Bases de données

Université Aboubakr Belkaïd de Tlemcen

L3 Génie Industriel

Intégrité référentielle

Base de données

etudiant | matiere | note

Filter: Edit: File:

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

- universite
  - Tables
    - etudiant
    - matiere
    - note
  - Views
  - Routines

etudiant | matiere | note

Filter: Edit: File:

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

etudiant | matiere | note

Filter: Edit: File:

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

Intégrité référentielle

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

etudiant | matiere | note

Filter: Edit: File:

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

etudiant | matiere | note

Filter: Edit: File:

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
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

etudiant | matiere | note

Filter: Edit: File:

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

B

etudiant | matiere | note

Filter: Edit: File:

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

etudiant | matiere | note

Filter: Edit: File:

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

B

etudiant | matiere | note

Filter: Edit: File:

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

etudiant | matiere | note

Filter: Edit: File:

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

B

etudiant | matiere | note

Filter: Edit: File:

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

```sql

CREATE TABLE ETUDIANT (
    idEtudiant INT NOT NULL,
    nom TEXT NOT NULL,
    CONSTRAINT pk PRIMARY KEY (idEtudiant)
);

CREATE TABLE MATIERE (
    idMatiere INT NOT NULL,
    nomMatiere TEXT NOT NULL,
    CONSTRAINT pk PRIMARY KEY (idMatiere)
);

CREATE TABLE NOTE (
    idEtudiant INT,
    idMatiere INT,
    NOTE INT NOT NULL,
    CONSTRAINT fk1 FOREIGN KEY (idEtudiant) REFERENCES ETUDIANT (idEtudiant),
    CONSTRAINT fk2 FOREIGN KEY (idMatiere) REFERENCES MATIERE (idMatiere)
);
```

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

```sql
DELETE FROM ETUDIANT
WHERE
    idEtudiant = 1;
```

1:5

Action Output

nse

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

```sql
CREATE TABLE ETUDIANT (
    idEtudiant INT NOT NULL,
    nom TEXT NOT NULL,
    CONSTRAINT pk PRIMARY KEY (idEtudiant)
);

CREATE TABLE MATIERE (
    idMatiere INT NOT NULL,
    nomMatiere TEXT NOT NULL,
    CONSTRAINT pk PRIMARY KEY (idMatiere)
);

CREATE TABLE NOTE (
    idEtudiant INT,
    idMatiere INT,
    NOTE INT NOT NULL,
    CONSTRAINT fk1 FOREIGN KEY (idEtudiant) REFERENCES ETUDIANT (idEtudiant)
        ON DELETE SET NULL,
    CONSTRAINT fk2 FOREIGN KEY (idMatiere) REFERENCES MATIERE (idMatiere)
        ON DELETE SET NULL
);
```

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

Script SQL pour la modification des contraintes

```sql
ALTER TABLE NOTE DROP FOREIGN KEY fk1; -- Supprime la contrainte fk1
ALTER TABLE NOTE DROP FOREIGN KEY fk2; -- Supprime la contrainte fk2

ALTER TABLE NOTE -- Ajoute la contrainte fk1
    ADD CONSTRAINT fk1
        FOREIGN KEY (idEtudiant) REFERENCES ETUDIANT(idEtudiant)
            ON DELETE SET NULL; -- Lors du DELETE de idEtudiant
                                -- NOTE(idEtudiant) est à NULL

ALTER TABLE NOTE
    ADD CONSTRAINT fk2
        FOREIGN KEY (idMatiere) REFERENCES MATIERE(idMatiere)
            ON DELETE SET NULL; -- Lors du DELETE de idMatiere
                                -- NOTE(idMatiere) est à NULL

```

Intégrité référentielle dans SQL - ON DELETE SET NULL

```sql
INSERT INTO ETUDIANT VALUES (12, 'Alba');
INSERT INTO NOTE VALUES (12, 1, 0);

DELETE FROM ETUDIANT
WHERE
    idEtudiant = 12;

SELECT
    *
FROM
    NOTE
where
    note = 0;
```

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

```sql
ALTER TABLE NOTE DROP FOREIGN KEY fk1;
ALTER TABLE NOTE DROP FOREIGN KEY fk2;

ALTER TABLE NOTE ADD CONSTRAINT fk1 FOREIGN KEY (idEtudiant)
    REFERENCES ETUDIANT(idEtudiant)
        ON DELETE CASCADE;

ALTER TABLE NOTE ADD CONSTRAINT fk2 FOREIGN KEY (idMatiere)
    REFERENCES MATIERE(idMatiere)
        ON DELETE CASCADE;

```

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

Intégrité référentielle dans SQL - ON DELETE CASCADE

```sql
INSERT INTO ETUDIANT VALUES (12, 'Alba');
INSERT INTO NOTE VALUES (12, 1, 0);

DELETE FROM ETUDIANT
WHERE
    idEtudiant = 12;

SELECT
    *
FROM
    NOTE
where
    note = 0;
```

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

```sql
UPDATE ETUDIANT
SET
    idEtudiant = 20
WHERE
    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

```sql
ALTER TABLE NOTE DROP FOREIGN KEY fk1;
ALTER TABLE NOTE DROP FOREIGN KEY fk2;

ALTER TABLE NOTE ADD CONSTRAINT fk1 FOREIGN KEY (idEtudiant)
    REFERENCES ETUDIANT(idEtudiant)
        ON UPDATE SET NULL;

ALTER TABLE NOTE ADD CONSTRAINT fk2 FOREIGN KEY (idMatiere)
    REFERENCES MATIERE(idMatiere)
        ON UPDATE SET NULL;
```

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

```sql
INSERT INTO ETUDIANT VALUES (12, 'Alba');
INSERT INTO NOTE VALUES (12, 1, 0);

UPDATE ETUDIANT
SET
    idEtudiant = 20
WHERE
    idEtudiant = 12;

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

```sql
ALTER TABLE NOTE DROP FOREIGN KEY fk1;
ALTER TABLE NOTE DROP FOREIGN KEY fk2;

ALTER TABLE NOTE ADD CONSTRAINT fk1 FOREIGN KEY (idEtudiant)
    REFERENCES ETUDIANT(idEtudiant)
        ON UPDATE CASCADE;

ALTER TABLE NOTE ADD CONSTRAINT fk2 FOREIGN KEY (idMatiere)
    REFERENCES MATIERE(idMatiere)
        ON UPDATE CASCADE;
```

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

```sql
INSERT INTO ETUDIANT VALUES (12, 'Alba');
INSERT INTO NOTE VALUES (12, 1, 0);

UPDATE ETUDIANT
SET
    idEtudiant = 20
WHERE
    idEtudiant = 12;

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