

Insertion des données

Nous allons aborder dans ce chapitre l'aspect LMD (langage de manipulation des données) de MySQL.

On a trois instructions :

- INSERT pour l'insertion d'enregistrement.
- UPDATE pour la modification de données.
- DELETE et TRUNCATE pour la suppression des données.

INSERT syntaxe

La syntaxe suivante permet l'insertion de données. Son inconvénient est qu'il faut renseigner toutes les colonnes pour quelle s'exécute correctement dans MySQL

```
INSERT INTO table  
VALUES ('valeur 1', 'valeur 2', ...);
```

Une autre syntaxe plus souple est la suivante, vous ne renseignez que les colonnes voulues. Attention cependant aux colonnes dont les valeurs sont obligatoires.

```
INSERT INTO table  
(nom_colonne_1, nom_colonne_2)  
VALUES ('valeur 1', 'valeur 2')
```

Dans ce cas seule les colonnes mentionnées seront enregistrées. Attention cependant aux colonnes dont les données sont obligatoires (NOT NULL).

Vous pouvez enregistrer plusieurs lignes à la fois :

```
INSERT INTO table (nom_colonne_1, nom_colonne_2, ...  
VALUES  
('valeur 1', 'valeur 2', ...),  
('valeur 11', 'valeur 22', ...),  
('valeur 111', 'valeur 222', ...),
```

Exercice insertion de données compagnies

Insérez les données suivantes dans la table compagnies :

comp	street	city	name	numStreet	status
AUS	sidney	Australie	AUSTRA Air	19	draft
CHI	chi	Chine	CHINA Air	NULL	draft
FRE1	beaubourg	France	Air France	17	draft
FRE2	paris	France	Air Electric	22	draft
SIN	pasir	Singapour	SIN A	15	draft

Exercice Ajouter une colonne created

Ajoutez la colonne created dans la table pilots. Ajoutez l'option suivante à cette colonne :

DEFAULT CURRENT_TIMESTAMP

Cette option permettra lors de l'insertion de spécifier la date de création d'un nouvel enregistrement.

Puis insérez les pilotes suivants :

certificate	numFlying	compagny	name	created
ct-1	90.0	AUS	Alan	2020-02-04 12:50:52
ct-10	90.0	FRE1	Tom	2020-02-04 12:50:52
ct-100	200.0	SIN	Yi	2020-02-04 12:50:52
ct-11	200.0	AUS	Sophie	2020-02-04 12:50:52
ct-12	190.0	AUS	Albert	2020-02-04 12:50:52
ct-16	190.0	SIN	Yan	2020-02-04 12:50:52
ct-56	300.0	AUS	Benoit	2020-02-04 12:50:52
ct-6	20.0	FRE1	Jhon	2020-02-04 12:50:52
ct-7	80.0	CHI	Pierre	2020-02-04 12:50:52

Exercice ajout d'une colonne et mise à jour

Ajoutez les colonnes birth_date, next_flight, num_jobs dans la table pilots. Trouvez les meilleurs types pour définir ces colonnes.

Mettez à jour les données manquantes en vous aidant du script ci-dessous. Vous trouverez dans le cours en ligne un fichier **Update SQL**. Il contient les données suivantes :

```
UPDATE `pilots`  
SET `birth_date` = '1978-02-04 00:00:00',  
    `next_flight` = '2020-12-04 09:50:52',  
    `num_jobs` = 10  
WHERE name = 'Yi';
```

```
UPDATE `pilots`  
SET `birth_date` = '1978-10-17 00:00:00',  
    `next_flight` = '2020-06-11 12:00:52',  
    `num_jobs` = 50  
WHERE name = 'Sophie';
```

```
UPDATE `pilots`  
SET `birth_date` = '1990-04-04 00:00:00',  
    `next_flight` = '2020-05-08 12:50:52',  
    `num_jobs` = 10  
WHERE name = 'Albert';
```

```
UPDATE `pilots`  
SET `birth_date` = '1998-01-04 00:00:00',  
    `next_flight` = '2020-05-08 12:50:52',  
    `num_jobs` = 30  
WHERE name = 'Yan';
```

```
UPDATE `pilots`  
SET `birth_date` = '2000-01-04 00:00:00',  
    `next_flight` = '2020-02-04 12:50:52',  
    `num_jobs` = 7  
WHERE name = 'Benoit';
```

```
UPDATE `pilots`  
SET `birth_date` = '2000-01-04 00:00:00',  
    `next_flight` = '2020-12-04 12:50:52',  
    `num_jobs` = 13  
WHERE name = 'Jhon';
```

```
UPDATE `pilots`  
SET `birth_date` = '1977-01-04 00:00:00',  
    `next_flight` = '2020-05-04 12:50:52',  
    `num_jobs` = 8  
WHERE name = 'Pierre';
```

name	num_jobs	next_flight	birth_date
Alan	30	2020-04-04 07:50:52	2001-03-04 00:00:00
Tom	10	2020-12-04 09:50:52	1978-02-04 00:00:00
Yi	50	2020-06-11 12:00:52	2000-04-14 00:00:00
Sophie	30	2020-10-14 12:50:52	1978-10-17 00:00:00
Albert	10	2020-05-08 12:50:52	1990-04-04 00:00:00
Yan	30	2020-04-04 12:50:52	1998-01-04 00:00:00
Benoit	7	2020-02-04 12:50:52	2000-01-04 00:00:00
Jhon	13	2020-12-04 12:50:52	2000-01-04 00:00:00
Pierre	8	2020-05-04 12:50:52	1977-01-04 00:00:00

Commande de suppression DELETE ou TRUNCATE

Ces deux commandes suppriment les enregistrements d'une table.

La commande truncate supprime des enregistrements en libérant éventuellement l'espace de stockage utilisé par la table.

La commande DELETE syntaxe :

```
DELETE FROM pilots [WHERE (condition)];
```

Différences entre DELETE et TRUNCATE

Truncate est similaire à DELETE. La commande TRUNCATE pour sa part ré-initialisera les valeurs auto-incrémentées. De plus TRUNCATE est plus optimisée dans son exécution et ne comporte pas de clause WHERE. Elle s'utilise seule :

```
TRUNCATE FROM pilots ;
```

Exercice sauvegarde et suppression (facultatif)

Créez la table new_pilots à l'aide de la commande suivante et supprimez toutes les lignes de la table pilots :

```
CREATE TABLE `new_pilots` (SELECT * FROM `pilots`);
```

Lorsque vous aurez supprimé les données de la table pilots récupérez les données de la table new_pilots pour les remettre dans la table pilots et supprimer la table new_pilots.

Utilisez la syntaxe suivante pour remettre les données dans la table pilots :

```
INSERT INTO
pilots (
    `certificate`,
    `numFlying`,
    `compagny`,
    `name`,
    `num_jobs`,
    `next_flight`,
    `birth_date`,
    `created`
) SELECT * FROM `new_pilots`;
```