

**Travail Pratique**  
associé au cours  
« **Entrepôts de données** »

5A  
Mars 2021

Bernard ESPINASSE

**Entrepôt de données**  
« **AcciRoute** »

**Analyse OLAP avec**  
**JRUBIK ou SAIKU**

**Sommaire**

1.	Objectif du TP .....	2
2.	Entrepôt de données « AcciRoute » .....	2
3.	Travail demandé .....	4
3.1	Installation du SGBD « PostGreSQL » sur vos machines .....	4
3.2	Implantation et alimentation de l'entrepôt « AcciRoute » .....	4
3.3	Installation et familiarisation avec la suite BI .....	4
3.4	Définition d'un schéma MONDRIAN .....	5
3.3	Connection de la suite BI avec votre entrepôt et son schéma de cube .....	6
3.4	Analyse OLAP de l'entrepôt : requêtes MDX.....	6
4.	Annexe 1 : Script de création de la base ROLAP de l'entrepôt.....	8

**RENDU DU TP : le rendu du TP sera fourni en respectant le cadre défini par le fichier « 3-Rendu-Vide-AcciRoute-11mars21.doc » fourni.**

# 1. Objectif du TP

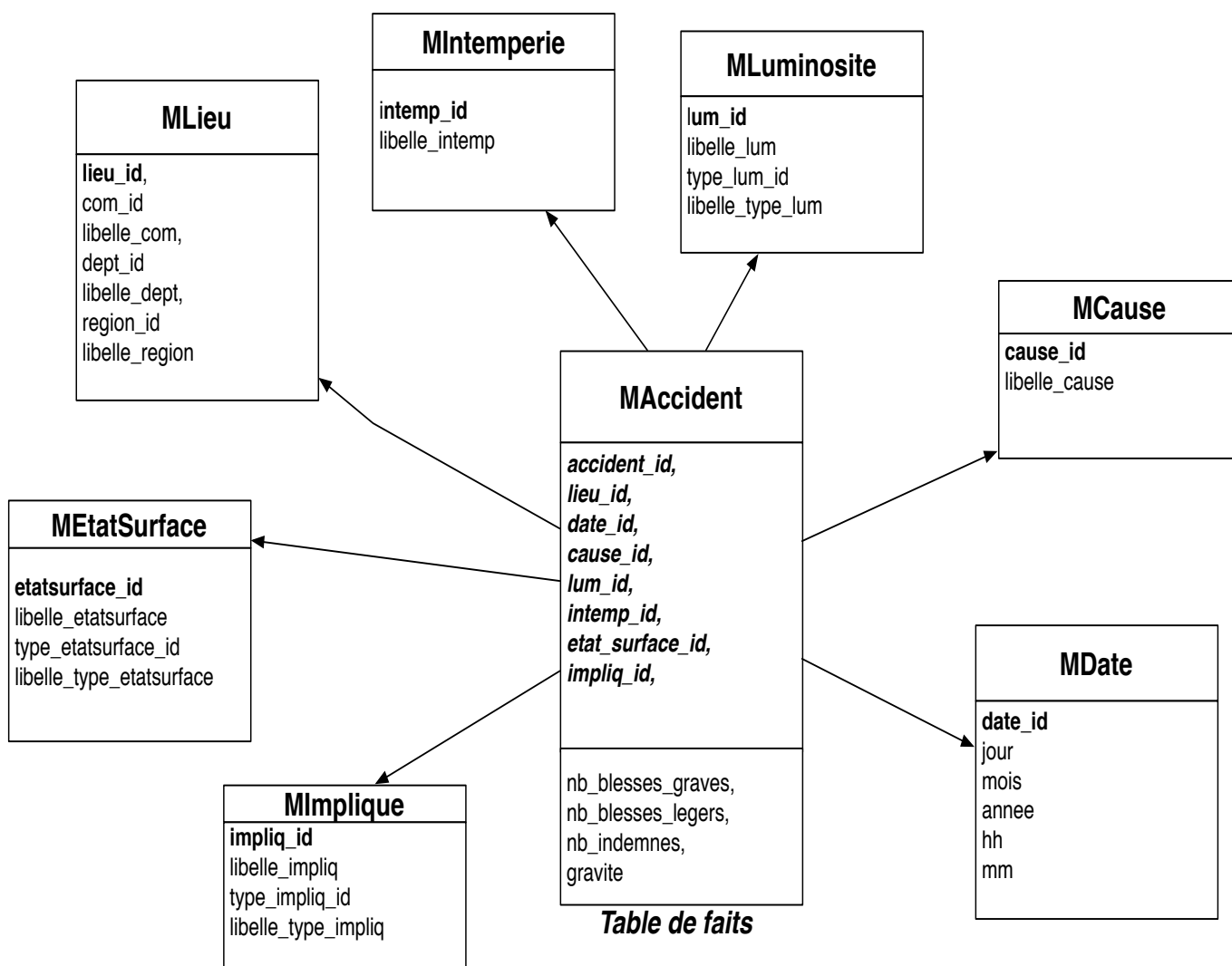
Ce TP a pour objectif de concevoir un entrepôt de données à partir de sources de données existantes selon la méthode, l'implémenter et l'exploiter.

Cet entrepôt, nommé « AcciRoute », concerne des données publiques liées à des accidents de la circulation routière (source de données).

Enfin, cet entrepôt sera exploité en analyse OLAP en langage MDX (vu en cours), au travers d'une suite Business Intelligence (BI) libre (JRUBIK, SAIKU, ...) qui devra être installée.

# 2. Entrepôt de données « AcciRoute »

L'entrepôt de données « AcciRoute » est structuré selon le schéma ROLAP suivant :



La structure des tables est la suivante :

### Table de faits

#### TABLE MAccident

**accident\_id** : Identifiant de l'accident

**lieu\_id** : clé étrangère vers table MLieu  
**date\_id** : clé étrangère vers table MDate  
**cause\_id** : clé étrangère vers table MCause  
**lum\_id** : clé étrangère vers table MLuminosite  
**intemp\_id** : clé étrangère vers table MIntemperie  
**etat\_surface\_id** : clé étrangère vers table MEtatDeSurface  
**impliq\_id** : clé étrangère vers table MImplique  
**nb\_blesses\_graves** : mesure 1 = nb de blessés graves dans l'accident  
**nb\_blesses\_legers** : mesure 1 = nb de blessés légers dans l'accident  
**nb\_indemnes** : mesure 1 = nb de personnes indemnes dans l'accident  
**gravite** : gravité de l'accident (de 1 - pas grave à 3 - très grave)

Remarque : les lieux des accidents (lieu\_id) ont été générés de façon **aléatoire**.

### *Tables de dimension*

#### **TABLE MLieu**

**lieu\_id** : Identifiant du lieu de l'accident  
**com\_id** : code commune  
**libelle\_com** : libellé commune  
**dept\_id** : code département  
**libelle\_dept** : libellé département  
**region\_id** : code région  
**libelle\_region** : libellé région

#### **TABLE MDate**

**date\_id** : code date  
**jour** : jour de l'accident  
**mois** : mois de l'accident  
**annee** : année de l'accident  
**hh** : heure de l'accident  
**mm** : minutes de l'accident

#### **TABLE MCause**

**cause\_id** : code cause  
**libelle\_cause** : libellé cause (dépassement à droite, en stationnement, piéton, ...)

#### **TABLE MEtatSurface**

**etatsurface\_id** : code état de surface  
**libelle\_etatsurface** : libellé état de surface (humide, enneigé, gravillons, ...)  
**type\_etatsurface\_id** : code type état de surface  
**libelle\_type\_etatsurface** : libellé type état de surface (dangereux, très dangereux, normal, ...)

#### **TABLE MImplique**

**impliq\_id** : code implication  
**libelle\_impliq** : libellé implication (piéton, bicyclette, bus, ...)  
**type\_impliq\_id** : code type implication  
**libelle\_type\_impliq** : libellé type implication (humain, véhicule, deux-roues)

#### **TABLE MIntemperie**

**intemp\_id** : code intempérie  
**libelle\_intemp** : libellé intempérie (beau temps, pluie forte, ...)

#### **TABLE MLuminosite**

**lum\_id** : code luminosité  
**libelle\_lum** : libellé luminosité (jour, demi-jour, ...)  
**type\_lum\_id** : code type luminosité (1 = jour, 2 = nuit)  
**libelle\_lyp\_lum** : libellé type luminosité (jour, nuit)

### **3. Travail demandé**

---

La suite BI libre utilisée est une suite libre, soit JRUBIK soit SAIKU téléchargeable sur la page Web du cours. Sur cette page vous y trouverez aussi, un manuel de mise en œuvre et/ou d'utilisation et d'autres ressources associées à leur mise en œuvre.

#### **3.1 Installation du SGBD « PostgreSQL » sur vos machines**

Vous téléchargerez le SGBD relationnel libre PostgreSQL et vous l'installerez sur vos machines (PC sous Windows, un Mac OS ou machine sous Linux). Téléchargement de PostgreSQL (version 9.4.26 suffisante) :

<https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>

#### **3.2 Implantation et alimentation de l'entrepôt « AcciRoute »**

Un script de création des tables de l'entrepôt et de chargement des données pour le SGBD PostgreSQL est fourni sur la page Web du cours.

#### **3.3 Installation et familiarisation avec la suite BI**

On vous demande d'installer une suite BI libre utilisée, JRUBIK ou SAIKU et de vous familiariser avec elles. Ces suites sont téléchargeables sur la page Web du cours. Sur cette page vous y trouverez aussi, un manuel de mise en œuvre et/ou d'utilisation et d'autres ressources associées à leur mise en œuvre.

Pour la suite JRUBIK, vous pouvez vous familiariser avec elle avec l'entrepôt FoodMart sous DERBY disponible, en suivant le manuel d'utilisation (disponible sur mon site en anglais et en français). Vous pouvez aussi découvrir le schéma Mondrian en xml utilisé. De même avec SAIKU, vous pourrez découvrir cette suite en la manipulant avec des entrepôts disponibles.

## Liens utiles pour JRUBIK :

1 - Lien source forge :

<http://rubik.sourceforge.net/jrubik/intro.html>

## Liens utiles pour SAIKU :

1 – SAIKU ANALYTICS 3.x documentation et manuel en ligne :

<https://saiku-documentation.readthedocs.io/en/latest/>

2 – Un exemple d’usage de SAIKU en biodiversité :

<https://www.youtube.com/watch?v=-As7F5e7hyU&feature=youtu.be>

### 3.4 Définition d’un schéma MONDRIAN

Les suites BI libres que sont JRUBIK et SAIKU, sont basées sur le serveur OLAP libre nommé MONDRIAN. Pour pouvoir utiliser ces suites sur votre entrepôt « AcciRoute », vous devrez définir un schéma de cube OLAP.

Ce cube définit les *dimensions* et *mesures* que vous utiliserez dans vos analyses OLAP sur votre entrepôt en langage MDX. Ci-dessous un schéma xml incomplet que vous devez compléter avant de l’utiliser dans votre suite BI.

Ci-dessous un schéma Mondrian en mdx pour l’utilisation OLAP de l’entrepôt « AcciRoute ». Ce schéma peut être utilisé pour se faire la main sur votre entrepôt de données.

#### Schéma cube xml basique « CubeAcciRoute0 » sur l’entrepôt « AcciRoute » :

```
<?xml version="1.0" encoding="UTF-8"?>
<Schema name="cubeacciroute0">
  <Cube name="cubeacciroute0">
    <Table name="maccident"/>

    <Dimension type="StandardDimension" name="Lieu" visible="true" highCardinality="true" foreignKey="lieu_id">
      <Hierarchy name="Lieu" hasAll="true" primaryKey="lieu_id">
        <Table name="mlieu"/>
        <Level name="Region" table="mlieu" column="libelle_region" uniqueMembers="true"/>
      </Hierarchy>
    </Dimension>
    <Dimension type="StandardDimension" name="Date" visible="true" highCardinality="true" foreignKey="date_id">
      <Hierarchy name="Date" hasAll="true" primaryKey="date_id">
        <Table name="mdate"/>
        <Level name="annee" table="mdate" column="annee" uniqueMembers="false"/>
        <Level name="mois" table="mdate" column="mois" uniqueMembers="false"/>
      </Hierarchy>
    </Dimension>
    <Dimension type="StandardDimension" name="Cause" visible="true" highCardinality="false" foreignKey="cause_id">
      <Hierarchy name="Cause" hasAll="true" primaryKey="cause_id">
        <Table name="mcause"/>
        <Level name="Libelle cause" table="mcause" column="libelle_cause" uniqueMembers="false"/>
      </Hierarchy>
    </Dimension>

    <Measure name="SUM of MAccident.accident_id" column="accident_id" aggregator="count" formatString="Standard"/>
    <Measure name="SUM of MAccident.nb_blesses_graves" column="nb_blesses_graves" aggregator="sum" formatString="Standard"/>
    <Measure name="SUM of MAccident.nb_blesses_legers" column="nb_blesses_legers" aggregator="sum" formatString="Standard"/>
    <Measure name="SUM of MAccident.nb_indemnes" column="nb_indemnes" aggregator="sum" formatString="Standard"/>
  </Cube>
```

</Schema>

Vous noterez que ce schéma de cube :

- ne concerne que les dimensions *Lieu*, *Date* et *Cause*. Il ne contient pas les dimensions : *Luminosité*, *Intemperie*, *Surface* et *Implique*.
- Ne prend pas en compte tous les niveaux des hiérarchies des dimensions

**On vous demande :**

- de bien comprendre comment est construit un schéma de cube xml pour MONDRIAN
- de créer, à partir de ce schéma, un autre ce schéma de cube intégrant les quatre autres dimensions (*Luminosité*, *Intemperie*, *Surface* et *Implique*) et tous les niveaux de hiérarchies et le tester. Ce schéma MONDRIAN en xml vous servira pour faire vos analyses OLAP.

### **3.3 Connection de la suite BI avec votre entrepôt et son schéma de cube**

La suite BI libre installée, JRUBIK ou SAIKU, il vous faudra accéder à votre entrepôt « AcciRoute », source de données, et rattacher le schéma MONDRIAN en xml de base fournit, puis avec celui que complété par vous-même. Suivre pour cela la procédure définie par la suite BI utilisée.

### **3.4 Analyse OLAP de l'entrepôt : requêtes MDX**

Une fois connecté à votre entrepôt de données « AcciRoute » et à votre schéma de cube MONDRIAN en xml, on vous demande de développer une dizaine requêtes d'analyse OLAP en langage MDX de votre choix, tirant profit au maximum de la puissance du langage MDX.

**A) Pour chaque requête, vous donnerez pour les 10 requêtes suivantes : 1) la formulation MDX, et 2) le tableau de résultats obtenu (cela pourra être un bout d'écran intégré dans votre rapport).**

- R1 : Pour chaque mois de l'année, quelle est le nombre de blessés légers et de blessés graves pour la région « Provence Alpes et Côte d'Azur » ?
- R2 : Quel est le nombre d'accidents par cause d'accident un refus de priorité et par mois de l'année 2015 et par région ?
- R3 : Quel est le nombre de blessés graves par commune et par mois les années 2015 pour le département « Bouches du Rhône » ?
- R4 : Quelle sont les communes les plus accidentogènes de la région « SAVOIE » (celles qui ont le plus de blessés) ?
- R5 : Quels sont les 10 communes ayant le plus de blessés grave par temps de neige en 2015 ?

- R6 : Sur quel type de surface et par quel type de temps il y a le plus de blessés graves ?
- R7 : Le nombre d'accidents par intempérie suivant la luminosité pour la région « Auvergne » ?
- R8 : Quelles sont les 10 communes les plus dangereuses par département de la région « Hauts de France » pour 2015 ? (Communes où il y a eu le plus d'accidents « très grave »)
- R9 : Quels sont les 5 départements les plus sûrs ? (Départements pour lesquels il y a le moins d'accidents)
- R10 : Le classement des 10 départements ayant le moins accidents très grave (3) en 2010, 2011, et 2012 ?

**B) En plus de ces 10 requêtes demandées, vous proposerez 10 autres requêtes MDX de votre choix.**

**Pour chacune des requêtes, vous donnerez : 1) l'objet de la requête, 2) la formulation MDX, et 3) le tableau de résultats obtenus. Pour cela vous pourrez intégrer dans votre rapport des bouts d'écran.**

# Annexes

## 4. Annexe 1 : Script de création de la base ROLAP de l'entrepôt

Ci-dessous le script de création de la base ROLAP de l'entrepôt « AcciRoute » pour le SGBD PostGreSQL. Il se peut que vous soyez amené à modifier légèrement les types de certains attributs de cette base de données (Int = Integer dans PostGreSQL).

### Table de faits

#### CREATE TABLE MAccident

```
(
  accident_id      Varchar (8),
  lieu_id          Varchar (8),
  date_id         Varchar (8),
  cause_id        Varchar (8),
  lum_id          Varchar (8),
  intemp_id       Varchar (8),
  etat_surface_id Varchar (8),
  impliq_id       Varchar (8),
  nb_blesses_graves Int,
  nb_blesses_legers Int,
  nb_indemnes     Int,
  gravite         Varchar (8)
);
```

### Tables de dimension

#### CREATE TABLE MLieu

```
(
  lieu_id      Varchar (8),
  com_id       Varchar(8),
  libelle_com  Varchar(60),
  dept_id     Varchar(8),
  libelle_dept Varchar(60),
  region_id   Varchar(8),
  libelle_region Varchar(60)
);
```

#### CREATE TABLE MDate

```
(
  date_id  Varchar (8),
  jour     Varchar (8),
  mois     Varchar (8),
```



```
    annee      Varchar (8),
    hh         Varchar (8),
    mm         Varchar (8)
);
```

#### **CREATE TABLE MCause**

```
(
    cause-id      Varchar (8),
    libelle_cause Varchar (80)
);
```

#### **CREATE TABLE MEtatSurface**

```
(
    etatsurface_id      Varchar (8),
    libelle_etatsurface Varchar (80),
    type_etatsurface_id Varchar (8),
    libelle_type_etatsurface Varchar (80)
);
```

#### **CREATE TABLE MImplique**

```
(
    impliq_id      Varchar (8),
    libelle_impliq Varchar (80),
    type_impliq_id Varchar (8),
    libelle_type_impliq Varchar (80)
);
```

#### **CREATE TABLE MIntemperie**

```
(
    intemp_id      Varchar (8),
    libelle_intemp Varchar (80)
);
```

#### **CREATE TABLE MLuminosite**

```
(
    lum_id      Varchar (8),
    libelle_lum Varchar (80),
    type_lum_id Varchar (8),
    libelle_type_lum Varchar (80)
);
```

—