

Contrôle Final

(Correction)

La durée du contrôle est de 1 heure 30. Cet énoncé contient 6 pages et les questions ne sont pas ordonnées par difficulté croissante.

Le seul document autorisé est une feuille A4 manuscrite recto-verso.

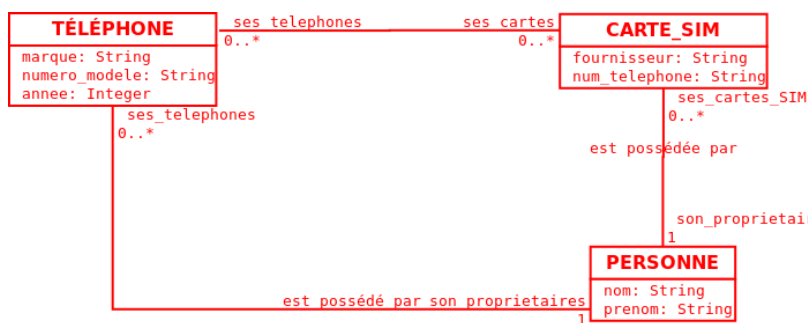
Si une question fermée (vrai/faux ou QCM) vous demande de **justifier** ou d'**expliquer** votre réponse, alors répondre sans justification vous rapportera 0 point.

1 Téléphones (15 minutes)

Un téléphone a une marque, un numéro de modèle, une année de commercialisation. Il est utilisé avec 0, 1 ou plusieurs “cartes SIM”. Une carte SIM est caractérisée par son fournisseur (Free, Orange, SFR, Bouygues...), son numéro de téléphone, et est associée à un et un seul propriétaire. Une personne peut posséder plusieurs téléphones et plusieurs cartes SIM.

- Proposez une modélisation en diagrammes de classe UML de ces spécifications.

Correction : D'autres solutions sont acceptables car pas explicites dans l'énoncé (par exemple, un téléphone possédé par plusieurs personnes).



- Transformez ensuite ce diagramme en un modèle relationnel, en précisant les clés primaires et étrangères. Pour cela, vous choisirez votre représentation (notation écrite parenthésée, schémas de table, etc.) mais elle devra bien sûr être lisible et claire.

Correction :

PERSONNE(id : Integer, nom : String, prenom : String)

TELEPHONE(id : Integer, marque : String, numero_modele : String, annee : Integer, propriétaire : Integer)
propriétaire est une clé étrangère faisant référence à PERSONNE.id

CARTE_SIM(num_telephone : String, fournisseur : String, propriétaire)
propriétaire est une clé étrangère faisant référence à PERSONNE.id

TELEPHONE_CARTE(tel_id, num_telephone)
tel_id est une clé étrangère faisant référence à TELEPHONE.id

num_telephone est une clé étrangère faisant référence à CARTE_SIM.num_telephone

2 Gestion des stages (15 minutes)

On souhaite créer une application de gestion des stages des élèves d'une école d'ingénieur. Un élève (numéro d'étudiant, nom, prénom) effectue plusieurs stages au cours de sa scolarité, chacun ayant une date de début, une date de fin, et s'effectuant dans une entreprise (nom, adresse). Une entreprise peut bien sûr accueillir plusieurs stagiaires. Un élève fait partie d'une spécialité, et on conserve l'information de l'année à laquelle il a rejoint cette spécialité.

Proposez une liste de dépendances fonctionnelles adaptée à ce problème, calculez-en la couverture minimale et proposez une modélisation relationnelle en 3ème forme normale.

Correction : *Beaucoup de possibilités possibles, par exemple :*

num_etudiant → *nom_etudiant, prenom_etudiant, specialite_etudiant, annee_etudiant*

nom_entreprise → *adresse_entreprise*

num_etudiant, date_debut, date_fin → *nom_entreprise_stage*

Puis cela donne (avec expansion de l'adresse) :

ENTREPRISE(*nom*, *num_rue*, *rue*, *code_postal*, *ville*)

ETUDIANT(*num_etudiant*, *nom*, *prenom*, *specialite*, *annee*)

STAGE(*id*, *num_etudiant*, *date_debut*, *date_fin*, *nom_entreprise*)

num_etudiant est une clé étrangère faisant référence à *ETUDIANT.num_etudiant*

num_entreprise est une clé étrangère faisant référence à *ENTREPRISE.nom*

3 Couverture minimale (20 minutes)

- Déterminez la couverture minimale de l'ensemble de dépendances fonctionnelles suivants. Expliquez chaque suppression d'élément ou de dépendance.

$\{F, I\} \rightarrow B$

$\{F, G\} \rightarrow I$

$F \rightarrow \{G, E\}$

$\{B, I\} \rightarrow A$

$\{A, B\} \rightarrow \{G, D\}$

$I \rightarrow M$

$G \rightarrow \{B, D\}$

$B \rightarrow \{F, E\}$

Correction : *Attention, plusieurs résultats possibles...*

(a) *Décomposition pour avoir un seul attribut à droite :*

$F \rightarrow G$

$F \rightarrow E$

$\{F, I\} \rightarrow B$

$\{B, I\} \rightarrow A$

$\{A, B\} \rightarrow G$

$\{A, B\} \rightarrow D$

$G \rightarrow B$

$G \rightarrow D$

$B \rightarrow F$

$B \rightarrow E$

$\{F, G\} \rightarrow I$

$$I \rightarrow M$$

(b) Suppression des attributs en surnombre à gauche (obtention de DF élémentaires) :

$F \rightarrow G$, donc par transitivité on transforme $\{F, G\} \rightarrow I$ en $\{F\} \rightarrow I$

$F \rightarrow I$, donc par transitivité on transforme $\{F, I\} \rightarrow B$ en $\{F\} \rightarrow B$

$B \rightarrow F$, et $F \rightarrow I$, donc par transitivité on transforme $\{B, I\} \rightarrow A$ en $\{B\} \rightarrow A$

Cela nous permet de transformer $\{A, B\} \rightarrow G$ en $B \rightarrow G$ et $\{A, B\} \rightarrow D$ en $B \rightarrow D$

(c) Suppression des DF redondantes :

i. $F \rightarrow G$, $G \rightarrow B$, et $B \rightarrow E$, donc par transitivité $F \rightarrow B$ et $F \rightarrow E$ sont déductibles.

ii. $\{A, B\} \rightarrow G$ et $G \rightarrow D$, donc par transitivité $\{A, B\} \rightarrow D$ est déductible.

iii. $B \rightarrow F$, $F \rightarrow G$, donc par transitivité $B \rightarrow G$ est déductible.

iv. $B \rightarrow F$, $F \rightarrow G$ et $G \rightarrow D$, donc par transitivité $B \rightarrow D$ est déductible.

Résultat final :

$$F \rightarrow G$$

$$G \rightarrow B$$

$$B \rightarrow A$$

$$G \rightarrow D$$

$$B \rightarrow F$$

$$B \rightarrow E$$

$$F \rightarrow I$$

$$I \rightarrow M$$

Attention, s'il est vrai que l'on peut tout déduire à partir de F , il ne s'agit pas d'une couverture minimale. À partir de la couverture minimale, on doit pouvoir reconstituer toutes les dépendances de départ, ce qui n'est pas le cas avec "F donne tout le reste". En conséquence, une table contenant tous les attributs avec F en clé primaire ne serait pas en 3FN.

2. Suivant ces dépendances fonctionnelles, les relations suivantes sont-elles en 1ère, 2ème ou 3ème forme normale ? On considèrera que chaque attribut est mono-valué.

Correction : Toutes les relations sont au moins en 1FN car les attributs sont supposés mono-valués

(a) $R_1(\underline{F}, I, G, E, B)$

Correction : 1FN car pas 2FN car G dépend de F

(b) $R_2(\underline{G}, D, B)$

Correction : 3FN

(c) $R_3(\underline{F}, I, M)$

Correction : 2FN car pas 3FN car M dépend de I

(d) $R_4(\underline{B}, A, F, E, I)$

Correction : 2FN car pas 3FN car I dépend de F

4 Additifs (40 minutes)

Un produit alimentaire contient des additifs, dont certains sont nocifs pour la santé. Chaque produit est conçu et vendu par une marque.

Le modèle relationnel SQL associé à cette situation est présenté ci-dessous :

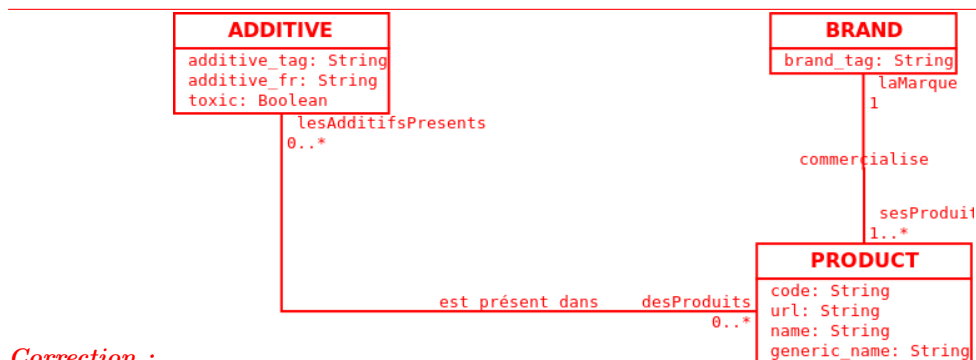
```
CREATE TABLE additive (
    additive_tag VARCHAR(10) PRIMARY KEY,      # identifiant
    additive_fr VARCHAR(50) NOT NULL UNIQUE,    # nom complet en français
    toxic        BOOLEAN NOT NULL              # nocif ou pas
);

CREATE TABLE brand (
    brand_tag VARCHAR(50) PRIMARY KEY          # identifiant
);

CREATE TABLE product (
    code        VARCHAR(13) PRIMARY KEY,        # code barre
    url         VARCHAR(256) UNIQUE,            # URL
    name        VARCHAR(256) NOT NULL,          # nom complet en français
    generic_name VARCHAR(256),                  # nom générique
    brand_tag    VARCHAR(50),                   # marque
    FOREIGN KEY (brand_tag) REFERENCES brand(brand_tag) ON DELETE RESTRICT
);

CREATE TABLE product_additive (
    product_code VARCHAR(13) NOT NULL,
    additive      VARCHAR(10) NOT NULL,
    PRIMARY KEY (product_code, additive),
    FOREIGN KEY (product_code) REFERENCES product(code),
    FOREIGN KEY (additive) REFERENCES additive(additive_tag)
);
```

1. Dessinez un diagramme UML correspondant au modèle relationnel décrit par ce code SQL.



Correction :

2. Répondez par vrai ou faux aux assertions suivantes. **Expliquez.**

- (a) Tous les produits doivent avoir une URL et un nom uniques.

Correction : Faux (URL UNIQUE mais nom pas forcément).

- (b) Plusieurs marques peuvent commercialiser le même produit.

Correction : Faux (Association BRAND-PRODUCT avec cardinalité une à plusieurs).

- (c) Si on supprime une marque de la base, tous les produits associés à cette marque seront automatiquement supprimés pour maintenir le respect de la contrainte d'intégrité.

Correction : *Faux ("ON DELETE RESTRICT" indique que cette suppression sera impossible si des produits sont associés)*

- (d) Si on supprime un produit de la base, toutes les instances associées à ce produit dans la table PRODUCT_ADDITIVE seront automatiquement supprimées pour maintenir le respect de la contrainte d'intégrité.

Correction : *Faux ("ON DELETE RESTRICT" est la consigne par défaut)*

3. À quelle question répondent les requêtes suivantes ?

- (a) `SELECT COUNT(*) from additive;`

Correction : *Combien y a-t-il d'additifs en tout ?*

- (b) `SELECT name FROM PRODUCT WHERE brand_tag = "marks-spencer";`

Correction : *Quels sont les produits de la marque "marks-spencer" ?*

- (c) `SELECT PA.additive
FROM PRODUCT P JOIN PRODUCT_ADDITIVE PA ON P.code = PA.product_code
WHERE P.name="Ginger snaps" AND P.brand_tag = "marks-spencer";`

Correction : *Quels sont les identifiants des additifs du produit nommé "Ginger snaps" de la marque "Marks & Spencer" ?*

- (d) `SELECT P.name, COUNT(*)
FROM PRODUCT P JOIN PRODUCT_ADDITIVE PA
ON P.code = PA.product_code WHERE P.brand_tag="kinder" GROUP BY P.name;`

Correction : *Dans chaque produit de la marque "kinder", combien y a-t-il d'additifs ?*

- (e) `SELECT AVG(nombre) FROM (
 SELECT COUNT(*) as nombre FROM PRODUCT P
 JOIN PRODUCT_ADDITIVE PA ON P.code = PA.product_code
 WHERE P.brand_tag="lu" GROUP BY P.code
);`

Correction : *Combien y a-t-il d'additifs en moyenne dans les produits de la marque "lu" ?*

- (f) `SELECT MAX(nombre) FROM (
 SELECT COUNT(*) as nombre FROM PRODUCT P
 JOIN PRODUCT_ADDITIVE PA ON P.code = PA.product_code GROUP BY P.code
);`

Correction : *Quel est le nombre maximum d'additifs présents dans un même produit, toutes marques confondues ? Attention, plusieurs produits peuvent avoir le même nom, y compris dans la même marque.*

```
(g) SELECT * FROM PRODUCT WHERE code IN (
      SELECT P.code
      FROM PRODUCT P JOIN PRODUCT_ADDITIVE PA ON P.code = PA.product_code
      GROUP BY P.code HAVING COUNT(*) = (
        SELECT MAX(nombre) FROM (
          SELECT COUNT(*) as nombre FROM PRODUCT P
          JOIN PRODUCT_ADDITIVE PA ON P.code = PA.product_code
          GROUP BY P.code
        )
      )
    );
```

Correction : Quels sont les produits qui contiennent ce nombre maximal d'additifs ?

```
(h) SELECT PA.additive, A.additive_fr, COUNT(*)
FROM PRODUCT_ADDITIVE PA JOIN ADDITIVE A ON PA.additive = A.additive_tag
GROUP BY additive HAVING COUNT(*) > 1000
ORDER BY COUNT(*);
```

Correction : Quels sont les additifs utilisés dans plus de 1000 produits ?
