TUTORIAL 5

1. In the following tables ADVISOR and TAUGHTBY are foreign keys referring to the table PROFESSOR. ROLLNO and COURSEID in ENROLLMENT refer to tables with primary keys of the same name.

STUDENT (ROLLNO, NAME, AGE, GENDER, ADDRESS, ADVISOR)
COURSE (COURSEID, CNAME, TAUGHTBY, CREDITS)
PROFESSOR (PROFID, PNAME, PHONE)
ENROLLMENT (ROLLNO, COURSEID, GRADE)

Write SQL expressions for the following queries:

(i) Names of courses taught by 'Prof. Raju'.
(ii) Names of students who have *not* enrolled for any course taught by 'Prof.Ganapathy'.
(iii) For each course, name of the course and number of students enrolled for the Course.

Answer

i. SELECT C. COURSEID, C. CNAME FROM COURSE C, PROFESSOR P WHERE C.TAUGHTBY=P.PROFESSORID AND P.PNAME='PROF. RAJU'

ii. SELECT S.NAME FROM STUDENT S WHERE NOT EXISTS (SELECT * FROM ENROLLMENT E, COURSE C WHERE C.TAUGHTBY='GANAPATHY' AND C.COURSEID = E.COURSEID AND E.ROLLNO=S.ROLLNO)

iii. SELECT CNAME ,COUNT(*) FROM COURSE C, ENROLLMENT E WHERE C. COURSED=E.COURSEID GROUP BY COURSE

2. Consider the following relations for bank database (Primary keys are underlined):

Customer (customer-name, customer-street, customer-city)

Branch (branch-name, branch-city, assets)

Account (account-number, branch-name, balance)

Depositor (customer-name, account-number)

Loan (loan-number, branch-name, amount)

Answer the following in SQL:

i) Create tables with primary keys and foreign keys

ii) Create an assertion for the sum of all loan amounts for each branch must be less than the sum of all account balances at the branch.

Answer

```
1. CREATE TABLE Customer (customer-name varchar (15),
customer-street varchar (20), customer-city varchar (30), primary key
(customer-name));
2. CREATE TABLE Branch (branch-name varchar (15), branch-
city varchar (30), assets integer, primary key (branch-name), check
(assets >= 0));
3. CREATE TABLE Account (account-number varchar (20), branch-
name varchar (15) REFERENCES Branch (branch-name), balance
integer, PRIMARY KEY (account-number));
4. CREATE TABLE Depositor (customer-name varchar (15)
REFERENCES Customer (customer-name), account-number varchar
(20) REFERENCES Account (account-number), PRIMARY KEY
(customer-name, account-number));
5. CREATE TABLE Loan (loan-number varchar (100), branch-name
varchar (15) REFERENCES Branch (branch-name), amount integer,
PRIMARY KEY (loan-number));
ii) Create assertion sum-constraint check
    (not exists (select * from Branch
        where (select sum (amount) from Loan
            where Loan.branch-name = Branch.branch-name) >=
                (select sum (balance) from Account
                    where Account.branch-name = Branch.branch-name)))
```