |
+-----+---------+ +
INSERT INTO PARTS VALUES(10,&#39;KEYBOARD&#39;,&#39;RED&#39;);
INSERT INTO PARTSVALUES(20,&#39;KEYBOARD&#39;,&#39;BLACK&#39;);
INSERT INTO PARTS VALUES(30,&#39;MOUSE&#39;,&#39;BLACK&#39;);
INSERT INTO PARTS VALUES(40,&#39;MOUSE&#39;,&#39;RED&#39;);
SELECT * FROM PARTS;

+-----+----------+ +
| PID | PNAME | COLOR |
+-----+----------+ +
| 10 | KEYBOARD | RED |
| 20 | KEYBOARD | BLACK |
| 30 | MOUSE | BLACK |
| 40 | MOUSE | RED |
+-----+----------+ +

INSERT INTO CATALOGVALUES(1,10,450.56);
INSERT INTO CATALOGVALUES(1,20,350.56);
INSERT INTO CATALOG VALUES(1,30,250.56);
INSERT INTO CATALOG VALUES(1,40,250);
INSERT INTO CATALOG VALUES(2,40,250);
INSERT INTO CATALOG VALUES(3,20,250);
INSERT INTO CATALOG VALUES(3,10,250);
SELECT * FROM CATALOG;
+------+------+ +
| SID | PID | COST |
+------+------+ +
| 1 | 10 | 450.56 |
| 1 | 20 | 350.56 |
| 1 | 30 | 250.56 |
| 1 | 40 | 250 |
| 2 | 40 | 250 |
| 3 | 20 | 250 |
| 3 | 10 | 250 |
+------+------+ +
a) SELECT P.PNAME FROM PARTS P,CATALOG C WHERE P.PID=C.PID;
+ +
| PNAME |
+ +
| KEYBOARD |
| KEYBOARD |
| KEYBOARD |
| KEYBOARD |
| MOUSE |
| MOUSE |
| MOUSE |
+ +
b) SELECT SNAME FROM SUPPLIERS WHERE SID IN (SELECT C.SID FROM CATALOG C
WHERE NOT EXISTS(SELECT P.
PID FROM PARTS P WHERE NOT EXISTS(SELECT C1.SID FROM CATALOG C1 WHERE
C1.SID=C.SID AND C1.PID=P.PID)));
(OR)
SELECT S.SNAME

FROM SUPPLIERS S
WHERE NOT EXISTS(
SELECT *
FROM PARTS P
WHERE NOT EXISTS (
SELECT *
FROM CATALOG C
WHERE C.SID=S.SID AND C.PID=P.PID
)
);
+ +
| SNAME |
+ +
| SANDEEP |
+ +
```

```
c) SELECT C.SID FROM CATALOG C,PARTS P WHERE P.COLOR=&#39;RED&#39; AND
C.PID=P.PID;
+ +
| SID | nbv2
+ +
| 1 |
| 3 |
| 1 |
| 2 |
+
```