

Partie 1

1) Les interrogations: requêtes transactionnelles vs analytiques

1)

Requête transactionnelles: Car la requête est faite pour un moment données, on se demande juste le nombre de place restante (pas de group by)

2)

Requête analytique: On regarde le nombre de place vendu pour chaque séance pour le dernier film de Star Wars

3)

Retourne le nombre de place vendu dans chaque cinéma, pour chaque mois et pour chaque film.

Requête analytique: on analyse le nombre de place vendu par rapport à des éléments choisis, cette requête peut être réutilisée plusieurs fois.

4)

Cette requête compte le nombre de ventes pour chaque créneau horaires.

Requête analytique: cette requête pourrait être utile pour savoir le nombre d'employés nécessaires

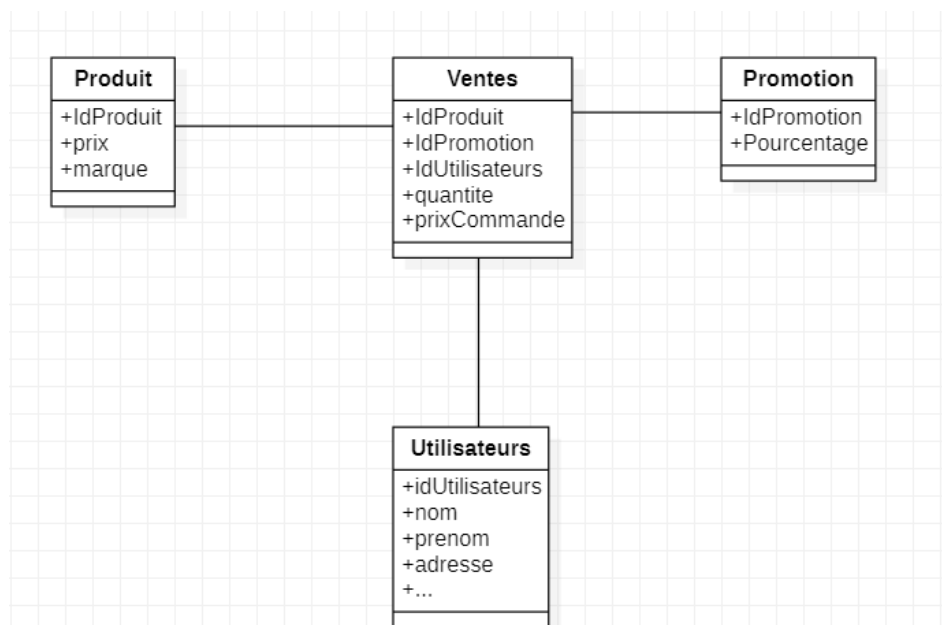
5)

Requetes tanalytique:

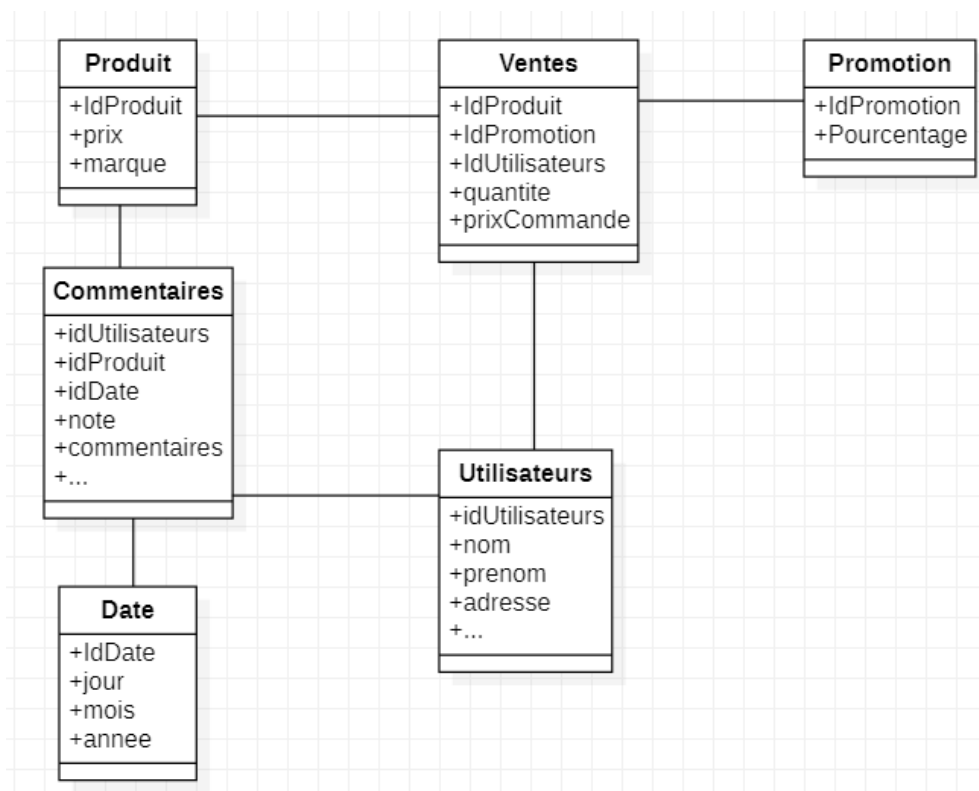
Car dans l'insertion de la value, on insere que des clés étrangère avec une valeur.

2) Un entrepôt de données pour Amazon

1)



2)



3)

- 1: La quantité de produit vendu chaque jour par produit
- 2: Le produit avec la plus grosse promotion
- 3: Les utilisateurs qui achete le plus de produit chaque mois

3) Requêtes Analytiques

1)

```
SELECT sum(montant_journalier)
from ventes_monoprix
group by(id_produit);
```

```
SELECT sum(montant_journalier)
from ventes_monoprix
group by ROLLUP(id_produit);
```

27 Lignes car lorsque l'on fait cette requête, on va obtenir les lignes obtenues sans le rollup puis une dernière ligne dans cette requête, qui sera la somme de tous les montant journalier

```
SELECT sum(montant_journalier)
from ventes_monoprix
group by CUBE(id_produit);
```

De meme pour cette requete avec CUBE

2)

```
SELECT sum(montant_journalier)
from ventes_monoprix
group by(id_produit, id_ville);
```

```
SELECT sum(montant_journalier)
from ventes_monoprix
group by ROLLUP(id_produit, id_ville);
```

Ici, on obtiendra toujours les ligne sélectionnées avec la requête sans aucun option, mais, on obtiendra en plus les lignes des montant total journalier de chaque produit et le montant total journalier de tous les produits vendue.

```
SELECT sum(montant_journalier)
from ventes_monoprix
group by CUBE(id_produit, id_ville);
```

Avec le CUBE en plus d'obtenir ce que l'on aura avec la requête précédente avec le ROLLUP, on obtiendra aussi le montant journalier total par ville

3)

```
SELECT sum(montant_journalier)
from ventes_monoprix
group by(id_produit, id_date);
```

```
SELECT sum(montant_journalier)
from ventes_monoprix
group by ROLLUP(id_produit, id_date);
```

On obtiendra toujours les ligne sélectionnées avec la requête sans aucun option, mais, on obtiendra en plus les lignes des montant total journalier de chaque produit et le montant total journalier de tous les produits vendue.

```
SELECT sum(montant_journalier)
from ventes_monoprix
group by CUBE(id_produit, id_date);
```

Avec le CUBE en plus d'obtenir ce que l'on aura avec la requête précédente avec le ROLLUP, on obtiendra aussi le montant journalier total par date

4)

```
SELECT avg(montant_journalier)
from ventes_monoprix
group by(id_magasin, id_date);
```

```
SELECT avg(montant_journalier)
```

```
from ventes_monoprix
group by ROLLUP(id_magasin, id_date);
```

On obtiendra toujours les ligne sélectionnées avec la requête sans aucun option, mais, on obtiendra en plus les lignes des moyennes de montant journalier de chaque produit et la moyenne du montant journalier de tous les produits vendue.

```
SELECT avg(montant_journalier)
from ventes_monoprix
group by CUBE(id_magasin, id_date);
```

Avec le CUBE en plus d'obtenir ce que l'on aura avec la requête précédente avec le ROLLUP, on obtiendra aussi la moyenne des montant journalier par date

5)

```
SELECT sum(montant_journalier)
from ventes_monoprix
group by(id_date);
```

```
SELECT sum(montant_journalier)
from ventes_monoprix
group by ROLLUP(id_date);
```

On obtiendra toujours les ligne sélectionnées avec la requête sans aucun option, mais, on obtiendra en plus les lignes des montant total journalier par date et le montant total journalier de tous les produits vendue.

```
SELECT sum(montant_journalier)
from ventes_monoprix
group by CUBE(id_date);
```

Rien ne changera de la requête avec le ROLLUP

6)

```
SELECT sum(montant_journalier)
from ventes_monoprix
group by(id_produit, id_ville, id_date);
```

```
SELECT sum(montant_journalier)
from ventes_monoprix
group by ROLLUP(id_produit, id_ville, id_date);
```

On obtiendra toujours les ligne sélectionnées avec la requête sans aucun option, mais, on obtiendra en plus les lignes des montant total journalier par produit et ville, e montant total journalier par produit et le montant total journalier de tous les produits vendue.

```
SELECT sum(montant_journalier)
from ventes_monoprix
group by CUBE(id_produit, id_ville, id_date);
```

Avec le CUBE en plus d'obtenir ce que l'on aura avec la requête précédente avec le ROLLUP, on obtiendra aussi le montant journalier total par date et par produit, le montant journalier total par ville et date, le montant journalier total par ville et le montant journalier total par date

Requêtes regroupant toutes les interrogations précédentes

```
SELECT sum(montant_journalier), avg(montant_journalier)
from ventes_monoprix
group by CUBE(id_produit, id_ville, id_date, id_magasin);
```

Cette requête nous répondra à toutes les interrogations posaient, mais pas que. On obtiendra aussi des réponses à des interrogations supplémentaires.

Partie 2

4) Classification des faits

1)

a) Fait transactionnel: On insère une ligne lors d'un événement en particuliers

b) Additif: car on peut faire des opérations calculatoire sur le x pour chaque attribut du fait

2)

a) Fait transactionnel: On insère une ligne lors d'un événement en particuliers

b) Semi-additif: car $x+p$ n'a pas de sens le chiffre d'affaires, car il est calculés pour tous les produits

3)

a) Snapshot: Car cela correspond à une photo à un instant qui ici est le jour du stock pour un produit donné dans le magasin m

b) Semi-additif: Car faire la moyenne d'un stock ne donne aucun mouvement sur le flux des produits

4)

a) Snapshot: Car on prend une photo pour chaque produit de chaque magasin une photo du stock

b) Non-Additif: car x est déjà en rapport avec les autres attributs

5)

a) Fait transactionnel: On insère une ligne lors d'un événement en particuliers

b) Non-additif: Aucune valeur ne peut être calculé

6)

a) Fait transactionnel: On insère une ligne lors d'un événement en particuliers

b) Non-additif: Car on ne peut pas faire la somme.

7)

a) Fait transactionnel: On insère une ligne lors d'un événement en particuliers

b) Additif: On peut calculer le temps passer par un employé au téléphone avec ses clients ou faire la moyenne des durée des appels.

8)

a) Fait transactionnel: On insère une ligne lors d'un événement en particuliers

b) Semi-additif: car $x+m$ n'a pas de sens

9)

a) Fait transactionnel: On insère une ligne lors d'un événement en particuliers

b) Non-additif