

BD

02/06/09

Corrigé de l'exercice.

Exercice 1.

 $R(A, B, C, D, E, F)$ $A, B \rightarrow C$ $D \rightarrow C$ $D \rightarrow E$ $CE \rightarrow F$ $E \rightarrow A$

A	B	C	D	E	F
w	1	i	110	m.	54
x	2	j	100	n.	52
w	1	i	110	m.	54
x	2	j	100	n.	52

 $D^+ = DCEFA$ $(CE)^+ = CEFA$

Clé candidate.

 $(BD)^+ = BDCEFA$

Exercice 2.

 $R(A, B, C, D)$

superclé

 $\overline{AB} \rightarrow C$

superclé

 $\overline{AB} \rightarrow D$

donc c'est en 3NF.

 $D \rightarrow \underline{B}$

attribut primaire

3NF?

il faut avoir les clés candidates $(AB)^+ = ABCD$ $(AD)^+ = ADCB$

{AB}, {AD} clés candidates

3NF si pour chaque $X \rightarrow A$ dépendance fonctionnelle.soit X est une superclé $\Rightarrow (X^+ = \text{tous les attributs})$ soit A un attribut primaire (1 attribut qui a 1 clé candidate)

ici les attributs primaire = {A, B, D}

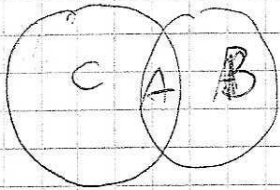
BCNF?

R est en BCNF si pr tout $x \rightarrow A \in F^t$ x est une superclé
la 3^{ème} dépendance car D n'est pas 1 superclé ($D \rightarrow B$)
donc ce n'est pas une BCNF

de Compos^{it}.

th Healt₄ R(A,B,C)

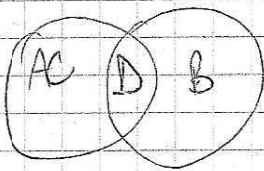
$A \rightarrow B \in F$



$$R_1 = \pi_{CA}(R)$$

$$R_2 = \pi_{AB}(R)$$

$D \rightarrow B$



→ on garde les colonnes AC, D de R
 $\pi_{ACD}(R) = R_1$

$$\pi_{BD}(R) = R_2$$

$$R = R_1 \bowtie R_2$$

dans la table R_2

D $\rightarrow$ B car D est la clé primaire

dans la table R_1

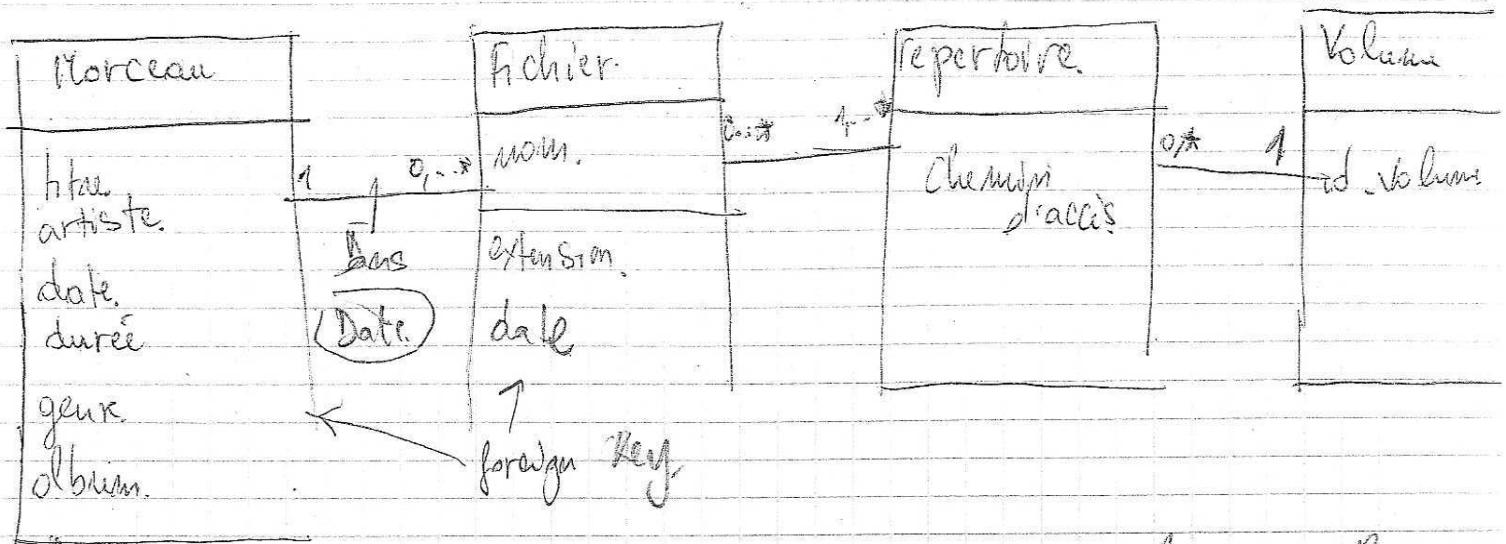
AD $\rightarrow$ C

il y a perte de dépendances fonctionnelles ($AB \rightarrow C$)

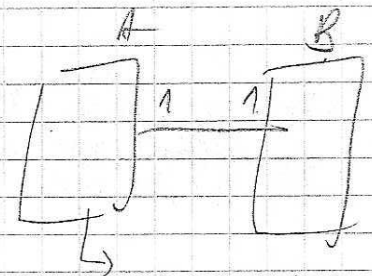
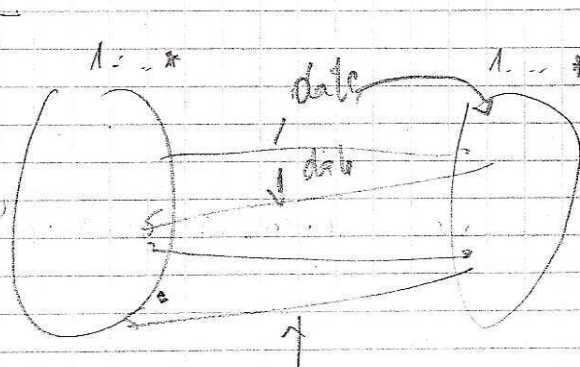
- trigger
- prix
- chemin
- Résultat
- extension
- rapport

Exercice 3

4



8. ou a :



date: foreign key entre A, soit en B

pas codé avec foreign key plutôt 1
table avec comme de primaire.
les des primaires de 2 autres

Exercice 4

int doc(10, 2) time-stamp.
Cours (action-id, valeur, quand)
possede (client-id, action-id, quantite, min)
mise-en-vente (client-id, action-id, quantite, date)

create trigger toto.

after insert

On Cours for each row

execute procedure vendre();

Create or replace function vendre () returns trigger as \$\$

declare

toto record;

vendre cursor for select

begin

open vendre for toto in client_id, action_id, quantite

loop

fetch vendre

into toto

exit when not found where New.value < ~~New~~.min and

New.action_id = action_id and quantite <= 0

update loop

update passed set quantite = 0

where client_id = toto.client_id and action_id = toto.action_id

insert insert into mise en vente

values (toto.client_id, toto.action_id, toto.quantite, current-
timestamp);

end

\$\$ language

END loop

END

\$\$ language 'plpgsql';

methode avec curseur.

new = la ligne à insérer

old : ligne avant modif et new ligne après modif

Exercice 6

Grant select insert on table action

to toto

with Grant options :

sert à donner le droit à toto sur la table actions, les droits select insert et la dernière ligne lui permet de donner des droits ~~act~~ et select et update.

Exercice 5

Begin ;

set transaction.

isolation level.

serializable ;

select ...

update ...

insert ...

commit // pour finir. end-if ;

Exercice 7

end.
pp langage 'plpgsql'!

table de indexage égale $\rightarrow$ action_id (=). B-arbre peut être utilisé.

B-arbre : inégalité $\rightarrow$ Date <

Date < et action_id =

parcours séquentiel à partir du 1er résultat

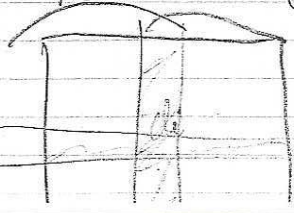
Exercice 8

distribu² de tables

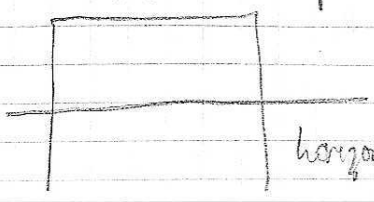
reparties de données, ds plusieurs endroits.

Clag ptes doit avoir en

commune la clé



découpage
vertical



horizontal