

CLEF PRIMAIRE et étrangère

PRIMARY KEY clef primaire, elle peut être composite (plusieurs nom de colonnes) <=> **UNIQUE NOT NULL INDEX** Rappelons qu'un **INDEX** est une structure supplémentaire créé par MySQL pour garder l'ordre des colonnes indexées.

FOREIGN KEY clef étrangère (éventuellement composite) elle se réfère à un **INDEX** ou clef primaire. Attention, on doit utiliser le moteur **InnoDB**, **MyISAM** n'est pas relationnel (pas de clef étrangère).

Elle permet de ne pas insérer de données qui n'auraient pas de sens dans les tables. **Elle permet l'intégrité des données.**

Pour créer une clef étrangère on doit :

- Choisir la ou les colonnes **FOREIGN KEY**
- Choisir dans l'autre table la ou les colonnes qui va/vont servir de référence(s), **REFERENCES**. La clef étrangère ne peut s'ajouter dans la description d'une colonne, on met sa définition en général en dessous de la clef primaire.

Contraintes à connaître sur les tables

La table pilots est physiquement relié à la table compagnies.

- Les contraintes sont les suivantes :
1. Vous ne pouvez pas ajouter ou modifier une ligne dans la table pilots dont la compagnie n'existe pas dans la table compagnies.
 2. Vous ne pouvez pas supprimer ou modifier une compagnie qui aurait des/une référence(s) dans la table pilots.

```
CREATE TABLE `pilots` (  
  `certificate` varchar(6) NOT NULL,  
  `numFlying` decimal(7,1),  
  `compagny` char(4),  
  `name` varchar(20) NOT NULL,  
  `plane` char(5),  
  `bonus` decimal(5,1),  
  `num_jobs` tinyint(3) unsigned,  
  `next_flight` datetime,  
  `birth_date` datetime,  
  `created` datetime DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (`certificate`),  
  UNIQUE KEY `un_name` (`name`),  
  KEY `fk_pilots_compagny` (`compagny`),
```

```

    CONSTRAINT `fk_pilots_compagny` FOREIGN KEY (`compagny`) REFERENCES `compagnies` (`comp`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci

```

Table compagnies :

```

CREATE TABLE `compagnies` (
  `comp` char(4) NOT NULL,
  `street` varchar(20) DEFAULT NULL,
  `city` varchar(20) DEFAULT NULL,
  `name` varchar(20) NOT NULL,
  `numStreet` tinyint(3) unsigned DEFAULT NULL,
  `status` enum('published','unpublished','draft') DEFAULT 'draft',
  PRIMARY KEY (`comp`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

Options des clefs étrangères

On a deux types : DELETE et UPDATE.

ON DELETE [RESTRICT| NO ACTION | SET NULL | CASCADE] permet de déterminer le comportement de MySQL en cas de suppression d'une référence.

ON UPDATE [RESTRICT| NO ACTION | SET NULL | CASCADE] permet de déterminer le comportement de MySQL en cas de modification d'une référence.

ON DELETE RESTRICT et NO ACTION comportement par défaut, ces deux options on le même effet dans MySQL uniquement mais pas dans les autres SGDB.

ON DELETE SET NULL dans ce cas NULL est substitué aux valeurs dont la référence est supprimée.

Exemple : si on a ON DELETE SET NULL sur la clef étrangère de la table pilots, et si on supprime une compagnie alors les pilotes qui avaient cette référence (comp de la table compagnies supprimé) ne sont pas supprimés et la valeur NULL est mise à la place de l'identifiant de la table compagnies dans la table pilots.

ON DELETE CASCADE comportement plus risqué, plus violent ! Elle supprime toutes les lignes. Par exemple si on supprime une compagnie dans la table compagnies reliée à la table pilots (c'est dans cette table que l'on a la clef étrangère) alors tous les pilotes référencés seront également supprimés.

Moins utiliser UPDATE RESTRICT et NO ACTION : empêche la modification si elle casse la contrainte (comportement par défaut).

SET NULL : met NULL partout où la valeur modifiée était référencée. CASCADE : modifie également la valeur là où elle est référencée.