

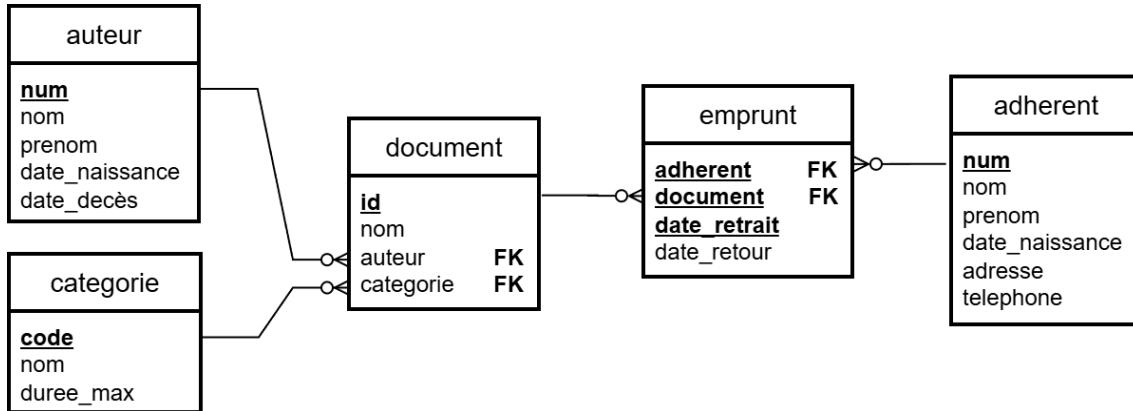
- L3 MIASHS/Ingémath/METIS
- Université Paris Cité
- Course Homepage



Examen du 9 janvier 2026 (3H)

Exercice 1

Voici le modèle relationnel du SI d'une médiathèque.



Ce SI enregistre les emprunts de documents par les adhérents de la médiathèque. Les documents sont de plusieurs catégories (livre, BD, CD, DVD). Chaque catégorie a une durée d'emprunt maximum spécifique exprimée en jours.

- Les attributs `auteur.num`, `categorie.code`, `document.id` et `adherent.num` sont des entiers ainsi que les attributs qui les référencent.
- Les attributs `emprunt.date_retrait`, `emprunt.date_retour` sont de type DATE et stockent les dates de retrait et de retour d'un emprunt de document par un adhérent. **Lorsqu'un document n'est pas encore rendu**, l'attribut `emprunt.date_retour` est vide.
- Les attributs `auteur.date_naissance`, `auteur.date_deces` et `adherent.date_naissance` sont de type DATE.
- L'attribut `categorie.duree_max` est un entier qui stocke le nombre de jours maximum d'un emprunt pour un document de la catégorie.

Questions

1. Écrire en SQL la requête de définition de la table `emprunt` en tenant compte du modèle relationnel et des indications précédentes.

 **Solution**

```
CREATE TABLE emprunt(
    adherent INTEGER,
    document INTEGER,
    date_retrait DATE,
    date_retour DATE,
    CONSTRAINT emprunt_pk PRIMARY KEY (adherent,document,date_retrait),
    CONSTRAINT emprunt_adherent_fk
        FOREIGN KEY (adherent)
        REFERENCES adherent(num),
    CONSTRAINT emprunt_document_fk
        FOREIGN KEY (document)
        REFERENCES document(id)
);
```

2. Écrire en SQL une requête qui liste les noms des documents dont l'auteur est Jules Verne.

 **Solution**

```
SELECT d.nom
FROM document d JOIN auteur a ON d.auteur = a.num
WHERE a.nom ILIKE 'Jules Verne';
```

3. Écrire en SQL une requête qui liste pour chaque auteur, le nombre de documents présents dans la médiathèque. On indiquera sur chaque ligne, le numéro de l'auteur, le nom de l'auteur et le nombre de documents.

 **Solution**

```
SELECT a.num, a.nom, COUNT(d.id) as nb_doc
FROM document d JOIN auteur a ON d.auteur = a.num
GROUP BY a.num, a.nom;
```

4. Écrire en SQL une requête qui liste toutes les informations des adhérents qui ont emprunté au moins un document qui n'a pas encore été rendu. Chaque adhérent ne doit apparaître qu'une fois.

 **Solution**

```
SELECT DISTINCT a.*
FROM emprunt e JOIN adherent a ON e.adherent = a.num
WHERE e.date_retour IS NULL;
```

5. Écrire en SQL une requête qui liste les id des documents de la catégorie nommée BD qui ont été empruntés au moins 10 fois.

Solution

```
SELECT d.id
FROM emprunt e
    JOIN document d ON e.document = d.id
    JOIN categorie c ON d.categorie = c.code
WHERE c.nom = 'BD'
GROUP BY d.id
HAVING COUNT(e.*) >=10 ;
```

6. Écrire en SQL une requête qui liste les numéros des adhérents qui n'ont jamais emprunté le document dont l'id est 200.

Solution

Avec une jointure externe et une CTE :

```
WITH emprunt_200 as
(
    SELECT DISTINCT adherent
    FROM emprunt
    WHERE document = 200
)
SELECT a.num
FROM adherent a LEFT JOIN emprunt_200 e ON a.num = e.adherent
WHERE e.adherent IS NULL;
```

On peut écrire une version avec IN, ou bien EXISTS, et une sous-requête :

```
SELECT num
FROM adherent
WHERE num NOT IN
(
    SELECT DISTINCT adherent
    FROM emprunt
    WHERE document = 200
);
```

Une solution équivalente avec EXCEPT et deux sous-requêtes :

```
( 
    SELECT num
    FROM adherent
)
EXCEPT
(
    SELECT DISTINCT adherent
    FROM emprunt
    WHERE document = 200
);
```

Remarque : les clauses DISTINCT ne sont pas obligatoires.

7. Écrire en SQL une requête qui donne, pour chaque catégorie, le nombre moyen d'emprunts par document durant l'année 2025. Sur chaque ligne, on indiquera seulement le code de la catégorie et le nombre moyen.

Indication : l'expression `emprunt.date_retrait BETWEEN '2025-01-01' AND '2025-12-31'` permet de tester si un emprunt a été fait durant l'année 2025.

 **Solution**

```
WITH nb_emprunt_2025_par_document AS
(
    SELECT d.id, d.categorie, COUNT(e.*) AS nb_emprunt
    FROM emprunt e JOIN document d ON e.document = d.id
    WHERE e.date_retrait BETWEEN '2025-01-01' AND '2025-12-31'
    GROUP BY d.id, d.categorie
)
SELECT categorie, AVG(nb_emprunt)
FROM nb_emprunt_2025_par_document
GROUP BY categorie;
```

8. Écrire en SQL, sans sous-requête, sans CTE (sans WITH) et sans INTERSECT, une requête qui liste les numéros des adhérents qui ont emprunté au moins une fois le document d'id 100 et au moins une fois celui d'id 200. Un numéro d'adhérent ne doit apparaître au plus qu'une fois.

 **Solution**

```
SELECT DISTINCT e1.adherent
FROM emprunt e1 JOIN emprunt e2 ON e1.adherent = e2.adherent
WHERE e1.document = 100 AND e2.document = 200;
```

9. Écrire en SQL une requête qui indique le document emprunté par le plus grand nombre d'adhérents différents.

Solution

```
WITH nb_adherent_par_document AS
(
    SELECT document, COUNT(DISTINCT adherent) nb_adherent
    FROM emprunt
    GROUP BY document
)
SELECT document, nb_adherent
FROM nb_adherent_par_document
WHERE nb_adherent =
(
    SELECT MAX(nb_adherent)
    FROM nb_adherent_par_document
) ;
```

ou bien

```
WITH nb_adherent_par_document AS
(
    SELECT document, COUNT(DISTINCT adherent) nb_adherent
    FROM emprunt
    GROUP BY document
)
SELECT document, nb_adherent
FROM nb_adherent_par_document
WHERE nb_adherent >= ALL
(
    SELECT nb_adherent
    FROM nb_adherent_par_document
) ;
```

10. (BONUS) Expliquer pourquoi l'attribut `emprunt.date_retrait` fait partie de l'identifiant de la relation `emprunt`.

Solution

Si `emprunt.date_retrait` ne fait pas partie de la clé primaire de la table `emprunt`, chaque occurrence (`emprunt.adherent, emprunt.document`) ne peut figurer qu'une fois dans la table `emprunt`. Il n'est alors pas possible de stocker l'historique des emprunts.

Exercice 2

On veut concevoir une base de données pour gérer des réservations de courts de tennis par les membres d'un club de sport.

Doivent être enregistrés dans la base :

- les membres du club avec un numéro unique, un nom, un prénom.
- les entraîneurs du club avec un numéro unique, un nom, un prénom.
- les installations avec un numéro unique, un nom, un type d'installation.
- les types d'installation (court intérieur, court extérieur, trainer, etc.) avec un code unique, un nom, une description.

La future interface de saisie d'une réservation se présentera ainsi :

- Saisie du numéro de membre.
- Saisie de la date-heure et de la durée de la réservation souhaitée.

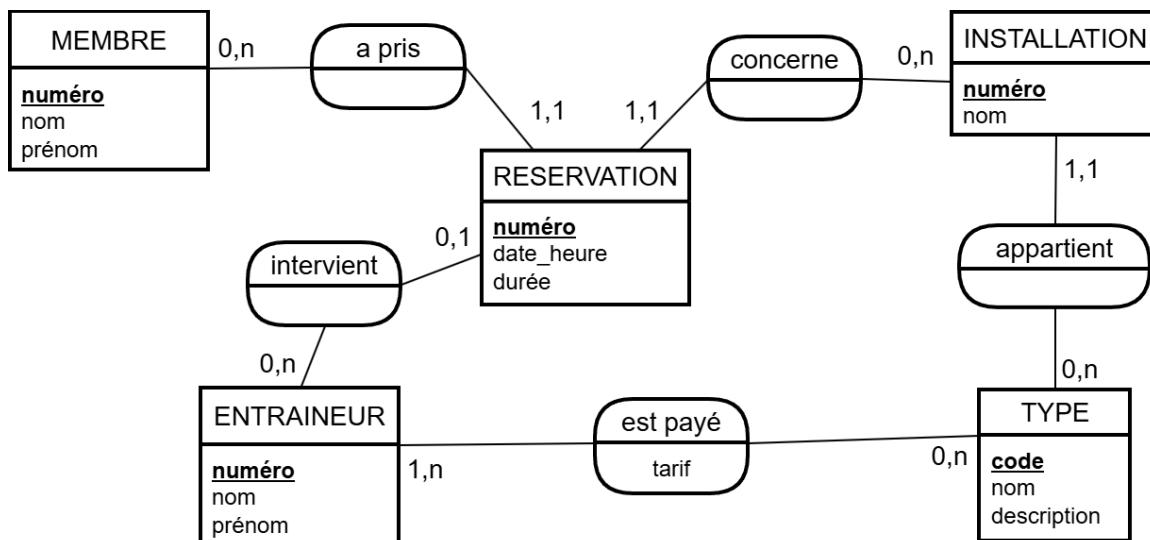
- Saisie du type d'installation souhaité. Après la saisie de ce champ, le système affichera les installations disponibles pour le type souhaité dans la plage horaire demandée et **le membre choisira une installation**. Si aucune installation n'est choisie ou n'est disponible, la réservation ne sera pas créée.
- Saisie de la demande d'un entraîneur : “oui/non”. Si le membre répond “oui”, le système affichera la liste des entraîneurs disponibles avec leurs tarifs horaires et **le membre choisira un entraîneur**. Si aucun entraîneur n'est choisi ou n'est disponible, la réservation ne sera pas créée à moins de répondre “non” à la demande d'un entraîneur.

Le **tarif horaire d'un entraîneur** dépend de l'entraîneur et du type d'installation sur laquelle il intervient.

Questions

1. Écrire, en modélisant les réservations par une entité, un diagramme E/A pour ce SI. Indiquer les cardinalités et justifier les cardinalités qui ajoutent une contrainte au modèle.

Solution

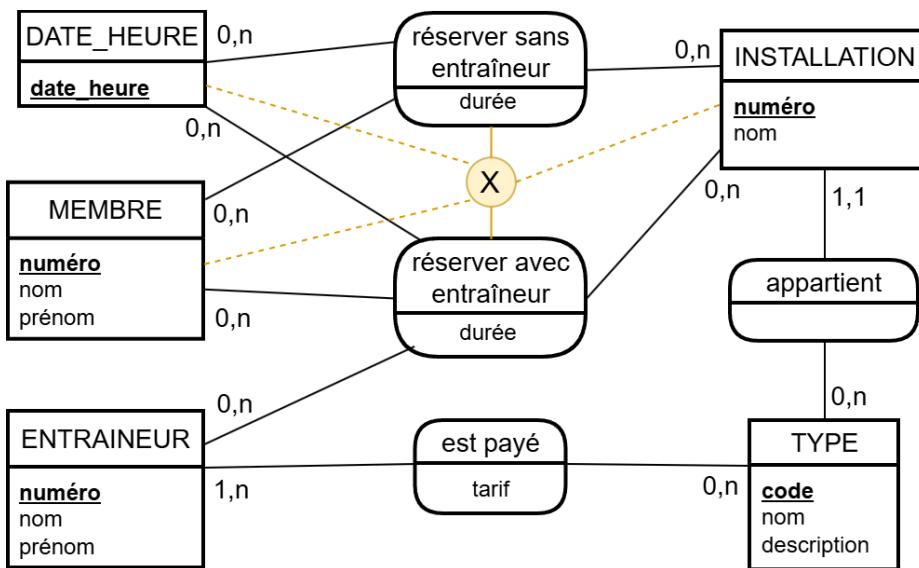


Justifications des cardinalités :

- association *a pris*. Côté *RESERVATION* : minimum et maximum 1 car dans le formulaire, une réservation est créée par un membre et un seul.
- association *concerne*. Côté *RESERVATION* :
 - minimum 1 : une réservation n'est pas créée si une installation disponible n'est pas choisie.
 - maximum 1 : le formulaire ne permet de choisir qu'une installation par réservation.
- association *intervient*. Côté *RESERVATION* :
 - minimum 0 : une réservation peut se faire sans entraîneur.
 - maximum 1 : le formulaire ne permet de choisir qu'un entraîneur par réservation.
- association *appartient*. Côté *INSTALLATION*, minimum et maximum 1 : l'énoncé dit qu'une installation à 1 type.
- association *est payé*. Côté *ENTRAINEUR* : l'énoncé ne détermine pas la cardinalité minimale. On peut supposer que chaque entraîneur a au moins un tarif pour une installation.

2. Écrire, en modélisant les réservations sans utiliser une entité, un diagramme E/A avancé pour ce SI. Indiquer les cardinalités (pas de justification demandée). Vérifier qu'un membre puisse réserver plusieurs fois la même installation. Ajouter si nécessaire des contraintes au diagramme.

Solution

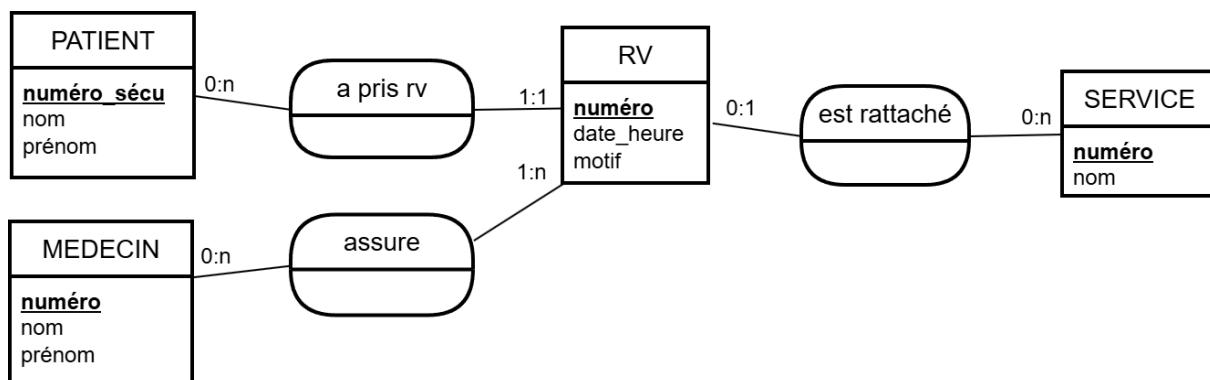


- Il faut créer une entité **DATE-HEURE** et la faire participer aux associations **réserver** pour permettre d'enregistrer plusieurs réservations d'un même membre pour la même installation.
- On doit créer deux associations **réserver** car il n'y a pas de participation optionnelle d'une entité (**ENTRAINEUR**) à une association.
- On peut ajouter une contrainte d'exclusion entre les deux associations **réserver** sur le triplet (**MEMBRE, INSTALLATION, *DATE-HEURE***), car un même membre ne peut pas réserver la même installation avec et sans entraîneur à la même date-heure. Mais cette contrainte ne décrit pas bien la réalité de la contrainte car la date-heure de début de la réservation ne suffit pas, il faut vérifier plus généralement une contrainte de non-chevauchement de toutes les réservations sur une installation.

Exercice 3

On veut créer une BDD pour enregistrer les RV d'une clinique médicale.

L'équipe de conception a élaboré le MCD suivant :



Indications complémentaires :

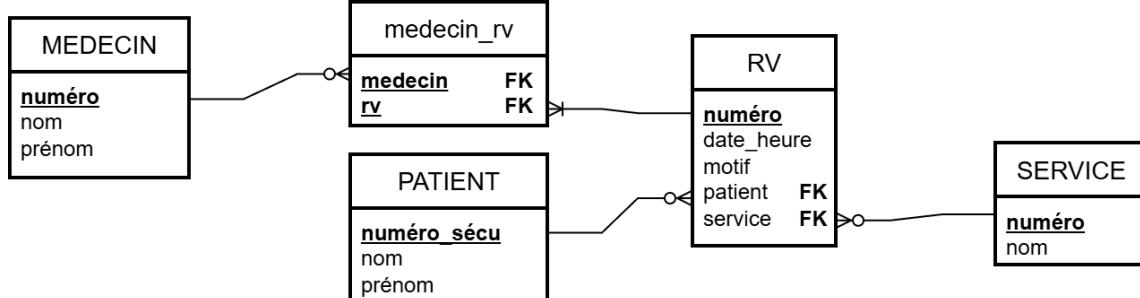
- dans certains cas particuliers, un RV nécessite la présence de plusieurs médecins.
- certains RV sont rattachés à un service, d'autres non.

Questions

- Convertir ce diagramme E/A en un modèle relationnel. Pas de justifications demandées.

Solution

On obtient ce modèle relationnel :



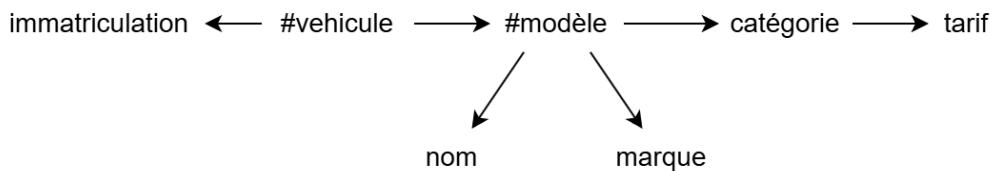
2. Quelles sont les clés étrangères sur lesquelles il faudra ajouter une contrainte NOT NULL ou UNIQUE lors de la définition des tables en SQL ? Justifier.

Solution

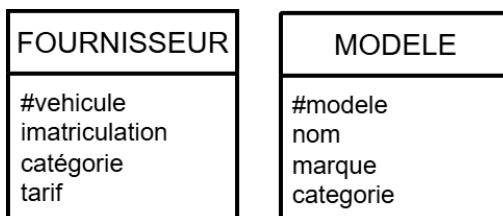
Il faut ajouter une contrainte NOT NULL sur l'attribut patient de la table rv car la cardinalité minimale de l'association *a pris rv* est 1 du côté de l'entité PATIENT.

Exercice 4

L'étude d'un SI a conduit à ce diagramme des DF :



L'équipe de conception vous a fourni ce schéma relationnel :



Questions

1. Déterminer si ce schéma relationnel entraîne des pertes de DF.

Solution

La DF `#vehicule → #modèle` nest pas préservée.

Pour tout l'exercice, on appelle Σ l'ensemble des DF données et on note de manière abrégée, F et M respectivement les schémas FOURNISSEUR et MODELE.

On justifie la perte de DF en montrant que la fermeture de $\{\#vehicule\}$ par rapport à $\pi_F(\Sigma) \cup \pi_M(\Sigma)$ ne contient pas `#modèle`. On note cette fermeture $\{\#vehicule\}_{F,M}^+$.

On utilise l'algorithme vu en cours.

initialisation $Z := \{\#vehicule\}$.

- $Z \cap F = \{\#vehicule\}$, $\{\#vehicule\}^+ = \{\#vehicule, immatriculation, #modèle, nom, marque, catégorie, tarif\}$, puis $\{\#vehicule\}^+ \cap F = F$, $Z := Z \cup F = F$.
- $Z \cap M = \emptyset$, Z est inchangé.

La répétition des deux étapes précédentes n'augmente pas Z donc $\{\#vehicule\}_{F,M}^+ = F$ qui ne contient pas `#modèle` (CQFD).

2. Pour chaque relation de ce schéma, déterminer les clés et vérifier si la relation est en FNBC, sinon donner la forme normale optimale qu'elle vérifie.

Solution

$\pi_F(\Sigma)$ est équivalent à

$$\Sigma_F = \{\#vehicule \rightarrow immatriculation, \#vehicule \rightarrow catégorie, catégorie \rightarrow tarif\}$$

- F a donc pour unique clé `#vehicule`. La DF `catégorie → tarif` ne respecte pas la FNBC car `catégorie` n'est pas une sur-clé de F . Elle ne respecte pas non plus la FN3 car `tarif` ne fait partie d'une clé de F .
- F est en FN2 car son unique clé ne contient qu'un attribut. C'est la forme normale optimale que F respecte.

$\pi_M(\Sigma)$ est équivalent à

$$\Sigma_M = \{\#modèle \rightarrow nom, \#modèle \rightarrow marque, \#modèle \rightarrow catégorie\}$$

- M a pour unique clé `#modèle`. Il n'y a pas de DF dans Σ_M qui ne respecte pas la FNBC donc M est en FNBC.

3. Pour les tables qui ne sont pas FNBC, illustrer les redondances par des exemples de tuples.

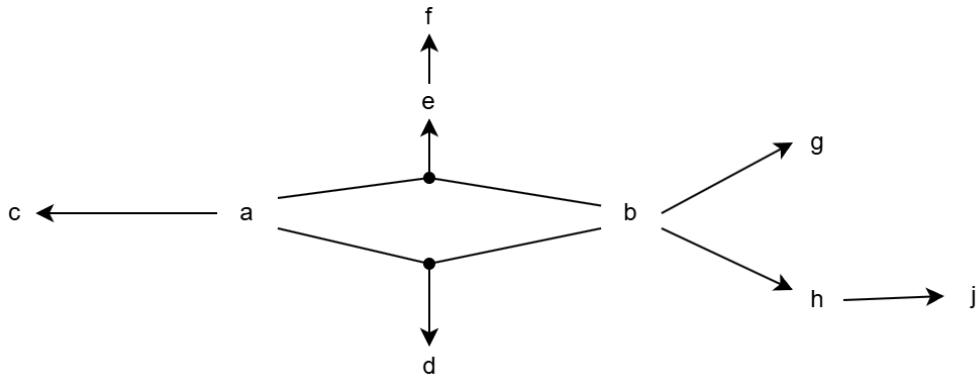
Solution

On peut illustrer les redondances de la relation `FOURNISSEUR` par les tuples :

#vehicule	immatriculation	catégorie	tarif
v1	i1	c1	t1
v2	i2	c1	t1
v3	i3	c1	t1

Exercice 5

Voici le diagramme des DF d'un SI :



Questions

- Proposer un schéma en FN3 pour ce SI.

Solution

Il s'agit de présenter le meilleur schéma en FN3, c'est à dire un schéma sans perte de DF et sans perte d'information.

On note Σ l'ensemble des DF données. Σ est irredundant donc on peut appliquer directement l'algorithme de décomposition vu en cours, on obtient le schéma

- $A_1(a, c)$, de clé a ,
- $A_2(a, b, d, e)$, de clé ab ,
- $A_3(e, f)$, de clé e ,
- $A_4(b, g, h)$, de clé b ,
- $A_5(h, j)$, de clé h .

Comme ab est une clé pour l'ensemble du SI, il n'est pas nécessaire d'ajouter un schéma pour éviter la perte d'information.

- Montrer que ce schéma est en FN3.

Solution

- $\pi_{A_1}(\Sigma)$ est équivalent à $\{a \rightarrow c\}$, $a \rightarrow c$ vérifie la FN3 et la FNBC pour A_1 car le déterminant est une clé de A_1 .
- $\pi_{A_2}(\Sigma)$ est équivalent à $\{ab \rightarrow ed\}$, $ab \rightarrow ed$ vérifie la FN3 et la FNBC pour A_2 car le déterminant est une clé de A_2 .
- $\pi_{A_3}(\Sigma)$ est équivalent à $\{e \rightarrow f\}$, $e \rightarrow f$ vérifie la FN3 et la FNBC pour A_3 car le déterminant est une clé de A_3 .
- $\pi_{A_4}(\Sigma)$ est équivalent à $\{b \rightarrow gh\}$, $b \rightarrow gh$ vérifie la FN3 et la FNBC pour A_4 car le déterminant est une clé de A_4 .
- $\pi_{A_5}(\Sigma)$ est équivalent à $\{h \rightarrow j\}$, $h \rightarrow j$ vérifie la FN3 et la FNBC pour A_5 car le déterminant est une clé de A_5 .

- Ce schéma est-il en FNBC ? Justifier.

Solution

Le schéma est en FNBC (justification dans la question précédente).