

# 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 connaîtra 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

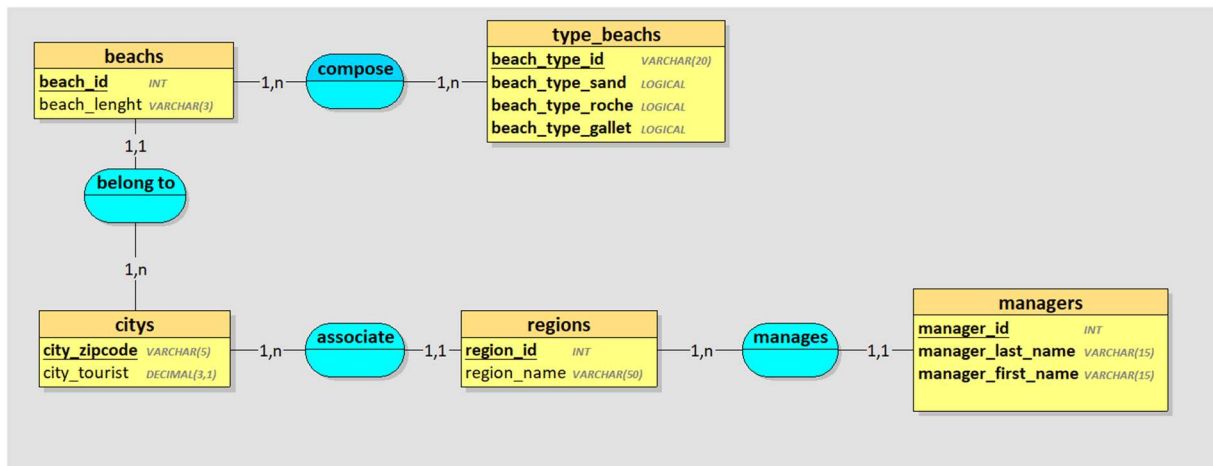
type\_beach\_id → type\_beach\_sand, type\_beach\_roche, type\_beach\_gallet

region\_id → region\_name

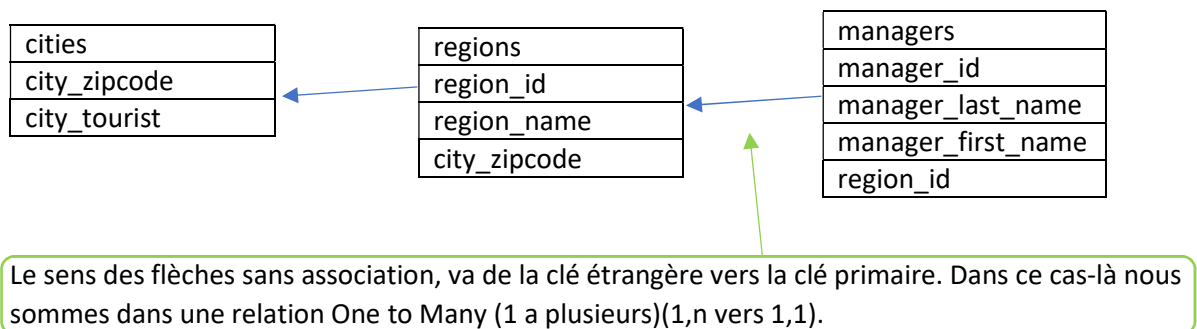
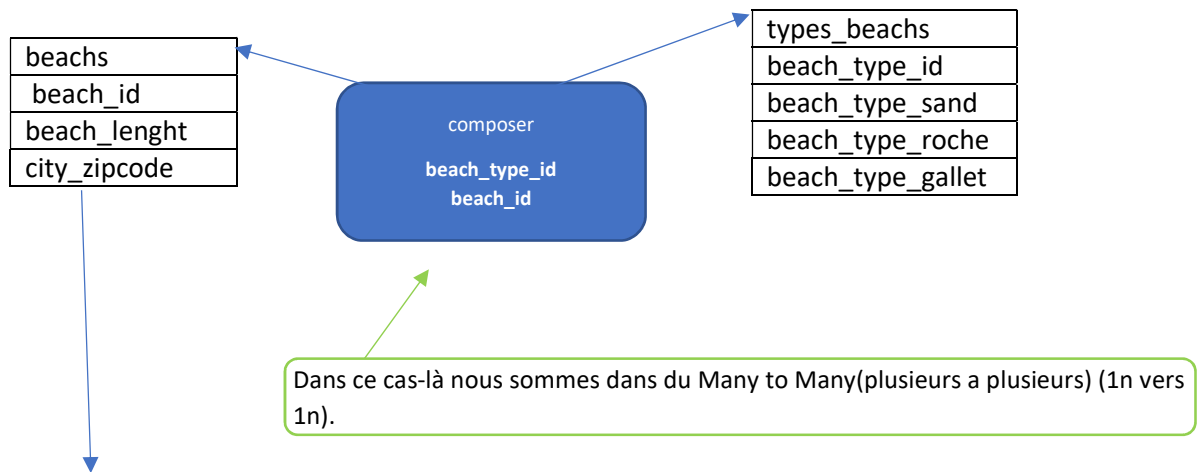
manager\_id → manager\_firstname, manager\_lastname

city\_zipcode → city\_tourist

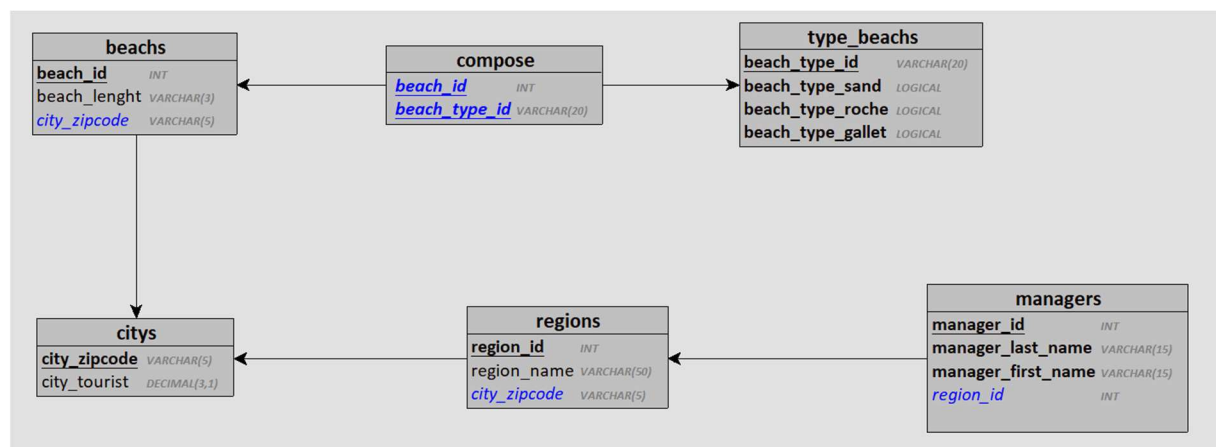
MCD (modèle conceptuel)



## MLD (modèle logique)



## 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);
beaches = (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)
);
```

```
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)
);
```