

Solutions de l'examen *Bases de données* 2020–2021

Exercice A : Algèbre relationnelle et SQL sans imbrication

1. Nom des équipements situés au repère GPS 345_78S

Algèbre relationnelle

$$\pi_{Nom}(\sigma_{GPS='345_78S'}(Equip))$$

SQL

```
SELECT Nom
FROM Equip
WHERE GPS = '345_78S';
```

Explication On sélectionne d'abord les tuples de *Equip* dont *GPS*='345_78S' (opérateur σ), puis on projette la colonne *Nom* (opérateur π).

2. Identifiants des épreuves de ski

Algèbre relationnelle

$$\pi_{IdE} \left(Epreuve \bowtie_{Epreuve.IdS=Sport.IdS} \sigma_{Nom='ski'}(Sport) \right)$$

SQL

```
SELECT E.IdE
FROM Epreuve AS E
JOIN Sport AS S
  ON E.IdS = S.IdS
WHERE S.Nom = 'ski';
```

Explication On filtre les sports de nom “ski”, on les joint à *Epreuve* selon *IdS*, puis on projette *IdE*.

3. Identifiants des épreuves individuelles au GPS 345_78S

Algèbre relationnelle

$$\begin{aligned} &\pi_{IdE} \left(\sigma_{TP='individuelle'}(Sport) \right. \\ &\quad \left. \bowtie_{Sport.IdS=Epreuve.IdS} Epreuve \right. \\ &\quad \left. \bowtie_{Epreuve.IdEq=Equip.IdEq} \sigma_{GPS='345_78S'}(Equip) \right) \end{aligned}$$

SQL

```
SELECT E.IdE
FROM Sport AS S
JOIN Epreuve AS E
    ON S.IdS = E.IdS
JOIN Equip AS Q
    ON E.IdEq = Q.IdEq
WHERE S.TP = 'individuelle'
    AND Q.GPS = '345_78S';
```

Explication On applique deux sélections successives (type “individuelle”, puis GPS) via deux jointures, puis on projette IdE.

4. Sports (nom) sans aucune épreuve

Algèbre relationnelle

$$\pi_{Nom}(Sport) - \pi_{Nom}(Sport \bowtie Epreuve)$$

SQL

```
SELECT Nom
FROM Sport
WHERE IdS NOT IN (
    SELECT DISTINCT S.IdS
    FROM Sport AS S
    JOIN Epreuve AS E
        ON S.IdS = E.IdS
);
```

Explication On prend la différence entre l'ensemble des sports et ceux apparaissant dans Epreuve.

5. Sports (nom, TP) ayant utilisé l'équipement de l'épreuve de ski individuelle du 14-02-2014

Algèbre relationnelle

$$\pi_{Nom,TP}\left(Sport \bowtie \left(\pi_{IdEq}(\sigma_{Nom='ski' \wedge TP='individuelle' \wedge Date='2014-02-14'}(Epreuve \bowtie Sport))\right)\right)$$

SQL

```
SELECT DISTINCT S.Nom, S.TP
FROM Sport AS S
JOIN Epreuve AS E
    ON S.IdS = E.IdS
WHERE E.IdEq = (
    SELECT IdEq
```

```

FROM Epreuve AS X
JOIN Sport AS Y
  ON X.IdS = Y.IdS
WHERE Y.Nom = 'ski'
      AND Y.TP = 'individuelle'
      AND X.Date = '2014-02-14'
)
;

```

Explication On isole d'abord l'IdEq de l'épreuve ciblée par une sous-requête, puis on sélectionne tous les sports l'ayant utilisé.

6. Équipements jamais utilisés (noms)

Algèbre relationnelle

$$\pi_{Nom}(Equip) - \pi_{Nom}(Equip \bowtie Epreuve)$$

SQL

```

SELECT Nom
FROM Equip
WHERE IdEq NOT IN (
  SELECT DISTINCT IdEq
  FROM Epreuve
);

```

Explication On liste les équipements dont l'identifiant n'apparaît pas dans la relation Epreuve.

7. Identifiants des équipements utilisés au moins deux fois

Algèbre relationnelle La condition « au moins deux fois » s'exprime par auto-mutlijoin :

$$\pi_{E1.IdEq}(\sigma_{E1.IdEq=E2.IdEq \wedge E1.IdE \neq E2.IdE}(\rho_{E1}(Epreuve) \times \rho_{E2}(Epreuve)))$$

SQL

```

SELECT IdEq
FROM Epreuve
GROUP BY IdEq
HAVING COUNT(*) >= 2;

```

Explication On regroupe par IdEq et on filtre ceux dont le nombre d'occurrences est au moins 2.

Exercice B : Algèbre relationnelle (division)

8. Lieux (GPS) où *tous* les sports ont été programmés

Soit

$$— R_1 = \pi_{IdS, IdEq}(Epreuve)$$

$$— R_2 = \pi_{IdS}(Sport)$$

$$— R_3 = \pi_{GPS, IdEq}(Equip)$$

On veut

$$\pi_{GPS}\left(R_3 \bowtie (R_1 \div R_2)\right)$$

où $R_1 \div R_2$ donne les *IdEq* associés à *tous* les *IdS*. Enfin on retrouve les GPS correspondants.

Exercice C : SQL avec agrégats et sous-requêtes

9. Nombre d'équipements au repère GPS 345_78S

```
SELECT COUNT(*) AS nb_equipements
FROM Equip
WHERE GPS = '345_78S';
```

10. Identifiants des équipements utilisés au moins 10 fois

```
SELECT IdEq
FROM Epreuve
GROUP BY IdEq
HAVING COUNT(*) >= 10;
```

11. Durée moyenne des épreuves par sport

```
SELECT S.IdS, S.Nom, AVG(E.Duree) AS duree_moyenne
FROM Sport AS S
JOIN Epreuve AS E
  ON S.IdS = E.IdS
GROUP BY S.IdS, S.Nom;
```

12. Lieux (GPS) où *tous* les sports ont été programmés (SQL)

```
SELECT Q.GPS
FROM Equip AS Q
WHERE NOT EXISTS (
  SELECT *
  FROM Sport AS S
  WHERE NOT EXISTS (
    SELECT *
    FROM Epreuve AS E
    WHERE E.IdS = S.IdS
    AND E.IdEq = Q.IdEq
  )
)
```

)
);

Anomalies de mise à jour

On considère la relation `Employe`(`Emp_Id`, `Nom`, `Salle`, `Extension`) dont l'instance est celle de la Figure 2.

1. Trois types d'anomalies

- **Anomalie de mise à jour (update anomaly)** : pour modifier l'extension de la salle A7, il faut mettre à jour deux tuples (Dupont, Petit) : risque d'incohérence.
- **Anomalie de suppression (delete anomaly)** : si on supprime la ligne de **Grand**, on perd l'information qu'il existe une extension 4501 pour la salle B4 si aucun autre employé n'est dans B4.
- **Anomalie d'insertion (insert anomaly)** : on ne peut pas enregistrer une nouvelle salle C1 avec extension 3200 avant qu'un employé ne l'occupe : la relation lie employés et salles.

2. Élimination des anomalies

On décompose en deux tables :

$$\begin{aligned} &Employe'(Emp_Id, Nom, Salle), \\ &Salle(Salle, Extension) \end{aligned}$$

Ainsi l'extension dépend uniquement de la salle.

3. Nécessité de la normalisation

La normalisation est justifiée si l'on souhaite garantir l'intégrité et éviter les anomalies à long terme. Toutefois, pour de très petites relations ou en cas de fortes contraintes de performance, on peut tolérer un certain dénormalisation.

Formes normales

Cas 1 : $R(ABCD)$, $F = \{C \rightarrow D, C \rightarrow A, B \rightarrow C\}$

1. B^+

On a $B \rightarrow C$, $C \rightarrow A$, $C \rightarrow D$, donc

$$B^+ = \{B, C, A, D\}.$$

2. F^+ (fermée de F)

Outre les dépendances initiales, on déduit :

$$B \rightarrow A, \quad B \rightarrow D, \quad (...)$$

(on liste l'ensemble des DF impliquées par transitivité)

3. Clés

Puisque B^+ contient tous les attributs, B est clé. Aucune autre singleton ne l'est.

4. 3FN ?

La DF $C \rightarrow A$ viole la 3FN car C n'est pas super-clé et A n'est pas partie d'une clé. Donc *non*.

5. Instance redondante

Par exemple :

B	C	A	D
b1	c1	a1	d1
b1	c1	a1	d2

On répète $a1$ à tort.

Cas 2 : $R(ABCDE)$, $F = \{A \rightarrow B, AB \rightarrow D, C \rightarrow E, D \rightarrow C, AC \rightarrow E\}$

1. Clés

On calcule $A^+ : A \rightarrow B$, puis $AB \rightarrow D$, $D \rightarrow C$, $C \rightarrow E$. Ainsi $A^+ = \{A, B, D, C, E\}$, donc A est clé.

2. FNBC ?

La DF $D \rightarrow C$ viole la BCNF (l'antécédent D n'est pas super-clé). Donc *non*.

3. Décomposition BCNF

On choisit la DF violant BCNF, $D \rightarrow C$, et on obtient :

$$R_1(D, C), \quad R_2(A, B, D, E)$$

Puis dans R_2 , on a $AB \rightarrow D$ qui viole BCNF, on décompose encore :

$$R_{21}(A, B, D), \quad R_{22}(A, B, E)$$

Arbre de décomposition :

$$R(ABCDE) \xrightarrow{D \rightarrow C} (R_1(DC), R_2(ABDE)) \xrightarrow{AB \rightarrow D} (R_{21}(ABD), R_{22}(ABE), R_1(DC))$$