BC+ = {B, C, G}

E+ = {E, D, B, C, G} //siêu khóa vì có đủ thuộc tính

Khóa E là khóa dự bị vì là siêu khóa có ít phần tử nhất

BCNF: tất cả vế trái của F là siêu khóa

* Không là BCNF

3NF: Với bất kỳ phụ thuộc hàm X → A trong F thì hoặc X là một siêu khóa của R, hoặc A là một thuộc tính khóa (và) Không có thuộc tính không khóa nào phụ thuộc hàm bắc cầu vào khóa của R

BC không là siêu khóa và G không là thuộc tính khóa => Không là 3NF

2NF: các thuộc tính không khóa đều phụ thuộc toàn phần vào khóa dự bị

Là 2NF

//ĐK là 2NF :

Mọi thuộc tính không khóa đều phụ thuộc toàn phần vào mọi khóa dự bị hoặc phụ thuộc bắc cầu vào mọi khóa dự bị.

b,

R1={BCG} F1={BC->G} K={BC}

R2={BCDEG} F2={E->DE, E->B, E->CD, E->G} K={E}

Câu 2:

CREATE TABLE CV(

cvid CHAR(20),

cvname CHAR(20),

sln INTEGER,

diachi CHAR(20),

PRIMARY KEY (cvid)

);

CREATE TABLE NLC(

nsxid CHAR(20),

lkid CHAR(20),

cvid CHAR(20),

sl INTEGER,

PRIMARY KEY (nsxid, lkid, cvid),

FOREIGN KEY(nsxid) REFERENCES NSX(nsxid),

FOREIGN KEY(lkid) REFERENCES LK(lkid),

FOREIGN KEY(cvid) REFERENCES CV(cvid),

);

SELECT NSX.diachi FROM NSX, LK, NLC

WHERE LK.mau = “xanh” and NSX.id=NLC.id and LK.id=NLC.id;

SELECT NSX.name FROM NSX, NLC, CV

WHERE CV.soluongnguoi>200 and CV.id = NLC.id and NSX.id=NLC.id;

INSERT INTO NSX(nsxid, lkid, cvid, sl) VALUES (“NSX1”, “LK2”, “CV30”, 25);

DELETE FROM NSX WHERE NOT IN (SELECT \* FROM NLC, NSX WHERE NLC.id=NSX.id)

