

Dépendances :

A B C D E

a1 b1 c1 d1 e1

a1 b2 c2 d2 e1

a2 b1 c3 d3 e1

a2 b1 c4 d3 e1

a3 b2 c5 d1 e1

A -> D = non (a1 -> d1 et a1 -> d2 = impossible)

AB -> D = oui (a1^b1 = d1, etc. tous différents)

=> Une entrée doit donner au plus une sortie

X est super clé de $\Pi XY (r)$ si X détermine fonctionnellement tous les autres attributs de la relation $\Pi XY (r)$, c'est à dire X détermine fonctionnellement Y.

Algorithme de vérification de dépendance :

1) Trier la table suivant X

2) Regarder que si $X_i = X_{i+1}$ alors $Y_i = Y_{i+1}$

Ou bien :

1) Stocker les paires (X,Y) (Ex : map en java)

2) Comparer les cas avec des X identiques : vérifier que les Y le soient

SELECT * FROM r, r as copier where r.X=copier.X and r.Y<> copieR.Y

Doit rendre un ensemble vide de tuple.

DF = {AB → C, C → E, E → C, C → D, AB → E}

AB -> E existe déjà (AB -> C + C -> E)

Reste C -> D --> C -> E --> E -> D

DF' = {AB → C, C → E, E → C, E → D}

Clé : C'est un ensemble d'attribut qui détermine fonctionnellement tous les autres attributs.

1NF : les attributs sont atomiques (une seule valeur par attribut + pas de répétitions)

Si répétitions -> séparer en plusieurs tables

2NF : attributs non clés ne peut pas dépendre d'une partie d'une clé mais de la totalité de la clé

Si clé en plusieurs parties, vérifier si chaque item dépend de l'intégralité de la clé.

Ex : (* = clé)

*N° de commande

*N° article

Désignation

Quantité commandée

-->

*N° commande

*N° article

Quantité commandée

+

*N° article

Désignation

-> 2 tables

3NF : attributs non clés ne peut pas dépendre d'un sous ensemble d'attribut non clés (la 2NF l'autorise) mais directement de la clé

Ex :

*N° commande

Date

N° client

Nom

-->

*N° commande

Date

N° client

+

*N° client

Nom

-> Mettre les champs dépendants par transitivité dans d'autres tables

Dépendance fonctionnelle : A -> B si chaque valeur de A n'a qu'une valeur associée de B.

Transitivité : Si A -> B et B -> alors A -> C.

GROUP BY B.IdOrganisme

HAVING COUNT(*) >=2

DROP VIEW IF EXISTS employees;

DROP TABLE IF EXISTS Depot, Brevet, Personne, Organisme, Pays ;

CREATE TABLE Pays (

IdPays CHAR(2) PRIMARY KEY ,

Nom VARCHAR(30));

CREATE TABLE Organisme(

IdOrganisme INT PRIMARY KEY,

NomOrganisme VARCHAR(30),

NbEmployes INT,

IdPays CHAR(2) REFERENCES Pays);

CREATE TABLE Personne (

IDPersonne INT PRIMARY KEY,

Nom VARCHAR(30),

Prenom VARCHAR(30) ,

Age INT,

IdPaysNaissance CHAR(2) REFERENCES Pays,

IdOrganisme INT REFERENCES Organisme);

CREATE TABLE Brevet (

NumB INT PRIMARY KEY CHECK (NumB >= 1000),

IdOrganisme INT REFERENCES Organisme,

IdPersonne INT REFERENCES Personne);

CREATE TABLE Depot (

NumB INT REFERENCES Brevet,

DateDpt DATE CHECK (DateDpt >= '1789-07-13'),

IdPays CHAR(2) REFERENCES Pays,

PRIMARY KEY (NumB,DateDpt,IdPays));

```

INSERT INTO Pays VALUES ('FR', 'France');
INSERT INTO Organisme VALUES (3,'HMAG',2, 'CH');
INSERT INTO Personne VALUES (30, 'Happart','Noemie',
65,'BE',2);

```

```

-- Q1: Noms des organismes de plus de 100 personnes
ayant depose un brevet avant l'an 2000
SELECT O.NomOrganisme FROM Organisme O
WHERE O.NbEmployes >= 100 AND O.IdOrganisme IN
( SELECT B.IdOrganisme FROM Brevet B WHERE
B.NumB IN
(SELECT D.NumB FROM Depot D WHERE
D.DateDpt < '2000-01-01'));

```

```

-- Q2 Afficher l'IdOrganisme des organismes qui ont
depose au moins un brevet dans chaque pays
SELECT B.IdOrganisme FROM Brevet B
EXCEPT
SELECT T.IdOrganisme FROM
(
SELECT O.IdOrganisme, P.IdPays FROM Organisme
O, Pays P
EXCEPT
SELECT B.IdOrganisme, D.IdPays FROM Brevet B,
Depot D
WHERE B.NumB = D.NumB
) AS T;

```

```

-- ou
SELECT O.IdOrganisme FROM Organisme O
WHERE NOT EXISTS
(SELECT * FROM Pays P WHERE P.IdPays NOT IN
(SELECT D.IdPays FROM Depot D WHERE
D.NumB IN
(SELECT B.NumB FROM Brevet B WHERE
B.IdOrganisme = O.IdOrganisme )));

```

-- Q3 Ecrire une requete SQL qui calcule par
IdOrganisme le nombre de brevets deposees par chaque
personne

```

SELECT B.IdOrganisme, B.IdPersonne, COUNT(*) FROM
Brevet B, Depot D
WHERE B.NumB = D.NumB
GROUP BY B.IdOrganisme, B.IdPersonne
ORDER BY B.IdOrganisme, B.IdPersonne ;

```

-- Q4 Ecrire une requete SQL qui calcule le nombre total
des employes qui travaillent dans les organismes qui ont
au moins deposees 2 brevets

```

CREATE OR REPLACE VIEW Employes AS
(SELECT O.IdOrganisme, COUNT(*) FROM Organisme
O, Brevet B, Depot D
WHERE O.IdOrganisme = B.IdOrganisme AND
B.NumB = D.NumB
GROUP BY O.IdOrganisme
HAVING COUNT(*) >=2);

```

```

-- ou
CREATE OR REPLACE VIEW Employes AS
(SELECT B.IdOrganisme FROM Brevet B
GROUP BY B.IdOrganisme
HAVING COUNT(*) >=2);

```

```

SELECT SUM(O.NbEmployes) FROM Organisme O
WHERE O.IdOrganisme IN (SELECT E.IdOrganisme
FROM Employes E);

```