T.P. II - B. d. d. Modifications, Jointures, Agrégats

I - Modification de bases de données

I.1 - Création d'une table

Définition 1 - CREATE TABLE ...

La requête `CREATE TABLE nom (col1 domaine1, col2 domaine2)` permet de créer la table `nom` dont les attributs sont `col1` et `col2` en spécifiant leur domaine.

∗ Pour créer la table `Entreprise` on pourrait utiliser :

```
CREATE TABLE Entreprise (
    NomEntreprise TEXT,
    Siren TEXT,
    AdresseEntreprise TEXT,
    CPEntreprise TEXT,
    VilleEntreprise TEXT,
    SecteurActivite TEXT,
    ContactEntreprise TEXT,
    PRIMARY KEY (Siren))
```

∗ Pour créer la table `Stagiaire` on pourrait utiliser :

```
CREATE TABLE Stagiaire (
    NirStagiaire TEXT,
    NomStagiaire TEXT,
    PrenomStagiaire TEXT,
    DatenaissanceStagiaire DATE,
    AdresseStagiaire TEXT,
    CPStagiaire TEXT,
    VilleStagiaire TEXT,
    TelStagiaire TEXT,
    SirenEntreprise TEXT,
    PRIMARY KEY (NirStagiaire),
    FOREIGN KEY (SirenEntreprise)
        REFERENCES Entreprise(Siren))
```

Exercice 1.

1. En utilisant la base des formations, écrire des requêtes permettant de créer :

a) la table `Formateur`.

b) la table `Anime`.

2. En utilisant la base des remontées mécaniques, écrire des requêtes permettant de créer :

a) la table `Clients`.

b) la table `Tarif_Forfait`.

I.2 - Modification d'une table

Définition 2 - Ajout d'un enregistrement

La requête `INSERT INTO table (col1, col2, ...) VALUES (val1, val2, ...)` permet d'insérer dans `table` l'enregistrement dont l'attribut `col1` vaut `val1`, l'attribut `col2` vaut `val2`,...

∗ Ajout d'un stagiaire :

```
INSERT INTO Stagiaire
(NirStagiaire, NomStagiaire, PrenomStagiaire,
  DatenaissanceStagiaire, AdresseStagiaire,
  CPStagiaire, VilleStagiaire, TelStagiaire,
  SirenEntreprise)
VALUES
("214053122002379", "Filine", "Agathe",
  2014-05-31, "3␣rue␣du␣boulot", "31500",
  "Toulouse", "06.25.56.48.75", "571261524")
```

∗ Ajout d'une entreprise :

```
INSERT INTO Entreprise
(NomEntreprise, Siren, AdresseEntreprise,
  CPEntreprise, VilleEntreprise,
  SecteurActivite, ContactEntreprise)
VALUES
("Lycée␣Saint␣Sernin", "193100419",
  "3␣place␣Saint␣Sernin", "31000", "Toulouse",
  "Education", "0310041b@ac-toulouse.fr")
```

Exercice 2.

1. En utilisant la base des formations, écrire des requêtes permettant d'ajouter :

a) une formation.

b) un inscrit à cette formation.

2. En reprenant la base des remontées mécaniques, écrire des requêtes permettant d'ajouter :

a) un client.

b) un badge associé à ce client.

c) un domaine skiable.

Définition 3 - Suppression d'un enregistrement

La requête `DELETE FROM table WHERE conditions` permet de supprimer tous les enregistrements de `table` qui satisfont `conditions`

∗ Suite à une erreur de saisie, pour supprimer toutes les formations dont la date de début est au 1er janvier 2023 :

```
DELETE FROM Formation
WHERE DateDebutFormation = 2023-01-01
```

∗ Suite à des avaries dans les locaux, pour supprimer toutes les formations ayant lieu à Agen :

```
DELETE FROM Formation
WHERE LieuFormation = "Agen"
```

∗ Suite à sa démission, pour supprimer toutes les formations assurées par le formateur dont le numéro de sécurité sociale est 1763104325012 :

```
DELETE FROM Anime
WHERE NirFormateur = "1763104325012"
```

Exercice 3.

1. En utilisant la base des formations, supprimer

a) les stagiaires nés le 30 février 1980 (entrés dans la base suite à une erreur de saisie).

b) l'entreprise dont le `Siren` est "571261524" car elle a fait faillite.

2. En utilisant la base des remontées mécaniques, supprimer

a) le(s) client(s) nommés "Dupont" et habitant à "Nemours"

b) les badges numéros "31415" et "2718"

Définition 4 - Modification des enregistrements

La requête `UPDATE table SET col1 = nvelle_valeur WHERE conditions` permet de modifier l'attribut `col1` en y attribuant la valeur `nvelle_valeur` pour tous les enregistrements de `table` qui satisfont les `conditions`.

∗ La nouvelle responsable des formations `Sécurité` sera `Marie` :

```
UPDATE Formation
SET ResponsableFormation = "Marie"
WHERE DesignationFormation = "Sécurité"
```

∗ L'entreprise de numéro `Siren` "571261524" déménage au "12 rue des congés" à "Castres" avec le code postal "81100"

```
UPDATE Entreprise
SET AdresseEntreprise = "12␣rue␣des␣congés",
    CPEntreprise = "81100",
    VilleEntreprise = "Castres"
WHERE Siren = "571261524"
```

Exercice 4.

1. En utilisant la base des formations,

a) décaler toutes les formations qui devaient commencer le 2023-01-02 au 2023-01-03.

b) pour les spécialités des formateurs renseignées comme étant "Maths", changer en "Mathématiques".

2. En utilisant la base des remontées mécaniques, suite à une inflation galopante, changer tous les forfaits dont le montant est 30 en 40.

II - Jointures

Dans la base Formation, la table `Inscrit` permet de connaître le numéro NIR des stagiaires inscrits à une formation mais elle ne permet pas de les contacter. En effet, le contact des stagiaires se trouve dans la table `Stagiaire`. Il faut donc coupler ces deux tables pour obtenir cette information. Dans les bases de données, cette opération s'appelle une *jointure*.

Définition 5 - ... INNER JOIN ...

La requête `SELECT * FROM table1 INNER JOIN table2` permet d'afficher le contenu des tables `table1` et `table2` dans lesquelles une jointure naturelle a été faite sur leur attribut commun.

* Pour obtenir les noms et numéros de téléphone de stagiaires inscrits à une formation :

```
SELECT NomStagiaire, TelStagiaire
FROM Stagiaire
INNER JOIN Inscrit
```

* Pour obtenir le nom des stagiaires ainsi que le nom de l'entreprise à laquelle ils sont rattachés :

```
SELECT NomStagiaire, NomEntreprise
FROM Stagiaire
INNER JOIN Entreprise
```

Exercice 5.

1. En utilisant la base des formations, écrire une requête pour déterminer

a) le nom des formateurs ainsi que le numéro de la formation qu'ils animent.

b) le nom et l'entreprise d'appartenance des formateurs.

2. En utilisant la base des remontées mécaniques, écrire une requête permettant de déterminer

a) les noms de domaines et le montant des forfaits sur ces domaines.

b) les numéros des badges et le nom de la ville d'origine des clients associés.

c) le nom des clients et la date de leur badge.

d) le numéro des badges et le nom du domaine associé.

III - Fonctions d'agrégation

Définition 6 - Agrégation

Les fonctions d'agrégation permettent d'obtenir des statistiques sur un ensemble d'enregistrements. On pourra retenir les fonctions `MIN`, `MAX`, `SUM`, `AVG` (moyenne se dit *average* en anglais), `COUNT(*)`.

* Pour compter le nombre d'entreprises présentes dans la base de donnée :

```
SELECT COUNT(*)
FROM Entreprise
```

* Pour obtenir le coût de la formation la moins chère :

```
SELECT MIN(CoutFormation)
FROM Formation
```

* Pour obtenir le coût de la formation la plus chère :

```
SELECT MAX(CoutFormation)
FROM Formation
```

* Pour obtenir le coût moyen d'une formation :

```
SELECT AVG(CoutFormation)
FROM Formation
```

Exercice 6.

1. En utilisant la base de donnée des formations, écrire une requête permettant de déterminer :

a) le nombre de formations disponibles dans la base.

b) la date de naissance du stagiaire le plus âgé.

c) le nombre de formations désignées "Sécurité" disponibles dans la base.

2. En utilisant la base des remontées de ski, écrire une requête permettant de déterminer :

a) le montant du forfait le moins cher.

b) le montant du forfait le plus cher.

c) le montant moyen d'un forfait.

d) le nombre de clients présents dans la base.

e) le nombre de domaines skiables gérés.

IV - Résumé

* Notions : Table, Enregistrement, Attribut, Domaine.
* Structure d'une requête :

```
SELECT ...
FROM ...
INNER JOIN ...
WHERE ...
```

* Fonctions d'agrégats : `AVG`, `MAX`, `MIN`, `SUM`.
* Création d'une table :

```
CREATE TABLE ... (... ..., ... ...)
```

* Modification d'un enregistrement

```
UPDATE ...
SET ... = ...
```

* Suppression d'enregistrements :

```
DELETE FROM ...
WHERE ...
```

* Ajout d'un enregistrement :

```
INSERT INTO ...
(..., ..., ...)
VALUES
(..., ..., ...)
```

V - Du côté des concours

Exercice 7. [HEC - Oral - 2023 - T1] On dispose d'une base de données regroupant des informations sur les cafés parisiens. Son schéma relationnel est le suivant :
- `Cafes` (`Nom` : TEXT, `Adresse` : TEXT, `Arrondissement` : INTEGER, `Prix_cafe` : INTEGER)
Écrire les requêtes SQL permettant d'afficher

1. la liste des cafés du 14^{e} arrondissement ;

2. le prix moyen du café dans les cafés parisiens. On pourra utiliser la fonction de moyenne `AVG()` ;

3. le nombre de cafés où le prix du café est à 1 euro. On pourra utiliser la fonction de comptage `COUNT()`.

Exercice 8. [Ecricome - ECT - 2024] On a accès à une base de données SQL répertoriant les espèces de poissons présentes dans un lac sous forme d'une table nommée `poissons` dont le schéma relationnel est le suivant.

poissons
id : INTEGER
espece : TEXT
quantite : INTEGER
taille : INTEGER
protection : INTEGER

Chaque enregistrement de la table correspond à une espèce de poissons. Les attributs de la table sont décrits de la manière suivante :

- `id` : un numéro permettant d'identifier l'espèce
- `espece` : le nom de l'espèce
- `quantite` : le nombre d'individus
- `taille` : la taille moyenne des individus (en mm)
- `protection` : le statut protégé ou nom de l'espèce (1 si protégée ; 0 sinon).

1. Identifier la clé primaire de la table `poissons`.

2. Une étude a été effectuée et il a été constaté une modification de la taille moyenne des goujons, valant désormais 610 mm.
Écrire une requête SQL permettant de mettre à jour la base de données concernant la taille moyenne des goujons (dont l'attribut `espece` est "goujon").

3. Écrire une requête SQL permettant d'afficher la liste des poissons que les pêcheurs peuvent pêcher, c'est-à-dire ceux qui ne sont pas protégés et dont la taille moyenne est supérieure à 125 mm.

Exercice 9. [HEC - Maths Appli - 2023 - Exercice 6] On considère une base de données de location de voitures dont le schéma relationnel est donné par :

- `voiture` (`id_voiture`, `marque`, `modele`, `etat`)
- `client` (`id_client`, `nom`, `prenom`, `adresse`, `num_permis`)
- `location` (`id_loc`, `voiture`, `client`, `date_debut`, `date_fin`)

1. Identifier les clés primaires et les éventuelles clés étrangères de chacune des tables.

2. L'état d'une voiture à louer est désigné par l'une de ces dénominations : « `neuf` », « `bon` », « `abîmé` », « `au garage` ». Déterminer la liste des voitures qui sont actuellement dans l'un des deux états.

3. Déterminer la liste des voitures (identifiant, marque et modèle) qui ont déjà été louées ou en cours de location.

4. Déterminer tous les clients (identifant, nom et prénom) qui ont loué une Renault. On pourra réaliser une requête imbriquée, de la forme :

```
SELECT *
FROM table1
WHERE attribut IN (
    SELECT attribut2 FROM table2)
```