

Question-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.
43

- 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.

Create the respective tables

```
create table student(snum int(20),sname varchar(20),major varchar(20),level varchar(3),age int(3),primary key(snum));
```

```
create table faculty(fid int,fname varchar(20),deptid int(4),primary key(fid));
```

```
create table class(name varchar(20),meetsat varchar(20),room varchar(3),fid int,primary key(name),foreign key(fid)references faculty(fid));
```

```
create table enrolled(snum int,cname varchar(20),primary key(snum,cname),foreign key(snum) references student(snum),foreign key(cname) references class(name));
```

Queries

- 1) SELECT DISTINCT S.SNAME FROM STUDENT S,CLASS C, ENROLLED E,FACULTY F WHERE S.SNUM=E.SNUM AND E.CNAME = C.NAME AND C.FID = F.FID AND F.FNAME='Prof.Harshith' AND S.LEVEL = 'JR';
- 2) SELECT C.NAME FROM CLASS C WHERE C.ROOM = 'R128' OR C.NAME IN (SELECT E.CNAME FROM ENROLLED E GROUP BY E.CNAME HAVING COUNT(*)>=5);
- 3) SELECT C.CNO,T.ISBN,T.TITLE FROM COURSE C,ADOPT A,TEXT T WHERE C.CNO=A.CNO AND C.DEPT='CSE' AND A.ISBN=T.ISBN AND EXISTS(SELECT COUNT(A1.CNO) FROM ADOPT A1 WHERE A1.CNO=C.CNO GROUP BY A1.CNO HAVING COUNT(*)>2) ORDER BY C.CNO,T.ISBN,T.TITLE;
- 4) SELECT FNAME FROM FACULTY F,CLASS C WHERE F.FID=C.FID GROUP BY C.FID HAVING COUNT(DISTINCT ROOM) = (SELECT COUNT(DISTINCT ROOM) FROM CLASS);
- 5) SELECT DISTINCT F.FNAME FROM FACULTY F WHERE 5 > (SELECT COUNT(E.SNUM) FROM CLASS C, ENROLLED E WHERE C.NAME = E.CNAME AND C.FID = F.FID);

Question-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.

Create the respective tables

```
create table flight( fno int,ffrom varchar(20),fto varchar(20),distance int(11),departs time,arrives time,price float , primary key(fno));
```

```
desc flight;
```

```
create table aircraft(aid int,aname varchar(20),crange int,primary key(aid));
```

```
desc aircraft;
```

```
create table employees(eid int, ename varchar(20),salary int, primary key(eid));
```

```
desc employees;
```

```
create table certified(eid int,aid int,primary key(eid,aid) , foreign key(eid) references  
employees(eid),foreign key(aid) references aircraft(aid));
```

```
desc certified;
```

Queries

- 1) INSERT INTO EMPLOYEES VALUES(4214,'AJITH',85000);
INSERT INTO EMPLOYEES VALUES(4215,'JUJARE',35000);
INSERT INTO EMPLOYEES VALUES(4216,'KUMAR',95000);
INSERT INTO AIRCRAFT VALUES(5604,"GOING",2600);
INSERT INTO AIRCRAFT VALUES(5605,"BOEING",2150);
INSERT INTO AIRCRAFT VALUES(5606,"BOEING DAUNTLESS",2500);
INSERT INTO CERTIFIED VALUES(4214,5604);
INSERT INTO CERTIFIED VALUES(4215,5605);
INSERT INTO CERTIFIED VALUES(4216,5606);

SELECT ANAME FROM AIRCRAFT A, EMPLOYEES E, CERTIFIED C WHERE
C.EID=E.EID AND C.AID=A.AID AND E.SALARY>80000;
- 2) INSERT INTO AIRCRAFT VALUES(5608,"MAYING",2300);
INSERT INTO CERTIFIED VALUES(4214,5605);
INSERT INTO CERTIFIED VALUES(4214,5608);
INSERT INTO CERTIFIED VALUES(4214,5606);
INSERT INTO AIRCRAFT VALUES(9801,'PAYING',3000);

```
INSERT INTO CERTIFIED VALUES(4215,9801);
INSERT INTO CERTIFIED VALUES(4214,9801);
```

```
SELECT E.EID, MAX(A.CRANGE) FROM EMPLOYEES E,CERTIFIED
C,AIRCRAFT A WHERE E.EID=C.EID AND C.AID=A.AID GROUP BY C.EID
HAVING COUNT(*)>3;
```

```
3) INSERT INTO FLIGHT VALUES
(8741,'BANGALORE','FRANKFURT',1500,'08:30:00','10:30:00',55000);
```

```
SELECT DISTINCT E.ENAME FROM EMPLOYEES E,CERTIFIED C WHERE
E.EID=C.EID AND E.SALARY<(SELECT MIN(PRICE) FROM FLIGHT F WHERE
F.FFROM='BANGALORE' AND F.FTO='FRANKFURT');
```

```
4) SELECT A.ANAME,AVG(E.SALARY) FROM AIRCRAFT A,EMPLOYEES
E,CERTIFIED C WHERE CRANGE>1000 AND C.EID=E.EID AND A.AID=C.AID
GROUP BY A.ANAME;
```

```
5) SELECT ENAME FROM AIRCRAFT A,CERTIFIED C,EMPLOYEES E WHERE
A.AID=C.AID AND C.EID=E.EID AND A.ANAME LIKE '%BOEING%';
```

```
6) INSERT INTO FLIGHT VALUES(8125,'BANGALORE','NEW
DELHI',5000,"08:30:00","11:30:00",15000);
```

```
INSERT INTO FLIGHT VALUES(8126,'BANGALORE','NEW
DELHI',5000,"09:30:00","12:30:00",16000);
```

```
INSERT INTO FLIGHT VALUES(8127,'BANGALORE','NEW
DELHI',5000,"10:30:00","13:30:00",17000);
```

```
INSERT INTO FLIGHT VALUES(8128,'BANGALORE','NEW
DELHI',2900,"10:30:00","13:30:00",17000);
```

```
SELECT A.AID FROM AIRCRAFT A WHERE A.CRANGE>=(SELECT  
MIN(F.DISTANCE) FROM FLIGHT F WHERE FFROM='BANGALORE' AND  
FTO='NEW DELHI');
```

www.vtuCS.com

Question-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)

44

- 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.

Create the respective tables

```
create table student ( regno varchar(20), name varchar(20), major varchar(10), bdate date,
primary key(regno) );
```

```
create table course ( cno int, cname varchar(20), dept varchar(20), primary key(cno) );
```

```
create table enroll ( regno varchar(20), cno int, marks int, sem int, primary key(sem,regno,cno),
foreign key(regno) references student(regno), foreign key(cno) references course(cno) );
```

create table text(isbn int, title varchar(30), publisher varchar(20), author varchar(20), primary key(isbn));

create table adopt(cno int, sem int, isbn int, foreign key(cno) references course(cno), foreign key(isbn) references text(isbn) , foreign key(sem) references enroll(sem));

Queries

- 1) INSERT INTO TEXT VALUES(3042,'NEW','PEARSON','AJITH');
INSERT INTO ADOPT VALUES(1009,7,3042);

 - 2) INSERT INTO TEXT VALUES(1523,"T1","BILL","AJITH");
INSERT INTO TEXT VALUES(1524,"T2","WILL","JUJARE");
INSERT INTO TEXT VALUES(1525,"T3","KILL","JOSEPH");
INSERT INTO TEXT VALUES(1526,"T4","MILL","KUMAR");
INSERT INTO COURSE VALUES(4221,"KING","CSE");
INSERT INTO COURSE VALUES(4226,"BING","ISE");
INSERT INTO COURSE VALUES(4229,"PING","ECE");
INSERT INTO STUDENT VALUES("1PE11CS010","AJITH","CS","02-02-1994");
INSERT INTO ENROLL VALUES("1PE11CS010",4221,100,5);
INSERT INTO ADOPT VALUES(4221,5,1523);
INSERT INTO ADOPT VALUES(4226,5,1524);
INSERT INTO COURSE VALUES(4214,"KING","CSE");
INSERT INTO ADOPT VALUES(4214,5,1526);
INSERT INTO ADOPT VALUES(4214,5,1525);
- SELECT C.CNO,T.ISBN,T.TITLE FROM COURSE C,ADOPT A,TEXT T WHERE C.CNO=A.CNO AND C.DEPT='CSE' AND A.ISBN=T.ISBN AND EXISTS(SELECT COUNT(A1.CNO) FROM ADOPT A1 WHERE A1.CNO=C.CNO GROUP BY A1.CNO HAVING COUNT(*)>2) ORDER BY C.CNO,T.ISBN,T.TITLE;

3) INSERT INTO TEXT VALUES(1292,"T9","BILL","AJITH");
INSERT INTO ADOPT VALUES(4214,5,1292);

SELECT DISTINCT C.DEPT FROM COURSE C,ADOPT A,TEXT T WHERE
C.CNO=A.CNO AND T.ISBN=A.ISBN AND T.PUBLISHER='BILL' GROUP BY
DEPT HAVING COUNT(A.ISBN)=(SELECT COUNT(*) FROM TEXT WHERE
PUBLISHER='BILL');

www.vtuCS.com

Question-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.

Create the respective tables

```
create table author ( aid int, aname varchar(20), acity varchar(20), acountry varchar(20), primary key(aid) );
```

```
create table publisher ( pid int, pname varchar(20), pcity varchar(20), pcountry varchar(20), primary key(pid) );
```

```
create table category (cid int, description varchar(30), primary key(cid) );
```

create table catalog(bid int, btitle varchar(30), aid int, pid int, cid int, year int, price int, primary key(bid), foreign key(aid) references author(aid), foreign key(pid) references publisher(pid), foreign key(cid) references category(cid));

create table orderdetails(oid int, bid int, quantity int, primary key(oid), foreign key(bid) references catalog(bid));

Queries

- 1) INSERT INTO AUTHOR VALUES(4204,'AJITH','BANGALORE','INDIA');
INSERT INTO AUTHOR VALUES(4206,'JUJARE','BANGALORE','INDIA');
INSERT INTO PUBLISHER VALUES(8404,'GREAT MAN','BANGALORE','INDIA');
INSERT INTO CATEGORY VALUES(1204,'THRILLER');
INSERT INTO CATALOG VALUES(1201,'MAN ON MOUNTAINS',4204,8404,1204,2005,3000);
INSERT INTO CATALOG VALUES(1202,'MAN ON HEIGHTS',4204,8404,1204,2006,2000);
INSERT INTO CATALOG VALUES(1203,'MAN ON GREATS',4204,8404,1204,2006,2000);
INSERT INTO CATALOG VALUES(1204,'MAN ON FLIES',4206,8404,1204,2005,30);

SELECT A.ANAME FROM AUTHOR A,CATALOG C WHERE A.AID=C.AID AND C.YEAR>2000 AND C.PRICE>(SELECT AVG(PRICE) FROM CATALOG) GROUP BY C.AID HAVING COUNT(*)>=2;
- 2) INSERT INTO ORDERDETAILS VALUES(0112,1204,15);
INSERT INTO ORDERDETAILS VALUES(0113,1201,3);
INSERT INTO ORDERDETAILS VALUES(0114,1202,4);

```
SELECT A.ANAME FROM AUTHOR A,ORDERDETAILS O,CATALOG C WHERE  
A.AID=C.AID AND C.BID=O.BID AND O.QUANTITY =(SELECT  
MAX(O.QUANTITY) FROM ORDERDETAILS O) ;
```

3) SELECT C.BID,C.PRICE*1.1 AS NEWPRICE FROM PUBLISHER P,CATALOG C
WHERE P.PID=C.PID AND P.PNAME='GREAT MAN';

```
insert into author values(420,"AJITH","bang","India");  
insert into author values(840,"Jujare","bang","India");  
insert into publisher values(920,"Wiley","chennai","India");  
insert into publisher values(120,"giley","chennai","India");  
insert into category values(320,"thriller");  
insert into catalog values(311,"Great",420,920,320,2007,3000);  
insert into catalog values(313,"Threat",420,120,320,2010,2000);  
insert into catalog values(349,"phat",840,920,320,2009,4000);  
insert into orderdetails values(711,311,5);  
insert into orderdetails values(782,349,10);  
insert into orderdetails values(342,313,7);
```

Question-4

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)

45

- 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.

Create the respective tables

```
create table branch(bname varchar(20) primary key, bcity varchar(20), assets float(11));
```

```
create table account(accno int(11) primary key, bname varchar(11), balance float, foreign key(bname) references branch(bname) on delete cascade);
```

```
create table customer(cname varchar(11) primary key, cstreet varchar(11), ccity varchar(11));
```

```
create table depositor(cname varchar(11), accno int(11), primary key(cname, accno), foreign key(cname) references customer(cname) on delete cascade, foreign key(accno) references account(accno) on delete cascade);
```

create table loan(loanno int(11) primary key,bname varchar(11) , amount float, foreign
kdasdsadasdey(bname) references branch(bname) on delete cascade);

create table borrower(cname varchar(11), loanno int(11),primary key(cname,loanno) , foreign
key(loanno) references loan(loanno) on delete cascade , foreign key(cname) references
customer(cname) on delete cascade);

```
1) INSERT INTO BRANCH VALUES("PESSE","BANGALORE",50000);
INSERT INTO BRANCH VALUES("JPNAGAR","BANGALORE",51000);
INSERT INTO ACCOUNT VALUES(1014,'PESSE',20000);
INSERT INTO ACCOUNT VALUES(1015,'JPNAGAR',21000);
INSERT INTO ACCOUNT VALUES(1016,'PESSE',20000);
INSERT INTO ACCOUNT VALUES(1017,'JPNAGAR',20000);
INSERT INTO CUSTOMER VALUES("AJITH","3RD STREET","BANGALORE");
INSERT INTO CUSTOMER VALUES("JUJARE","4TH STREET","BANGALORE");
INSERT INTO DEPOSITOR VALUES("AJITH",1014);
INSERT INTO DEPOSITOR VALUES("AJITH",1016);
INSERT INTO DEPOSITOR VALUES("AJITH",1015);
INSERT INTO DEPOSITOR VALUES("JUJARE",1017);
```

```
SELECT D.CNAME FROM DEPOSITOR D,ACCOUNT A WHERE
A.BNAME='PESSE' AND A.ACCNO=D.ACCNO GROUP BY D.CNAME HAVING
COUNT(*)>=2;
```

```
2) INSERT INTO BRANCH VALUES("PESSE","BANGALORE",50000);
INSERT INTO BRANCH VALUES("JPNAGAR","BANGALORE",51000);
```

```
SELECT D.CNAME FROM ACCOUNT A,DEPOSITOR D,BRANCH B WHERE  
D.ACCNO=A.ACCNO AND A.BNAME=B.BNAME AND B.BCITY='BANGALORE'  
GROUP BY D.CNAME HAVING COUNT(DISTINCT B.BNAME)=(SELECT  
COUNT(DISTINCT B.BNAME) FROM BRANCH B WHERE  
B.BCITY='BANGALORE');
```

3) DELETE FROM ACCOUNT WHERE BNAME IN (SELECT B.BNAME FROM
BRANCH B WHERE B.BCITY='BANGALORE');

WWW.VTUCS.COM