

02 - Gestion d'une BDD MySQL

Ressources

L'écriture de scripts SQL pour créer des tables, insérer des données et effectuer des mises à jour constitue une compétence fondamentale. Cela permet de définir la structure de la base de données et de maintenir des données à jour.

Documentation basique : <https://www.w3schools.com/mysql/default.asp>

Documentation officielle : <https://www.mysqltutorial.org/>

Document ultime : <https://dev.mysql.com/doc/refman/8.0/en/>

Création d'une base de données

```
CREATE DATABASE [IF NOT EXISTS] database_name  
[CHARACTER SET charset_name]  
[COLLATE collation_name];
```

```
CREATE DATABASE testDB;
```

Création d'une table

```
CREATE TABLE productlines (  
    productLine varchar(50),  
    textDescription varchar(4000) DEFAULT NULL,  
    htmlDescription mediumtext,  
    image mediumblob,  
    PRIMARY KEY (productLine)  
);
```

Suppression d'une table

```
DROP TABLE IF EXISTS productlines;
```

Insertion de données dans une table

```
INSERT INTO  
    productlines(productLine, textDescription, htmlDescription, image)
```

```
VALUES ('Classic Cars',
```

```
        'Attention car enthusiasts: Make your wildest car  
ownership dreams come true. Whether you are looking for classic  
muscle cars, dream sports cars or movie-inspired miniatures, you  
will find great choices in this category. These replicas feature  
superb attention to detail and craftsmanship and offer features  
such as working steering system, opening forward compartment,  
opening rear trunk with removable spare wheel, 4-wheel independent  
spring suspension, and so on. The models range in size from 1:10  
to 1:24 scale and include numerous limited edition and several  
out-of-production vehicles. All models include a certificate of  
authenticity from their manufacturers and come fully assembled and  
ready for display in the home or office.',
```

```
        NULL,
```

```
        NULL) ;
```

Mise à jour de données dans une table

```
UPDATE productlines
```

```
    SET textDescription = 'Pour les fins connaisseurs !'
```

```
    WHERE productLine = 'Classic Cars';
```

Suppression de données dans une table

```
DELETE from productlines
```

```
    WHERE productCode = '10';
```

Clés primaires et Contraintes

L'utilisation d'index et de contraintes vise à optimiser les performances de la base de données et à garantir l'intégrité des données. Ces éléments contribuent à une gestion plus efficace et sécurisée.

```
CREATE TABLE products (
```

```
    productCode varchar(15),
```

```
    productName varchar(70) NOT NULL,
```

```
    productLine varchar(50) NOT NULL,
```

```
    productScale varchar(10) NOT NULL,
```

```
    productVendor varchar(50) NOT NULL,
```

```
    productDescription text NOT NULL,
```

```
    quantityInStock smallint(6) NOT NULL,
```

```
buyPrice decimal(10,2) NOT NULL,  
MSRP decimal(10,2) NOT NULL,  
PRIMARY KEY (productCode),  
FOREIGN KEY (productLine) REFERENCES productlines (productLine)  
);
```

On peut également, le faire en 2 étapes :

```
CREATE TABLE products (  
    productCode varchar(15),  
    productName varchar(70) NOT NULL,  
    productLine varchar(50) NOT NULL,  
    productScale varchar(10) NOT NULL,  
    productVendor varchar(50) NOT NULL,  
    productDescription text NOT NULL,  
    quantityInStock smallint(6) NOT NULL,  
    buyPrice decimal(10,2) NOT NULL,  
    MSRP decimal(10,2) NOT NULL,  
    PRIMARY KEY (productCode)  
) ENGINE=INNODB;
```

```
ALTER TABLE products (  
ADD  
    FOREIGN KEY (productLine) REFERENCES productlines (productLine)  
);
```

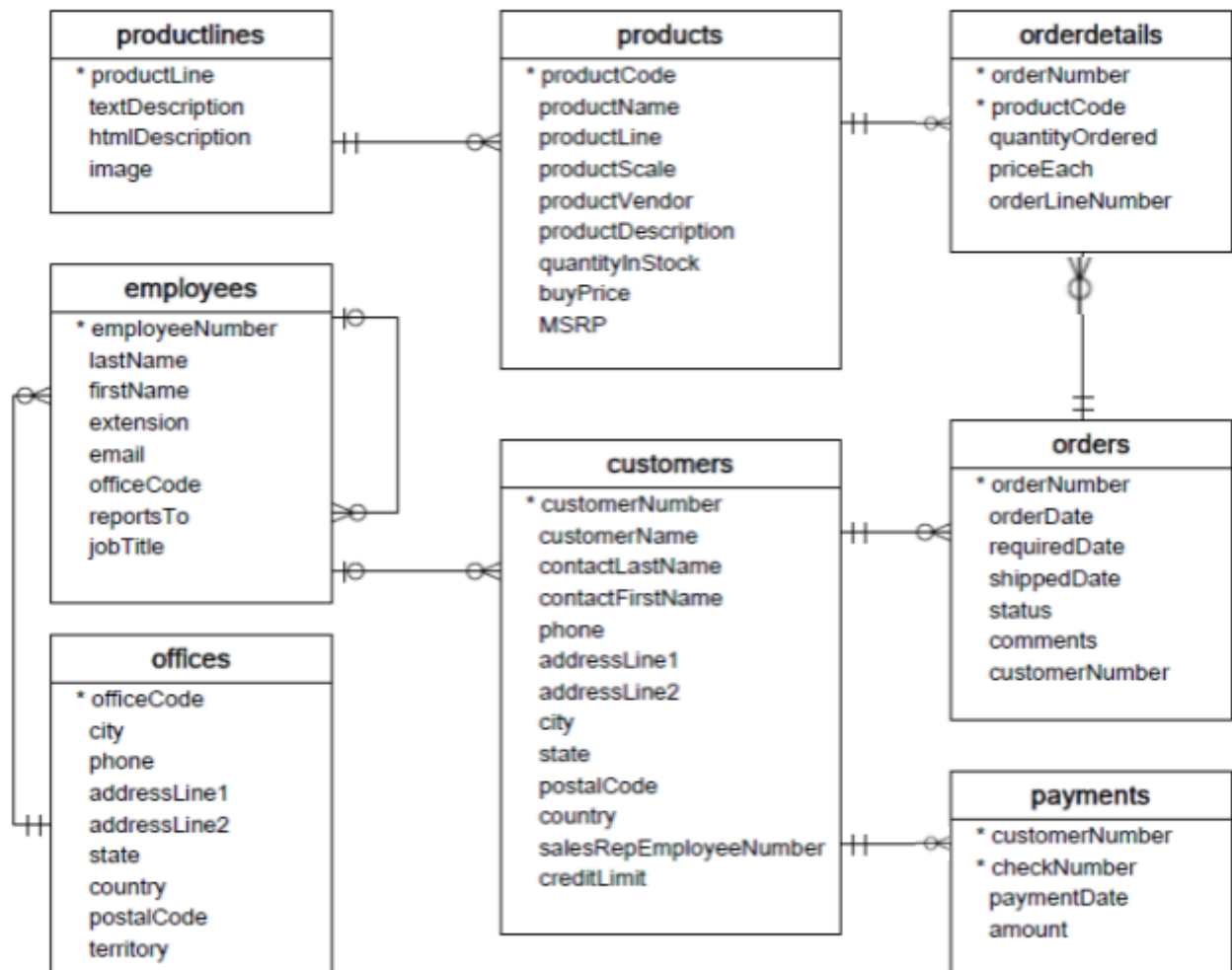
Exemple de base de données : “MySQL Sample Database”

Les tables de la base

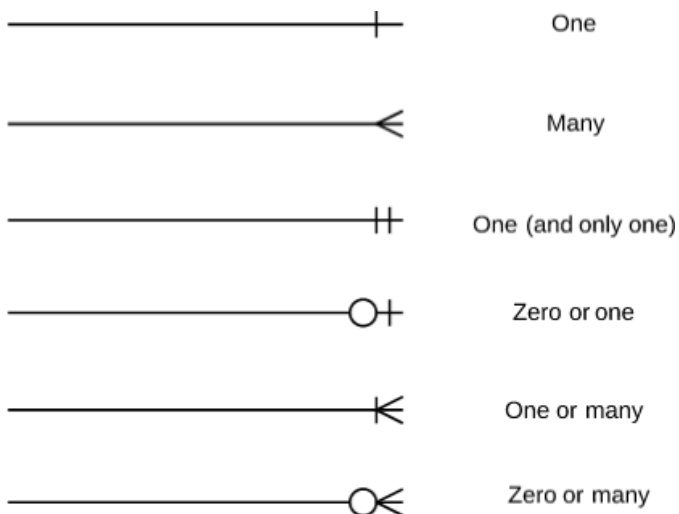
- **Customers** : données clients.
- **Products** : liste de modèles réduits de voitures.
- **ProductLines** : lignes de produits (ou catégories).
- **Orders** : bons de commande passés par les clients.
- **OrderDetails** : lignes de commande pour chaque bon de commande.
- **Payments** : paiements effectués par les clients.
- **Employees** : employés et informations de structure (notamment les responsables).
- **Offices** : magasins.

Le schéma MLD de la base

On parle aussi de diagramme Entité/Relation (E/R)



La notation des cardinalités est dite E/R (Entité/Relation) :



Téléchargement et installation de la base

<https://www.mysqltutorial.org/getting-started-with-mysql/mysql-sample-database/>

1. Comprendre la base
2. Déterminer l'ordre de précédence des tables
3. Vérifier le script
4. Exécuter le script d'installation

Requêtes de sélection

Sous-requêtes

Liste des employés dirigés par 'Bow','Anthony'

Technique :

1. créer une requête pour extraire l'ID de Anthony Bow
2. lister des employés qui reportsTo à cet ID

```
select lastName, firstName
from employees
where reportsTo = (
    select E2.employeeNumber
    from employees E2
    where E2.lastName = 'Bow' and E2.firstName = 'Anthony')
order by lastName;
```

Liste des produits dont le prix est le prix le plus élevé

Technique :

1. créer une requête pour extraire le prix maximum
2. lister les produit qui ont ce prix

```
select productName, buyPrice
from products
where buyPrice = (
    select MAX(buyPrice)
    from products
);
```

Jointures

Liste des salariés avec le nom de la ville dans laquelle se trouve leur bureau

```
select employees.lastName, employees.firstName, offices.city
from employees
left join offices on employees.officeCode = offices.officeCode;
```

Listes des nom et fonction des employés, avec le nom de leur supérieur (N+1)

```
select E.lastName, E.firstName, E.jobTitle, M.lastName
from employees E
left join employees M on E.reportsTo = M.employeeNumber;
```

Listes des nom et fonction des employés, avec le nom de leur N+1 et N+2

```
select E.lastName, E.firstName, E.jobTitle,
       M1.lastName as "N+1", M2.lastName as "N+2"
from employees E
left join employees M1 on E.reportsTo = M1.employeeNumber
left join employees M2 on M1.reportsTo = M2.employeeNumber;
```