CSCI 4707 Homework2

Hao Wang

wang5167@umn.edu

5086487

B.1.

(1)

a.

CREATE TABLE Customer

(

custid Integer,

PRIMARY KEY (custid)

);

CREATE TABLE Product

(

pid Integer,

PRIMARY KEY (pid)

);

CREATE TABLE Buy

(

custid Integer,

pid Integer,

tid Integer,

PRIMARY KEY (custid,pid),

FOREIGN KEY (custid) REFERENCES Customer,

FOREIGN KEY (pid) REFERENCES Product

);

b.

CREATE TABLE Customer

(

custid Integer,

PRIMARY KEY (custid)

);

CREATE TABLE Product

(

pid Integer,

PRIMARY KEY (pid)

);

CREATE TABLE Transaction

(

tid Integer,

PRIMARY KEY (tid)

);

CREATE TABLE Buy

(

custid Integer,

pid Integer,

tid Integer,

PRIMARY KEY (custid,pid,tid),

FOREIGN KEY (custid) REFERENCES Customer,

FOREIGN KEY (pid) REFERENCES Product,

FOREIGN KEY (tid) REFERENCES Transaction

);

c.

CREATE TABLE Customer

(

custid Integer,

PRIMARY KEY (custid)

);

CREATE TABLE Product

(

pid Integer,

PRIMARY KEY (pid)

);

CREATE TABLE Buy

(

custid Integer,

pid Integer,

tid Integer,

PRIMARY KEY (custid,pid),

FOREIGN KEY (custid) REFERENCES Customer,

FOREIGN KEY (pid) REFERENCES Product

);

d.

CREATE TABLE Product

(

pid Integer,

PRIMARY KEY (pid)

);

CREATE TABLE Customer

(

custid Integer,

pid Integer NOT NULL,

tid Integer,

PRIMARY KEY (custid,tid),

FOREIGN KEY (pid) REFERENCES Product

);

2.

CREATE TABLE Authors

(

name char(30),

PRIMARY KEY (name)

);

CREATE TABLE Publishers

(

name char(30),

PRIMARY KEY (name)

);

CREATE TABLE Categories

(

name char(30),

PRIMARY KEY (name)

);

CREATE TABLE Books

(

ISBN char(30),

name char(30) NOT NULL,

pname char(30) NOT NULL,

PRIMARY KEY (ISBN),

FOREIGN KEY (pname) REFERENCES Pubishers

);

CREATE TABLE Authored

(

ISBN char(30),

aname char(30),

PRIMARY KEY (ISBN,aname),

FOREIGN KEY (ISBN) REFERENCES Books,

FOREIGN KEY (aname) REFERENCES Authors

);

CREATE TABLE Belongs

(

ISBN char(30),

cname char(30),

PRIMARY KEY (ISBN,cname),

FOREIGN KEY (ISBN) REFERENCES Books,

FOREIGN KEY (cname) REFERENCES Categories

);

CREATE TABLE Parent

(

parname char(30),

chiname char(30),

PRIMARY KEY (chiname),

FOREIGN KEY (parname) REFERENCES Parent,

FOREIGN KEY (chiname) REFERENCES Parent

);

3

CREATE TABLE Professor

(

SSN Integer,

Salary Integer,

Phone Integer,

PRIMARY KEY (SSN)

);

CREATE TABLE Department

(

DNO Integer,

Name char(30),

Budget Integer,

ChairSSN Integer,

PRIMARY KEY (DNO),

FOREIGN KEY(ChairSSN) REFERENCES Professor

);

CREATE TABLE Grad\_Student

(

Name char(30),

AdvisorSSN Integer NOT NULL,

PRIMARY KEY (Name),

FOREIGN KEY(AdvisorSSN) REFERENCES Professor

ON DELETE CASCADE

);

C Table: Students S, Courses C, Onestop O

1

a.

b.

c.

d.

e.

2.

a.

Var1:

Var2:

Answer:

b.

3.

D

1b

SELECT P1.custid FROM Purchases P1, Purchases P2

WHERE P1.purchaseMethod < P2.purchaseMethod AND P1.custid = P2.custid

EXCEPT

SELECT P1.custid FROM Purchases P1, Purchases P2, Purchases P3,

WHERE P1.purchaseMethod < P2.purchaseMethod AND P2.purchaseMethod < P3.purchaseMethod AND P1.custid = P2.custid AND P2.custid = P3.custid;

2.

SELECT S1.barId, S2.barId FROM Serves S1, Serves S2

WHERE S1.barId < S2.barId AND S1.beerId = S2.beerId

GROUP BY S1.barId, S2.barId

HAVING COUNT(\*) = (SELECT COUNT(S3.beerId) FROM Serves S3

WHERE S3.barId = S1.barId) AND COUNT(\*) = (SELECT COUNT (S4.barId) FROM Serves S4 WHERE S4.barId, S2.barId);

4.

SELECT A.name FROM Actors A,

WHERE A.aid IN (SELECT C.aid FROM Cast C, Directories D, Movies M

WHERE M.mid = C.mid AND D.did = M.did AND D.name = "Spielberg"

EXCEPT

SELECT C1.aid FROM Cast C1, Directories D1, Movies M1

WHERE M.mid = C.mid AND D.did = M.did AND D.name != "Spielberg");