# Méthode merise

## Dictionnaire de données - MCD – MLD + SQL

### Exercice Plages

Une région voyant son activité touristique grandir, souhaite mettre en place une structure permettant de suivre l'état de ses plages.

Dans un premier temps, elle souhaite connaître toutes ses plages :

- Chaque plage appartient à une ville

- Pour une plage, on connaîtra :

• Sa longueur en km

• La nature du terrain : sable fin, rochers, galets, ... sachant qu'il peut y avoir des plages avec sable et rochers

Le suivi se fera par département (uniquement les départements de la région) :

- Un responsable région sera nommée : on en connaitra son nom et son prénom.

- Une ville est identifiée par son code postal et le nombre de touristes annuel qu'elle reçoit doit être connu.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| entité | mnémonique | signification | type | longueur | remarques / contraintes |
| beachs | beach\_id | plage associée à une ville | a | 20 | identifiant, obligatoire |
| beach\_lenght | longueur de la plage en km | an | 3 | >=0, obligatoire |
| type\_beachs | type\_beach\_id | identifiant de la plage | n | 10 | identifiant |
| type\_beach\_sand | terrain de la plage (sable) | b | 1 | obligatoire |
| type\_beach\_roche | terrain de la plage (roche) | b | 1 | obligatoire |
| type\_beach\_gallet | terrain de la plage (galet) | b | 1 | obligatoire |
| regions | region\_id | identifiant de la région | n | 11 | identifiant, obligatoire |
| region\_name | nom de la région | an | 20 | obligatoire |
| managers | manager\_id | identifiant du responsable | n | 11 | identifiant, obligatoire |
| manager\_firstname | prénom du responsable | a | 15 | obligatoire |
| manager\_lastname | nom du responsable | a | 15 | obligatoire |
| cities | city\_zipcode | code postale de la ville | an | 5 | identifiant, obligatoire |
| city\_tourist | nombre de touriste en millions | n | 3 | >=0, obligatoire |

**Règles de gestion :**

Une ou plusieurs plages est composé de 1 ou plusieurs type de terrain

Un ou plusieurs types de terrain compose 1 ou plusieurs plages

Une plage appartient à 1 ou plusieurs villes

Une ou plusieurs villes peuvent appartenir à 1 et 1 seule plage

Une ou plusieurs villes sont associées à 1 et 1 seule région

Une région est associée à 1 ou plusieurs villes

Une ou plusieurs régions est géré par 1 et 1 seul manager

Un manager gère 1 ou plusieurs régions

**Dépendance fonctionnelle :**

beach\_city🡪 beach\_lenght, beach\_lenght

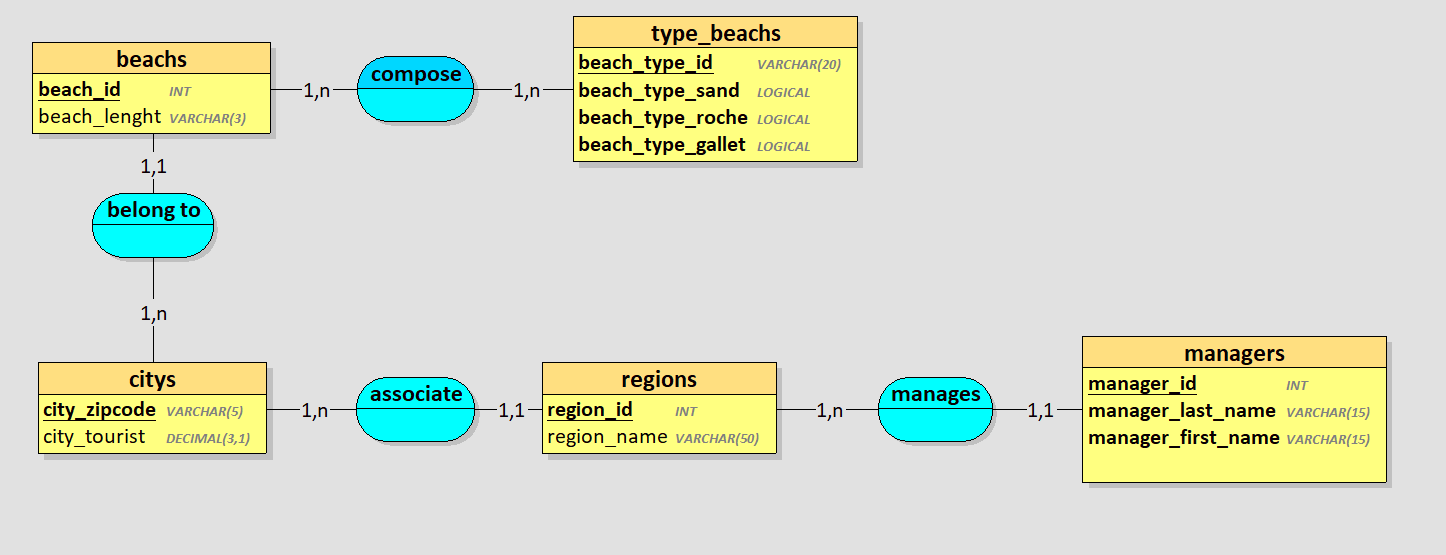
type\_beach\_id🡪 type\_beach\_sand, type\_beach\_roche, type\_beach\_gallet

region\_id🡪 region\_name

manager\_id🡪 manager\_firstname, manager\_lastname

city\_zipcode🡪 city\_tourist

MCD (modèle conceptuel)



MLD (modèle logique)

|  |
| --- |
| types\_beachs |
| beach\_type\_id |
| beach\_type\_sand |
| beach\_type\_roche |
| beach\_type\_gallet |

composer

**beach\_type\_id**

**beach\_id**

|  |
| --- |
| beachs |
| beach\_id |
| beach\_lenght |
| city\_zipcode |

Dans ce cas-là nous sommes dans du Many to Many(plusieurs a plusieurs) (1n vers 1n).

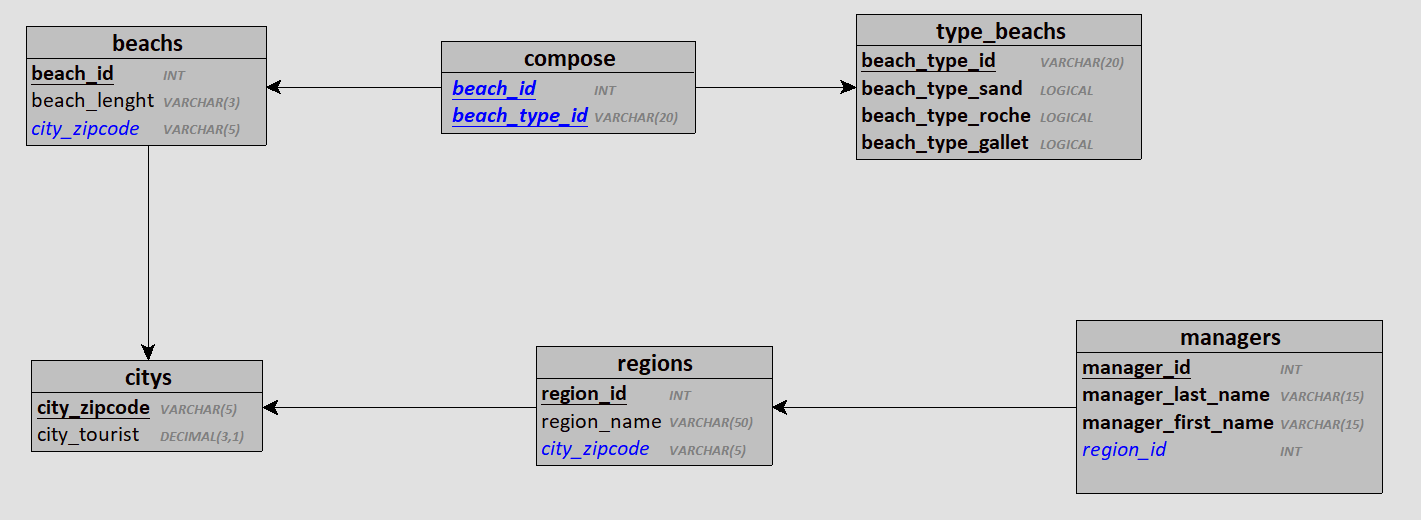
|  |
| --- |
| managers |
| manager\_id |
| manager\_last\_name |
| manager\_first\_name |
| region\_id |

|  |
| --- |
| cities |
| city\_zipcode |
| city\_tourist |

|  |
| --- |
| regions |
| region\_id |
| region\_name |
| city\_zipcode |

Le sens des flèches sans association, va de la clé étrangère vers la clé primaire. Dans ce cas-là nous sommes dans une relation One to Many (1 a plusieurs)(1,n vers 1,1).

MLD de looping



MLD VERSION TEXTE

**type\_beachs = (beach\_type\_id *INT***, beach\_type\_sand ***LOGICAL***, beach\_type\_roche ***LOGICAL***, beach\_type\_gallet ***LOGICAL*);**

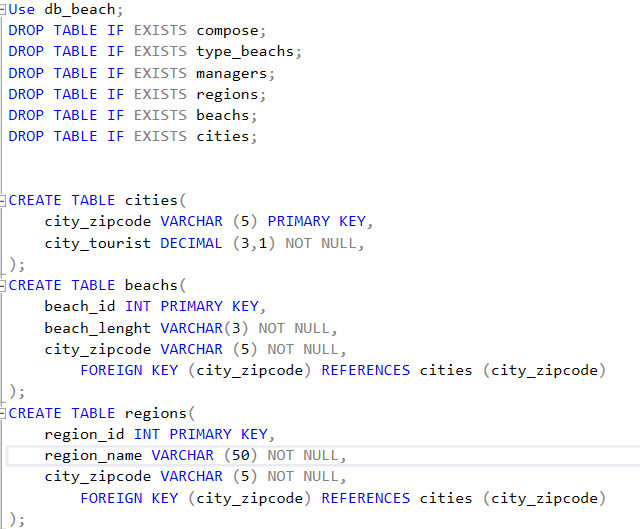
**cities = (city\_zipcode *VARCHAR(5)***, city\_tourist ***DECIMAL(3,1)*);**

**regions = (region\_id *INT***, region\_name ***VARCHAR(50)****, #city\_zipcode***);**

**beachs = (beach\_id *INT***, beach\_lenght ***VARCHAR(3)****, #city\_zipcode***);**

**managers = (manager\_id *INT***, **manager\_last\_name *VARCHAR(15)***, **manager\_first\_name *VARCHAR(15)****, #region\_id***);**

**compose = (*#beach\_id, #beach\_type\_id*);**

SQL SERVER

* Dans un premier temps créer la base de données
* Créer une requête / enregistrer la requête
* Créer les requêtes SQL avec le MLD (bon courage ;))
* Le drop des tables est inversé par rapport à la création 🡪

Use db\_beach;

DROP TABLE IF EXISTS compose;

DROP TABLE IF EXISTS type\_beachs;

DROP TABLE IF EXISTS managers;

DROP TABLE IF EXISTS regions;

DROP TABLE IF EXISTS beachs;

DROP TABLE IF EXISTS cities;

CREATE TABLE cities(

city\_zipcode VARCHAR (5) PRIMARY KEY,

city\_tourist DECIMAL (3,1) NOT NULL,

);

CREATE TABLE beachs(

beach\_id INT PRIMARY KEY,

beach\_lenght VARCHAR(3) NOT NULL,

city\_zipcode VARCHAR (5) NOT NULL,

FOREIGN KEY (city\_zipcode) REFERENCES cities (city\_zipcode)

);

CREATE TABLE regions(

region\_id INT PRIMARY KEY,

region\_name VARCHAR (50) NOT NULL,

city\_zipcode VARCHAR (5) NOT NULL,

FOREIGN KEY (city\_zipcode) REFERENCES cities (city\_zipcode)

);

CREATE TABLE managers(

manager\_id INT PRIMARY KEY IDENTITY(1,1),

manager\_last\_name VARCHAR (15) NOT NULL,

manager\_first\_name VARCHAR (15) NOT NULL,

region\_id INT NOT NULL,

FOREIGN KEY (region\_id) REFERENCES regions (region\_id)

);

CREATE TABLE type\_beachs(

beach\_type\_id INT PRIMARY KEY,

beach\_type\_sand BIT NOT NULL,

beach\_type\_roche BIT NOT NULL,

beach\_type\_gallet BIT NOT NULL,

);

CREATE TABLE compose(

beach\_id INT NOT NULL,

beach\_type\_id INT NOT NULL,

PRIMARY KEY (beach\_id,beach\_type\_id),

FOREIGN KEY (beach\_id) REFERENCES beachs (beach\_id),

FOREIGN KEY (beach\_type\_id) REFERENCES type\_beachs (beach\_type\_id)

);