

IFT 187 – Éléments de base de données

Examen #1 (formatif)

- ◇ Enseignant Luc Lavoie
- ◇ Émission 2020-10-03

Notes et consignes

- ◇ Cet examen d'une durée de 110 minutes est individuel. Il comporte cinq questions.
- ◇ La correction est basée, entre autres, sur le fait que chacune des réponses soit claire (c'est-à-dire lisible et compréhensible), exacte (c'est-à-dire précise et sans erreurs), complète (c'est-à-dire comprenant tous les éléments requis et tous les éléments nécessaires) et concise (c'est-à-dire sans éléments superflus).
- ◇ La documentation personnelle est limitée à une seule feuille recto verso au format lettre ou A4.
- ◇ Aucun équipement informatique, électronique ou de communication n'est autorisé.
- ◇ **Justifier les réponses.**
- ◇ **Remettre le questionnaire et le cahier avant de quitter la salle d'examen.**

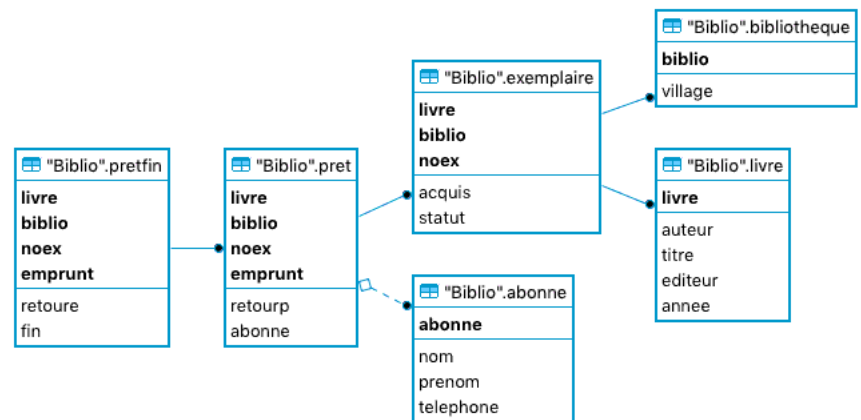
Pondération

Question	Q1	Q2	Q3	Q4	Q5	Total
Poids	/ 20	/ 30	/ 30	/ 10	/ 10	/ 100

1 Programmation SQL (LDD)

Mise en contexte

La société Bibliovik assure la gestion du réseau des bibliothèques sur le territoire du Nunavik. Elle désire conserver un inventaire des exemplaires de livre qu'elle possède, une liste de ses abonnés et un registre des emprunts. Un livre est décrit par un code, un auteur, un titre, un éditeur et une année de parution. Une bibliothèque est décrite par un code et le nom du village où elle se trouve. Chaque exemplaire est décrit par une référence au livre, une référence à la bibliothèque qui le possède, son numéro d'exemplaire, sa date d'acquisition et son statut (disponible, perdu, en restauration). Finalement, on consigne les abonnés (numéro d'abonné, nom, prénom, numéro de téléphone), les prêts (exemplaire, abonné emprunteur, date d'emprunt, date de retour prévu). Pour le suivi des emprunts, il faut en outre en conserver la fin (perdu, retourné) et la date.



Question

Écrire le script de création SQL des tables requises. Spécifier clairement le prédicat de chaque table. Justifier le choix du type des attributs, lorsqu'approprié. Traiter adéquatement les clés candidates, les clés référentielles et les contraintes requises. Indiquer clairement vos hypothèses.

2 Évaluation de requêtes SQL (LMD)

Mise en contexte

Soit le schéma Provik et les données en annexe.

Questions

Pour chacune des requêtes suivantes : (a) décrire en une phrase ce qui est calculé *et* (b) calculer et transcrire le contenu de la table résultante.

[2.1]

```
SELECT codeP, nom, cat
FROM Projet
WHERE budget >= 100
```

[2.2]

```
SELECT codeP, COUNT(DISTINCT NMP)
FROM Tache
GROUP BY codeP
```

[2.3]

```
SELECT Projet.codeP, COUNT(DISTINCT NMP)
FROM Projet LEFT JOIN Tache USING (codeP)
GROUP BY Projet.codeP
```

[2.4]

```
SELECT DISTINCT Projet.codeP, P.codeP
FROM Projet JOIN Projet AS P USING (debut)
WHERE (Projet.codeP < P.codeP)
```

[2.5]

```
WITH
A AS
( SELECT NMP
  FROM Membre
    JOIN Tache USING (NMP)
    JOIN Projet USING (codeP)
  GROUP BY NMP
  HAVING COUNT (DISTINCT Projet.cat) > 1
)
SELECT NMP, nom, prenom
FROM Membre JOIN A USING (NMP)
```

[2.6]

```
WITH
X AS
( SELECT DISTINCT NMP, codeP, sum(part) AS total
  FROM Tache JOIN Membre USING (NMP)
  GROUP BY NMP, codeP
)
SELECT DISTINCT X1.NMP
FROM X AS X1
WHERE NOT EXISTS
( SELECT *
  FROM X AS X2
  WHERE (X1.codeP = X2.codeP) AND (X1.NMP <> X2.NMP) AND (X1.total < X2.total)
)
```

3 Programmation de requêtes SQL (LMD)

Mise en contexte

Soit le schéma et les données en annexe.

Questions

Proposer une requête SQL pour chacune des demandes suivantes :

- [3.1] Quel est le nombre d'heures travaillées du mois de janvier 2016 (tous projets confondus) ?
Donner la valeur.
- [3.2] Quels sont les membres qui ont travaillé avant leur date d'embauche ?
Donner le NMP, le nom, le prénom de ceux-ci.
- [3.3] Quels sont les projets ayant consommé (jusqu'à présent) plus de 50 % de leur budget ?
Donner le code, le nom, la catégorie, le total consommé et le budget de chacun de projets.
- [3.4] Quels sont les projets ayant le plus grand nombre de tâches complétées ?
Donner le code, le nom, la catégorie et le nombre tâches.
- [3.5] Réduire le budget des projets de 2 heures si le budget original est de moins de 25 heures et de 15 % pour les autres.
- [3.6] Supprimer les projets dont la date de fin est dépassée et qui n'ont aucune tâche associée.

4 Modèle relationnel

Mise en contexte

Le modèle relationnel définit l'attribut comme une partie (identificateur, type).

Questions

- [4.1] Pourquoi est-il essentiel d'avoir (toujours) un identificateur ?
- [4.2] Pourquoi est-il essentiel d'avoir (toujours) un type ?

5 Intégrité référentielle

Mise en contexte

Les clés référentielles (étrangères) sont un mécanisme important de modélisation relationnelle. Elles reposent sur la définition préalable de clés candidates. La clé référentielle établit un lien entre les attributs d'une relation et la clé candidate d'une autre relation.

Question

- [5.1] Pourquoi un attribut annulable ne peut-il faire partie d'une clé qu'elle soit candidate ou référentielle ?

Annexe

La société Provik assure la gestion de projets sur le territoire du Nunavik. Elle désire faire le suivi tâches réalisées par les membres du personnel au sein de chacun des projets.

Un projet est caractérisé par un code (unique) de projet, un nom, une date de début, une date de fin, une catégorie et un budget (nombre d'heures de travail allouées). Les membres du personnel sont caractérisés par un numéro (unique), leur nom, leur prénom et leur date d'embauche. Chaque membre du personnel consigne à la fin de chaque journée ouvrée les tâches réalisées par projet en donnant le nombre d'heures consacrées et la description de la tâche.

Considérer les trois tables décrites par les énoncés suivants :

```
CREATE TABLE Membre (
  NMP          CHAR(4)          NOT NULL,      -- numéro de membre du personnel
  nom          VARCHAR(40)     NOT NULL,
  prenom       VARCHAR(40)     NOT NULL,
  embauche     DATE            NOT NULL,      -- date d'embauche
  CONSTRAINT Membre_cc0      PRIMARY KEY (NMP),
  CONSTRAINT Membre_NMP     CHECK (NMP SIMILAR TO 'E[0-9]{3}'),
  CONSTRAINT Membre_nom     CHECK (nom SIMILAR TO '[:alpha:][:space:]*-+'),
  CONSTRAINT Membre_prenom  CHECK (prenom SIMILAR TO '[A-Za-z][:space:]*-+')
);
CREATE TABLE Projet (
  codeP        CHAR(4)          NOT NULL,      -- code de projet
  nom          VARCHAR(32)     NOT NULL,
  budget       NUMERIC(6)      NOT NULL,      -- budget total en heures
  cat          CHAR(3)         NOT NULL,
  debut        DATE            NOT NULL,
  fin          DATE            NOT NULL,
  CONSTRAINT Projet_cc0     PRIMARY KEY (codeP),
  CONSTRAINT Projet_codeP  CHECK (codeP SIMILAR TO 'P[0-9]{3}'),
  CONSTRAINT Projet_nom    CHECK (nom SIMILAR TO '[A-Za-z0-9][:space:]*-+'),
  CONSTRAINT Projet_cat    CHECK (cat SIMILAR TO '[A-Za-z0-9]{3}'),
  CONSTRAINT Projet_budget CHECK (1<=budget),
  CONSTRAINT Projet_dates  CHECK (debut<fin) --> ou (debut<=fin)
);
CREATE TABLE Tache (
  NMP          CHAR(4)          NOT NULL,
  codeP        CHAR(4)          NOT NULL,
  jour         DATE            NOT NULL,
  part         NUMERIC(2)      NOT NULL,      -- participation en heures
  description   TEXT           NOT NULL,
  CONSTRAINT Tache_cc0     PRIMARY KEY (NMP, codeP, jour),
  CONSTRAINT Tache_ce1    FOREIGN KEY (NMP) REFERENCES Membre,
  CONSTRAINT Tache_ce2    FOREIGN KEY (codeP) REFERENCES Projet,
  CONSTRAINT Tache_part   CHECK (part BETWEEN 1 AND 16)
  -- un membre ne peut travailler plus de 16 heures par jour
);
```

Initialement, les données sont les suivantes :

Membre			
nmp	nom	prenom	embauche
E001	Smith	Paul	2015-03-01
E002	Dupont	Jean	2015-01-01
E003	Durand	Maude	2014-01-01
E004	Clark	Joe	2013-08-01
E005	El Khoury	Alexandre	2015-06-01

Projet					
codep	nom	budget	cat	debut	fin
P001	Portail	100	dev	2014-01-01	2016-06-30
P002	Boutique	50	exp	2015-01-01	2016-06-30
P003	Velox	20	dev	2016-01-01	2016-06-30
P004	Velox	160	maj	2017-01-01	2017-06-30
P005	Plox	1000	dev	2017-01-01	2018-08-30
P006	Redox	10	maj	2017-06-01	2017-06-30

Tâche				
nmp	codep	jour	part	description
E001	P001	2014-01-01	3	anomalie #03
E001	P002	2015-01-01	12	test #05
E001	P003	2016-01-01	4	conception IRT
E001	P004	2017-01-01	12	anomalie #12
E001	P005	2017-01-01	11	documentation
E001	P005	2017-01-02	11	suivi
E002	P001	2014-01-02	8	anomalie #33
E002	P002	2015-01-02	4	test #33
E003	P002	2016-01-02	12	suivi
E004	P002	2016-01-02	7	test #12
E004	P004	2017-01-02	3	documentation
E004	P005	2017-01-02	4	suivi

Fin de l'annexe