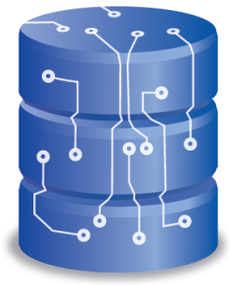


Bases de données

Ecole Marocaine des Sciences de l'Ingénieur

© Yousra Lembachar

Questions sur les premières parties?

Chapitre 4

SQL

PLAN

- Création de tables
- Insertion de données
- Mise à jour de données
- Suppression de données

SQL

- Structured Query Language
- Langage de manipulation et de consultation d'une BD
- Supporté par la plupart des SGBD

Requêtes de création et de manipulation

Les clauses CREATE/DROP

```
CREATE SCHEMA nomSchema;

CREATE  TABLE nomTable(

        attribut1  type, attribut2 type, ...

        contrainte1, contrainte2, ...);
```

type: INT, TEXT, CHAR, VARCHAR(n),..

```
DROP TABLE nomTable;
```

Contrainte d'intégrité: La clé primaire

CONSTRAINT nomContrainte

PRIMARY KEY (attribut1 [, attribut2,...])

Contrainte d'intégrité: La clé étrangère

CONSTRAINT nomContrainte

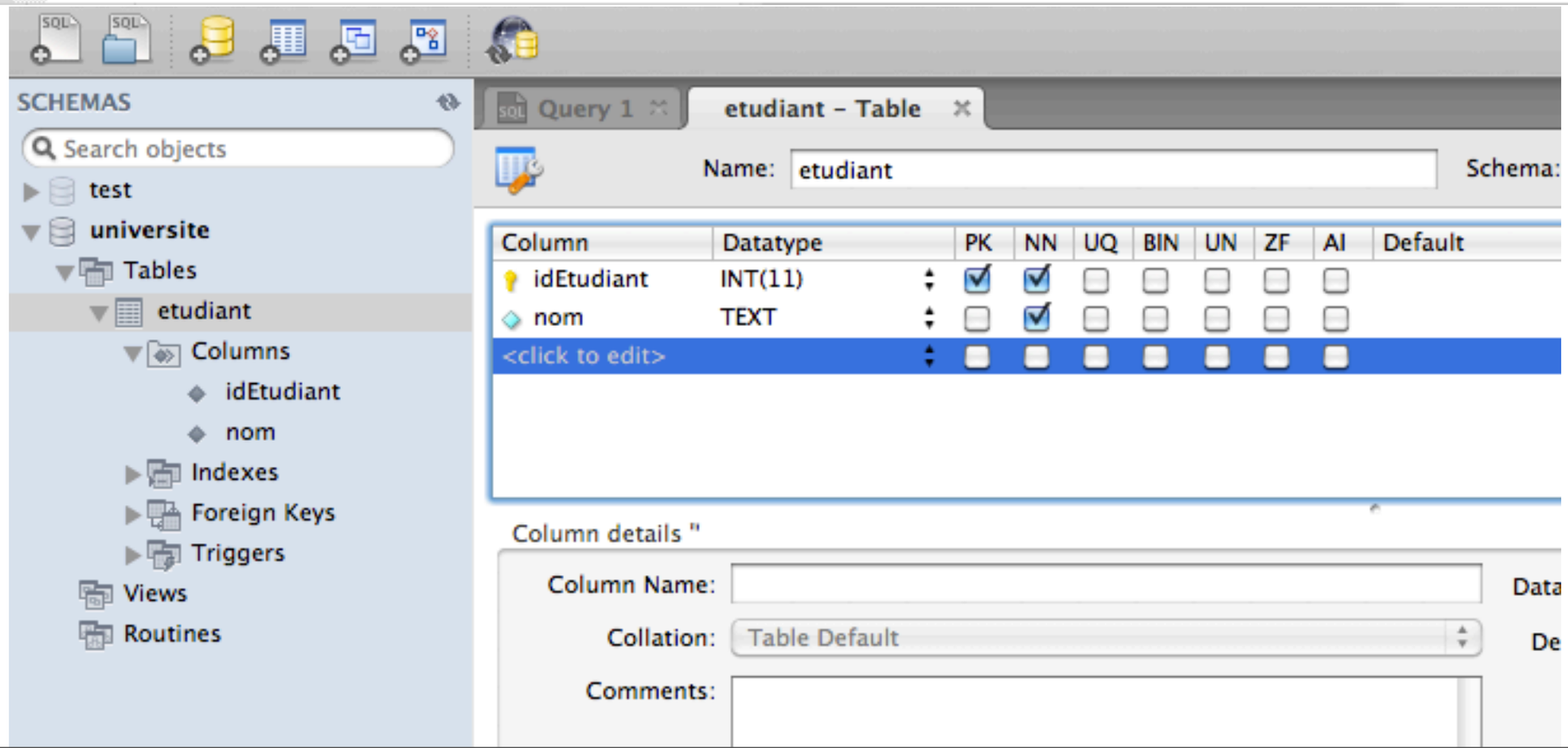
FOREIGN KEY (attribut)

REFERENCES nomTable(attribut)

La clause CREATE

Query 1 | note – Table

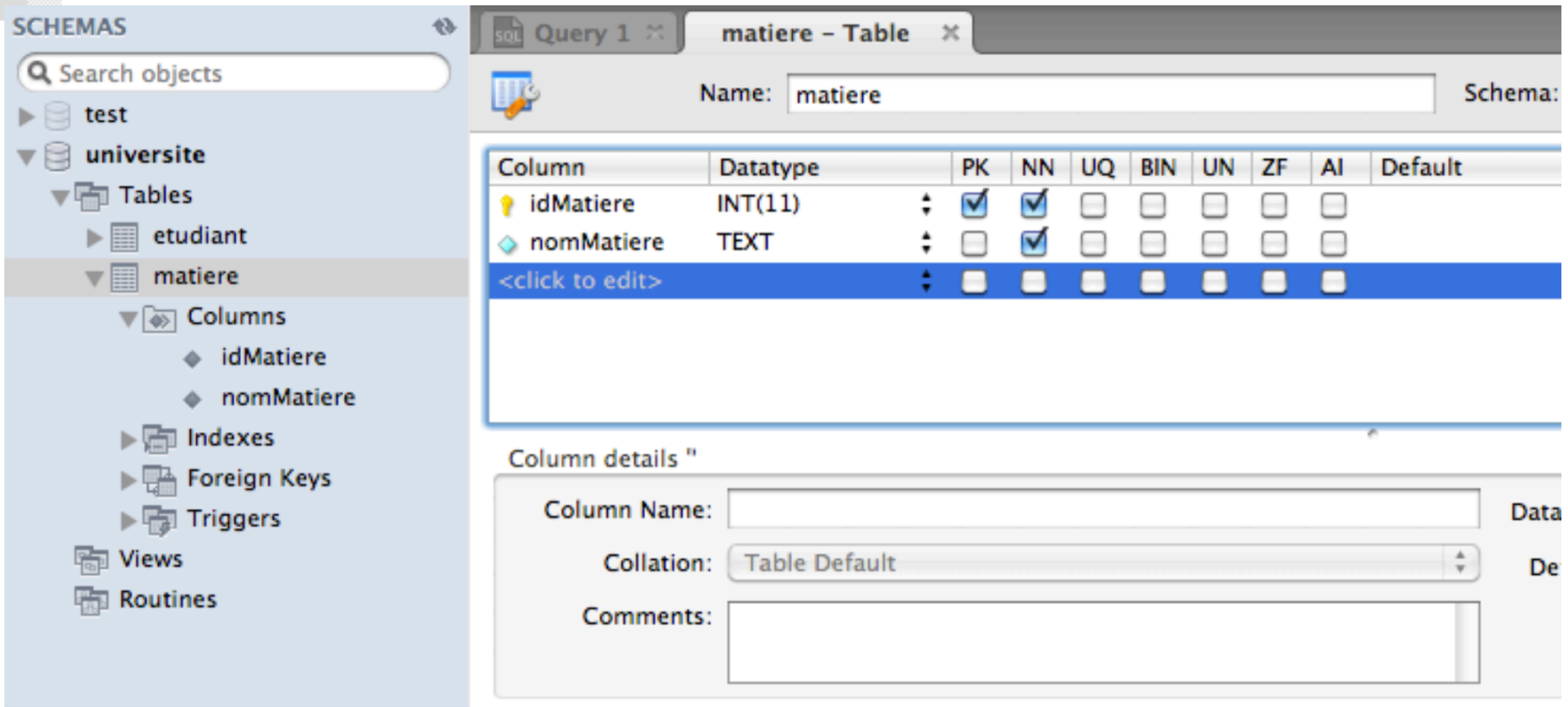
```sql
CREATE TABLE etudiant (
    idEtudiant INT NOT NULL,
    nom TEXT NOT NULL,
    CONSTRAINT pk1 PRIMARY KEY (idEtudiant)
)
```



La clause CREATE

Query 1 | etudiant – Table

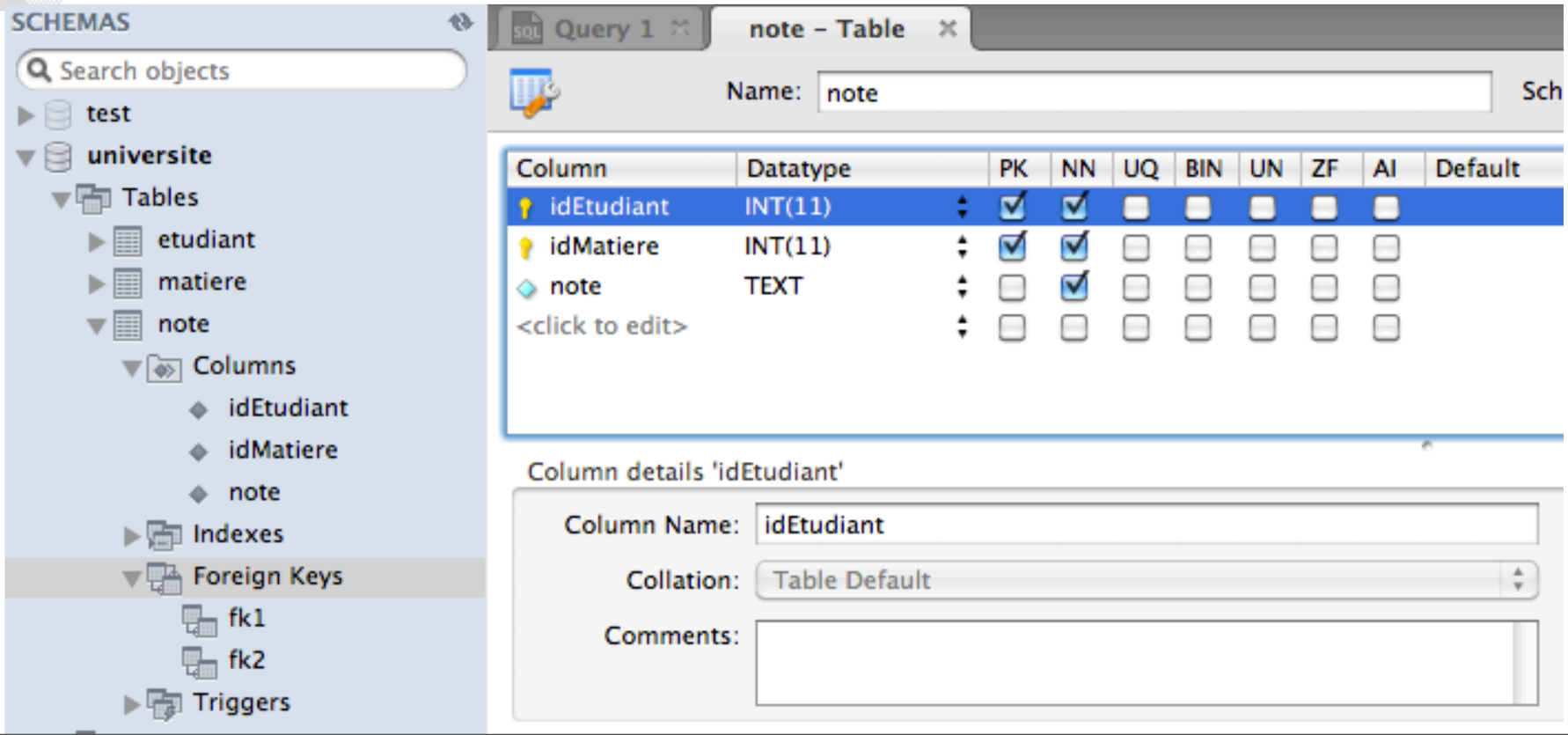
```
1  CREATE TABLE matiere (
2      idMatiere INT NOT NULL,
3      nomMatiere TEXT NOT NULL,
4      CONSTRAINT pk1 PRIMARY KEY (idMatiere)
5  )
```



La clause CREATE

Query 1 | matiere – Table

```
1  CREATE TABLE note (
2      idEtudiant INT NOT NULL,
3      idMatiere INT NOT NULL,
4      note TEXT NOT NULL,
5      CONSTRAINT pk1 PRIMARY KEY (idEtudiant , idMatiere),
6      CONSTRAINT fk1 FOREIGN KEY (idEtudiant)
7          REFERENCES etudiant (idEtudiant),
8      CONSTRAINT fk2 FOREIGN KEY (idMatiere)
9          REFERENCES matiere (idMatiere)
10 )
```



La clause INSERT

```
INSERT INTO    nomTable [(col1, col2, ...)]
VALUES (valeurAttr1, valeurAttr2, ...)
```

La clause INSERT

Query 1 | note – Table

```
1 • INSERT INTO ETUDIANT VALUES (1, 'Samantha');
2 • INSERT INTO ETUDIANT VALUES (2, 'Francis');
3 • INSERT INTO ETUDIANT VALUES (3, 'Charles');
4 • INSERT INTO ETUDIANT VALUES (4, 'Penelope');
5 • INSERT INTO ETUDIANT VALUES (5, 'Craig');
6 • INSERT INTO ETUDIANT VALUES (6, 'Steve');
7 • INSERT INTO ETUDIANT VALUES (7, 'Dave');
```

Query 1 | etudiant

Filter: Edit: File:

idEtudiant	nom
1	Samantha
2	Francis
3	Charles
4	Penelope
5	Craig
6	Steve
7	Dave
	NULL

La clause INSERT

Query 1 | etudiant

```
1 • INSERT INTO ETUDIANT values (1, 'Neil');
```

41:1

Action Output

	Time	Action	Response
1	18:40:30	INSERT INTO ETUDIANT values (1, 'Neil')	Error Code: 1062. Duplicate entry '1' for key 'PRIMARY'

Contrainte d'intégrité (clé primaire) violée!

La clause INSERT

Query 1

```
1  INSERT INTO MATIERE values (1, 'Maths');
2  INSERT INTO MATIERE values (2, 'Psychologie');
3  INSERT INTO MATIERE values (3, 'Nutrition');
4  INSERT INTO MATIERE values (4, 'Art Moderne');
5  INSERT INTO MATIERE values (5, 'Art Contemporatin');
6  INSERT INTO MATIERE values (6, 'Art Classique');
7
```

Query 1 | matiere

Filter: Edit: File:

idMatiere	nomMatiere
1	Maths
2	Psychologie
3	Nutrition
4	Art Moderne
5	Art Contempo…
6	Art Classique
	NULL

La clause UPDATE

UPDATE nomTable

SET attribut = valeur

WHERE condition

La clause UPDATE

Query 1 | matiere

Filter: Edit: File:

idMatiere	nomMatiere
1	Maths
2	Psychologie
3	Nutrition
4	Art Moderne
5	Art Contempo...
6	Art Classique
	NULL

Avant la requête

Query 1 | matiere

```
1  UPDATE MATIERE
2  SET
3      nomMatiere = 'Sociologie'
4  WHERE
5      idMatiere = 2
```

Query 1 | matiere

Filter: Edit: File:

idMatiere	nomMatiere
1	Maths
2	Sociologie
3	Nutrition
4	Art Moderne
5	Art Contempo...
6	Art Classique
	NULL

Après la requête

La clause DELETE

DELETE FROM nomTable

WHERE condition

La clause DELETE

Query 1 | matiere

Filter: | Edit: | File:

idMatiere	nomMatiere
1	Maths
2	Sociologie
3	Nutrition
4	Art Moderne
5	Art Contempo…
6	Art Classique
	NULL

Avant la requête

Query 1 | matiere

```
1 • DELETE FROM MATIERE
2   WHERE
3       idMatiere = 2
```

Query 1 | matiere

Filter: | Edit: | File:

idMatiere	nomMatiere
1	Maths
3	Nutrition
4	Art Moderne
5	Art Contempo…
6	Art Classique
	NULL

Après la requête

Requêtes de consultation

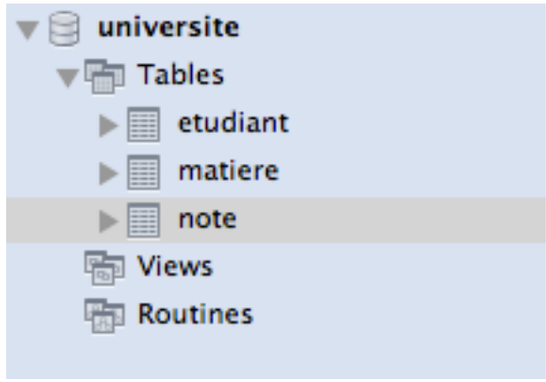
La clause SELECT

SELECT attribut1, attribut2, ...

FROM table1, table2, ...

WHERE condition

La clause SELECT



Query 1 | etudiant | matiere | note

Filter: Edit: File:

idEtudiant	nom
1	Samantha
2	Francis
3	Charles
4	Penelope
5	Craig
6	Steve
7	Dave
	NULL

Query 1 | etudiant | matiere | note

Filter: Edit: File:

idMatiere	nomMatiere
1	Maths
2	Sociologie
3	Nutrition
4	Art moderne
5	Art contempo...
6	Art classique
	NULL

Query 1 | etudiant | matiere | note

Filter: Edit: File:

idEtudiant	idMatiere	note
1	1	15
1	2	20
2	1	20
3	1	13
4	1	14
5	1	10
6	1	8
	NULL	NULL

La clause SELECT

Query 1

```
1  SELECT
2      *
3  FROM
4      ETUDIANT
```

Tous les attributs de la table Etudiant

Equivalent à ETUDIANT en algèbre relationnelle

Query 1

Filter: Edit: File:

idEtudiant	nom
1	Samantha
2	Francis
3	Charles
4	Penelope
5	Craig
6	Steve
7	Dave
	NULL

La clause SELECT - Opérateur de sélection

```
1  SELECT
2      *
3  FROM
4      ETUDIANT
5  WHERE
6      idEtudiant > 4;
7
```

Les étudiants avec un identifiant > 4

$$\sigma_{idEtudiant > 4} \, ETUDIANT$$

1:1

Filter: Edit: File:

idEtudiant	nom
5	Craig
6	Steve
7	Dave
	NULL

La clause SELECT - Opérateur de sélection

Les étudiants qui s'appellent Samantha

$$\sigma_{nom = \ 'Samantha'} ETUDIANT$$

```
1 • SELECT
2       *
3   FROM
4       ETUDIANT
5   WHERE
6       nom = 'Samantha';
7
```

8:6

Filter: Edit: File:

idEtudiant	nom
1	Samantha
	NULL

La clause SELECT - Opérateur de sélection

Les étudiants dont le nom commence avec 'S'

$$\sigma_{nom\ like\ 'S\%'}\ ETUDIANT$$

```sql
SELECT
    *
FROM
    ETUDIANT
WHERE
    nom like 'S%';
```

19:6

Filter: Edit: File:

idEtudiant	nom
1	Samantha
6	Steve
	NULL

La clause SELECT - Opérateur de projection

Query 1

```
1 • SELECT
2       nom
3   FROM
4       ETUDIANT
```

Les noms des étudiants

$$\pi_{nom} ETUDIANT$$

1:1

Filter: File:

nom
Samantha
Francis
Charles
Penelope
Craig
Steve
Dave

La clause SELECT - Opérateur de projection

Query 1

```
1  SELECT
2      nom
3  FROM
4      ETUDIANT
5  WHERE
6      idEtudiant = 4
```

Le nom de l'étudiant avec l'identifiant 4

$$\pi_{nom}(\sigma_{idEtudiant=4} ETUDIANT)$$

Query 1

Filter: File:

NOM
Penelope

La clause SELECT - Opérateur de projection

Query 1

```
1  SELECT
2      nomMatiere
3  FROM
4      MATIERE
5  WHERE
6     nomMatiere like '%Art%'
```

Les matières dont le nom contient la chaîne 'Art'

Query 1

Filter: File:

nomMatiere
Art moderne
Art contemporain
Art classique

La clause SELECT - Opérateur de projection

Query 1

```
1 • SELECT
2       note/2
3   FROM
4       NOTE
```

Les notes divisées par 2

9:4

Filter: File:

note/2
10.0000
6.5000
7.0000
5.0000
4.0000
4.0000

La clause SELECT - Opérateur de projection

Notion d'alias

Query 1

```
1 • SELECT
2       note/2 as noteSur10
3   FROM
4       NOTE
```

Les notes divisées par 2

24:2

Filter: File:

noteSur10
10.0000
6.5000
7.0000
5.0000
4.0000
4.0000

Le produit cartésien et les jointures

```
1 • SELECT
2       *
3   FROM
4       ETUDIANT, NOTE
```

ETUDIANT X NOTE

19:4

Filter: File:

idEtudiant	nom	idEtudiant	idMatiere	note
1	Samantha	2	1	20
1	Samantha	3	1	13
1	Samantha	4	1	14
1	Samantha	5	1	10
1	Samantha	6	1	8
1	Samantha	1	4	8
2	Francis	2	1	20
2	Francis	3	1	13
2	Francis	4	1	14
2	Francis	5	1	10
2	Francis	6	1	8
2	Francis	1	4	8
3	Charles	2	1	20
3	Charles	3	1	13
3	Charles	4	1	14
3	Charles	5	1	10

Le produit cartésien et les jointures

Query 1

```
1  SELECT
2      nom
3  FROM
4      ETUDIANT,
5      NOTE
6  WHERE
7      ETUDIANT.idEtudiant = NOTE.idEtudiant
8
```

Le nom de tous les étudiants ayant une note

$$\pi_{nom} ETUDIANT \bowtie NOTE$$

Query 1

Filter: File:

nom
Samantha
Samantha
Francis
Charles
Penelope
Craig
Steve

etudiant

Filter: Edit

idEtudiant	nom
1	Samantha
2	Francis
3	Charles
4	Penelope
5	Craig
6	Steve
7	Dave
NULL	

note

Filter: Edit:

idEtudiant	idMatiere	note
1	1	15
1	2	20
2	1	20
3	1	13
4	1	14
5	1	10
6	1	8
NULL	NULL	NULL

Le produit cartésien et les jointures

Query 1

```
1  SELECT distinct
2      nom
3  FROM
4      ETUDIANT,
5      NOTE
6  WHERE
7      ETUDIANT.idEtudiant = NOTE.idEtudiant
8
```

Le nom de tous les étudiants ayant une note sans duplication

Query 1

Filter: File:

nom
Samantha
Francis
Charles
Penelope
Craig
Steve

Le produit cartésien et les jointures

Query 1

```
1  SELECT distinct
2      nom
3  FROM
4      ETUDIANT,
5      NOTE
6  WHERE
7      ETUDIANT.idEtudiant = NOTE.idEtudiant
8      AND note > 15
9
```

Le nom de tous les étudiants ayant une note > 15 sans duplication

$$\pi_{nom}(\sigma_{note>15} ETUDIANT \bowtie NOTE)$$

Query 1

Filter: File:

nom
Samantha
Francis

etudiant

Filter: Edit

idEtudiant	nom
1	Samantha
2	Francis
3	Charles
4	Penelope
5	Craig
6	Steve
7	Dave
	NULL

note

Filter: Edit:

idEtudiant	idMatiere	note
1	1	15
1	2	20
2	1	20
3	1	13
4	1	14
5	1	10
6	1	8
	NULL	NULL

Le produit cartésien et les jointures

Query 1

```
1  SELECT
2      nom, note
3  FROM
4      ETUDIANT,
5      NOTE
6  WHERE
7      ETUDIANT.idEtudiant = NOTE.idEtudiant
8      AND note > 15
9
```

Le nom et la note de tous les étudiants ayant une note > 15

$$\pi_{nom,note}(\sigma_{note>15} ETUDIANT \bowtie NOTE)$$

Query 1

Filter: File:

nom	note
Samantha	20
Francis	20

Le produit cartésien et les jointures

Query 1

```sql
SELECT
    nom, note, nomMatiere
FROM
    ETUDIANT,
    NOTE,
    MATIERE
WHERE
    ETUDIANT.idEtudiant = NOTE.idEtudiant
        AND MATIERE.idMatiere = NOTE.idMatiere
        AND note > 15
```

Query 1

Filter: File:

nom	note	nomMatiere
Francis	20	Maths
Samantha	20	Sociologie

Le nom, la note et la matière de tous les étudiants ayant une note > 15

$$\pi_{nom,note,nomMatiere}(\sigma_{note>15} ETUDIANT \bowtie NOTE \bowtie MATIERE)$$

Le produit cartésien et les jointures

Query 1

```sql
SELECT
    nom, note
FROM
    ETUDIANT,
    NOTE,
    MATIERE
WHERE
    ETUDIANT.idEtudiant = NOTE.idEtudiant
        AND MATIERE.idMatiere = NOTE.idMatiere
        AND note > 10
        AND nomMatiere = 'Maths'
```

Le nom et la note de tous les étudiants ayant une note >10 en Maths

Query 1

Filter: File:

nom	note
Samantha	15
Francis	20
Charles	13
Penelope	14

$$\pi_{nom,note,nomMatiere}(\sigma_{note>10 \wedge nomMatiere='Maths'} ETUDIANT \bowtie NOTE \bowtie MATIERE)$$

Le produit cartésien et les jointures

Query 1

```sql
SELECT
    idEtudiant, nom, note
FROM
    ETUDIANT,
    NOTE,
    MATIERE
WHERE
    ETUDIANT.idEtudiant = NOTE.idEtudiant
        AND MATIERE.idMatiere = NOTE.idMatiere
        AND note > 10
        AND nomMatiere = 'Maths'
```

160% 19:8

Action Output

		Time	Action	Response
✖	1	14:02:03	SELECT idEtudiant, nom, note FROM ETUDIANT, NOTE, ...	Error Code: 1052. Column 'idEtudiant' in field list is ambiguous

L'identifiant, le nom et la note de tous les étudiants ayant une note >10 en Maths

La colonne idEtudiant est ambigüe!

Le produit cartésien et les jointures

Query 1

```
1  SELECT
2      ETUDIANT.idEtudiant, nom, note
3  FROM
4      ETUDIANT,
5      NOTE,
6      MATIERE
7  WHERE
8      ETUDIANT.idEtudiant = NOTE.idEtudiant
9          AND MATIERE.idMatiere = NOTE.idMatiere
10         AND note > 10
11         AND nomMatiere = 'Maths'
```

L'identifiant, le nom et la note de tous les étudiants ayant une note >10 en Maths

Query 1

Filter: File:

idEtudiant	nom	note
1	Samantha	15
2	Francis	20
3	Charles	13
4	Penelope	14

La colonne idEtudiant n'est plus ambigüe!

Le produit cartésien et les jointures

Query 1

```sql
SELECT
    ETUDIANT.idEtudiant, nom, note
FROM
    ETUDIANT,
    NOTE,
    MATIERE
WHERE
    ETUDIANT.idEtudiant = NOTE.idEtudiant
        AND MATIERE.idMatiere = NOTE.idMatiere
        AND note > 10
        AND nomMatiere = 'Maths'
ORDER BY note;
```

Query 1

Filter: File:

idEtudiant	nom	note
3	Charles	13
4	Penelope	14
1	Samantha	15
2	Francis	20

L'identifiant, le nom et la note de tous les étudiants ayant une note >10 en Maths, par ordre ascendant des notes

Le produit cartésien et les jointures

Query 1

```sql
SELECT
    ETUDIANT.idEtudiant, nom, note
FROM
    ETUDIANT,
    NOTE,
    MATIERE
WHERE
    ETUDIANT.idEtudiant = NOTE.idEtudiant
        AND MATIERE.idMatiere = NOTE.idMatiere
        AND note > 10
        AND nomMatiere = 'Maths'
ORDER BY note desc;
```

Query 1

Filter: File:

idEtudiant	nom	note
2	Francis	20
1	Samantha	15
4	Penelope	14
3	Charles	13

L'identifiant, le nom et la note de tous les étudiants ayant une note >10 en Maths, par ordre descendant des notes