Base de données évoluées
Projet entrepôt de donnée

CARON Dylan

2017-2018

Sommaire

1 Dataset utilisé

Le dataset utilisé ici décrit les intempéries qui se sont produits aux États-Unis. Pour le projet, j'ai seulement utilisé le fichier concernant l'année 1994. Par conséquent chacune des lignes de ce fichier décrit le type de d'intempérie, cela peut être une tornade, de la grêle, de l'orage et autre, l'endroit ou cette intempérie à frappée et à quelle date (à la minute près). Il décrit les dommages causés par cette dernière en précisant s'il s'agit d'infrastructure ou de champs.

2 Entrepôt de donnée

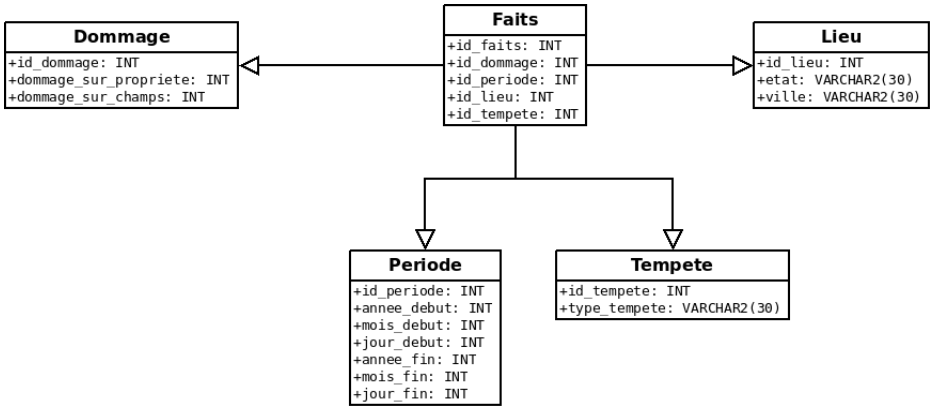
2.1 Description de l'entrepôt

L'entrepôt que j'en ai déduit est une base de donnée en étoile composé de 5 tables que voici :

1. Table Dommage: Celle ci donne les chiffres des dégâts estimés en dollars
2. Table Periode: Elle donne la période durant laquelle l'intempérie à eu lieu. Ici la précision retenue est au jour près.
3. Table Tempete: Elle permet d'identifier la nature de l'intempérie, à savoir grêle, orage, tornade ...
4. Table Lieu: Donne la localisation. Ici la table donne l'état et la ville ou s'est déroulé l'intempérie.
5. Table Faits: Cette dernière contient les identifiants de chacune des 4 tables précédemment évoquée.

L'entrepôt à été réalisé avec ORACLE et donc en SQL.

2.2 Schéma de l'entrepôt



Remarque : Chaque attribut commençant par "id_" est clé de sa table.

3 Intégration des données

3.1 Fonctionnement

Pour l'intégration, j'ai décidé d'utiliser un programme C++ programmer par moi-même. Basiquement le programme lit caractère par caractère une ligne du fichier CSV et insère les caractères dans les structures prévues à cet effet. Puis à la fin de la lecture d'une ligne, le code va écrire dans un fichier les ligne INSERT correspondant aux données lues.

3.2 Défauts

Un des principales de cette méthode est que certaine cas non géré par mon programme résulte en l'absence d'un tuple dans la future BDD. Par conséquent certain tuple du CSV ne sont pas insérés dans ma BDD à cause de ce problème. Cependant ce problème reste assez léger, car celui-ci concerne seulement 40 tuples sur les 16000 présents dans le CSV soit 0.25%.

De plus, par manque de temps, il a plusieurs dans Dommage qui sont identiques. En effet avec mon programme C++ si deux tempêtes n'infligent aucun dégâts alors dans la table Dommage, j'aurais deux lignes avec 0 sur l'attribut dommage_sur_propriete et dommage_sur_champs avec un id_dommage différent. Le problème est assez simple à régler, il suffit de stocker ce qu'on a déjà inséré et vérifier que ce qu"on va insérer n'existe pas déjà.

4 Requêtes OLAP

Requête 1 :

```
SELECT mois_debut, type_tempete, SUM(dommage_sur_propriete)
FROM Faits f, Tempete t, Periode p, Dommage d
WHERE f.id_tempete = t.id_tempete AND f.id_periode = p.id_periode
      AND f.id_dommage = d.id_dommage
GROUP BY ROLLUP(mois_debut, type_tempete);
```

Première des neufs requêtes utilisées, ici le but est d'utiliser ROLLUP, la requête permet de voir trois choses :

- Les dégâts causées (sur les infrastructures) par un type de tempête sur un mois particulier.
- Les dégâts causées (sur les infrastructures) par l'ensemble des tempêtes sur un mois particulier.
- Les dégâts causées (sur les infrastructures) par l'ensemble des tempêtes sur l'année.

Requête 2 :

```
SELECT mois_debut, type_tempete,
 SUM(dommage_sur_propriete + dommage_sur_champs) AS dommage_cause
FROM Faits f, Tempete t, Periode p, Dommage d
WHERE f.id_tempete = t.id_tempete AND f.id_periode = p.id_periode
      AND f.id_dommage = d.id_dommage
GROUP BY CUBE(mois_debut, type_tempete);
```

Ici le but est d'utiliser CUBE, la requête est quasi identique à la précédente mais cette dernière permet de ressortir aussi les dégâts causées (cette fois, c'est sur les infrastructures et les champs) par un type de tempête sur l'année.

Requête 3 :

```
SELECT etat, mois_debut, COUNT(*) AS nb_tempete
FROM Periode p, Lieu l, Faits f
WHERE f.id_lieu = l.id_lieu AND f.id_periode = p.id_periode
GROUP BY (mois_debut,etat)
ORDER BY (mois_debut);
```

Ici, j'utilise un simple GROUP BY pour récupérer le nombre de tempête par état et par mois.

Requête 4 :

```
SELECT etat, mois_debut, SUM(nb_tempete) AS nb_tempete
FROM (
        SELECT etat, mois_debut, COUNT(*) AS nb_tempete
        FROM Periode p, Lieu l, Faits f
        WHERE f.id_lieu = l.id_lieu
              AND f.id_periode = p.id_periode
        GROUP BY (mois_debut,etat)
        ORDER BY (mois_debut)
        )
GROUP BY GROUPING SETS((etat),(mois_debut),());
```

Cette requête permet d'obtenir le nombre de tempête par État durant l'année, le nombre de tempête par mois aux Etat-Unis ainsi que le nombre de tempêtes aux États-Unis durant l'année.

Requête 5 :

```
SELECT etat, SUM(dommage_sur_champs),
       rank() over (order by SUM(dommage_sur_champs) desc)
FROM Faits f, Lieu l, Dommage d
WHERE f.id_lieu = l.id_lieu AND f.id_dommage = d.id_dommage
GROUP BY (etat);
```

Donne un classement des États ayant reçues le plus de dégâts subis sur les champs.

Requête 6 :

```
SELECT etat, damage_property
FROM (
        SELECT etat, SUM(dommage_sur_propriete) AS damage_property
        FROM Faits f, Lieu l, Dommage d
        WHERE f.id_lieu = l.id_lieu AND f.id_dommage = d.id_dommage
        GROUP BY (etat)
        ORDER BY SUM(dommage_sur_propriete) DESC
        )
WHERE ROWNUM <=10;
```

La requête renvoie les 10 États ayant subis le plus de dégâts sur leurs infrastructures.

Requête 7 :

```
SELECT etat, nb_tempete,
       NTILE(4) over(order by nb_tempete DESC) AS Quart
FROM (
        SELECT etat, COUNT(*) AS nb_tempete
        FROM Lieu l, Faits f
        WHERE f.id_lieu = l.id_lieu
        GROUP BY (etat)
        );
```

Permet de savoir si un États se trouve dans le premier quart des États ayant subi le plus de tempête.

Requête 8 :

```
SELECT mois_debut, etat, SUM(nb_tempete) AS nb_temp,
       GROUPING(mois_debut) AS mois, GROUPING(etat) AS state
FROM (
        SELECT mois_debut, etat, COUNT(*) AS nb_tempete
        FROM Lieu l, Faits f, Periode p, Dommage d
        WHERE f.id_lieu = l.id_lieu AND f.id_periode = p.id_periode
              AND f.id_dommage = d.id_dommage
              AND dommage_sur_propriete ='5000000'
        GROUP BY (etat,mois_debut)
        )
GROUP BY GROUPING SETS((mois_debut),(etat))
ORDER BY nb_temp DESC;
```

Liste le nombre de tempête ayant causer 5M de dollars par État ou par mois.

Requête 9 :

```
SELECT mois_debut, COUNT(*) AS nb_tempete,
       SUM(COUNT(*)) over(order by mois_debut
                          rows unbounded preceding) AS Cumul
FROM Faits f, Periode p
WHERE f.id_periode = p.id_periode
GROUP BY mois_debut;
```

Une requête dont le résultat à été déjà aperçu dans les précédentes requêtes, cependant le but est d'utiliser WINDOW donc cette requête sort le nombre de tempête par mois avec un cumul, ce qui nous permet de savoir, par exemple, combien de tempêtes ont eu les six premier mois.