

Prérequis : Modèle conceptuel de données (entité-association), modèle relationnel, bases du langage SQL.

Durée : 1 h 50

TD 3 – Requêtes SQL

Description du système d'informations

La direction des études des Mines de Nancy a décidé d'informatiser la gestion des emplois du temps. Chaque étudiant est caractérisé par son numéro d'étudiant, son nom, son prénom et son âge. Chaque cours est identifié de façon unique par un sigle (SI033, MD021, ...) et possède un intitulé (bases de données, mathématiques discrètes, ...) ainsi qu'un enseignant responsable. On connaît également le nombre de séances de chaque cours. Les enseignants sont caractérisés par un identifiant alphanumérique, leur nom et leur prénom. Enfin, chaque séance est identifiée par le cours ainsi que le numéro de la séance (séance 3 du cours SI033, séance 1 du cours de MD021, ...), le type d'intervention (CM, TD, TP), la date, l'heure de début et l'heure de fin auxquelles la séance a lieu ainsi que la salle et l'enseignant qui dispense la séance. Les étudiants s'inscrivent aux cours auxquels ils souhaitent assister.

Schéma relationnel retenu

Les clés primaires sont soulignées et les clés étrangères sont en italique.

```

etudiant ( numero , nom , prenom , age )
enseignant ( id , nom , prenom )
cours ( sigle , intitulé , responsable , nombreSeances )
seance ( cours , numero , type , date , salle , heureDebut , heureFin , enseignant )
inscription ( etudiant , cours )

```

Requêtes simples

- i*) Écrire les requêtes de création des tables « Étudiant » et « Séance ».
- ii*) Inscrivez l'étudiant ('10372', 'Léponge', 'Bob', 20) au cours ('LOG015', 'Logique', 'jh1908').
- iii*) Cherchez le nom et le prénom de tous les étudiants de moins de 20 ans.
- iv*) Cherchez le nom et le prénom de l'enseignant responsable du cours de Statistiques.
- v*) Cherchez le nom et le prénom de tous les étudiants inscrits au cours de Probabilités.
- vi*) Déterminez le nombre d'enseignants intervenant dans le cours de Modélisation Stochastique.
- vii*) Où et quand a lieu le premier cours d'Algèbre linéaire ?
- viii*) Affichez un « emploi du temps » du cours de Logique.
- ix*) Pour chaque enseignant, indiquez le nombre de cours dans lesquels il intervient (restreignez les réponses à l'ensemble des enseignants qui interviennent dans au moins deux cours).

Requêtes imbriquées

- i*) Ajoutez un cours magistral de Logique le 14 décembre avec Jacques Herbrand en salle S250 de 14h à 18h.

- ii) Listez les étudiants inscrits à aucun cours.
- iii) Combien d'étudiants (différents) ont assistés à au moins une séance animée par Leonhard Euler ?

Syntaxe SQL

Sélection

```

SELECT liste_d_attributs ← { DISTINCT, AS
                             +, -, *, %
                             AVG, MAX, MIN, SUM, COUNT
FROM noms_de_tables
[ WHERE liste_de_critères ] ← { <, <=, >, >=, =, <>
                                BETWEEN, IN, IS NULL
                                LIKE, AND, OR, NOT
[ GROUP BY liste_d_attributs ]
[ HAVING liste_de_critères ]
[ ORDER BY liste_d_attributs ];
    
```

Exemple :

```

SELECT SUM(p.gain)
FROM Participe p, Jockey j
WHERE p.Numero_jockey = j.Numero_jockey
AND j.nom like 'Jean-Claude Dusse';
    
```

Création de tables

```

CREATE TABLE nom_de_la_table (
    nom_de_l_attribut type [ liste_de_contraintes_d_attribut ]
    nom_de_l_attribut type [ liste_de_contraintes_d_attribut ]
    ...
    liste_de_contraintes_de_table
);
    
```

où :

- $type \in \left\{ \begin{array}{l} \text{VARCHAR}(n) \text{ où } n \in \mathbb{N}, \text{ INT, } \\ \text{DATE, TIME} \end{array} \right\}$
- $contrainte_d_attribut \in \left\{ \begin{array}{l} \text{NULL, NOT NULL,} \\ \text{DEFAULT valeur,} \\ \text{CHECK(nom_de_l_attribut in domaine_de_définition) } \end{array} \right\}$
- $contrainte_de_table \in \left\{ \begin{array}{l} \text{PRIMARY KEY(liste_d_attributs),} \\ \text{FOREIGN KEY(nom_de_l_attribut) ...} \\ \text{... REFERENCES nom_de_la_table(nom_de_l_attribut) } \end{array} \right\}$

Exemple :

```
CREATE TABLE cours (
  sigle          VARCHAR(20)  NOT NULL,
  intitule       VARCHAR(128) NOT NULL,
  responsable    VARCHAR(50)  NOT NULL,
  nombreSeances INT          NOT NULL DEFAULT '0',
  PRIMARY KEY (sigle),
  FOREIGN KEY (responsable) REFERENCES enseignant(id)
);
```

Suppression de table

```
DROP TABLE nom_de_la_table;
```

Insertion

```
INSERT INTO nom_de_la_table ( attribut_1, attribut_2, ... )
VALUES( valeur_1, valeur_2, ... );
```

Requêtes imbriquées / sous-requêtes

Une sous-requête est une commande SELECT dans une autre commande. Par exemple :

```
SELECT * FROM table1 WHERE id IN (SELECT id FROM table2);
```

On dit que la sous-requête est imbriquée dans la requête externe. Il est possible d'imbriquer des requêtes dans des sous-requêtes. Une sous-requête doit toujours être entre parenthèses.

Voici un exemple de commande qui montre les principaux avantages des sous-requêtes et de leur syntaxe :

```
SELECT r1
FROM t1
WHERE s11 = (
  SELECT COUNT(*) FROM t2
  WHERE NOT EXISTS (
    SELECT * FROM t3
    WHERE r3 =
      (SELECT 50,11*s1 FROM t4 WHERE r5 in (SELECT * FROM t5) AS t5)
  )
);
```

EXISTS teste simplement si la requête interne retourne une ligne. NOT EXISTS teste si la requête interne ne retourne aucun résultat.

Prérequis : Modèle conceptuel de données (entité-association), modèle relationnel, bases du langage SQL.

Durée : 1 h 50

TD 3 – Requêtes SQL

Correction

Description du système d'informations

La direction des études des Mines de Nancy a décidé d'informatiser la gestion des emplois du temps. Chaque étudiant est caractérisé par son numéro d'étudiant, son nom, son prénom et son âge. Chaque cours est identifié de façon unique par un sigle (SI033, MD021, ...) et possède un intitulé (bases de données, mathématiques discrètes, ...) ainsi qu'un enseignant responsable. Les enseignants sont caractérisés par un identifiant alphanumérique, leur nom et leur prénom. Enfin, chaque séance est identifiée par le cours ainsi que le numéro de la séance (séance 3 du cours SI033, séance 1 du cours de MD021, ...), le type d'intervention (CM, TD, TP), la date, l'heure de début et l'heure de fin auxquelles la séance a lieu ainsi que la salle et l'enseignant qui dispense la séance. Les étudiants s'inscrivent aux cours auxquels ils souhaitent assister.

Schéma relationnel retenu

Les clés primaires sont soulignées et les clés étrangères sont en italique.

```

Etudiant ( numéro , nom , prenom , âge )
Enseignant ( id , nom , prenom )
Cours ( sigle , intitulé , responsable, NombreSéances )
Séance ( cours , numéro , type , date , salle , heureDébut , heureFin , enseignant )
Inscription ( etudiant , cours )

```

Requêtes simples

- i) Écrire les requêtes de création des tables « Etudiant » et « Séance ».

Réponse :

```

CREATE TABLE Etudiant (
  numero    VARCHAR(20)  PRIMARY KEY,
  nom       VARCHAR(50)  NOT NULL,
  prenom    VARCHAR(50)  NOT NULL,
  age       INT          NOT NULL CHECK(age > 0)
);

CREATE TABLE Séance (
  cours     VARCHAR(20)  NOT NULL,
  numero    VARCHAR(50)  NOT NULL,
  type      VARCHAR(2)   NOT NULL CHECK(type in "CM", "TD", "TP"),
  date      DATE         NOT NULL,
  salle     VARCHAR(10)  NOT NULL,

```

```

heureDebut TIME          NOT NULL,
heureFin   TIME          NOT NULL CHECK(heureFin > heureDebut),
enseignant VARCHAR(20)  NOT NULL,
FOREIGN KEY (cours)      REFERENCES Cours(sigle),
FOREIGN KEY (enseignant) REFERENCES Enseignant(id),
PRIMARY KEY (cours,numero)
);

```

ii) Inscrivez l'étudiant ('10372','Léponge','Bob',20) au cours ('LOG015','Logique','jh1908').

Réponse :

```
INSERT INTO Inscription VALUES ("10372","LOG015");
```

iii) Cherchez le nom et le prénom de tous les étudiants de moins de 20 ans.

Réponse :

```
SELECT nom, prenom FROM Etudiant WHERE age < 20;
```

iv) Cherchez le nom et le prénom de l'enseignant responsable du cours de Statistiques.

Réponse :

```

SELECT nom, prenom
FROM   Enseignant, Cours
WHERE  responsable = id
      AND intitule LIKE "Statistiques";

```

v) Cherchez le nom et le prénom de tous les étudiants inscrits au cours de Probabilités.

Réponse :

```

SELECT e.nom, e.prenom
FROM   etudiant e, inscription i, cours c
WHERE  e.numero = i.etudiant
      AND i.cours = c.sigle
      AND c.intitule like "Probabilites";

```

vi) Déterminez le nombre d'enseignants intervenant dans le cours de Modélisation Stochastique.

Réponse :

```

SELECT count(DISTINCT enseignant)
FROM   Seance, Cours
WHERE  sigle = cours
      AND intitule LIKE "%Modelisation%";

```

vii) Où et quand a lieu le premier cours d'Algèbre linéaire ?

Réponse :

```

SELECT date, salle, heureDebut, heureFin
FROM   Seance, Cours
WHERE  sigle = cours
      AND numero = 1
      AND intitule LIKE "Algebre lineaire";

```

viii) Affichez un « emploi du temps » du cours de Logique.

Réponse :

```
SELECT numero, date, salle, heureDebut, heureFin, e.nom, e.prenom
FROM Seance, Cours, Enseignant e
WHERE sigle = cours
      AND enseignant = id
      AND intitule LIKE "Logique"
ORDER BY date,heureDebut;
```

- ix)* Pour chaque enseignant, indiquez le nombre de cours dans lesquels il intervient (restreignez les réponses à l'ensemble des enseignants qui interviennent dans au moins deux cours).

Réponse :

```
SELECT e.nom, e.prenom, count(distinct cours)
FROM Seance, Cours, Enseignant e
WHERE sigle = cours
      AND enseignant = e.id
GROUP BY e.id
HAVING count(distinct cours)>1
```

Requêtes imbriquées

- i)* Ajoutez un cours magistral de Logique le 14 décembre avec Jacques Herbrand en salle S250 de 14h à 18h.

Réponse :

```
INSERT INTO 'Seance' VALUES (
  (SELECT sigle FROM 'Cours'
   WHERE intitule LIKE 'Logique'),
  (SELECT nombreSeances+1 FROM 'Cours'
   WHERE intitule LIKE "Logique"),
  'CM', '2008-12-14', 'S250', '14:00', '18:00',
  (SELECT id FROM 'Enseignant'
   WHERE nom like "Herbrand"
   AND prenom = "Jacques")
);
UPDATE 'cours' SET nombreSeances = nombreSeances+1 WHERE intitule LIKE 'Logique';
```

- ii)* Listez les étudiants inscrits à aucun cours.

Réponse :

```
SELECT e.nom, e.prenom
FROM Etudiant e
WHERE NOT EXISTS
  (SELECT * FROM Inscription i
   WHERE i.etudiant = e.numero
  );
```

- iii)* Combien d'étudiants (différents) ont assistés à au moins une séance animée par Leonhard Euler ?

Réponse :

```
SELECT COUNT(DISTINCT e.numero)
FROM Etudiant e, Inscription i
WHERE i.etudiant = e.numero
```

```
AND EXISTS(  
  SELECT s.cours  
  FROM   Enseignant e, Seance s  
  WHERE  e.id = s.enseignant  
         AND s.cours = i.cours  
         AND e.nom LIKE "Euler"  
         AND e.prenom LIKE "Leonhard"  
);
```