

Bases de données

Ecole Marocaine des Sciences de l'Ingénieur

© Yousra Lembachar

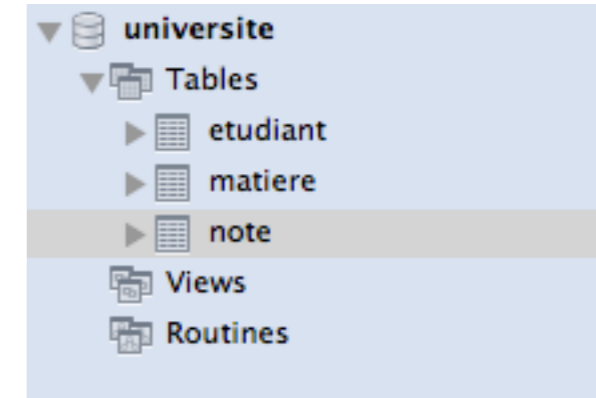
Intégrité référentielle

Base de données

idEtudiant	nom
1	Samantha
2	Francis
3	Charles
4	Penelope
5	Craig
6	Steve
7	Dave
9	Cameron
10	Patricia
11	Patricia
	NULL

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

idEtudiant	idMatiere	note
6	1	8
11	3	7
10	1	5
10	2	4
2	1	20
1	2	20
1	1	15
4	1	14
3	1	13
5	1	10
	NULL	NULL



Définition:

Etant données deux tables A et B, si une information dans B référence une information dans A, l'information référencée existe dans la table A.

Intégrité référentielle

A = Etudiant

idEtudiant	nom
1	Samantha
2	Francis
3	Charles
4	Penelope
5	Craig
6	Steve
7	Dave
9	Cameron
10	Patricia
11	Patricia
	NULL

B = Note

idEtudiant	idMatiere	note
6	1	8
11	3	7
10	1	5
10	2	4
2	1	20
1	2	20
1	1	15
4	1	14
3	1	13
5	1	10
15	1	9

Violation de l'intégrité référentielle

1. Intégrité référentielle de NOTE(idEtudiant) vers ETUDIANT(idEtudiant)
=> Chaque valeur dans NOTE(idEtudiant) existe dans ETUDIANT(idEtudiant)
2. Le contraire n'est pas nécessaire! Chaque valeur dans ETUDIANT(idEtudiant) n'existe pas forcément dans NOTE(idEtudiant)
3. ~ Contrainte de clé étrangère (peut être constituée de plusieurs attributs)
4. L'attribut référencé est toujours unique et généralement une clé primaire

Intégrité référentielle - Insertion

A

idEtudiant	nom
1	Samantha
2	Francis
3	Charles
4	Penelope
5	Craig
6	Steve
7	Dave
9	Cameron
10	Patricia
11	Patricia
	NULL

B

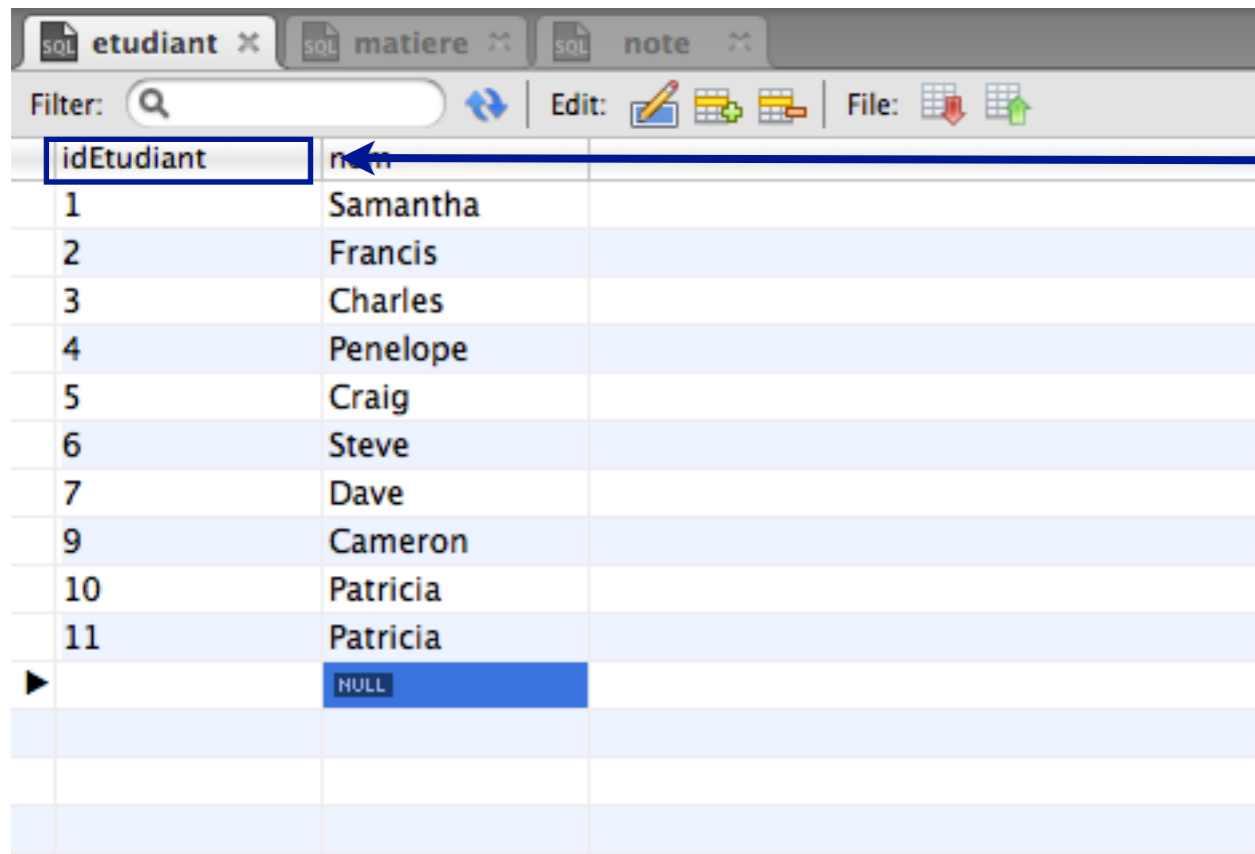
idEtudiant	idMatiere	note
6	1	8
11	3	7
10	1	5
10	2	4
2	1	20
1	2	20
1	1	15
4	1	14
3	1	13
5	1	10
	NULL	NULL

1. Insertion dans table A => OK
2. Insertion dans table B => OK si l'information référencée existe dans A, sinon NOK

On peut insérer (7,1,10) mais on ne peut pas insérer (12,1,10)

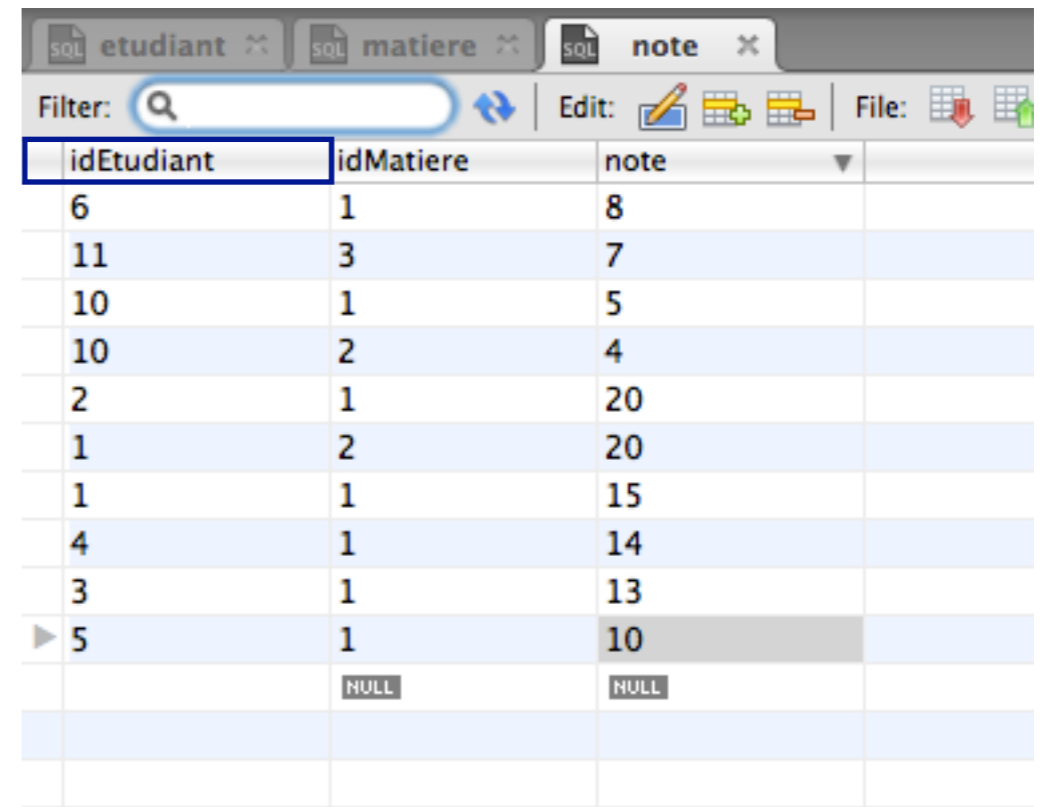
Intégrité référentielle - Suppression

A



idEtudiant	nom
1	Samantha
2	Francis
3	Charles
4	Penelope
5	Craig
6	Steve
7	Dave
9	Cameron
10	Patricia
11	Patricia
	NULL

B



idEtudiant	idMatiere	note
6	1	8
11	3	7
10	1	5
10	2	4
2	1	20
1	2	20
1	1	15
4	1	14
3	1	13
5	1	10
	NULL	NULL

1. Suppression dans table B => OK
2. Suppression dans table A => NOK si l'information à supprimer existe dans B, sinon OK

On peut supprimer n'importe quelle ligne dans NOTE mais on ne peut pas supprimer les étudiants 1, 2, 3, 4, 5, 6, 10 et 11 dans ETUDIANT

Intégrité référentielle - Mise à jour

A

idEtudiant	nom
1	Samantha
2	Francis
3	Charles
4	Penelope
5	Craig
6	Steve
7	Dave
9	Cameron
10	Patricia
11	Patricia
	NULL

B

idEtudiant	idMatiere	note
6	1	8
11	3	7
10	1	5
10	2	4
2	1	20
1	2	20
1	1	15
4	1	14
3	1	13
5	1	10
	NULL	NULL

1. Mise à jour dans table B => OK si l'information mise à jour existe dans A, sinon NOK
2. Mise à jour dans table A => NOK si l'information mise à jour existe dans B et se retrouve sans référence, sinon OK

1. On peut mettre à jour idEtudiant dans NOTE à condition que le nouvel ID soit égal à l'un des identifiants dans ETUDIANT
2. On ne peut pas mettre à jour les identifiants des étudiants qui ont une note

Intégrité référentielle dans SQL

- ON DELETE
 - RESTRICT
 - SET NULL
 - CASCADE
- ON UPDATE
 - RESTRICT
 - SET NULL
 - CASCADE

Intégrité référentielle dans SQL - ON DELETE RESTRICT

Script SQL pour la création de la BD

```
5
6 • CREATE TABLE ETUDIANT (
7     idEtudiant INT NOT NULL,
8     nom TEXT NOT NULL,
9     CONSTRAINT pk PRIMARY KEY (idEtudiant)
10 );
11
12 • CREATE TABLE MATIERE (
13     idMatiere INT NOT NULL,
14     nomMatiere TEXT NOT NULL,
15     CONSTRAINT pk PRIMARY KEY (idMatiere)
16 );
17
18 • CREATE TABLE NOTE (
19     idEtudiant INT,
20     idMatiere INT,
21     NOTE INT NOT NULL,
22     CONSTRAINT fk1 FOREIGN KEY (idEtudiant) REFERENCES ETUDIANT (idEtudiant),
23     CONSTRAINT fk2 FOREIGN KEY (idMatiere) REFERENCES MATIERE (idMatiere)
24 );
```

Restrict est le comportement par défaut: Suppression dans A
=> NOK si l'information à supprimer existe dans B, sinon OK

Intégrité référentielle dans SQL - ON DELETE RESTRICT

```
1 • DELETE FROM ETUDIANT
2 WHERE
3     idEtudiant = 1;
4
5
```

1:5

Action Output

Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`universite`.`note`, CONSTRAINT `fk1` FOREI...

Suppression de `idEtudiant = 1` dans `ETUDIANT` ne passe pas car l'étudiant a une note et que le mode est `RESTRICT`.

Intégrité référentielle dans SQL - ON DELETE SET NULL

Script SQL pour la création de la BD


```
6 • CREATE TABLE ETUDIANT (  
7     idEtudiant INT NOT NULL,  
8     nom TEXT NOT NULL,  
9     CONSTRAINT pk PRIMARY KEY (idEtudiant)  
10 );  
11  
12 • CREATE TABLE MATIERE (  
13     idMatiere INT NOT NULL,  
14     nomMatiere TEXT NOT NULL,  
15     CONSTRAINT pk PRIMARY KEY (idMatiere)  
16 );  
17  
18 • CREATE TABLE NOTE (  
19     idEtudiant INT,  
20     idMatiere INT,  
21     NOTE INT NOT NULL,  
22     CONSTRAINT fk1 FOREIGN KEY (idEtudiant) REFERENCES ETUDIANT (idEtudiant)  
23     ON DELETE SET NULL,  
24     CONSTRAINT fk2 FOREIGN KEY (idMatiere) REFERENCES MATIERE (idMatiere)  
25     ON DELETE SET NULL  
26 );
```

SET NULL: Suppression dans A => Si l'information à supprimer existe dans B, elle est mise à NULL

Intégrité référentielle dans SQL - ON DELETE SET NULL

```
1 • INSERT INTO ETUDIANT VALUES (12, 'Alba');
2 • INSERT INTO NOTE VALUES (12, 1, 0);
3
4 • DELETE FROM ETUDIANT
5 WHERE
6     idEtudiant = 12;
7
8 • SELECT
9     *
10 FROM
11     NOTE
12 where
13     note = 0;
14
```

160% 1:1

Filter: File: 

idEtudiant	idMatiere	NOTE
▶	1	0

idEtudiant = 12 est mis à NULL

Intégrité référentielle dans SQL - ON DELETE CASCADE

Script SQL pour la modification des contraintes


```
1 • ALTER TABLE NOTE DROP FOREIGN KEY fk1;  
2 • ALTER TABLE NOTE DROP FOREIGN KEY fk2;  
3  
4 • ALTER TABLE NOTE ADD CONSTRAINT fk1 FOREIGN KEY (idEtudiant)  
5     REFERENCES ETUDIANT(idEtudiant)  
6     ON DELETE CASCADE;  
7  
8 • ALTER TABLE NOTE ADD CONSTRAINT fk2 FOREIGN KEY (idMatiere)  
9     REFERENCES MATIERE(idMatiere)  
10    ON DELETE CASCADE;  
11  
12
```

CASCADE: Suppression dans A => Si l'information à supprimer existe dans B, elle est supprimée

Intégrité référentielle dans SQL - ON DELETE CASCADE

```
1 • INSERT INTO ETUDIANT VALUES (12, 'Alba');
2 • INSERT INTO NOTE VALUES (12, 1, 0);
3
4 • DELETE FROM ETUDIANT
5 WHERE
6     idEtudiant = 12;
7
8 • SELECT
9     *
10 FROM
11     NOTE
12 where
13     note = 0;
14
```

160% 6:12

Filter: File: 

idEtudiant	idMatiere	NOTE

La ligne avec idEtudiant = 12 est supprimée des deux tables NOTE et ETUDIANT

Intégrité référentielle dans SQL

- ON DELETE
 - RESTRICT ✓
 - SET NULL ✓
 - CASCADE ✓
- ON UPDATE
 - RESTRICT
 - SET NULL
 - CASCADE

Intégrité référentielle dans SQL - ON UPDATE RESTRICT

Script SQL pour la création de la BD

```
5
6 • CREATE TABLE ETUDIANT (
7     idEtudiant INT NOT NULL,
8     nom TEXT NOT NULL,
9     CONSTRAINT pk PRIMARY KEY (idEtudiant)
10 );
11
12 • CREATE TABLE MATIERE (
13     idMatiere INT NOT NULL,
14     nomMatiere TEXT NOT NULL,
15     CONSTRAINT pk PRIMARY KEY (idMatiere)
16 );
17
18 • CREATE TABLE NOTE (
19     idEtudiant INT,
20     idMatiere INT,
21     NOTE INT NOT NULL,
22     CONSTRAINT fk1 FOREIGN KEY (idEtudiant) REFERENCES ETUDIANT (idEtudiant),
23     CONSTRAINT fk2 FOREIGN KEY (idMatiere) REFERENCES MATIERE (idMatiere)
24 );
```

Restrict est le comportement par défaut: Mise à jour dans A
=> NOK si l'information à mettre à jour existe dans B, sinon OK

Intégrité référentielle dans SQL - ON UPDATE RESTRICT

```
1 • UPDATE ETUDIANT
2 SET
3     idEtudiant = 20
4 WHERE
5     idEtudiant = 1;
```

250%	20:5		
Action Output			
	Time	Action	Response
✘ 1	16:35:31	UPDATE ETUDIANT SET idEtudiant = 20 WHERE idEtudiant = 1	Error Code: 1451. Cannot delete or update a parent row: a foreign key cons...

Mise à jour de idEtudiant = 1 dans ETUDIANT ne passe pas car l'étudiant a une note et que le mode est RESTRICT.

Intégrité référentielle dans SQL - ON UPDATE SET NULL

Script SQL pour la modification des contraintes

```
1 • ALTER TABLE NOTE DROP FOREIGN KEY fk1;  
2 • ALTER TABLE NOTE DROP FOREIGN KEY fk2;  
3  
4 • ALTER TABLE NOTE ADD CONSTRAINT fk1 FOREIGN KEY (idEtudiant)  
5     REFERENCES ETUDIANT(idEtudiant)  
6     ON UPDATE SET NULL;  
7  
8 • ALTER TABLE NOTE ADD CONSTRAINT fk2 FOREIGN KEY (idMatiere)  
9     REFERENCES MATIERE(idMatiere)  
10    ON UPDATE SET NULL;  
11  
12
```

SET NULL: Mise à jour dans A => Si l'information à mettre à jour existe dans B, elle est mise à NULL

Intégrité référentielle dans SQL - ON UPDATE SET NULL

```
1 • INSERT INTO ETUDIANT VALUES (12, 'Alba');
2 • INSERT INTO NOTE VALUES (12, 1, 0);
3
4 • UPDATE ETUDIANT
5 SET
6     idEtudiant = 20
7 WHERE
8     idEtudiant = 12;
9
10 • SELECT * FROM NOTE WHERE note = 0;
```

160% 34:10

Filter: File: 

idEtudiant	idMatiere	NOTE
▶	1	0

idEtudiant = 12 est mis à NULL

Intégrité référentielle dans SQL - ON UPDATE CASCADE

Script SQL pour la modification des contraintes


```
1 • ALTER TABLE NOTE DROP FOREIGN KEY fk1;  
2 • ALTER TABLE NOTE DROP FOREIGN KEY fk2;  
3  
4 • ALTER TABLE NOTE ADD CONSTRAINT fk1 FOREIGN KEY (idEtudiant)  
5     REFERENCES ETUDIANT(idEtudiant)  
6     ON UPDATE CASCADE;  
7  
8 • ALTER TABLE NOTE ADD CONSTRAINT fk2 FOREIGN KEY (idMatiere)  
9     REFERENCES MATIERE(idMatiere)  
10    ON UPDATE CASCADE;  
11  
12
```

CASCADE: Mise à jour dans A => Si l'information à mettre à jour existe dans B, elle est mise à jour aussi

Intégrité référentielle dans SQL - ON UPDATE CASCADE

```
1 • INSERT INTO ETUDIANT VALUES (12, 'Alba');
2 • INSERT INTO NOTE VALUES (12, 1, 0);
3
4 • UPDATE ETUDIANT
5 SET
6     idEtudiant = 20
7 WHERE
8     idEtudiant = 12;
9
10 • SELECT * FROM NOTE WHERE note = 0;
```

160% 1:1

Filter: File: 

idEtudiant	idMatiere	NOTE
▶ 20	1	0

La ligne avec idEtudiant = 12 est mis à jour dans les deux tables
NOTE et ETUDIANT