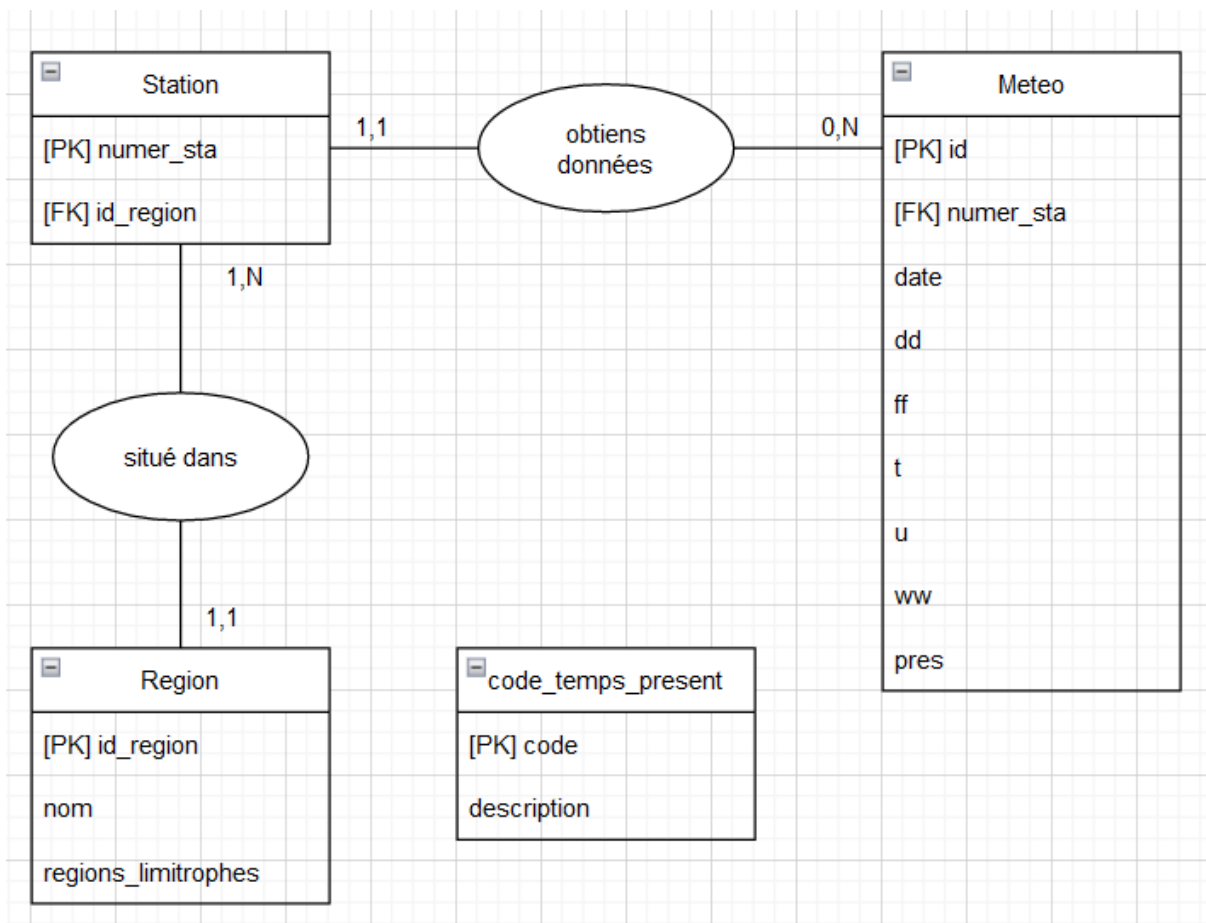


## Réalisation d'une base de données météorologique

### PARTIE 1

#### 1) Conception de la base de données.

##### a) Schéma conceptuel



##### b) Schéma logique ou relationnel

###### Description :

La table "station" possède la clé primaire "numer\_sta" et un champ "region\_id" qui est une clé étrangère référençant la table "region".

La table "region" possède la clé primaire "id".

La table "meteo" possède une clé primaire composite composée de la clé étrangère "numer\_sta" référençant la table "station" et du champ "date".

La table "code\_temps\_present" possède une clé primaire "code"

#### **MLD :**

station(numer\_sta PK, region\_id FK)

region(id PK, nom, region\_limitrophes)

meteo(numer\_sta PK FK, date PK, dd, ff, t, u, ww, pres)

code\_temps\_present(code PK, description)

\*PK = Primary Key, FK = Foreign key

### **c) Justification des choix de conception**

Développement de façon incrémental : la première idée étant d'avoir la liste des stations, et pour chacune d'entre elles différentes valeurs par date de prélèvement. Ces deux tables principales ont d'abord été faites. J'ai ensuite greffé la table région qui ajoute une information supplémentaire à la station, ainsi que code\_temps\_present. Cette dernière contient la liste de tous les codes ainsi que sa description, pour permettre une lecture des valeurs ww de la table meteo.

## **PARTIE 2**

### **1) Création de la base - CreateTable.sql**

J'ai choisi d'insérer les données à l'aide de la méthode LOAD DATA INFILE. J'ai utilisé l'option "LOCAL".

SCRIPT :

-- Création de la table "region" avec sa clé primaire "id"

```
CREATE TABLE region (  
id INT PRIMARY KEY,  
nom VARCHAR(50),  
region_limitrophes VARCHAR(200)  
);
```

-- Création de la table "station" avec sa clé primaire "numer\_sta" et la clé étrangère "region\_id" référençant la table "region"

```
CREATE TABLE station (  
numer_sta INT PRIMARY KEY,  
region_id INT,  
FOREIGN KEY (region_id) REFERENCES region(id)  
);
```

-- Création de la table "code\_temps\_present" avec sa clé primaire "code"

```
CREATE TABLE code_temps_present (  
code INT PRIMARY KEY,  
description VARCHAR(255)  
);
```

-- Création de la table "meteo" avec sa clé primaire composite composée de la clé étrangère "numer\_sta" référençant la table "station" et du champ "date"

```
CREATE TABLE meteo (  
numer_sta INT,  
date DATE,  
dd INT,  
ff INT,  
t FLOAT,  
u INT,  
ww INT,  
pres INT,  
PRIMARY KEY (numer_sta, date),  
FOREIGN KEY (numer_sta) REFERENCES station(numer_sta)  
);
```

-- Ajout de contraintes d'intégrité référentielle

```
ALTER TABLE station ADD CONSTRAINT fk_region_id FOREIGN KEY (region_id)  
REFERENCES region(id);  
ALTER TABLE meteo ADD CONSTRAINT fk_numer_sta FOREIGN KEY (numer_sta)  
REFERENCES station(numer_sta);  
ALTER TABLE meteo ADD CONSTRAINT fk_code FOREIGN KEY (ww) REFERENCES  
code_temps_present(code);
```

## 2) Insertion des données - InsertData.sql

Les données sont importées via CSV pour les tables station et météo via le fichier SYNOP. Elles le sont également pour la table code\_temps\_present. Le CSV contient une colonne "code" et sa "description".

### I. La table région est remplie avec le code suivant :

```
-- Création des données pour la table "region"
INSERT INTO region (id, nom, region_limitrophes)
VALUES (1, 'Ile-de-France', '2,3,4'),
       (2, 'Nord', '1,3,5'),
       (3, 'Est', '1,2,4,5'),
       (4, 'Ouest', '1,3'),
       (5, 'Sud', '2,3');
```

### II. Insertion des données de code temps présent, soit ww dans meteo :

```
LOAD DATA LOCAL INFILE '/var/lib/mysql-files/code_wmo.csv'
INTO TABLE code_temps_present
CHARACTER SET UTF8
FIELDS TERMINATED BY ';'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES
(code, description);
```

### III. Insertion des donnée SYNOP :

```
LOAD DATA LOCAL INFILE '/var/lib/mysql-files/donnees_meteo.csv'
INTO TABLE station
CHARACTER SET UTF8
FIELDS TERMINATED BY ';'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES
-- si data = mq alors NULL
(@numer_sta, @region)
SET numer_sta = NULLIF(@numer_sta,'mq'),
region_id = NULLIF(@region,'mq');
```

```

LOAD DATA LOCAL INFILE '/var/lib/mysql-files/donnees_meteo.csv'
INTO TABLE meteo
CHARACTER SET UTF8
FIELDS TERMINATED BY ';'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES
-- si data = mq alors NULL
(@numer_sta, @date, @dd, @ff, @t, @u, @ww, @pres)
SET numer_sta = NULLIF(@numer_sta,'mq'),
date = NULLIF(@date,'mq'),
dd = NULLIF(@dd,'mq'),
ff = NULLIF(@ff,'mq'),
t = NULLIF(@t,'mq'),
u = NULLIF(@u,'mq'),
ww = NULLIF(@ww,'mq'),
pres = NULLIF(@pres,'mq');

```

On voit ici comment le **remplacement des valeurs mq** ont été défini par NULL directement à l'importation.

### 3) Requêtes demandées - Requetes.sql

voir fichier

### 4) Procédures - Procedures.sql

voir fichier