

Informatique tronc commun

Devoir n° 04

Première partie, sur papier

10 juin 2017

Un professeur d'informatique a créé une base de données pour gérer les notes de ses interrogations hebdomadaires. Pour cela, il a créé trois tables : `etudiants`, `interros`, `notes`. Voici les commandes SQL ayant permis de créer ces tables.

```
CREATE TABLE etudiants (  
    -- table des données des étudiants  
    id INTEGER, -- identifiant de l'étudiant  
    nom VARCHAR NOT NULL, -- nom de l'étudiant  
    prenom VARCHAR NOT NULL, -- prénom de l'étudiant  
    date_naissance DATE NOT NULL, -- date de naissance de l'étudiant, format AAAA-MM-JJ  
    PRIMARY KEY (id)  
);  
  
CREATE TABLE interros (  
    -- table des données des interros  
    id INTEGER, -- identifiant de l'interro  
    titre VARCHAR, -- titre de l'interro  
    sujet VARCHAR, -- sujet de l'interro  
    date DATE, -- date du jour où a été donnée l'interro, format AAAA-MM-JJ  
    PRIMARY KEY (id)  
);  
  
CREATE TABLE notes (  
    -- table des notes des étudiants aux interros  
    id_etudiant INTEGER NOT NULL, -- identifiant de l'étudiant  
    id_interro INTEGER NOT NULL, -- identifiant de l'interro  
    note INTEGER NOT NULL, -- note obtenue  
    PRIMARY KEY (id_etudiant, id_interro),  
    FOREIGN KEY (id_etudiant) REFERENCES etudiants,  
    FOREIGN KEY (id_interro) REFERENCES interros  
);
```

Q1 Peut-on donner plusieurs notes au même étudiant et pour la même interrogation ?

Q2 Donner une requête SQL traduisant l'opération suivante, exprimée dans le vocabulaire d'algèbre relationnelle usuel :

$$\pi_{\text{sujet}}(\sigma_{\text{id}=1}(\text{interros})).$$

- Q3** Écrire une requête SQL permettant d'obtenir la liste des noms et prénoms des étudiants de la classe.
- Q4** Écrire une requête SQL permettant d'obtenir la liste des prénoms des étudiants de la classe, sans doublon.
- Q5** Les enfants du professeur se sont amusés à rentrer des données factices, que le professeur aimerait retrouver afin de les effacer ensuite. Écrire une requête SQL permettant d'obtenir la liste des identifiants des étudiants ayant pour nom "**reinedesneiges**".
- Q6** Écrire une requête SQL permettant d'obtenir la date de naissance de l'étudiant le plus jeune de la classe.
- Q7** Écrire une requête SQL permettant d'obtenir la liste des noms et prénoms des étudiants ayant obtenu au moins un 20 à une des interrogations.
- Q8** Écrire une requête SQL permettant d'obtenir la liste des noms, prénoms d'étudiants et titres d'interrogations pour chaque note de 0 obtenue.
- Q9** Écrire une requête SQL permettant d'obtenir la liste des noms et prénoms des étudiants, avec la moyenne des notes de chaque étudiant.
- Q10** Écrire une requête SQL permettant d'obtenir la liste des noms et prénoms des étudiants, suivis pour chaque étudiant du nombre d'interrogations rendues.
- Q11** Écrire une requête SQL permettant d'obtenir le nom, le prénom et le nombre de copies rendues par l'étudiant ayant rendu le plus de copies (on suppose qu'il n'y en a qu'un). On rappelle que l'instruction **LIMIT k** permet de tronquer une table à ses **k** premières lignes.
- Q12** Écrire une requête SQL permettant d'obtenir la liste des noms et prénoms des étudiants ayant rendu au moins 10 copies, suivis du nombre de copies rendues.
- Q13** Écrire une requête SQL permettant d'obtenir la liste des titres des interrogations, suivie pour chaque interrogation du nombre d'étudiants qui n'ont pas rendu de copies.
- Q14** Écrire une requête SQL permettant d'obtenir la liste des titres des interrogations pour lesquelles tous les étudiants ont rendu une copie.