Bases de données

Un aperçu de SQL par l'exemple

BD100
v121d

2020-09-08

Département d'informatique
Faculté des sciences

UNIVERSITÉ DE SHERBROOKE

Christina.Khnaisser@USherbrooke.ca
http://info.USherbrooke.ca/ckhnaisser
Luc.Lavoie@USherbrooke.ca
http://info.USherbrooke.ca/llavoie

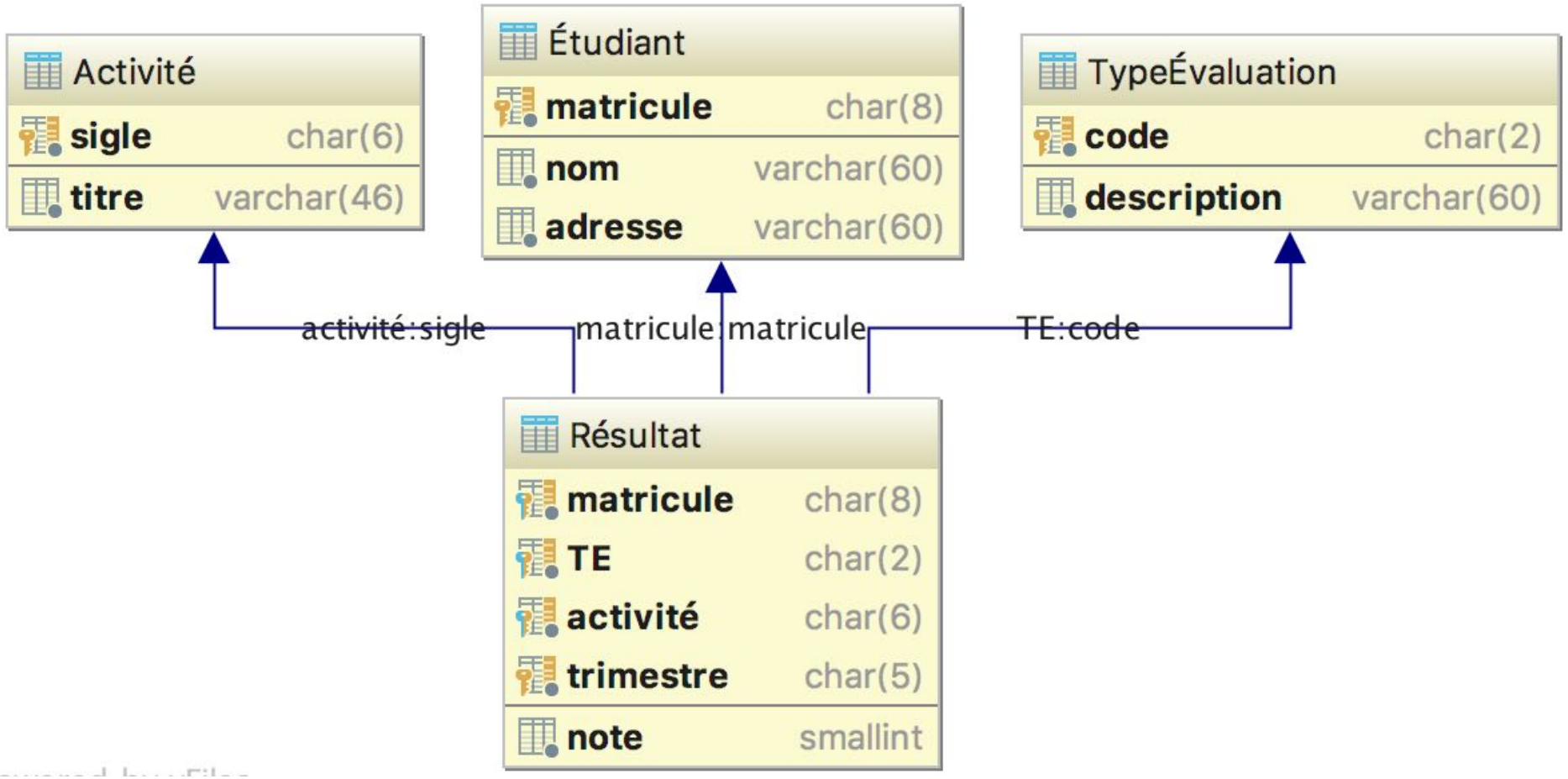
Plan

- Évaluation
  - Rappels
    - Schéma relationnel
    - Exemple de données
  - Schéma SQL
    - Itérations 0, 1, 2, 3
  - Exemple d'insertion de données
  - Requêtes
- Les colles du prof

Solution — Quels sont les prédicats ?
Deuxième essai

- Activité :
  - L'activité de sigle « sigle », décrite par le titre « titre », est offerte par l'UdeS.
- Étudiant :
  - L'étudiant dont le matricule est « matricule », le nom est « nom » et l'adresse est « adresse » est admis à l'UdeS.
- TE :
  - Le type d'évaluation de code « code », décrit par la description « description » est autorisée à l'UdeS.
- Résultat :
  - Le résultat pour l'évaluation « TE » dans le cadre de l'activité « activité » au trimestre « trimestre », décrit par la note « note », a été obtenu par l'étudiant dont le matricule est « matricule ».
  - L'étudiant dont le matricule est « matricule » est inscrit à l'activité « activité » au trimestre « trimestre » à l'UdeS.

Évaluation
Diagramme relationnel



Évaluation — Exemple de données

Étudiant

matricule	nom	adresse
15113150	Paul	>ᐅᖏᓂᑐᖅ
15112354	Éliane	Blanc-Sablon
15113870	Mohamed	Tadoussac
15110132	Sergeï	Chandler

TypeÉvaluation

code	description
IN	Examen intra
FI	Examen final
TP	Travail pratique
PR	Projet

Activité

sigle	titre
IFT159	Analyse et programmation
IFT187	Éléments de bases de données
IMN117	Acquisition des médias numériques
IGE401	Gestion de projets
GMQ103	Géopositionnement

Résultat

matricule	TE	activité	trimestre	note
15113150	TP	IFT187	20133	80
15112354	FI	IFT187	20123	78
15113150	TP	IFT159	20133	75
15112354	FI	GMQ103	20123	85
15110132	IN	IMN117	20123	90
15110132	IN	IFT187	20133	45
15112354	FI	IFT159	20123	52

Évaluation – Itération 0
Schéma relationnel et script SQL (1/2)

Activité {
sigle : Texte;
titre : Texte
}

Étudiant {
matricule : Texte;
nom : Texte;
adresse : Texte
}

```
CREATE TABLE Activite (
    sigle       CHAR(6) NOT NULL,
    titre       VARCHAR(46) NOT NULL
  );

CREATE TABLE Etudiant (
    matricule   CHAR(8) NOT NULL,
    nom         VARCHAR(60) NOT NULL,
    adresse     VARCHAR(60) NOT NULL
  );
```

Évaluation – Itération 0
Schéma relationnel et script SQL (2/2)

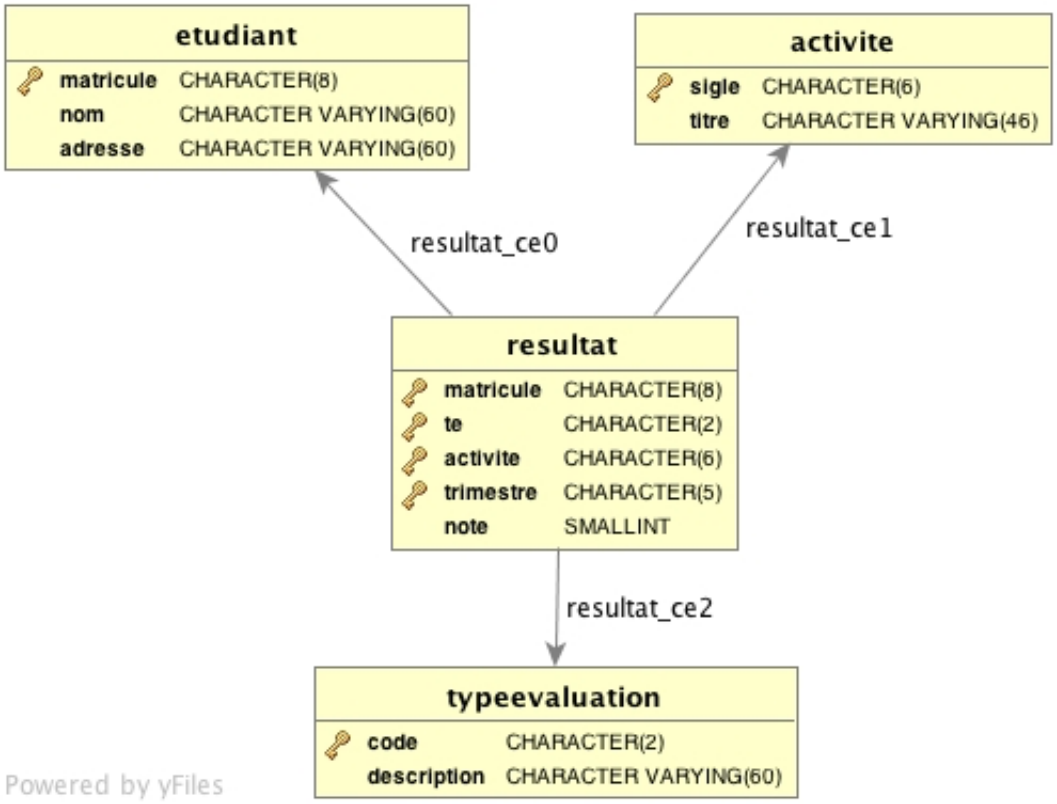
TypeÉvaluation {
code : Texte;
description : Texte
}

```
CREATE TABLE TypeEvaluation (
    code          CHAR(2) NOT NULL,
    description   VARCHAR(60) NOT NULL
);
```

Résultat {
matricule : Texte;
TE : Texte;
activité : Texte;
trimestre : Entier;
note : Entier [0..100]
}

```
CREATE TABLE Resultat (
    matricule     CHAR(8) NOT NULL,
    TE            CHAR(2) NOT NULL,
    activite      CHAR(6) NOT NULL,
    trimestre     CHAR(5) NOT NULL,
    note          INTEGER NOT NULL
);
```

Évaluation – Itération 1
Ajouter des clés : le diagramme relationnel



Évaluation – Itération 1
Ajouter des clés : le script SQL (1/2)

```
CREATE TABLE Activite (
    sigle         CHAR(6) NOT NULL,
    titre         VARCHAR(46) NOT NULL,
    PRIMARY KEY (sigle)
  );
CREATE TABLE Etudiant (
    matricule     CHAR(8) NOT NULL,
    nom           VARCHAR(60) NOT NULL,
    adresse       VARCHAR(60) NOT NULL,
    PRIMARY KEY (matricule)
  );
CREATE TABLE TypeEvaluation (
    code          CHAR(2) NOT NULL,
    description   VARCHAR(60) NOT NULL,
    PRIMARY KEY (code)
  );
```

Évaluation – Itération 1
Ajouter des clés : le script SQL (2/2)

```
CREATE TABLE Resultat (
  matricule     CHAR(8) NOT NULL,
  TE            CHAR(2) NOT NULL,
  activite      CHAR(6) NOT NULL,
  trimestre     CHAR(5) NOT NULL,
  note          SMALLINT NOT NULL,
  PRIMARY KEY (matricule, activite, TE, trimestre),
  FOREIGN KEY (matricule)
    REFERENCES Etudiant (matricule),
  FOREIGN KEY (activite)
    REFERENCES Activite (sigle),
  FOREIGN KEY (TE)
    REFERENCES TypeEvaluation (code)
  );
```

Évaluation – Itération 2
Ajouter des identifiants aux clés (1/2)

```
CREATE TABLE Activite (
    sigle         CHAR(6) NOT NULL,
    titre         VARCHAR(46) NOT NULL,
    CONSTRAINT Activite_cc0 PRIMARY KEY (sigle)
  );
CREATE TABLE Etudiant (
    matricule     CHAR(8) NOT NULL,
    nom           VARCHAR(60) NOT NULL,
    adresse       VARCHAR(60) NOT NULL,
    CONSTRAINT Etudiant_cc0 PRIMARY KEY (matricule)
  );
CREATE TABLE TypeEvaluation (
    code          CHAR(2) NOT NULL,
    description   VARCHAR(60) NOT NULL,
    CONSTRAINT TypeEvaluation_cc0 PRIMARY KEY (code)
  );
```

Évaluation – Itération 2
Ajouter des identifiants aux clés (2/2)

```
CREATE TABLE Resultat (
  matricule     CHAR(8) NOT NULL,
  TE            CHAR(2) NOT NULL,
  activite      CHAR(6) NOT NULL,
  trimestre     CHAR(5) NOT NULL,
  note          SMALLINT NOT NULL,
  CONSTRAINT Resultat_cc0
    PRIMARY KEY (matricule, activite, TE, trimestre),
  CONSTRAINT Resultat_cr0 FOREIGN KEY (matricule)
    REFERENCES Etudiant (matricule),
  CONSTRAINT Resultat_cr1 FOREIGN KEY (activite)
    REFERENCES Activite (sigle),
  CONSTRAINT Resultat_cr2 FOREIGN KEY (TE)
    REFERENCES TypeEvaluation (code)
  );
```

Évaluation – Itération 3
Ajouter de contraintes générales

- Sigle de cours :
  - trois lettres majuscules suivies de trois chiffres.
- Matricule :
  - huit chiffres.
- Code de type d'évaluation :
  - deux lettres.
- Trimestre :
  - l'année suivie d'un chiffre : 1 (hiver), 2 (été) ou 3 (automne).
- Note :
  - entier compris en 0 et 100.

Évaluation – Itération 3
Ajouter des contraintes générales (1/4)

```
CREATE TABLE Activite (
    sigle         CHAR(6) NOT NULL,
    titre         VARCHAR(46) NOT NULL,
    CONSTRAINT Activite_cc0 PRIMARY KEY (sigle),
 -- On s'assure que les trois premiers caractères
 -- du sigle sont des lettres latines majuscules et
 -- les trois derniers des chiffres.
    CONSTRAINT Activite_sigle CHECK (
      sigle SIMILAR TO '[A-Z]{3}[0-9]{3}'
      )
  );
```

Évaluation – Itération 3
Ajouter des contraintes générales (2/4)

```
CREATE TABLE Etudiant (
    matricule     CHAR(8) NOT NULL,
    nom           VARCHAR(60) NOT NULL,
    adresse       VARCHAR(60) NOT NULL,
    CONSTRAINT Etudiant_cc0 PRIMARY KEY (matricule),
 -- On s'assure que le matricule soit composé
 -- de huit chiffres.
    CONSTRAINT Etudiant_matricule CHECK (
      matricule SIMILAR TO '[0-9]{8}'
      )
  );
```

Évaluation – Itération 3
Ajouter des contraintes générales (3/4)

```
CREATE TABLE TypeEvaluation (
    code          CHAR(2) NOT NULL,
    description   VARCHAR(60) NOT NULL,
    CONSTRAINT TypeEvaluation_cc0 PRIMARY KEY (code),
 -- On s'assure que le code soit composé de
 -- deux lettres latines.
    CONSTRAINT TypeEvaluation_code CHECK (
      code SIMILAR TO '[A-Za-z]{2}'
      )
  );
```

Évaluation – Itération 3
Ajouter des contraintes générales (4/4)

```
CREATE TABLE Resultat (
  matricule     CHAR(8) NOT NULL,
  TE            CHAR(2) NOT NULL,
  activite      CHAR(6) NOT NULL,
  trimestre     CHAR(5) NOT NULL,
  note          SMALLINT NOT NULL,
  CONSTRAINT Resultat_cc0
    PRIMARY KEY (matricule, activite, TE, trimestre),
  CONSTRAINT Resultat_cr0 FOREIGN KEY (matricule)
    REFERENCES Etudiant (matricule),
  CONSTRAINT Resultat_cr1 FOREIGN KEY (activite)
    REFERENCES Activite (sigle),
  CONSTRAINT Resultat_cr2 FOREIGN KEY (TE)
    REFERENCES TypeEvaluation (code),
   CONSTRAINT Resultat_note CHECK (note BETWEEN 0 AND 100),
 -- Les trimestres sont encodés en suffixant le no du
 -- trimestre à l'année.
    CONSTRAINT Resultat_trimestre CHECK
     (trimestre SIMILAR TO '[0-9]{4}[1-3]{1}')
 );
```

Évaluation
Exemple d'insertion de données

```
INSERT INTO Activite (sigle, titre) VALUES
    ('IFT159', 'Analyse et programmation'),
    ('IFT187', 'Éléments de bases de données'),
    ('IMN117', 'Acquisition des médias numériques'),
    ('IGE401', 'Gestion de projets'),
    ('GMQ103', 'Géopositionnement');
```

Évaluation
Requêtes

1. Quels sont les étudiants inscrits en IFT 187?
2. Quels sont les étudiants inscrits à une activité d'informatique à l'automne 2013?
3. Quels étaient étudiants en situation d'échec au final à l'automne 2012?
4. Produire le relevé de notes d'Éliane.
5. Quels étudiants ne sont inscrits à aucune activité?

Évaluation – R1
Quels sont les étudiants inscrits en IFT 187?

- Clarification
  - La formulation utilisée pourrait indiquer qu'on s'intéresse au seul trimestre courant. Le requérant nous précise cependant qu'il vise toutes les inscriptions depuis la première offre de l'activité.
- Entête
  - InscritsIFT187 {matricule : Texte}
- Requête
  - (Résultat σ (activité='IFT187')) π {matricule}

Évaluation – R1
Script SQL

```
SELECT DISTINCT matricule
FROM Resultat
WHERE activite='IFT187'
```

- Rappels
  - La requête n'est correcte que si :
    - matricule est une clé référentielle de Résultat vers Étudiant;
    - « on » définit qu'une inscription n'est en vigueur que si l'étudiant a complété une première évaluation et que celle-ci a été saisie;
  - Si la première condition est naturelle, la deuxième l'est moins et soulève une interrogation quant à la justesse de la modélisation.

ÉVALUATION – R2
QUELS SONT LES ÉTUDIANTS INSCRITS À UNE ACTIVITÉ D'INFORMATIQUE À L'AUTOMNE 2013?

- Clarification
  - Une activité d'informatique est définie comme étant toute activité dont le sigle débute par le préfixe 'IFT'. Nous supposons qu'il existe une fonction préfixe définie sur les chaines de caractères (le type Texte).
  - "Automne 2013" doit être recodé sous la forme de l'entier 20133.
- Entête
  - InscritsIFT {matricule : Texte}
- Requête
  - (Résultat σ (préfixe(activité,3)='IFT') $\wedge$ trimestre='20133') π {matricule}

Évaluation – R2
Script SQL

```
SELECT matricule
FROM Resultat
WHERE SUBSTRING(activite, 1, 3)='IFT'
  AND trimestre = '20133';
```

Avec doublons

matricule
15113150
15113150
15110132

```
SELECT DISTINCT matricule
FROM Resultat
WHERE SUBSTRING(activite, 1, 3)='IFT'
  AND trimestre = '20133';
```

Sans doublons

matricule
15113150
15110132

Évaluation – R3
Quels étaient étudiants en situation d'échec au final à l'automne 2012?

- Clarification
  - Une situation d'échec est une note inférieure à 60.
  - Un final est un 'Examen final' représenté par le code 'FI'.
  - « Automne 2012 » doit être recodé sous la forme de l'entier 20123.
- Entête
  - Échecs20123 {matricule : Texte}
- Requête
  - (Résultat σ (note<60 $\wedge$ TE='FI' $\wedge$ trimestre='20123'))π {matricule}

Évaluation – R3
Script SQL

```
SELECT DISTINCT matricule
FROM Resultat
WHERE note < 60
  AND TE = 'FI'
  AND trimestre = '20123';
```

Évaluation – R4
Produire le relevé de notes d'Éliane.

- Clarification
  - Le matricule a été introduit pour différencier les homonymes. Un relevé produit sur la seule base du nom est donc susceptible d'être inexact. En conséquence, **nous demanderons à Éliane** son matricule (nous ne consultons pas la base de données).
- Entête
  - RelevéÉliane
    { TE : TypeEval; activité : SigleCours;
    trimestre : Trimestre; note : Note }
- Requête
  - (Résultat σ (matricule='15112354')) π {TE, activité, trimestre, note}

ÉVALUATION – R4
SCRIPT SQL

```
SELECT DISTINCT TE, activite, trimestre, note
FROM Resultat
WHERE matricule = '15112354';
```

Pourquoi n'est-ce pas nécessaire, mais néanmoins pas inexact, de spécifier DISTINCT ?

Évaluation – R4 (bis)
Produire le relevé de notes d'Éliane.

- Clarification
  - On demande d'ajouter le titre de l'activité dans le relevé.
- Entête
  - Relevé2Éliane {TE : Texte; *sigle : Texte;* titre : Texte; trimestre : Entier; note : Entier [0..100]}
- Requête
  - Remarquons qu'il est nécessaire de joindre la relation Activité pour obtenir le titre et que l'attribut de jointure n'y porte pas le même nom que dans la relation Résultat.
  - (((Résultat σ (matricule='15112354'))
    ρ {activité → sigle})
    ⋈ Activité)
    π {TE, *sigle*, titre, trimestre, note}

Évaluation – R4 (bis)
Script SQL

```
SELECT TE, sigle, titre, trimestre, note
FROM Resultat
        JOIN Activite ON (activite = sigle)
WHERE matricule = '15112354';
```

Évaluation – R5
Quels étudiants ne sont inscrits à aucune activité?

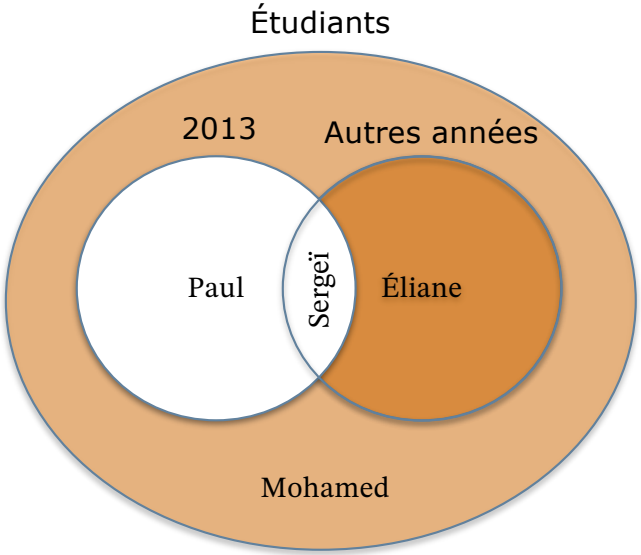
- Clarification
  - Encore une fois, il est nécessaire de faire préciser la période à couvrir, en terme de trimestres. Supposons que ce soit les trois trimestres de l'année 2013, la question devient donc :
    - Quels étudiants ne sont inscrits à aucune activité en 2013?
  - Supposons également qu'on désire avoir un maximum d'information sur ces étudiants et pas seulement leur matricule (à savoir tous les attributs disponibles dans la relation Étudiant).
- Entête
  - NonInscrits2013
    {matricule : Texte; nom : Texte; adresse : Texte}

Évaluation – R5 (suite)
Quels étudiants ne sont inscrits à aucune activité en 2013?

- Requête
  - Nous réalisons cette requête en calculant d'abord les étudiants inscrits à au moins une activité en 2013 puis en les soustrayant de l'ensemble des étudiants.
  - Inscrits2013 :
    (Résultat σ ('20131'≤trimestre≤'20133')) π {matricule}
  - NonInscrits2013 :
    Étudiant – (Étudiant $\bowtie$ Inscrits2013)
  - ou, en extension :
    Étudiant –
    (Étudiant $\bowtie$
    ((Résultat σ ('20131'≤trimestre≤'20133')) π {matricule}))

Évaluation – R5 : Script SQL version 2

```
SELECT matricule, nom, adresse
FROM Etudiant
EXCEPT
  SELECT matricule, nom, adresse
  FROM Etudiant
    JOIN (
          SELECT matricule
          FROM Resultat
          WHERE '20131' <= trimestre
            AND trimestre <= '20133'
        ) AS Inscrit2013
    USING(matricule);
```



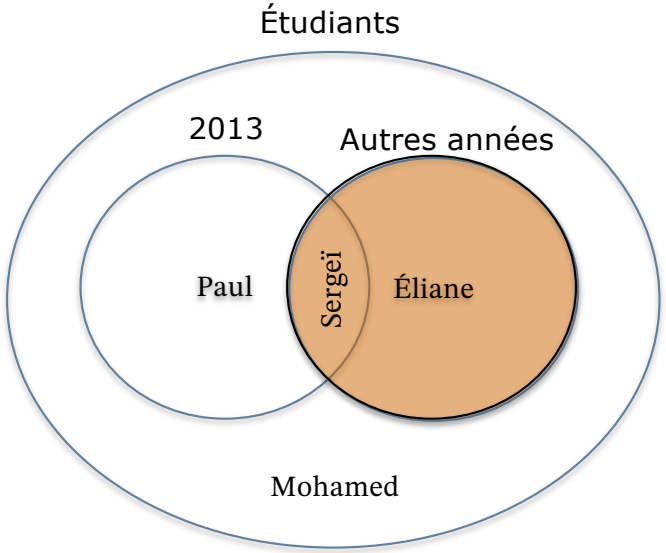
matricule	nom	adresse
15112354	Éliane	Blanc-Sablon
15113870	Mohamed	Tadoussac

Évaluation – R5 (bis)
Quels étudiants ne sont inscrits à aucune activité en 2013?

- Justin remarque qu'il est possible de simplifier l'expression précédente si on admet qu'au sein du modèle la négation de la proposition
  (20131≤trimestre≤20133)
  s'exprime par
  (trimestre<20131 ⋁ 20133<trimestre)
- Il obtient alors
  (Étudiant
  ⋈(Résultat σ (trimestre<'20131' ⋁ '20133'<trimestre)))
  π {matricule, nom, adresse}
- A-t-il raison ?
  - NON!

Évaluation – R5 (bis) : Script SQL

```
SELECT DISTINCT matricule, nom, adresse
FROM Etudiant
  JOIN (
        SELECT matricule
        FROM Resultat
        WHERE trimestre < '20131'
           OR '20133' > trimestre
      ) AS NonInscrit2013
  USING(matricule);
```



matricule	nom	adresse
15112354	Éliane	Blanc-Sablon
15110132	Sergeï	Chandler

Évaluation – R5 (bis)
Pourquoi ?

- Parce qu'un étudiant peut s'inscrire lors de plusieurs trimestres, on ne peut pas logiquement conclure qu'un étudiant inscrit en 2013 ne s'est inscrit à aucune autre année :
  - Par exemple, Sergeï s'est inscrit en 2012 et en 2013.
- On ne peut pas logiquement conclure qu'un étudiant qui ne s'est pas inscrit en 2013 s'est inscrit lors d'une autre année :
  - Par exemple, Mohamed est étudiant, mais ne s'est encore inscrit à aucune activité.

Les colles du prof

- Quelle est la différence entre un schéma, un diagramme et un script ?
- Y a-t-il une seule façon de traduire une expression relationnelle en script SQL ?
- Comment décririez-vous l'instruction SELECT ?
- Est-il pertinent de formuler une requête d'abord sous forme d'expression relationnelle puis de la traduire en script SQL ?
- Est-il préférable de formuler directement une requête en script SQL ?
- Comparer les prédicats utilisés dans BD100 avec ceux proposés par l'exemple déposé dans le répertoire public du cours. Décrire la portée des différences, indiquer les "meilleures" formulations et motiver les choix.