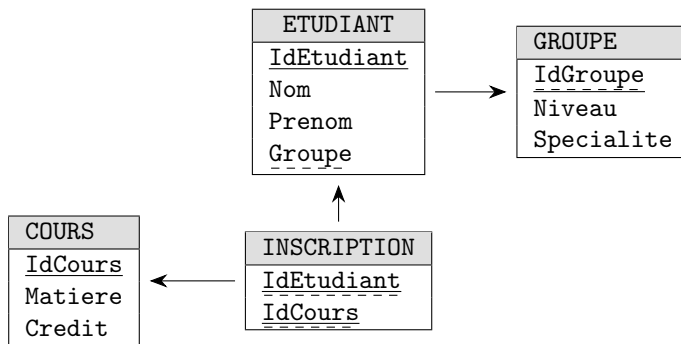


Université

Schéma de la base de données

On introduit un schéma pour représenter le fonctionnement d'une université. Dans la présentation des tables, un attribut souligné par une ligne continue fait partie de la clé primaire (PRIMARY KEY) de sa relation et un attribut souligné par une ligne pointillée fait partie d'une clé étrangère (FOREIGN KEY). Une flèche d'une première table vers une seconde signifie que la première table a une clé étrangère utilisant la clé primaire de la seconde.



Exercices

Avant de commencer, connectez-vous à Oracle sur votre machine virtuelle en suivant les instructions disponibles sur Moodle.

Exercice 1. Création du schéma

Question 1. Commencer par la création de la table GROUPE avec la commande suivante :

```
CREATE TABLE GROUPE (
  IdGroupe VARCHAR(10),
  Niveau VARCHAR2(20),
  Specialite VARCHAR2(20),
  CONSTRAINT cle_primaire_groupe PRIMARY KEY (IdGroupe));
```

Insérer le groupe d'identifiant L2INFO, de niveau licence 2 et de spécialité informatique. Pouvez-vous insérer une autre groupe avec le même identifiant ? Pourquoi ?

Question 2. Ajouter la table ETUDIANT avec :

```
CREATE TABLE ETUDIANT (
  IdEtudiant NUMBER,
  Prenom VARCHAR2(20),
  Nom VARCHAR2(20),
  Groupe VARCHAR2(10) NOT NULL,
  CONSTRAINT cle_primaire_etudiant PRIMARY KEY (IdEtudiant),
  CONSTRAINT cle_etrangere_etudiant_groupe FOREIGN KEY (Groupe) REFERENCES GROUPE
    (IdGroupe));
```

Ajouter l'étudiante Ada Lovelace au groupe des licences 2 info.

Question 3. On souhaite ajouter Margaret Hamilton au groupe des licences 1 math/info. Faites les modifications nécessaires pour faire cette ajout.

Question 4. Créer les tables `INSCRIPTION` et `COURS` en suivant les exemples précédents. Ajouter les informations suivantes dans ces deux tables :

- un cours d'anglais d'identifiant `ANG` avec 2 crédits
- un cours de base de données d'identifiant `BD` avec 5 crédits
- un cours de programmation d'identifiant `PROG` avec 6 crédits

Question 5. Tester les contraintes de ces deux tables avec opérations suivantes. Imaginer à l'avance la réponse et l'état de la base de données avant d'exécuter les opérations. Décrire en français les opérations qui ont un sens.

```
INSERT INTO INSCRIPTION VALUES (0, 'ANG');  
— une deuxième fois  
INSERT INTO INSCRIPTION VALUES (0, 'ANG');  
INSERT INTO INSCRIPTION VALUES (1, 'ANG');  
INSERT INTO INSCRIPTION VALUES (2, 'BD');  
— avec une faute de frappe  
INSERT INTO INSCRIPTION VALUES (1, 'PORG');
```

Question 6. Vider la table `INSCRIPTION`.

Exercice 2. Premières transactions

On souhaite que la base de données respecte la contrainte suivante :

**Tous les étudiants sont soit inscrits à des cours pour un total d'au moins 7 crédits
soit inscrits à aucun cours**

Nous allons utiliser deux sessions d'Oracle en parallèle avec les rôles suivants :

- **Session 1** opère les modifications, c'est celle de l'administrateur
- **Session 2** observe très régulièrement l'état de la base. Aucun état incohérent ne doit jamais être visible depuis cette session.

Question 7. Dans la session 2, rédiger une requête pour obtenir le total nombre de crédit des cours auquel chaque étudiante est inscrite.

Question 8. Ada Lovelace souhaite s'inscrire au cours d'anglais et de base de données. Procéder à son inscription depuis la session 1 à l'aide d'une transaction. À chaque opération, vérifiez le total des crédits par étudiant depuis la session 2. Si vous constatez un état incohérent, videz la table `INSCRIPTION` et recommencez.

Question 9. Margaret Hamilton souhaite être inscrite au cours d'anglais et de programmation. Commencer une première transaction et annulez la après avoir exécuté votre première requête. Vérifiez le total des crédits par étudiant depuis les deux sessions.

Exercice 3. Délégué de groupe

On souhaite que chaque groupe ait un délégué. Pour ce faire, on commence par ajouter une colonne à la table `GROUPE` avec l'opération suivante :

```
ALTER TABLE GROUPE ADD Delegate NUMBER;
```

Question 10. Regarder les groupes où les délégués ne sont pas définis, c'est à dire `NULL`. Ajouter manuellement un délégué pour chacun de ces groupes. Une fois que vous avez terminé, modifier la colonne pour interdire la valeur `NULL` :

```
ALTER TABLE GROUPE MODIFY Delegate NUMBER NOT NULL;
```

Question 11. On souhaite maintenant ajouter une contrainte sur la table **GROUPE** pour assurer que tous les groupes ont un étudiant comme délégué. Pour ajouter cette contrainte, utiliser une commande de la forme suivante où **CONSTRAINT** nom ... est de la même forme que l'introduction d'une contrainte dans la création d'une table.

ALTER TABLE GROUPE ADD CONSTRAINT nom ...

Pour supprimer la contrainte en cas d'erreur :

ALTER TABLE GROUPE DROP CONSTRAINT nom ;
--

Question 12. On souhaite ajouter un étudiant Kurt Gödel comme le délégué et appartenant au nouveau groupe des licences 2 mathématiques. Proposer une transaction pour ajouter l'étudiant et le groupe. Que se passe-t-il ?

Question 13. Question difficile

L'administration de cette base de données se rend compte qu'il n'y a pas de contrainte pour s'assurer que le délégué d'un groupe soit un étudiant du même groupe. Proposer une modification de la base de données (notamment le schéma) pour que les contraintes d'intégrité valident cette propriété.