Les contraintes d'intégrité

- Les **contraintes d'intégrité** aident à la cohérence et à l'exactitude des données.
- Elles font partie de la définition des tables.

NOT NULL	une valeur doit être définie
UNIQUE	une valeur n'apparaît qu'une seule fois dans une colonne
DEFAULT	valeur par défaut
CHECK	vérifier une assertion logique (pas avec MySQL)
PRIMARY KEY	clé primaire
FOREIGN KEY	clé étrangère

Exemple d'utilisation de UNIQUE

```
CREATE TABLE Students (
   id   INT NOT NULL AUTO_INCREMENT,
   firstName VARCHAR(20),
   dateOfBirth DATE,
   email  VARCHAR(50) UNIQUE,
   PRIMARY KEY(id)
);
```

```
mysql> INSERT INTO Students VALUES (NULL,'Linda','1996-08-22','lynda@u-pec.fr')
Query OK, 1 row affected (0,01 sec)

mysql> INSERT INTO Students VALUES (NULL,'Linda','1997-05-03','lynda@u-pec.fr')
ERROR 1062 (23000): Duplicate entry 'lynda@u-pec.fr' for key 'email'
```

Exemple d'utilisation de DEFAULT

```
CREATE TABLE Students (
   id   INT NOT NULL AUTO_INCREMENT,
   firstName VARCHAR(20),
   dateOfBirth DATE,
   email  VARCHAR(50) UNIQUE,
   dateOfCreation DATETIME DEFAULT NOW(),
  PRIMARY KEY(id)
);
```

```
mysql> INSERT INTO Students(firstName,email) VALUES ('Jenny','jenny2@u-pec.fr');
Query OK, 1 row affected (0,01 sec)

mysql> SELECT * FROM Students;
+----+-----------+-------------+-----------------+---------------------+
| id | firstName | dateOfBirth | email           | dateOfCreation      |
+----+-----------+-------------+-----------------+---------------------+
|  1 | Jenny     | NULL        | jenny2@u-pec.fr | 2016-09-22 13:35:57 |
+----+-----------+-------------+-----------------+---------------------+
```

Exemple d'utilisation de CHECK

```
CREATE TABLE Students (
   id  INT NOT NULL AUTO_INCREMENT,
   firstName VARCHAR(20),
   dateOfBirth DATE,
   email  VARCHAR(50) UNIQUE,
   dateOfCreation DATETIME DEFAULT NOW(),
  PRIMARY KEY(id),
  CHECK(id >0)
);
```

```
mysql> INSERT INTO Students(id, firstName,email) VALUES (-2,'Jenny','jenny3@u-pec.fr');
mysql> SELECT * FROM Students;
+----+-----------+-------------+----------------+---------------------+
| id | firstName | dateOfBirth | email          | dateOfCreation      |
+----+-----------+-------------+----------------+---------------------+
|  1 | Jenny     | NULL        | jenny2@u-pec.fr | 2016-09-22 13:35:57 |
+----+-----------+-------------+----------------+---------------------+
| -2 | Jenny     | NULL        | jenny3@u-pec.fr | 2016-09-22 13:42:45 |
+----+-----------+-------------+----------------+---------------------+
```

- L'insertion n'est pas faite avec SQL Server ou Oracle, mais elle est faite avec MySQL.

Produit cartésien

Le produit cartésien de deux ensembles A et B : $A \times B = \{(a, b) | a \in A, b \in B\}$.

Le produit cartésien de deux tables R_1 et R_2

Table obtenue avec les attributs de R_1 et les attributs de R_2 et toutes les combinaisons possibles entre les tuples de R_1 et les tuples de R_2.

- Exemple : `personnes`

nom	prenom
Bertin	Pierre
Durand	Sophie

et `institutions`

nom	ville
université	Paris
hôpital	Lyon

- Le calcul du produit cartésien s'effectue avec l'opération **CROSS JOIN**.

```
mysql> SELECT * FROM personnes CROSS JOIN institutions;
+--------+--------+------------+-------+
| nom    | prenom | nom        | ville |
+--------+--------+------------+-------+
| Bertin | Pierre | université | Paris |
| Durand | Sophie | université | Paris |
| Bertin | Pierre | hôpital    | Lyon  |
| Durand | Sophie | hôpital    | Lyon  |
+--------+--------+------------+-------+
```

Clé étrangère

FOREIGN KEY

Une **clé** étrangère est un attribut ou un ensemble d'attributs d'une table qui dont les valeurs sont celles d'une clé primaire d'une autre table.

- Elle établit une relation entre les tables.
- Ses valeurs peuvent apparaître plusieurs fois ou être **NULL**.

```
CREATE TABLE toys (
  toy_id INT
    NOT NULL
     auto_increment,
  toy VARCHAR(20) i
     DEFAULT NULL,
  PRIMARY KEY(toy_id)
);
```

```
mysql> SELECT * FROM toys;
+--------+----------------+
| toy_id | toy            |
+--------+----------------+
|      1 | hula hoop      |
|      2 | balsa glider   |
|      3 | toy soldiers   |
|      4 | harmonica      |
|      5 | baseball cards |
|      6 | tinker toys    |
|      7 | etch-a-sketch  |
|      8 | slinky         |
+--------+----------------+
```

```
CREATE TABLE boys (
  boy_id INT NOT NULL auto_increment,
  boy VARCHAR(20) DEFAULT NULL,
  toy_id INT DEFAULT NULL,
  PRIMARY KEY(boy_id),
  FOREIGN KEY(toy_id) REFERENCES toys(toy_id)
);
```

```
mysql> SELECT * FROM boys;
+--------+--------+--------+
| boy_id | boy    | toy_id |
+--------+--------+--------+
|      1 | Davey  |      3 |
|      2 | Bobby  |      5 |
|      3 | Beaver |      2 |
|      4 | Richie |      1 |
|      6 | Johnny |      4 |
|      5 | Billy  |      2 |
|      7 | Tony   |   NULL |
+--------+--------+--------+
```

Jointure

JOIN

- L'opérateur **JOIN** permet de combiner les données de deux tables en ajoutant des conditions pour ne pas obtenir tout le produit cartésien.
- La jointure la plus courante est la **jointure interne d'égalité** ou **équi-jointure**.

Exemple de jointure interne d'égalité :

- **SELECT** boys.boy, toys.toy **FROM** boys **INNER JOIN** toys **on** boys.toy_id=toys.toy_id ;
  (ou ancienne syntaxe **SELECT** boys.boy,toys.toy **FROM** boys,toys **WHERE** boys.toy_id=toys.toy_id;)

```
mysql> SELECT boys.boy,toys.toy FROM boys INNER JOIN toys ON boys.toy_id=toys.toy_id;
+--------+-----------------+
| boy    | toy             |
+--------+-----------------+
| Davey  | toy soldiers    |
| Bobby  | baseball cards  |
| Beaver | balsa glider    |
| Richie | hula hoop       |
| Johnny | harmonica       |
| Billy  | balsa glider    |
+--------+-----------------+
mysql> SELECT boys.boy,boys.toy_id,toys.toy_id,toys.toy FROM boys INNER JOIN toys  ON boys.toy_id=toys.toy_id;
+--------+--------+--------+-----------------+
| boy    | toy_id | toy_id | toy             |
+--------+--------+--------+-----------------+
| Davey  |      3 |      3 | toy soldiers    |
| Bobby  |      5 |      5 | baseball cards  |
| Beaver |      2 |      2 | balsa glider    |
| Richie |      1 |      1 | hula hoop       |
| Johnny |      4 |      4 | harmonica       |
| Billy  |      2 |      2 | balsa glider    |
+--------+--------+--------+-----------------+
```

```
mysql> SELECT * FROM boys;
+--------+--------+--------+
| boy_id | boy    | toy_id |
+--------+--------+--------+
|      1 | Davey  |      3 |
|      2 | Bobby  |      5 |
|      3 | Beaver |      2 |
|      4 | Richie |      1 |
|      6 | Johnny |      4 |
|      5 | Billy  |      2 |
|      7 | Tony   |   NULL |
+--------+--------+--------+
```

```
mysql> SELECT * FROM toys;
+--------+-----------------+
| toy_id | toy             |
+--------+-----------------+
|      1 | hula hoop       |
|      2 | balsa glider    |
|      3 | toy soldiers    |
|      4 | harmonica       |
|      5 | baseball cards  |
|      6 | tinker toys     |
|      7 | etch-a-sketch   |
|      8 | slinky          |
+--------+-----------------+
```

Les différentes jointures internes

INNER JOIN

- On peut faire des jointures avec les autres opérateurs de comparaison.

SELECT boys.boy, toys.toy **FROM**
boys **INNER JOIN** toys **on** boys.toy_id<>toys.toy_id
WHERE boy='billy';

```
+-------+----------------+
| boy   | toy            |
+-------+----------------+
| Billy | hula hoop      |
| Billy | toy soldiers   |
| Billy | harmonica      |
| Billy | baseball cards |
| Billy | tinker toys    |
| Billy | etch-a-sketch  |
| Billy | slinky         |
+-------+----------------+
```

```
mysql> SELECT * FROM boys;
+--------+--------+--------+
| boy_id | boy    | toy_id |
+--------+--------+--------+
|      1 | Davey  |      3 |
|      2 | Bobby  |      5 |
|      3 | Beaver |      2 |
|      4 | Richie |      1 |
|      6 | Johnny |      4 |
|      5 | Billy  |      2 |
|      7 | Tony   |   NULL |
+--------+--------+--------+
```

```
mysql> SELECT * FROM toys;
+--------+----------------+
| toy_id | toy            |
+--------+----------------+
|      1 | hula hoop      |
|      2 | balsa glider   |
|      3 | toy soldiers   |
|      4 | harmonica      |
|      5 | baseball cards |
|      6 | tinker toys    |
|      7 | etch-a-sketch  |
|      8 | slinky         |
+--------+----------------+
```

Jointure externe à gauche

LEFT OUTER JOIN

- On prend tous les éléments de la table de gauche.

SELECT boys.boy, toys.toy **FROM**
boys **LEFT OUTER JOIN** toys **on** boys.toy_id=toys.toy_id ;

```
+--------+----------------+
| boy    | toy            |
+--------+----------------+
| Davey  | toy soldiers   |
| Bobby  | baseball cards |
| Beaver | balsa glider   |
| Richie | hula hoop      |
| Johnny | harmonica      |
| Billy  | balsa glider   |
| Tony   | NULL           |
+--------+----------------+
```

```
mysql> SELECT * FROM boys;
+--------+--------+--------+
| boy_id | boy    | toy_id |
+--------+--------+--------+
|      1 | Davey  |      3 |
|      2 | Bobby  |      5 |
|      3 | Beaver |      2 |
|      4 | Richie |      1 |
|      6 | Johnny |      4 |
|      5 | Billy  |      2 |
|      7 | Tony   |   NULL |
+--------+--------+--------+
```

```
mysql> SELECT * FROM toys;
+--------+----------------+
| toy_id | toy            |
+--------+----------------+
|      1 | hula hoop      |
|      2 | balsa glider   |
|      3 | toy soldiers   |
|      4 | harmonica      |
|      5 | baseball cards |
|      6 | tinker toys    |
|      7 | etch-a-sketch  |
|      8 | slinky         |
+--------+----------------+
```

Jointure externe à droite

RIGHT OUTER JOIN

- On prend tous les éléments de la table de droite.

SELECT boys.boy,toys.toy **FROM**
boys **RIGHT OUTER JOIN** toys **on** boys.toy_id=toys.toy_id ;

```
+--------+----------------+
| boy    | toy            |
+--------+----------------+
| Richie | hula hoop      |
| Beaver | balsa glider   |
| Billy  | balsa glider   |
| Davey  | toy soldiers   |
| Johnny | harmonica      |
| Bobby  | baseball cards |
| NULL   | tinker toys    |
| NULL   | etch-a-sketch  |
| NULL   | slinky         |
+--------+----------------+
```

```
mysql> SELECT * FROM boys;
+--------+--------+--------+
| boy_id | boy    | toy_id |
+--------+--------+--------+
|      1 | Davey  |      3 |
|      2 | Bobby  |      5 |
|      3 | Beaver |      2 |
|      4 | Richie |      1 |
|      6 | Johnny |      4 |
|      5 | Billy  |      2 |
|      7 | Tony   |   NULL |
+--------+--------+--------+
```

```
mysql> SELECT * FROM toys;
+--------+----------------+
| toy_id | toy            |
+--------+----------------+
|      1 | hula hoop      |
|      2 | balsa glider   |
|      3 | toy soldiers   |
|      4 | harmonica      |
|      5 | baseball cards |
|      6 | tinker toys    |
|      7 | etch-a-sketch  |
|      8 | slinky         |
+--------+----------------+
```

Jointure externe des deux côtés

FULL OUTER JOIN

- On prend tous les éléments des deux tables.
- N'est pas supporté par MySQL. On la construit avec **UNION**.

```
(SELECT boys.boy, toys.toy FROM boys
  LEFT OUTER JOIN toys on boys.toy_id=toys.toy_id)
UNION
(SELECT boys.boy, toys.toy FROM boys
  RIGHT OUTER JOIN toys on boys.toy_id=toys.toy_id);
```

```
+--------+----------------+
| boy    | toy            |
+--------+----------------+
| Davey  | toy soldiers   |
| Bobby  | baseball cards |
| Beaver | balsa glider   |
| Richie | hula hoop      |
| Johnny | harmonica      |
| Billy  | balsa glider   |
| Tony   | NULL           |
| NULL   | tinker toys    |
| NULL   | etch-a-sketch  |
| NULL   | slinky         |
+--------+----------------+
```

```
mysql> SELECT * FROM boys;
+--------+--------+--------+
| boy_id | boy    | toy_id |
+--------+--------+--------+
|      1 | Davey  |      3 |
|      2 | Bobby  |      5 |
|      3 | Beaver |      2 |
|      4 | Richie |      1 |
|      6 | Johnny |      4 |
|      5 | Billy  |      2 |
|      7 | Tony   |   NULL |
+--------+--------+--------+
```

```
mysql> SELECT * FROM toys;
+--------+----------------+
| toy_id | toy            |
+--------+----------------+
|      1 | hula hoop      |
|      2 | balsa glider   |
|      3 | toy soldiers   |
|      4 | harmonica      |
|      5 | baseball cards |
|      6 | tinker toys    |
|      7 | etch-a-sketch  |
|      8 | slinky         |
+--------+----------------+
```

Un enfant, plusieurs jouets

```
CREATE TABLE kids (
  kid_id INT NOT NULL auto_increment,
  kid VARCHAR(20) NOT NULL, i
  PRIMARY KEY (kid_id) i
);
```

```
mysql> SELECT * FROM kids;
+--------+--------+
| kid_id | kid    |
+--------+--------+
|      1 | maria  |
|      2 | sofia  |
|      3 | kelly  |
|      4 | john   |
|      5 | Philip |
|      6 | Adam   |
|      7 | Ryan   |
+--------+--------+
```

```
CREATE TABLE kidsOwnToys (
  kid_id INT NOT NULL,
  toy_id INT NOT NULL,
  PRIMARY KEY(kid_id,toy_id),
  FOREIGN KEY(kid_id) REFERENCES kids(kid_id),
  FOREIGN KEY(toy_id) REFERENCES toys(toy_id) i
);
```

```
mysql> SELECT * FROM kidsOwnToys;
+--------+--------+
| kid_id | toy_id |
+--------+--------+
|      1 |      3 |
|      3 |      4 |
|      1 |      5 |
|      3 |      5 |
|      4 |      8 |
+--------+--------+
mysql> SELECT * FROM toys;
+--------+----------------+
| toy_id | toy            |
+--------+----------------+
|      1 | hula hoop      |
|      2 | balsa glider   |
|      3 | toy soldiers   |
|      4 | harmonica      |
|      5 | baseball cards |
|      6 | tinker toys    |
|      7 | etch-a-sketch  |
|      8 | slinky         |
+--------+----------------+
```

```
SELECT kid,toy FROM kids
  INNER JOIN kidsOwnToys
    ON kids.kid_id=kidsOwnToys.kid_id
  INNER JOIN toys
    ON kidsOwnToys.toy_id=toys.toy_id;
```

```
+-------+----------------+
| kid   | toy            |
+-------+----------------+
| maria | toy soldiers   |
| kelly | harmonica      |
| maria | baseball cards |
| kelly | baseball cards |
| john  | slinky         |
+-------+----------------+
```

Auto-jointure

```
CREATE TABLE employee (
  id INT NOT NULL auto_increment,
  name VARCHAR(20) NOT NULL,
  manager_id INT,
  PRIMARY KEY (id)
);
```

```
mysql> SELECT * FROM employee;
+----+--------+------------+
| id | name   | manager_id |
+----+--------+------------+
|  1 | maria  |          5 |
|  2 | sofia  |          3 |
|  3 | kelly  |       NULL |
|  4 | john   |          6 |
|  5 | Philip |          3 |
|  6 | Adam   |          3 |
|  7 | Ryan   |          6 |
+----+--------+------------+
```

```
SELECT e.name AS Employee,
  m.name AS Manager
FROM employee e
  INNER JOIN employee m
    ON m.id=e.manager_id;
```

```
+----------+---------+
| Employee | Manager |
+----------+---------+
| maria    | Philip  |
| sofia    | kelly   |
| john     | Adam    |
| Philip   | kelly   |
| Adam     | kelly   |
| Ryan     | Adam    |
+----------+---------+
```

```
SELECT e.name AS Employee,
  IFNULL(m.name, 'Top_Manager') AS Manager
FROM employee e
  LEFT OUTER JOIN employee m
    ON m.id=e.manager_id;
```

```
+----------+-------------+
| Employee | Manager     |
+----------+-------------+
| maria    | Philip      |
| sofia    | kelly       |
| kelly    | Top Manager |
| john     | Adam        |
| Philip   | kelly       |
| Adam     | kelly       |
| Ryan     | Adam        |
+----------+-------------+
```

Effacer des enregistrements

DELETE

- **DELETE FROM** *myTable* **WHERE** *myConditions*
- Si on ne met pas de clause **WHERE**, tous les enregistrements de la table sont effacés.

DELETE FROM doughnut_ratings **WHERE date** < '2007-06-01';

```
mysql> SELECT  * FROM doughnut_ratings;
+-----------------+----------+------------+--------------+--------+----------------+
| location        | time     | date       | type         | rating | comments       |
+-----------------+----------+------------+--------------+--------+----------------+
| Krispy King     | 08:50:00 | 2007-09-27 | plain glazed |     10 | almost perfect |
| Duncan's Donuts | 08:59:00 | 2007-08-25 | NULL         |      6 | greasy         |
+-----------------+----------+------------+--------------+--------+----------------+
```

```
mysql> SELECT * FROM doughnut_ratings;
+-----------------+----------+------------+--------------+--------+-----------------+
| location        | time     | date       | type         | rating | comments        |
+-----------------+----------+------------+--------------+--------+-----------------+
| Krispy King     | 08:50:00 | 2007-09-27 | plain glazed |     10 | almost perfect  |
| Duncan's Donuts | 08:59:00 | 2007-08-25 | NULL         |      6 | greasy          |
| Starbuzz Coffee | 07:35:00 | 2007-05-24 | cinnamon cake |      5 | stale, but tasty |
| Duncan's Donuts | 07:03:00 | 2007-04-26 | jelly        |      7 | not enough jelly |
+-----------------+----------+------------+--------------+--------+-----------------+
```

Mettre à jour des enregistrements

UPDATE

- **UPDATE** *myTable* **SET** *myColumn1*= *myValue1*,*myColumn2*=*myValue2*,... **WHERE** *myConditions*

```
UPDATE doughnut_ratings SET type='chocolate iced glazed'
  WHERE type IS NULL;
```

```
mysql> SELECT * FROM doughnut_ratings;
+-----------------+----------+------------+----------------------+--------+------------------+
| location        | time     | date       | type                 | rating | comments         |
+-----------------+----------+------------+----------------------+--------+------------------+
| Krispy King     | 08:50:00 | 2007-09-27 | plain glazed         |     10 | almost perfect   |
| Duncan's Donuts | 08:59:00 | 2007-08-25 | chocolate iced glazed |      6 | greasy           |
| Starbuzz Coffee | 07:35:00 | 2007-05-24 | cinnamon cake        |      5 | stale, but tasty |
| Duncan's Donuts | 07:03:00 | 2007-04-26 | jelly                |      7 | not enough jelly |
+-----------------+----------+------------+----------------------+--------+------------------+
```

```
mysql> SELECT * FROM doughnut_ratings;
+-----------------+----------+------------+---------------+--------+------------------+
| location        | time     | date       | type          | rating | comments         |
+-----------------+----------+------------+---------------+--------+------------------+
| Krispy King     | 08:50:00 | 2007-09-27 | plain glazed  |     10 | almost perfect   |
| Duncan's Donuts | 08:59:00 | 2007-08-25 | NULL          |      6 | greasy           |
| Starbuzz Coffee | 07:35:00 | 2007-05-24 | cinnamon cake |      5 | stale, but tasty |
| Duncan's Donuts | 07:03:00 | 2007-04-26 | jelly         |      7 | not enough jelly |
+-----------------+----------+------------+---------------+--------+------------------+
```

Modifier une table

ALTER

- **ALTER TABLE** *myTable instructions*
- La syntaxe des instructions dépend des SGBDs.

DROP	supprimer une colonne ou une clé
ADD	ajouter une colonne
CHANGE	modifier le type et le nom d'une colonne
MODIFY	modifier le type ou la position d'une colonne
RENAME	renommer une table

```
mysql> DESC  Students;
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| id          | int(11)     | NO   | PRI | NULL    | auto_increment |
| firstName   | varchar(20) | YES  |     | NULL    |                |
| dateOfBirth | date        | YES  |     | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
```

```
ALTER TABLE Students DROP dateOfBirth;
```

```
mysql> DESC Students;
+-----------+-------------+------+-----+---------+----------------+
| Field     | Type        | Null | Key | Default | Extra          |
+-----------+-------------+------+-----+---------+----------------+
| id        | int(11)     | NO   | PRI | NULL    | auto_increment |
| firstName | varchar(20) | YES  |     | NULL    |                |
+-----------+-------------+------+-----+---------+----------------+
```

Enlever une clé étrangère et une colonne

```
SHOW CREATE TABLE boys;
+-------+-----------------------------------------------------------------------+
| boys  | CREATE TABLE `boys` (
  `boy_id` int(11) NOT NULL AUTO_INCREMENT,
  `boy` varchar(20) COLLATE utf8_unicode_ci DEFAULT NULL,
  `toy_id` int(11) DEFAULT NULL,
  PRIMARY KEY (`boy_id`),
  KEY `toy_id` (`toy_id`),
  CONSTRAINT `boys_ibfk_1` FOREIGN KEY (`toy_id`) REFERENCES `toys` (`toy_id`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci |
+-------+-----------------------------------------------------------------------+
```

```
ALTER TABLE boys DROP FOREIGN KEY boys_ibfk_1;
ALTER TABLE boys DROP toy_id;
```

```
+-------+-----------------------------------------------------------------------+
mysql> SHOW CREATE TABLE boys;
| boys  | CREATE TABLE `boys` (
  `boy_id` int(11) NOT NULL AUTO_INCREMENT,
  `boy` varchar(20) COLLATE utf8_unicode_ci DEFAULT NULL,
  PRIMARY KEY (`boy_id`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci |
+-------+-----------------------------------------------------------------------+
```

Ajouter une clé et une colonne

```
mysql> SHOW CREATE TABLE Modules;
+---------+-----------------------------------------------------------+
| Modules | CREATE TABLE `Modules` (
  `moduleName` varchar(50) CHARACTER SET utf8 DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+---------+-----------------------------------------------------------+
```

```
ALTER TABLE Modules
  ADD id INT NOT NULL AUTO_INCREMENT,
  ADD PRIMARY KEY(id);
```

```
+---------+-----------------------------------------------------------+
mysql> SHOW CREATE TABLE Modules;
| Modules | CREATE TABLE `Modules` (
  `moduleName` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
  `id` int(11) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+---------+-----------------------------------------------------------+
```

Modifier le nom et le type d'une colonne

```
+---------+-------------------------------------------------------------+
mysql> SHOW CREATE TABLE Modules;
| Modules | CREATE TABLE `Modules` (
  `moduleName` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
  `id` int(11) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+---------+-------------------------------------------------------------+
```

```
ALTER TABLE Modules
  CHANGE id moduleId TINYINT NOT NULL AUTO_INCREMENT;
```

```
mysql> SHOW CREATE TABLE Modules;
+---------+-------------------------------------------------------------+
| Modules | CREATE TABLE `Modules` (
  `moduleName` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
  `moduleId` tinyint(4) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`moduleId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+---------+-------------------------------------------------------------+
```

- On peut seulement diminuer la taille du type d'une colonne, si tous les enregistrements de la table sont compatibles avec la nouvelle taille.

Modifier le type et la position d'une colonne

```
mysql> SHOW CREATE TABLE Modules;
+---------+-----------------------------------------------------------------+
| Modules | CREATE TABLE `Modules` (
  `moduleName` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
  `moduleId` tinyint(4) NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`moduleId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+---------+-----------------------------------------------------------------+
```

```
ALTER TABLE Modules
  MODIFY moduleId INT NOT NULL AUTO_INCREMENT FIRST;
```

```
mysql> SHOW CREATE TABLE Modules;
+---------+-----------------------------------------------------------------+
| Modules | CREATE TABLE `Modules` (
  `moduleId` int(11) NOT NULL AUTO_INCREMENT,
  `moduleName` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
  PRIMARY KEY (`moduleId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+---------+-----------------------------------------------------------------+
```

```
ALTER TABLE Modules
  ADD Ects TINYINT AFTER moduleName;
```

```
mysql> SHOW CREATE TABLE Modules;
+---------+-----------------------------------------------------------------+
| Modules | CREATE TABLE `Modules` (
  `moduleId` int(11) NOT NULL AUTO_INCREMENT,
  `moduleName` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
  `Ects` tinyint(4) DEFAULT NULL,
  PRIMARY KEY (`moduleId`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+---------+-----------------------------------------------------------------+
```

Renommer une table

```
mysql> show tables;
+--------------------+
| Tables_in_courses2 |
+--------------------+
| Modules            |
| Students           |
+--------------------+
```

```
ALTER TABLE Students RENAME TO Undergraduates;
```

```
mysql> show tables;
+--------------------+
| Tables_in_courses2 |
+--------------------+
| Modules            |
| Undergraduates     |
+--------------------+
```