

# TP3

---

TP3.pdf

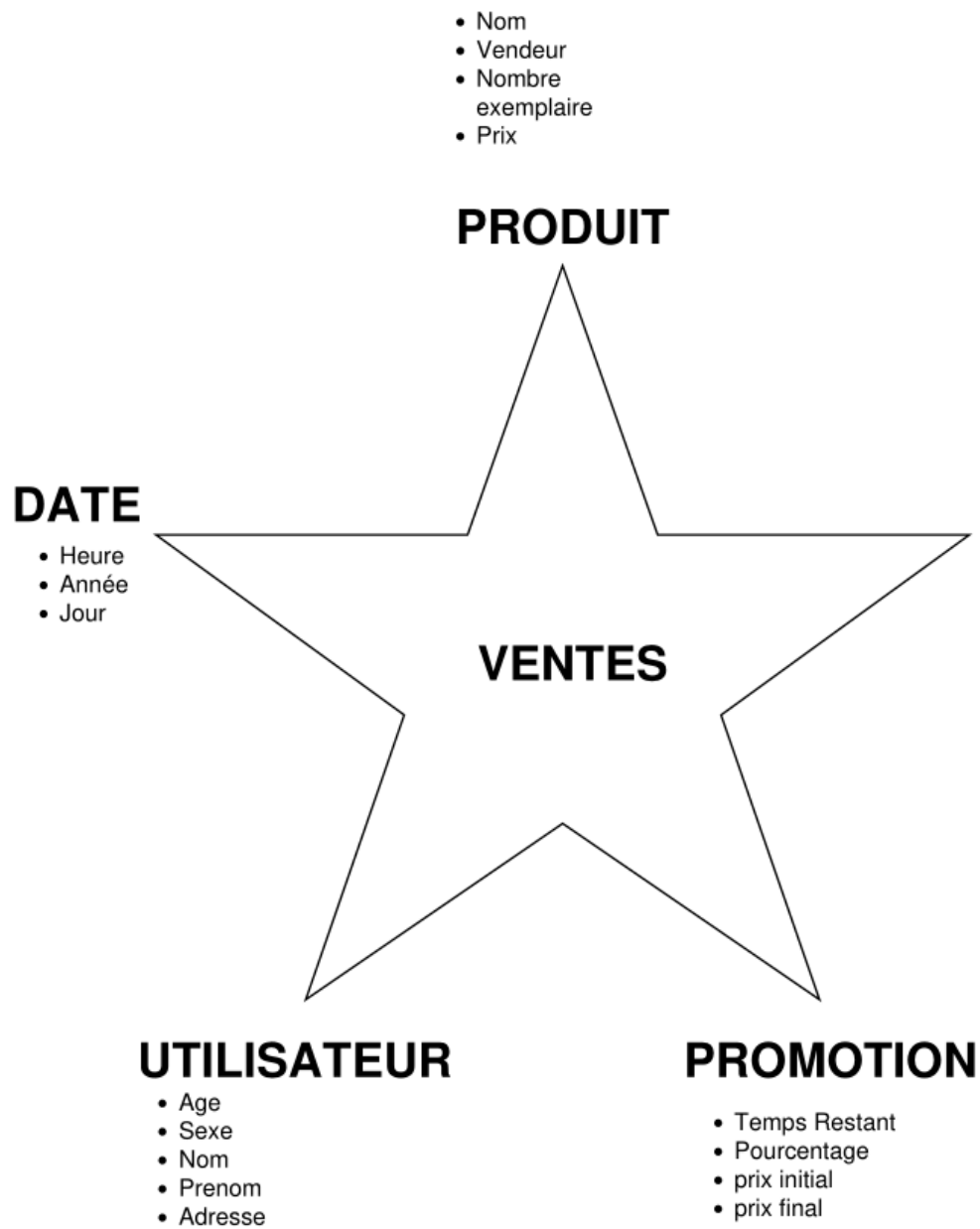
---

## Exercice 1

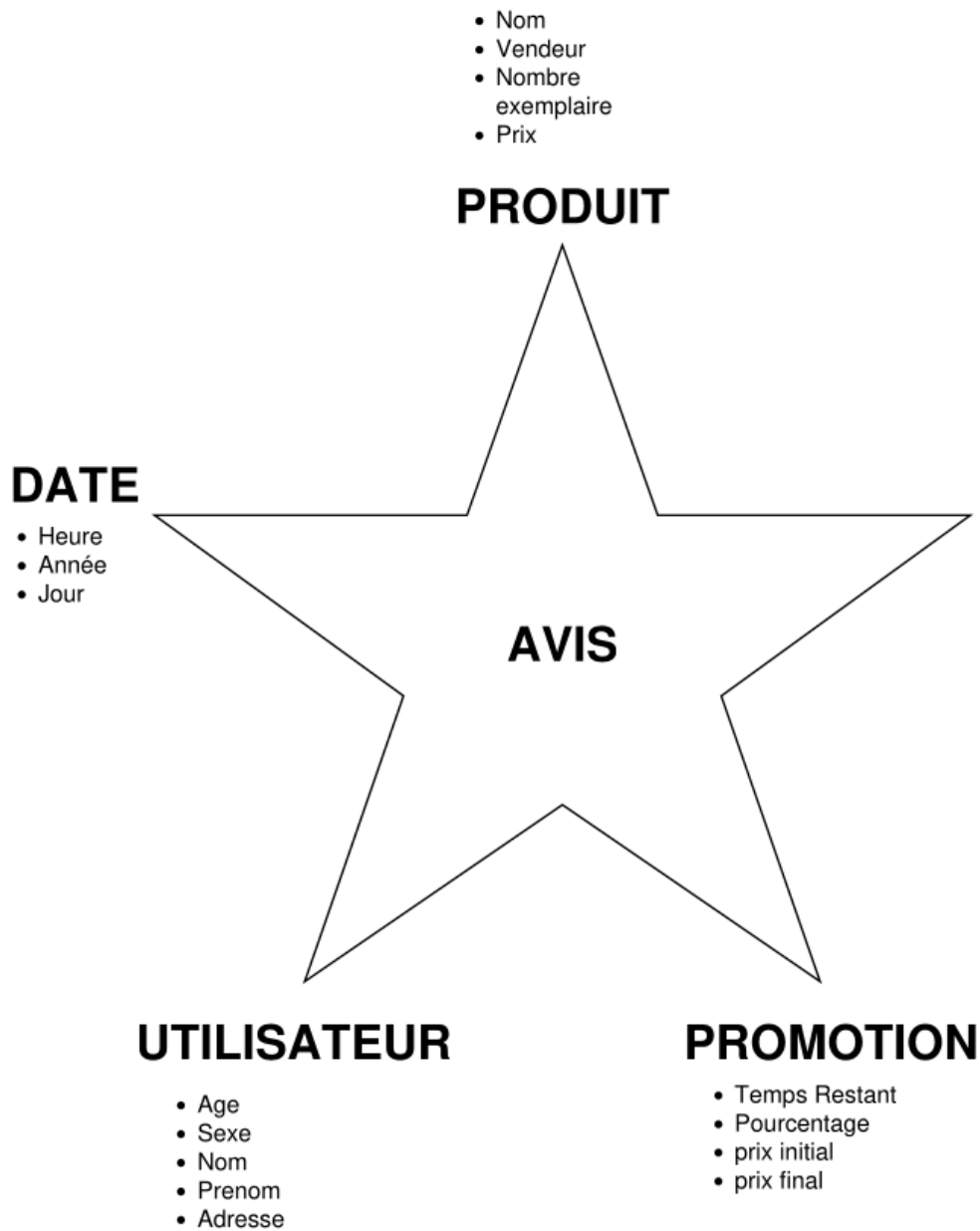
1. Transactionnel : On cherche une information dans la base de données assez précise, ici dans une ville, un film et une heure particulière
2. Analytique : On cherche à faire une analyse pour savoir si on aurait du faire une certaine action, ici augmenter le nombre de place et de séance pour *Star Wars*
3. Le nombre de spectateurs par titre de film, par nom de cinéma et par mois. C'est une requête analytique car on peut faire des statistiques sur la fréquentation des différents films en fonction des cinémas (par exemples des régions) et des périodes (par exemple Noël etc)
4. La fréquentation des cinémas en fonction du temps. C'est une requête analytique également car on peut analyser quand les gens vont le plus au cinéma
5. Insertion d'une vente de place. C'est une requête transactionnel car les insert to sont plus propice aux base de données relationnels par rapport aux entrepôts de données

## Exercice 2

- 1.



2.



3.

```
SELECT Num_Tel FROM UTILISATEUR WHERE SEXE = "FEMME" AND AGE = 20  
AND ADRESSE = "MONTPELLIER";
```

- Produits les plus vendus par mois
- Nombre de ventes par mois d'un produit
- Produit le plus vendu par région

### Exercice 3

## 1. Donner le montant total des ventes par produit

```
SELECT id_produit, SUM(montant_journalier)
FROM ventes_monoprix GROUP BY id_produit;
-- Exemple : Z = 19 796 575
```

```
SELECT id_produit, SUM(montant_journalier) FROM ventes_monoprix
GROUP BY rollup(id_produit);
-- Pareil que la requête de base
```

```
SELECT id_produit, SUM(montant_journalier) FROM ventes_monoprix
GROUP BY cube(id_produit);
-- Pareil que la requête de base
```

## 2. Donner le montant total des ventes par produit et par ville

```
SELECT id_produit, id_ville, SUM(montant_journalier) FROM ventes_monopr
ix
GROUP BY id_produit, id_ville;
-- Exemple : Produit Q dans la ville Y = 556 974
```

```
SELECT id_produit, id_ville, SUM(montant_journalier) FROM ventes_monopr
ix
GROUP BY rollup(id_produit, id_ville);
-- Pareil que la requête de base avec en plus montant par produit
-- Exemple : Z = 19 796 575
```

```
SELECT id_produit, id_ville, SUM(montant_journalier) FROM ventes_monopr
ix
```

```
GROUP BY cube(id_produit, id_ville);  
-- Pareil que rollup avec en plus total par ville  
-- Exemple : ville Z = 20 344 082
```

### 3. Donner le montant total des ventes par produit et par jour

```
SELECT id_produit, id_date, SUM(montant_journalier) FROM ventes_monop  
rix  
GROUP BY id_produit, id_date;  
-- Exemple : Produit Z à la date Y = 911 710
```

```
SELECT id_produit, id_date, SUM(montant_journalier) FROM ventes_monop  
rix  
GROUP BY rollup(id_produit, id_date);  
-- Pareil que la requête de base avec en plus montant par produit  
-- Exemple : Z = 19 796 575
```

```
SELECT id_produit, id_date, SUM(montant_journalier) FROM ventes_monop  
rix  
GROUP BY cube(id_produit, id_date);  
-- Pareil que rollup avec en plus total par date  
-- Exemple : date Z = 17 305 200
```

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

```
SELECT id_magasin, id_date, AVG(montant_journalier) FROM ventes_mono  
prix  
GROUP BY id_magasin, id_date;  
-- Exemple : Produit Z à la date Y = 60 780
```

```
SELECT id_magasin, id_date, AVG(montant_journalier) FROM ventes_mono  
prix
```

```
GROUP BY rollup(id_magasin, id_date);  
-- Pareil que la requête de base avec en plus moyenne par produit  
-- Exemple : Z = 49 921
```

```
SELECT id_produit, id_date, AVG(montant_journalier) FROM ventes_monoprix  
ix  
GROUP BY cube(id_produit, id_date);  
-- Pareil que rollup avec en plus moyenne par date  
-- Exemple : date Z = 47 153
```

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

```
SELECT id_ville, id_date, SUM(montant_journalier) FROM ventes_monoprix  
GROUP BY id_ville, id_date;  
-- Exemple : Ville E à la date O = 438 836
```

```
SELECT id_ville, id_date, SUM(montant_journalier) FROM ventes_monoprix  
GROUP BY rollup(id_ville, id_date);  
-- Pareil que la requête de base avec en plus total par ville  
-- Exemple : ville Z = 20 344 082
```

```
SELECT id_ville, id_date, SUM(montant_journalier) FROM ventes_monoprix  
GROUP BY cube(id_ville, id_date);  
-- Pareil que rollup avec en plus total par date  
-- Exemple : date Z = 17 305 200
```

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

```
SELECT id_produit, id_ville, id_date, SUM(montant_journalier)  
FROM ventes_monoprix  
GROUP BY id_produit, id_ville, id_date;  
-- Exemple : Produit N dans la ville G à la date I = 80 212
```

```
SELECT id_produit, id_ville, id_date, SUM(montant_journalier)
FROM ventes_monoprix
GROUP BY rollup(id_produit, id_ville, id_date);
-- Pareil que la requête de base avec en plus total par ville / produit
-- et total par produit
-- Exemple : Produit Z dans la ville Z = 740 819
```

```
SELECT id_produit, id_ville, id_date, SUM(montant_journalier)
FROM ventes_monoprix
GROUP BY cube(id_produit, id_ville, id_date);
-- Pareil que rollup avec en plus tous les triplets possible, soit total
-- par produit, total par ville, total par date, total par produit / ville
-- total par produit / date, total par ville / date et total par
-- produit / ville / date
-- Exemple : Produit Z de la ville Z à la date R = 145 361
-- Exemple : Produit Z à la date X = 860 106
```

## Exercice 4

1. Transactionnel et additif
2. Snapshot et additif
3. Snapshot et semi-additif
4. Snapshot et semi-additif
5. Transactionnel et void-additif
6. Transactionnel et non-additif
7. Transactionnel et additif
8. Snapshot et additif
9. Snapshot et non-additif