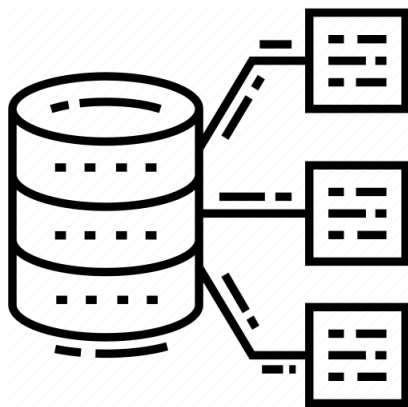




DEPARTEMENT INFORMATIQUE  
DE LA FACULTE DES SCIENCES

Quentin Yeché (21520370), Yanis Allouch (21708237)

## Rapport du TP N°3 : Entrepôts de données



**HMIN122M — Entrepôts de données et Big-Data**

Référent: Federico Ulliana et Anne-Muriel  
Chifolleau

2020

## Table des matières

|   |  |   |
|---|--|---|
| 1 | Les interrogations : requêtes transactionnelles vs analytiques | 3 |
| 2 | Un entrepôt de données pour Amazon                             | 4 |
| 3 | Requêtes Analytiques   | 5 |
| 4 | Classification des faits                                       | 6 |
| 5 | Annexe   | 8 |

## Introduction

L'objectif est d'étudier des cas pratiques d'étude de Bid-Data et de Data Warehousing, notamment au travers d'une étude simplifiée des ventes de Amazon. Nous abordons les différences entre les requêtes transactionnelles et analytiques ainsi que la construction de modèle relationnelle en étoile.

L'environnement de travail est composé de [Draw.io](#) pour la schématisation, ORACLE sur l'instance *pmaster* de la faculté des sciences pour le RDB<sup>1</sup> de test.

---

1. Relational Data Base

## 1 Les interrogations : requêtes transactionnelles vs analytiques

Considérons le cas d'un système d'information d'une chaîne nationale de cinémas multisalles. Pour chacune des requêtes suivantes, identifiez s'il s'agit d'une requête transactionnelle ou analytique. Justifiez vos réponses, en faisant des hypothèses si nécessaire.

1. Restent-ils des billets à Montpellier pour la séance de 20 heures du film "Logan" ?
  - Requête transactionnelle : en effet on interroge notre bdd pour obtenir une réponse binaire OUI/NON. Un client de ce cinéma peut très probablement poser cette question dans l'objectif de passer une transaction.
2. Aurait-on éventuellement pu proposer des plus grandes salles et plus de séances pour le dernier film de Star Wars ?
  - Requête analytique : en effet on cherche à faire de l'optimisation et de la stratégie en utilisant des données déjà existantes. Enfin, le client ne peut pas être à l'origine de cette requête.
- 3.

```
1 SELECT Film.titre, Cinema.nom, Date.mois, COUNT(Place.placeID)
2 FROM Film, Ventes, Cinema, Place, Temps, Date
3 WHERE Ventes.filmID = Film.filmID AND Ventes.cinemaID = Cinema.cinemaID
4 AND Ventes.tempsID = Temps.tempsID AND Place.cinemaID = Cinema.cinemaID
5 AND Ventes.dateID = Date.dateID
6 GROUP BY Film.titre, Cinema.nom, Date.mois
```

- Requête analytique : en effet on cherche à comparer le nombre de place par film par cinéma et par mois, chose qu'un client ou un utilisateur n'est pas voué à connaître.

4.

```
1 SELECT Temps.creneau, COUNT(*)
2 FROM Ventes, Temps
3 WHERE Ventes.tempsID = Temps.tempsID
4 GROUP BY Temps.creneau
```

- Requête analytique : en effet on cherche à comparer le nombre de ventes pour chaque créneau. Une utilisation possible de cette information serait de connaître s'il est nécessaire d'augmenter les effectifs en caisse pour potentiellement augmenter le nombre de ventes.

5.

```
1 INSERT INTO Ventes
2 VALUES ('film1', 'cinema24', 'date', 'temps3', 'place44', '7.50')
```

- Requête transactionnelle : en effet on effectue une insertion qui correspond simplement à la vente d'un ticket de cinéma.

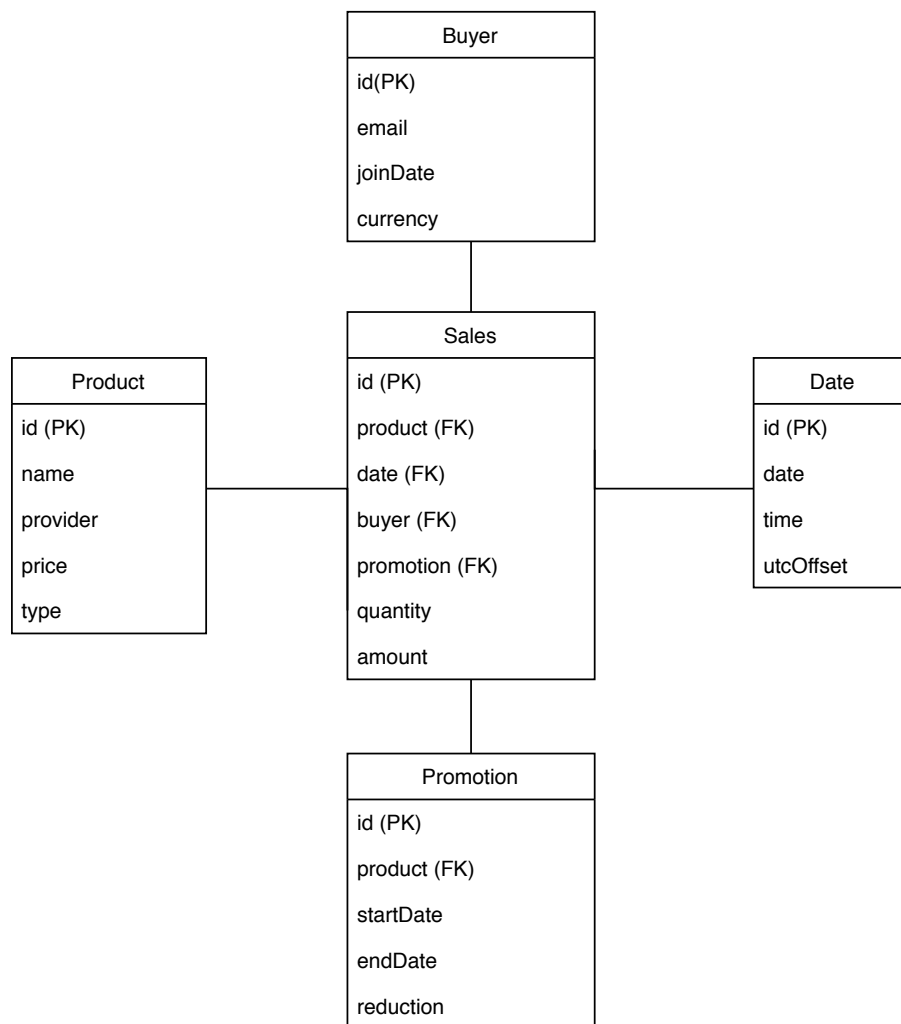


FIGURE 1 – Modèle en étoile - Question 2.1

## 2 Un entrepôt de données pour Amazon

1. Proposez un modèle en étoile pour un entrepôt de données permettant l'analyse des ventes dans Amazon. Le modèle doit pouvoir permettre une analyse temporelle des ventes en fonction des *produits*, des *utilisateurs*, ainsi que des *promotions* mises en place.

Voir [Figure 1](#).

2. Comment feriez vous pour intégrer les commentaires client dans ce modèle ?

Les commentaires, et plus généralement les avis des utilisateurs (qui peuvent également inclure les évaluations produit sur une échelle), seraient l'objet d'une autre table de faits. Cette table pourrait partager des dimensions avec d'autres tables de faits. Pour garder la métaphore du modèle en étoile, on parle alors de constellation.

- Proposez trois requêtes analytiques pour le modèle de données en étoile que vous avez conçu.

```
1 SELECT promotion, sum(quantity), sum(amount) FROM Sales JOIN Date on date
   =Date.id
2 WHERE date.mois = "decembre"
3 GROUP BY promotion;
4
```

```
1 SELECT promotion, avg(amount), avg(quantity) from Sales JOIN Date on date
   =Date.id
2 WHERE date.mois = "janvier";
3
```

```
1 SELECT product, sum(quantity) FROM Sales
2 GROUP BY product HAVING sum(quantity) >= ALL(SELECT sum(quantity) FROM
   Sales GROUP BY product);
3
```

### 3 Requêtes Analytiques

Considérez la table de faits (simplifiée) suivante, enregistrant les ventes journalières chez Monoprix (attention : un script de création et remplissage de la base est à disposition dans la page Moodle du cours).

```
1 CREATE TABLE ventes_monoprix (
2   id_date VARCHAR(10) NOT NULL,
3   id_produit VARCHAR(10) NOT NULL,
4   id_magasin VARCHAR(10) NOT NULL,
5   id_ville VARCHAR(10) NOT NULL,
6   montant_journalier NUMBER(10,2) NOT NULL
7 );
```

Exprimez en SQL les interrogations suivantes, à l'aide de l'opérateur GROUP-BY.

- Donner le montant total des ventes par produit.

```
1 select sum(montant_journalier) from ventes_monoprix
2 group by id_produit;
```

- Donner le montant total des ventes par produit et par ville.

```
1 select sum(montant_journalier) from ventes_monoprix
2 group by (id_produit,id_ville);
```

- Donner le montant total des ventes par produit et par jour.

```
1 select sum(montant_journalier) from ventes_monoprix
2 group by (id_produit, id_date);
```

4. Donner la moyenne du montant des ventes par magasin et par jour.

```
1 select avg(montant_journalier) from ventes_monoprix
2 group by (id_produit, id_date);
```

5. Donner le montant total des ventes par ville par jour.

```
1 select sum(montant_journalier) from ventes_monoprix
2 group by (id_ville, id_date);
```

6. Donner le montant total des ventes par produit, ville et jour.

```
1 select sum(montant_journalier) from ventes_monoprix
2 group by (id_produit, id_ville, id_date);
```

Enfin, testez les options ROLLUP et CUBE et comparer les résultats. Pourrait-on regrouper les interrogations grâce à ces options ?

Les opérateurs ROLLUP et CUBE de Oracle calculent des lignes qui vont au-delà de l'objectif des requêtes précédentes prises indépendamment. Prenons comme cas d'exemple la requête N°1 et voyons comment l'utilisation du *Rollup* ajoute de l'information à notre questionnement de base.

En effet on peut regrouper certaine requêtes ensemble et obtenir les mêmes résultats avec moins de requêtes.

En détail :

```
1 select sum(montant_journalier) from ventes_monoprix
2 group by cube (id_produit, id_ville, id_date);
3
```

Cette requête effectuera un GROUP BY sur toutes les permutations de n'importe quel nombre de colonnes parmi *id\_produit*, *id\_ville*, *id\_date*. Elle nous donne donc les résultats des requêtes 1,2,3,5 et 6.

## 4 Classification des faits

Considérez la liste de faits ci-dessous, chacun correspondant à un tuple de la table des faits de l'entrepôt.

- a) Précisez s'il s'agit d'un fait transactionnel ou d'un snapshot. Justifiez votre réponse.
- b) Précisez si (l'éventuelle) mesure correspondante est additive, semi-additive ou non-additive. Justifiez votre réponse.
  1. Un fait (j, p, c,m, x) existe lorsqu'un produit p est acheté par un client c le jour j au magasin m. La mesure x correspond au prix total.
    - a) Fait transactionnel, qui a pour but d'ajouter de l'information à l'entrepôt de donnée.
    - b) La mesure x est additive, en effet elle peut-être calculer avec ces semblables.
  2. Un fait (j, p,m, x) existe lorsqu'un produit p est acheté le jour j au magasin m. La mesure correspond au chiffre d'affaires.

- a) Fait transactionnel, qui a pour but d'ajouter de l'information à l'entrepôt de donnée.
  - b) La mesure  $x$  est additive, en effet elle peut-être calculer avec ces semblables.
3. Un fait  $(j, p, m, x)$  existe pour chaque combinaison de produit  $p$ , magasin  $m$  et jour  $j$ . La mesure  $x$  correspond au stock de  $p$  en  $m$  le jour  $j$ .
- a)
  - b) La mesure  $x$  est additive, en effet elle peut-être calculer avec ces semblables.
4. Un fait  $(j, p, m, x)$  existe pour chaque combinaison de produit  $p$ , magasin  $m$  et jour  $j$ . La mesure  $x$  correspond au nombre de ventes de  $p$  en  $m$  cumulées depuis le début de l'année jusqu'au jour  $j$ .
- a)
  - b) La mesure  $x$  est semi-additive, en effet elle peut-être calculer avec ces semblables sous conditions qu'on respecte le contrat "le nombre de produit cumulées depuis le début de l'année jusqu'au jour  $j$ ".
5. Un fait  $(c, e, j)$  existe lorsqu'un appel du client  $c$  le jour  $j$  est traité par l'employé  $e$ . Aucune mesure n'existe.
- a)
  - b)
6. Un fait  $(c, j, x)$  existe lorsqu'un client  $c$  le jour  $j$  laisse une note sur un produit acheté. La mesure  $x$  est la note donnée par le client.
- a)
  - b)
7. Un fait  $(c, e, j, x)$  existe lorsqu'un appel du client  $c$  le jour  $j$  est traité par l'employé  $e$ . La mesure  $x$  est la durée de l'appel en secondes.
- a)
  - b)
8. Un fait  $(m, b, j, x)$  existe lorsque la monnaie  $m$  est changée à la banque  $b$  le jour  $j$ . La mesure  $x$  est le montant total de la monnaie changée en euros.
- a)
  - b)
9. Un fait  $(m, b, j, x)$  existe lorsque la monnaie  $m$  est changée à la banque  $b$  le jour  $j$ . La mesure  $x$  est le cours de change moyen de  $m$  en euros pour toutes les transactions du jour  $j$ .
- a)
  - b)



## **5 Annexe**