

IE2 de BDD - correction

Exercice 1 - TRC

Question 1.1 (1pt)

La deuxième ligne nous indique que nous recherchons des identifiants de vols $t[vid]$ qui peuvent être volés ($v[distance] \leq a[portee]$) avec des avions affectés à des pilotes p (jointure sur les aid).

Les troisième et quatrième lignes nous indiquent que ces vols ont une autre propriété. Si on considère un vol v , tout avion qui pourrait le voler avec d'autres pilotes p_2 implique que $p_2 = p$ (avec $c2[eid] = c[eid]$). En d'autres termes, il n'y a pas d'autres pilotes que p qui peuvent le voler.

On conclue alors que cette requête renvoie les identifiants de vols qui ne peuvent être volés que par un seul pilote.

Question 1.2 (1pt)

On va modifier la requête TRC pour faire apparaître une forme que l'on peut écrire en SQL.

On supprime l'implication à l'aide de la formule de De Morgan :

$\{t|\exists v \in vols, \exists a \in avions, \exists c \in certifications :$

$t[vid] = v[vid] \text{ and } a[aid] = c[aid] \text{ and } v[distance] \leq a[portee] \text{ and}$

$\forall c2 \in certifications, \forall a2 \in avions : not(c2[aid] = a2[aid] \text{ and } v[distance] \leq a2[portee]) \text{ or } c2[eid] = c[eid]\}$

On supprime les "pour tout" :

$\{t|\exists v \in vols, \exists a \in avions, \exists c \in certifications :$

$t[vid] = v[vid] \text{ and } a[aid] = c[aid] \text{ and } v[distance] \leq a[portee] \text{ and}$

$\nexists c2 \in certifications, \nexists a2 \in avions : not(not(c2[aid] = a2[aid] \text{ and } v[distance] \leq a2[portee]) \text{ or } c2[eid] = c[eid])\}$

On supprime les doubles négations :

$\{t|\exists v \in vols, \exists a \in avions, \exists c \in certifications :$

$t[vid] = v[vid] \text{ and } a[aid] = c[aid] \text{ and } v[distance] \leq a[portee] \text{ and}$

$\nexists c2 \in certifications, \nexists a2 \in avions : c2[aid] = a2[aid] \text{ and } v[distance] \leq a2[portee]) \text{ and } c2[eid] \neq c[eid]\}$

On peut maintenant écrire la requête en SQL ainsi :

```
SELECT vid FROM vols v, avions a NATURAL JOIN certifications c WHERE
v.distance<=a.portee and not EXISTS(
  SELECT * FROM avions a2 NATURAL JOIN certifications c2 WHERE
  a2.portee>=v.distance and c2.eid<>c.eid
);
```

Exercice 2 - Normalisation

Dans tout cet exercice, les attributs formant la clé d'une relation sont soulignés.

Question 2.1 (1pt)

On considère $R(\underline{MedecinID}, \underline{PatientID}, \underline{Date}, Diagnose, CodeTraitement, Tarif)$.

On définit les dépendances fonctionnelles par

$F = \{ [\underline{MedecinID}, \underline{PatientID}, \underline{Date}] \rightarrow [CodeTraitement, Diagnose], [CodeTraitement] \rightarrow [Tarif] \}$.

Question 2.2 (1pt)

La relation R est en 2NF car il n'y a pas de dépendances partielles.

Question 2.3 (1pt)

La relation R n'est pas en 3NF car l'attribut non-primaire $Tarif$ est dépendant transitivement de la clé : $[\underline{MedecinID}, \underline{PatientID}, \underline{Date}] \rightarrow [CodeTraitement] \rightarrow [Tarif]$.

On décompose R en (R_1, R_2) qui est 3NF avec

$R_1 = (\underline{MedecinID}, \underline{PatientID}, \underline{Date}, Diagnose, CodeTraitement)$ et

$R_2 = (\underline{CodeTraitement}, Tarif)$.