

## SOUAPIEBE Edmond-Prosper - 19B057FS

19B057FS.sql

```
1  --Creation de la base de données
2  --SOUAPIEBE Edmond-Prosper 19B075FS
3  CREATE DATABASE IF NOT EXISTS `19B057FS`;
4  USE `19B057FS`;
5
6  -- Cree la table `Legume` avec pour clé primaire `id`
7  CREATE TABLE `Legume` (
8      `id` INT(10) NOT NULL AUTO_INCREMENT,
9      `nom` VARCHAR(255) NOT NULL,
10     `couleur` VARCHAR(255),
11     PRIMARY KEY(`id`)
12 );
13
14 -- Cree la table `Region` avec pour clé primaire `id`
15 CREATE TABLE `Region` (
16     `id` INT(10) NOT NULL AUTO_INCREMENT,
17     `nom` VARCHAR(255) NOT NULL,
18     `population` INT(10) NOT NULL,
19     PRIMARY KEY(`id`)
20 );
21
22 -- Cree la table `Cultiver` qui est le resultat de
23 -- l'association `Legume-Region` avec pour clé primaire `id`
24 -- et pour clés étrangères `idLegume` et `idRegion`
25 CREATE TABLE `Cultiver` (
26     `id` INT(10) NOT NULL AUTO_INCREMENT,
27     `idLegume` INT(10) NOT NULL,
28     `idRegion` INT(10) NOT NULL,
29     PRIMARY KEY(`id`)
30 );
31
32 ALTER TABLE `Cultiver` ADD CONSTRAINT `fk_legume_cultiver`
33 FOREIGN KEY(`idLegume`) REFERENCES `Legume`(`id`)
34 ON DELETE CASCADE ON UPDATE RESTRICT;
35
36 ALTER TABLE `Cultiver` ADD CONSTRAINT `fk_region_cultiver`
37 FOREIGN KEY(`idRegion`) REFERENCES `Region`(`id`)
38 ON DELETE CASCADE ON UPDATE RESTRICT;
39
40 -- Creation des index sur les champs `nom` des tables
41 CREATE INDEX `indexNomLegume` ON `Legume` (`nom`);
42 CREATE INDEX `indexNomRegion` ON `Region` (`nom`);
43
44 -- Creation des TRIGGERS
45
46 -- Trigger qui modifie le nom avant l'insertion dans la
47 -- table Legume lorsque le nom du legume est vide
48 CREATE TRIGGER `declencheur_insert`
49 BEFORE INSERT ON `Legume` FOR EACH ROW
50 BEGIN
51     IF NEW.nom = '' OR NEW.nom = 'sans nom' THEN
52         SET NEW.nom = 'legume_sans_nom';
53     END IF;
54 END;
```

```
55
56 -- Trigger qui modifie le nom du legume en `legume_fruit`
57 -- lorsque lors d'un UPDATE nom = anana ou a mangue
58 CREATE TRIGGER `declencheur_update`
59 BEFORE UPDATE ON `Legume` FOR EACH ROW
60 BEGIN
61     IF NEW.nom = 'anana' OR NEW.nom = 'mangue'
62     THEN SET NEW.nom = 'legume_fruit';
63     END IF;
64 END;
65
66 -- Trigger qui avant toute suppression dans la table
67 -- Region, supprime aussi les relations entre un
68 -- legume et cette region
69 CREATE TRIGGER `declencheur_delete`
70 BEFORE DELETE ON `Region` FOR EACH ROW
71 BEGIN
72     DELETE FROM `Cultiver` WHERE `idRegion` = OLD.id;
73 END;
74
75 -- Fin de la creation de la bd 19B057FS --
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0217 seconds.)

-- Creation de la base de données -- SOUAPIEBE Edmond-Prosper 19B075FS

[ Edit inline ] [ Edit ] [ Create PHP code ]

⚠ Error: #1046 No database selected

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0012 seconds.)

USE `19B057FS`;

[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.3854 seconds.)

-- Cree la table `Legume` avec pour clé primaire `id` `CREATE TABLE `Legume` (`id` INT(10) NOT NULL AUTO_INCREMENT, `nom` VARCHAR(255) NOT NULL, `couleur` VARCHAR(255), PRIMARY KEY(`id`));`

[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.3201 seconds.)

-- Cree la table `Region` avec pour clé primaire `id` `CREATE TABLE `Region` (`id` INT(10) NOT NULL AUTO_INCREMENT, `nom` VARCHAR(255) NOT NULL, `population` INT(10) NOT NULL, PRIMARY KEY(`id`));`

Your SQL query has been executed successfully.

`SHOW INDEX FROM `legume`;`

☐ Profiling [ Edit inline ] [ Edit ] [ Create PHP code ] [ Refresh ]

+ Options

Table	Non_unique	Key_name	Seq_in_index	Column_name
legume	0	PRIMARY	1	id
legume	1	indexNomLegume	1	nom

Your SQL query has been executed successfully.

`SHOW INDEX FROM `region`;`

☐ Profiling [ Edit inline ] [ Edit ] [ Create PHP code ] [ Refresh ]

+ Options

Table	Non_unique	Key_name	Seq_in_index	Column_name
region	0	PRIMARY	1	id
region	1	indexNomRegion	1	nom

## CREATION DES DECLENCHEURS

✓ Trigger `declencheur\_insert` has been created.

```
CREATE TRIGGER `declencheur_insert` BEFORE INSERT ON `legume`  
= 'legume_sans_nom'; END IF; END;
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ Trigger `declencheur\_update` has been created.

```
CREATE TRIGGER `declencheur_update` BEFORE UPDATE ON `legume`  
= 'legume_fruit'; END IF; END;
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ Trigger `declencheur\_delete` has been created.

```
CREATE TRIGGER `declencheur_delete` BEFORE DELETE ON `region`
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

## TEST D'UN DECLENCHEUR

✓ Showing rows 0 - 1 (2 total, Query took 0.0028 seconds.)

```
SELECT * FROM `legume`
```







☐ Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

```
UPDATE `legume` SET `nom` = 'mangue' WHERE `legume`.`id` = 1;
```

[ Edit inline ] [ Edit ] [ Create PHP code ]

☐ Show all | Number of rows: 25 ▾ Filter rows:

+ Options

					id	nom	couleur
<input type="checkbox"/>		Edit		Copy		Delete 1	legume_fruit vert
<input type="checkbox"/>		Edit		Copy		Delete 2	tasba vert