

Collectif francophone pour l'enseignement libre de l'informatique

Modélisation, conception et exploitation de bases de données

Auto-évaluation de niveau 1

MCED_N1_AE

Christina KHNAISSER (christina.khnaisser@usherbrooke.ca) Luc LAVOIE (luc.lavoie@usherbrooke.ca)

(les auteurs sont cités en ordre alphabétique nominal)

CoFELI/Scriptorum/AE_01-N1, version 1.0.2.b AVEC RÉPONSES, en date du 2025-08-29 — version applicable au 2025-08-30 —

Sommaire

Le présent test vise à évaluer les acquis en modélisation, conception et exploitation de données (MCED).

Historique

diffusion	resp.	description
2025-08-29	LL	Reformulation de certaines réponses grâce à l'inclusion d'extraits des modules de MCED.
2025-08-21	LL	Corrections mineures.
2025-01-09	LL	Corrections mineures.
2024-09-15	CK	Ajout de la question facultative sur les ontologies.
2024-04-15	LL	Adaptation au cadre du CoFELI.
2023-08-15	LL	Version initiale utilisée en UdeS_IGE487_2023-1.

Table des matières

1. Préambule	4
1.1. Processus	4
1.2. Barèmes et critères	4
2. Questions	5
2.1. Théorie relationnelle	5
2.2. Modélisation logique sur la base de la théorie relationnelle	6
2.3. Modélisation conceptuelle	7
2.4. Information, connaissances et données	9
2.5. Langage SQL	10
2.6. Interprétation SQL	13
2.7. Programmation SQL	14
A Annexe	16

1. Préambule

Le présent test vise à évaluer les acquis élémentaires en modélisation, conception et exploitation de données (MCED) couverts par les activités de niveau 1 telles que IFT187 (Sherbrooke) ou INFO221 (Douala). Ces acquis sont préalables aux activités de niveau 2 telles que IGE487 (Sherbrooke) et INFO321 (Douala).

Le test porte donc essentiellement sur la théorie relationnelle, la modélisation de données, le langage SQL et les systèmes de gestion de bases de données relationnelles (SGBDR).

Le test a été conçu comme une évaluation formative, réalisée d'abord de façon autonome, puis collaborative.

1.1. Processus

Le processus d'évaluation comporte quatre étapes.

Première étape [requise, en solo, devrait être réalisée en quatre heures ou moins]

- 1. Répondre au mieux à l'ensemble des questions, sans consulter de références, ni recourir à de l'aide externe. Conserver ce premier jeu de réponses. [Moins de deux heures].
- 2. Améliorer les réponses en recourant de façon autonome à des références externes (manuel, notes de cours, sites, etc.). Conserver ce deuxième jeu de réponses. [Moins d'une heure].
- 3. Évaluer chacune des réponses de chacun des jeux à l'aide du solutionnaire à être transmis par la personne enseignante et du barème présenté ci-après. [Moins d'une heure].

Deuxième étape [fortement recommandée, en équipe de 2 à 4 personnes, moins de deux heures]

- 4. Discuter des questions qui demeurent mal comprises ou problématiques. Suite aux discussions, corriger les réponses au besoin et produire un troisième jeu de notes.
- 5. Saisir les trois jeux de notes individuelles selon les modalités prescrites par la personne enseignante.

Troisième étape [facultative, en classe après compilation des notes, moins d'une heure]

- 6. Prendre connaissance des sources et outils complémentaires proposés par la personne enseignante.
- 7. À partir des indications données par la personne enseignante, évaluer le niveau d'atteinte des acquis et, si nécessaire, établir un plan de formation complémentaire et le volume de travail induit.

Quatrième étape [facultative, en solo ou en équipe, durée variable]

8. Prendre rendez-vous avec la personne enseignante afin de confirmer les évaluations et le choix des mesures compensatoires.

1.2. Barèmes et critères

Chaque réponse doit être notée:

- 2, pour une bonne réponse (parfois, il n'y a pas qu'une bonne réponse);
- 1, pour une réponse partielle ou marginalement incorrecte, mais suffisante pour qu'elle puisse être auto-complétée ou auto-corrigée après prise en compte de références adéquates;
- 0, pour une réponse insuffisante.

L'évaluation repose sur deux critères:

- atteinte d'un seuil minimal requis (seuil R) dans chacune des sept rubriques;
- atteinte d'un seuil global (seuil G) pour l'ensemble des questions.

Les seuils seront communiqués par la personne enseignante à la troisième étape.

2. Questions

2.1. Théorie relationnelle

TR01

Quel rapport y a-t-il entre la logique et la théorie relationnelle?

▼ Réponse

La théorie relationnelle (TR) vise à établir une équivalence entre les relations algébriques et les prédicats logiques. Les modèles dérivés de la TR ont généralement pour but de faciliter l'automatisation du calcul des prédicats en suivant un système logique particulier, notamment:

- la logique dite classique (logique de Boole, bi-valuée);
- une logique tri-valuée (dont celles de Kleene, de Priest ou de SQL);
- une logique tetra-valuée (dont celles Belnap ou de Codd).

TR02

Quelles sont les concepts fondamentaux de la théorie relationnelle?

V Réponse

Sur la base des définitions de valeur et de type postulées par la Théorie des types, la Théorie relationnelle définit les concepts suivants:

- l'attribut (un couple formé d'un identifiant et d'un type),
- le tuple (un ensemble d'attributs dénotant une proposition logique),
- la relation (un ensemble de tuples dénotant un prédicat logique),
- la base (un ensemble de variables contraintes par une expression logique).

Le module TMR_02-Fondements, disponible sur le site de cours, offre une présentation plus complète et formelle de la théorie relationnelle et de certains de ses modèles.

TR03

Quels sont les opérateurs primitifs de l'algèbre relationnelle?

V Réponse

- Opérations ensemblistes : union « | | », intersection « ∩ » et différence « ».
- Opérations relationnelles: projection $\langle \pi \rangle$, restriction $\langle \sigma \rangle$, jointure $\langle \bowtie \rangle$.
- Opération structurelle: renommage «ρ».

Notes

- La restriction est souvent nommée «sélection», bien que cela puisse entrainer une confusion avec l'opérateur SELECT de SQL qui ne lui est pas équivalent.
- L'intersection peut être omise puisqu'elle s'obtient à partir de la différence : $R \cap S = (R (R S))$.
- Certains préfèrent inclure le produit cartésien ensembliste «×» en lieu et place de la jointure relationnelle qui s'obtient dès lors en combinant le produit cartésien et la restriction :

$$R \bowtie S = (soit \ C := def(R) \cap def(S) : (R \times S) \ \sigma(\land [c \in C](R.c = S.c))).$$

• Inversement, le produit cartésien peut s'exprimer comme une simple jointure entre deux relations n'ayant pas d'attributs communs:

ANTE
$$id(R) \cap id(S) = \{\} : R \bowtie S = R \times S.$$

• En théorie, il est possible de réduire le nombre d'opérations du noyau algébrique à deux opérations, mais ce n'est guère pratique, car peu expressif et peu lisible.

Une présentation plus complète et formelle est disponible dans le module TMR_02-Fondements, disponible sur le site de cours.

2.2. Modélisation logique sur la base de la théorie relationnelle

ML01

Qu'est-ce que la première forme normale (1FN)? En quoi est-elle importante?

V Réponse

Une relation est en 1FN si et seulement si toute valeur associée à un attribut est du type associé à cet attribut. Cette propriété simplifie l'expression des dépendances fonctionnelles et contribue à établir la validité *a priori* de toute expression (proposition, prédicat, contrainte, assertion, requête, etc.) relative au modèle (à la base de données) dont fait partie la relation.

Pratiquement, elle permet de formuler toute expression en termes des seules valeurs légitimes sans devoir expliciter le traitement des valeurs illégitimes.

ML02

Qu'est-ce que la forme normale de Boyce-Codd (FNBC)? En quoi est-elle importante?

▼ Réponse

Une relation est en FNBC si et seulement si toutes ses dépendances fonctionnelles applicables non triviales sont issues d'une de ses clés. Cette propriété réduit considérablement la redondance de données (les informations redondantes étant susceptibles d'être modifiées de façon incohérente), facilitant ainsi l'atteinte de la validité, de la cohérence, de l'efficacité et de l'efficience du modèle (de la base de données) dont fait partie la relation.

ML03

Qu'est-ce que la cinquième forme normale (5FN)? En quoi est-elle importante?

V Réponse

Une relation R est en cinquième forme normale (5FN) si et seulement si chacune des dépendances de projection-jointure qui lui sont applicables est induite par une des clés de R.

Cette propriété garantit que, lors d'une décomposition par projection-jointure, suivie d'une recomposition, aucun tuple ne sera ni perdu ni ajouté. Cette propriété est nécessaire afin de garantir la validité et la cohérence du modèle (de la base de données) dont fait partie la relation.

En pratique, trois théorèmes facilitent grandement l'application de la 5FN :

- Une relation comportant une seule clé stricte est en 5FN si et seulement si la seule dépendance de projection-jointure applicable est induite par la clé.
- Toute relation en 3FN (*a fortiori* en FNBC) dont au moins un attribut ne participe à aucune clé stricte de R est en FNPJ.
- Toute relation en FNBC dont tous les attributs en sont une clé stricte est en FNPJ.

Note

- 5FN \Rightarrow FNBC \Rightarrow 1FN.
- En théorie, il suffit donc de monter la 5FN pour obtenir la FNBC et la 1FN.
- En pratique, il est souvent plus efficient de procéder par étapes :
 - ° d'abord, typer adéquatement les attributs (et obtenir la 1FN),

- ° puis, s'assurer que toutes les dépendances fonctionnelles sont issues des clés (et obtenir la FNBC),
- ° enfin, vérifier la trivialité de dépendance de projection-jointure des relations totales non singulières (souvent très peu nombreuses) pour établir la 5FN.

ML04

Qu'est-ce que la sixième forme normale (6FN)?

En quoi est-elle importante?

V Réponse

Une relation est en 6FN si et seulement si, quelle que soit la DJ à laquelle elle satisfait, cette dépendance est triviale.

Cette propriété est nécessaire afin de permettre l'historicisation indépendante de chacun des DJ.

Note

• 6FN \Rightarrow 5FN \Rightarrow FNBC \Rightarrow 1FN.

ML05

Quelles sont les principales stratégies de modélisation des données manquantes? Quels sont leurs avantages et inconvénients respectifs?

V Réponse

Attendu un modèle (une base de données) en 5FN, les deux principales stratégies sont:

- la décomposition relationnelle (DR) par projection-jointure (DPJ) ou par restriction-union (DRU);
- la mise en treillis des types (MTT).

Les principaux avantages de la DR sont:

- le maintien de la logique classique en support au raisonnement relatif au modèle;
- la différenciation entre donnée manquante et attribut non applicable;
- la capacité de modéliser les causes des manques.

Le principal inconvénient de la DR est:

• l'augmentation du nombre de relations requises pour le modèle et la complexification corolaire de certaines requêtes.

Le principal avantage de la MTT est:

• le maintien de la structure et du nombre de relations requises par le modèle.

Les principaux inconvénients de la MTT sont:

- le recours à une logique modale au sein du modèle et la complexification corolaire des raisonnements comme de leur expression;
- l'incapacité de la MTT de modéliser la différence entre donnée manquante et attribut non applicable... sans avoir recours à une logique comportant au moins quatre valeurs;
- l'incapacité (à elle seule) de modéliser les causes des manques... sans avoir recours à la DR.

2.3. Modélisation conceptuelle

MC01

Quel est le rôle de la modélisation conceptuelle?

V Réponse

Le modèle conceptuel (MC) est un modèle intermédiaire entre le modèle de connaissances et le modèle

logique. Le MC est généralement élaboré afin de faciliter :

- la communication entre les analystes du domaine et les analystes informatiques;
- l'établissement de la liste des entités (classes) et des attributs pertinents au problème ;
- l'établissement de la liste des associations (relations, axiomes) entre les entités;
- la détermination des contraintes propres au contexte (métier, organisation, usages, etc.);
- l'établissement d'un niveau d'abstraction tel que toutes les notions requises par le domaine sont couvertes, mais seulement celles-là.

MC02

Quelles sont les caractéristiques les plus importantes dans le choix d'une méthode de modélisation conceptuelle?

V Réponse

Parmi les nombreuses caractéristiques souhaitables, les plus importantes sont vraisemblablement:

- l'accessibilité des concepts et de la notation en regard des parties prenantes participantes;
- la capacité de traduire les artéfacts en un modèle logique équivalent, et ce, aussi automatiquement que possible.

MC03

Quelles sont les caractéristiques les plus importantes dans le choix d'une notation de modélisation conceptuelle?

▼ Réponse

Parmi les nombreuses caractéristiques souhaitables, les plus importantes sont vraisemblablement :

- la couverture du méta-modèle conceptuel;
- la non-ambigüité des artéfacts produits;
- la capacité de garder la trace et la motivation des changements.

MC04

Qu'est-ce qui distingue la modélisation conceptuelle par association (« entité-association) de celle par classe (« orientée objets »)?

V Réponse

En modélisation par association, les entités sont identifiées par des clés alors qu'en modélisation par classes, les entités (objets) sont identifiées par leur emplacement (adresse). La différence de définition de plusieurs des opérations applicables en découle.

MC05

La modélisation conceptuelle est-elle une étape obligatoire entre le problème et la solution adéquate (ici un modèle logique de données)?

V Réponse

Non.

À preuve, il est possible de passer directement d'un modèle de connaissances (par exemple un modèle ontologique) à un modèle logique. Il est également possible de dériver directement un modèle logique de l'énoncé de problème, de la spécification des exigences ou des entretiens avec les analystes du domaine.

2.4. Information, connaissances et données

IC01

Quel est le lien entre l'information, les connaissances et les données?



information

Élément de connaissance susceptible d'être transmis au moyen d'une suite de signes.

Élément de connaissance représentable par une donnée.

connaissance

- (1) Faculté mentale produisant une assimilation par l'esprit d'un contenu objectif; résultat de cette opération.
- (2) Traduction d'un contenu objectif en signes.

La connaissance est une possession symbolique des choses; elle comprend un nombre arbitraire de degrés. La connaissance (assimilation rationnelle, méthodique et universelle) est parfois opposée au savoir (assimilation empirique et chaotique).

donnée

Valeur associée à une représentation (donc apte à être traitée par ordinateur).

représentation

Au sens général, « Action de rendre sensible quelque chose au moyen d'une figure, d'un symbole, d'un signe. Cette figure, ce symbole, ce signe. » [Antidote, 2025-08-25]

Dans un contexte informatique, on distingue une représentation interne (une suite de signaux) d'une représentation externe (une suite de signes).

signe

«Unité linguistique constituée par l'association d'une forme sonore ou graphique (signifiant) et d'un contenu conceptuel (signifié). » [Larousse 2025-08-25]. Syn. symbole.

signal

Phénomène physique mesurable, donc suffisamment stable pour être acquis et converti en signe.

IC02

La méthode de caractérisation des problèmes d'information dite des 3V repose sur trois critères; elle a été étendue à cinq critères (5V) puis à huit critères (8V). Quels sont-ils?

T Réponse

Les quatre propriétés classiques

volume

° quantité de données devant être stockées;

variété

° diversité et de la complexité des types utilisés par le modèle;

vélocité

° débit de données entrantes et sortantes :

véracité

° marge d'erreur, incertitude et inexactitude des données.

Les quatre propriétés complémentaires

valeur

° richesse analytique des données, mais aussi valeur économique, politique, sociale ou sociétale;

· variabilité

° relativité des données en regard du temps, de l'espace et de la perspective des agents;

virtualité

° distribution des sources de données;

vertu

° lois et règlements applicables à la protection et l'utilisation des données; principes de gouvernance éthique des données.

IC03

L'ingénierie des exigences consiste à déterminer (et spécifier) les exigences applicables à une solution adéquate en regard d'un problème. Proposer une définition opérationnalisable de l'adéquation?

V Réponse

Un traitement (un programme, une application, un service, une logithèque, un modèle de données, une base de données, etc.) est adéquat s'il répond aux huit propriétés suivantes:

Propriétés absolues

- 1. Cohérence (en regard de la théorie)
- 2. Validité (en regard du modèle)
- 3. Efficacité (en regard du traitement)

Propriétés relatives

- 4. Complétude (subordonnée à la cohérence)
- 5. Efficience (subordonnée à la validité et l'efficacité)
- 6. Évolutivité (subordonnée à l'efficience)

Méta-propriétés

- 7. Réfutabilité
- 8. Acceptabilité (éthique)

Règles de pratique

Les propriétés absolues découlent (essentiellement?) des exigences fonctionnelles.

Les propriétés relatives découlent (principalement?) des exigences non fonctionnelles.

Les meta-propriétés découlent de l'épistémologie et de l'éthique.

Les huit propriétés doivent être confirmées par

- la preuve (si possible),
- les tests (selon les moyens) et
- la revue (dans tous les cas).

Réflexion

L'inclusion de l'acceptabilité porte parfois à débat. Elle semble toutefois difficilement contournable, tant en biologie qu'en informatique, étant donné le lien direct entre certaines de leurs sous-disciplines (biologie humaine, écologie, systèmes d'information, intelligence artificielle, etc.) et l'espèce humaine ainsi que l'impact potentiel sur ses individus. N'en serait-il pas de même pour la physique (nucléaire, par exemple) et la chimie (pétrolière, par exemple)? Les mathématiques y échappent-elles lorsque les statistiques en ont été séparées?

Bonne réflexion et bon débat!

2.5. Langage SQL

LS01

Qu'est-ce qu'un schéma (CREATE SCHEMA)?

V Réponse

Un schéma est un ensemble de définitions SQL doté de son propre espace d'identifiants au sein d'une base de données. Le schéma est particulièrement utile pour mettre en oeuvre le concept de module et pour permettre l'échange et la réutilisation de composants de modélisation. Il est également le support privilégié pour structurer le contrôle d'accès.

LS02

Qu'est-ce qui distingue un type (CREATE TYPE) d'un domaine (CREATE DOMAIN)?



Selon la théorie des types

Le type est un ensemble fini de valeurs. Un type de base est un type dont les valeurs lui sont propres (une valeur appartenant à un type de base n'appartient à aucun autre type de base). Le sous-type définit un sous-ensemble (de valeurs) d'un (autre) type par ailleurs défini. La portée du sous-ensemble est prescrite par une contrainte (une expression booléenne) sur les valeurs du type de base.

Une valeur d'un type de base est forcément incompatible avec celles d'un autre type de base. Une valeur d'un sous-type est forcément compatible avec celles du type de base à partir duquel le sous-type a été (directement ou indirectement) défini.

En SQL

Un TYPE est un type de base.

Un DOMAIN est un sous-type.

La fonction CAST permet de faire des conversions explicites entre valeurs de types différents.

Tous les dialectes ne proposent pas les constructeurs CREATE TYPE et CREATE DOMAIN.

Tous les dialectes ajoutent des exceptions (incohérentes avec la théorie des types) à celles déjà nombreuses formulées par le standard ISO!

LS03

Définir la contrainte référentielle (FOREIGN KEY).

▼ Réponse

Une clé (d'une relation R) est un sous-ensemble d'attributs déterminant un tuple unique au sein de la relation.

Autrement dit, soit X une clé de R:

```
\#R = \#(R \pi X)
```

Une clé est dite stricte (ou *candidate* en anglais)si (et seulement si) aucun attribut ne peut être retiré sans qu'elle perde sa propriété de clé.

Une clé référentielle (d'une relation R en regard d'une relation S) est un sous-ensemble d'attributs de R dont les valeurs sont restreintes aux seules valeurs effectives d'une clé de la relation S.

Autrement dit, soit X une clé référentielle de R sur S:

```
((R \pi X) \subseteq (S \pi X)) \wedge (\#S = \#(S \pi X))
```

Rappel

• π désigne la projection et # la cardinalité.

Note

- SQL distingue deux catégories de clés strictes, **la** clé primaire (PRIMARY KEY) et **les** clés secondaires (UNIQUE).
- Il y a toujours une clé primaire, il peut ne pas y avoir de clé secondaire.

LS04

Qu'est-ce qui distingue une vue (CREATE VIEW) d'une fonction (CREATE FUNCTION)?

V Réponse

La vue est essentiellement une relation définie (en compréhension) par une expression (SELECT). Une fonction est un traitement paramétrable qui a pour but de calculer une valeur sans modifier l'état (de la base de données). Lorsque le type de cette valeur est une relation, alors la fonction est une généralisation de la vue admettant le paramétrage.

Note

- 1. La vérification que la fonction ne modifie pas l'état de la base de données n'est généralement pas prise en charge par les SGBDR.
- 2. La table est essentiellement une relation définie (en extension) par l'ensemble de ses tuples.

LS05

Qu'est-ce qui distingue une fonction (CREATE FUNCTION) d'une procédure (CREATE PROCEDURE)?

V Réponse

La fonction est un traitement qui a pour but de calculer une valeur sans modifier l'état de la base de données. La procédure est un traitement qui a pour but de modifier l'état du SGBDR. Les deux sont paramétrables.

LS06

Quels sont les principaux usages des automatismes (CREATE TRIGGER)?

V Réponse

- 1. Mettre en oeuvre les assertions (CREATE ASSERTION) lorsqu'elles ne font pas partie du dialecte mis en oeuvre par le SGBDR.
- 2. Mettre en oeuvre les commandes d'insertion (INSERT), de retrait (DELETE) et de modification (UPDATE) applicables aux vues.
- 3. Mettre en oeuvre un automatisme (traitement déclenché par une modification de l'état de la base de données), plutôt que par un appel explicite (comme pour les procédures et les fonctions).

LS07

Quelle stratégie de traitement des données manquantes le langage SQL utilise-t-il? Comment se compare-t-elle avec les stratégies relationnelles recommandées?

V Réponse

Les concepteurs de SQL ont choisi une stratégie mixte alliant MTT (mise en treillis des types) et DR (composition relationnelle).

Afin de permettre le traitement des données strictement manquantes, SQL adopte une structure de type inspiré des semi-treillis en ajoutant à tous les types une valeur (NULL) se comportant parfois comme un

infimum, parfois comme un supremum. De cette structure découle l'obligation d'utiliser une logique à trois valeurs (FALSE, TRUE, UNKNOWN), mais les instructions WHERE et CHECK n'utilisent pas la même logique, ajoutant une complexité inutile à la solution.

Pour les attributs non applicables (NA), SQL ne propose aucun mécanisme particulier. Il faut donc avoir recours à la modélisation et à la décomposition (soit de projection-jointure, soit de projection-union).

La situation se corse lorsqu'un attribut est parfois inapplicable, mais peut aussi comporter des valeurs manquantes. Dès lors, la solution la plus simple consiste le plus souvent à modéliser l'attribut non applicable (NA) et la valeur manquante (NULL) par modélisation (et décomposition).

Remarquons que la solution de modélisation par décomposition ne nécessite ni structure de semitreillis, ni valeur spéciale (NULL), ni logique à trois valeurs. En outre, cette solution permet de préserver la cause du manque (et non seulement le fait du manque). La solution mixte proposée par SQL est non seulement complexe, mais **incorrecte**: elle comporte des failles importantes découlant de l'utilisation de deux logiques (à trois valeurs) variant selon le contexte (WHERE et CONSTRAINT).

Les inconvénients sont très nombreux, dont ceux-ci:

- l'obligation d'avoir recours à **deux** logiques différentes dans les vérifications et les tests;
- le manque d'orthogonalité dans le traitement, ce qui occasionne une complexité conséquente;
- la grande difficulté d'automatiser la détection des erreurs de traitement en présence de données manquantes.

LS08

Pourquoi les opérations de jointure (JOIN, JOIN ON, JOIN USING...) ne sont-elles pas commutatives en SQL?

V Réponse

Parce que la structure du tuple est une liste ordonnée d'attributs en SQL, plutôt qu'un ensemble d'attributs, comme le postule la Théorie relationnelle. La jointure SQL induit donc un ordre différent selon que l'on joint A à B, plutôt que B à A. Les résultats ne sont donc pas égaux.

Note

En conséquence de quoi, les concepteurs de SQL ont été amenés à retirer l'opération d'égalité entre tuples et entre relations!!!

2.6. Interprétation SQL

En fonction du modèle et des données en annexe, pour chacune des requêtes suivantes:

- 1. décrire en une phrase ou deux ce qui est calculé;
- 2. calculer et transcrire le contenu de la table résultante.

IS01

```
SELECT DISTINCT id, nom
FROM Produit
WHERE marque = '3ITC'
```

IS02

```
SELECT produit, COUNT(DISTINCT acheteur)
FROM Transaction
GROUP BY produit
```

IS03

```
SELECT id, MAX(nom) AS n, SUM(quantite*prix)/SUM(quantite) AS m
FROM Produit JOIN Transaction ON (id=produit)
GROUP BY id
```

IS04

```
SELECT DISTINCT A.acheteur FROM Transaction AS B ON (A.acheteur=B.vendeur)
```

IS05

```
WITH
A AS
( SELECT acheteur AS id
FROM Transaction
GROUP BY acheteur
HAVING COUNT(DISTINCT produit) = 1
)
SELECT DISTINCT id, nom, ville
FROM Organisation JOIN A USING (id)
```

IS06

```
WITH

X AS (SELECT DISTINCT acheteur AS id FROM Transaction)

SELECT A.id, A.nom

FROM Organisation AS A JOIN X USING (id)

WHERE NOT EXISTS

( SELECT 1

FROM Organisation AS V join Transaction AS T ON (V.id=T.vendeur)

WHERE (A.id=T.acheteur) AND (V.ville='Montréal')

)
```

V Réponses

Voir le fichier source Mercure_script-2023-1_psql.sql.

2.7. Programmation SQL

En fonction du modèle de données en annexe, proposer une requête SQL pour chacune des demandes suivantes.

PS01

Quels sont les types de produits ayant engendré des achats dépassant 2000 CAD?

PS02

Ouelle est la valeur totale des achats du mois de décembre 2016?

PS03

Quelle est la valeur totale des achats de chaque organisation? Donner l'identifiant (id) de l'organisation et la valeur totale.

PS04

Les acheteurs dont la valeur totale des achats dépasse 20 000 CAD. Donner l'identifiant et la valeur totale des achats de chacun.

PS05

Majorer de 10 % le cout des produits achetés uniquement par des manufacturiers.

PS06

Supprimer les produits qui n'ont jamais été achetés.



Voir le la trace d'exécution Mercure_script-2023-1.txt.

A. Annexe

Le gouvernement du Québec désire assurer la traçabilité de certains produits sensibles sur son territoire. Pour ce faire, il a mis en place un mécanisme de déclaration des transactions portant sur ces produits et instauré un registre des organisations habilitées à effectuer ces transactions.

```
CREATE DOMAIN Produit_code AS
 VARCHAR(3) CHECK (length(VALUE)=3);
CREATE DOMAIN Montant AS
 NUMERIC(12,2) CHECK (VALUE >= 0);
CREATE TABLE Produit
-- Le produit «id» est désigné par le nom «nom», est de type «type», de marque «marque»
-- et dont le cout «cout» est donné en dollars canadiens (CAD).
-- Le cout est ici un cout unitaire moyen de référence auguel on peut comparer le
-- prix effectif consenti lors d'une transaction donnée.
(
            Produit_code NOT NULL,
 id
            VARCHAR(20)
                          NOT NULL,
 nom
            VARCHAR(20)
                          NOT NULL,
 type
 marque
            VARCHAR(20)
                          NOT NULL,
 cout
            Montant
                          NOT NULL,
 CONSTRAINT Produit_cc0 PRIMARY KEY (id)
);
CREATE DOMAIN Org_code AS
 VARCHAR(3) CHECK (length(VALUE)=3);
CREATE TABLE Organisation
-- L'organisation «id» est enregistrée; elle porte le nom «nom», a son siège social
-- dans la ville «ville» et offre des services de type «type».
 id
            Org_code
                        NOT NULL,
 nom
            VARCHAR(20) NOT NULL,
            VARCHAR(60) NOT NULL,
 ville
            VARCHAR(20) NOT NULL,
 CONSTRAINT Organisation_cc0 PRIMARY KEY (id),
 CONSTRAINT Organisation_cc1 UNIQUE (nom, ville)
);
CREATE TABLE Transaction
-- L'organisation «acheteur» a acquis auprès de l'organisation «vendeur» un nombre
-- «quantité» de produit «produit» au prix unitaire de «prix» en date du «date»;
-- le prix est donné en CAD.
-- La valeur totale d'un achat est donnée par quantite*prix.
(
 acheteur Org_code
                          NOT NULL,
 vendeur
            Org_code
                          NOT NULL,
            Produit_code NOT NULL,
 produit
                          NOT NULL,
           DATE
 date
 quantite NUMERIC(9)
                          NOT NULL,
                          NOT NULL,
            Montant
 CONSTRAINT Transaction_cc0 PRIMARY KEY (acheteur, vendeur, produit, date),
 CONSTRAINT Transaction_ce1 FOREIGN KEY (acheteur) REFERENCES Organisation (id),
 CONSTRAINT Transaction_ce2 FOREIGN KEY (vendeur) REFERENCES Organisation (id),
 CONSTRAINT Transaction_ce3 FOREIGN KEY (produit) REFERENCES Produit (id)
);
```

```
INSERT INTO Produit
   (id,
               nom,
                                    type,
                                                             marque,
                                                                              cout)
VALUES
   ('P31',
                'Câble #1',
                                    'Électricité',
                                                              'IOCT',
                                                                                 10),
   ('P32', 'Câble #2',
                                    'Électricité',
                                                              'IOCT',
                                                                                 20),
   ('P33', 'Câble #3',
('P34', 'Tôle',
('P35', 'Gouttière',
                                                              '3ITC',
                                    'Électricité',
                                                                                 50),
                                    'Toiture',
                                                              'Rano',
                                                                                 80),
                                    'Toiture',
                                                              'Rano',
                                                                                 30),
   ('P36', 'Circuit-A', 'Électronique', '3ITC', 10), ('P37', 'Circuit-B', 'Électronique', 'Ducharme', 20), ('P38', 'Circuit-C', 'Électronique', 'Ducharme', 40);
INSERT INTO Organisation
   (id,
               nom,
                                               ville,
                                                                          type)
VALUES
   ('ABC', 'Société ABC',
                                                'Montréal',
                                                                           'Manufacturier'),
   ('DEF', 'Compagnie DEF'
                                                'Victoriaville', 'Assembleur'),
   ('GHI', 'Entreprises GHI',
                                                                          'Manufacturier'),
                                                'Québec',
   ('JKL', 'Jos, Karl & Lou',
                                                'Montréal',
                                                                           'Grossiste'),
   ('MNO', 'Marie Normandeau',
                                               'Sherbrooke',
                                                                           'Detaillant'),
   ('PQR', 'PQR SARL',
                                               'Sherbrooke',
                                                                           'Utilisateur'),
   ('STU', 'Compagnie STU',
                                                'Montréal'.
                                                                           'Transformateur'),
   ('VWX', 'Entreprises VWX',
                                                'Drummondville', 'Assembleur');
INSERT INTO Transaction
   (acheteur, vendeur, produit, date, quantite, prix)
VALUES
  ('GHI', 'DEF', 'P38', DATE '2006-12-01', 500, 80), ('JKL', 'GHI', 'P32', DATE '2006-12-02', 400, 40), ('VWX', 'JKL', 'P31', DATE '2006-12-03', 100, 90), ('MNO', 'JKL', 'P32', DATE '2006-12-04', 200, 30), ('VWX', 'ABC', 'P36', DATE '2006-12-07', 25, 10), ('PQR', 'DEF', 'P37', DATE '2006-12-08', 50, 40), ('JKL', 'ABC', 'P36', DATE '2006-12-09', 25, 20), ('JKL', 'ABC', 'P37', DATE '2006-12-09', 50, 30);
```

Produit le 2025-08-29 06:24:28 -0400



Collectif francophone pour l'enseignement libre de l'informatique