

INFORMATIQUE DECISIONNELLE Data warehousing

Entrepôt de données

Binôme :

DJEBIEN Tarik

RAKOTOBE Sitraka Eric

(Pour Exo 1 Q4 + Exo 3 Q3)

Groupe : MIAGE FA

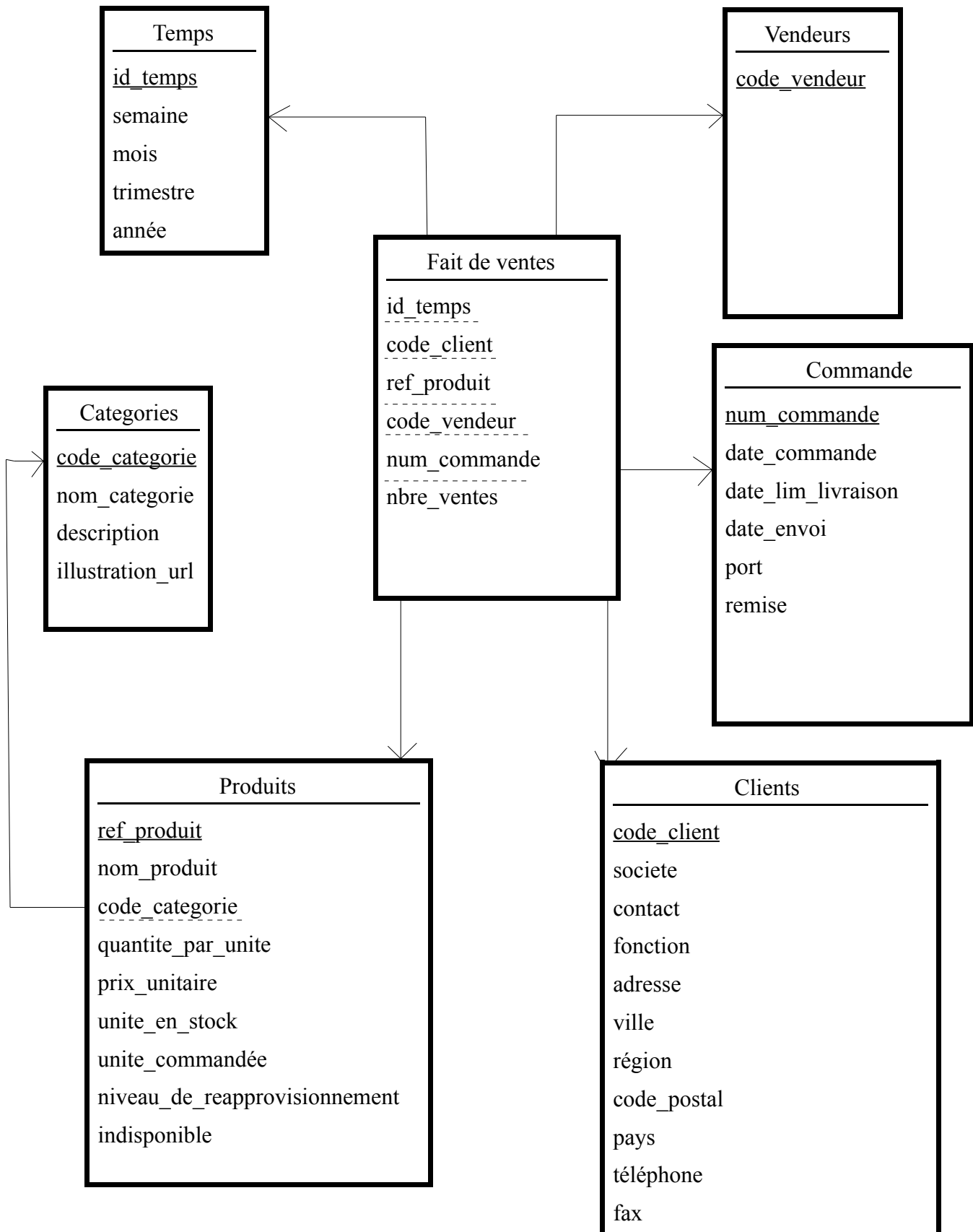
Date: Novembre 2011

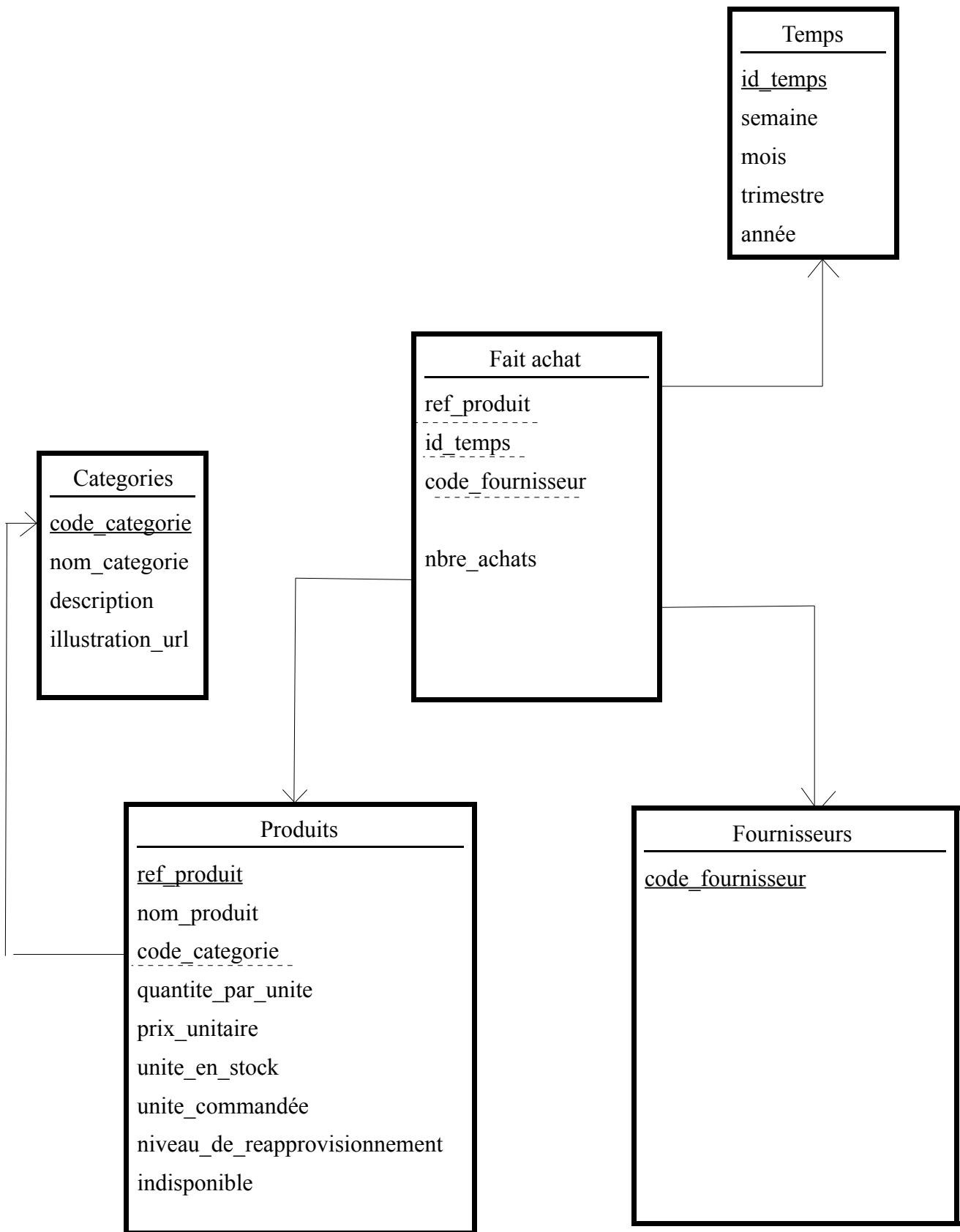


Exercice 1 : Schéma en étoiles et rapports

Question 1 : voir /TP_ID/Exercice1/q1/question1.sql

Question 2.





Question 3.

Création de la base de données:

TEMPS (ID_TEMPS, SEMAINE, MOIS, TRIMESTRE, ANNEE)

VENDEUR (CODE_VENDEUR)

CATEGORIE (CODE_CATEGORIE, NOM_CATEGORIE, DESCRIPTION,
ILLUSTRATION_URL)

PRODUIT (REF_PRODUIT, NOM_PRODUIT, CODE_CATEGORIE,
QUANTITE_UNITE, PRIX_UNITE, UNITE_STOCK,
UNITE_COMMANDEE, NIVEAU_REAPPRO, INDISPONIBLE)

COMMANDE (NUM_COMMANDE, DATE_COMMANDE, DATE_LIM_LIVRAISON,
DATE_ENVOI, PORT, REMISE)

CLIENT (CODE_CLIENT, SOCIETE, CONTACT, FONCTION, ADRESSE, VILLE,
REGION, CODE_POSTAL, PAYS, TELEPHONE, FAX)

FOURNISSEUR (CODE_FOURNISSEUR)

FAIT_VENTES (ID_TEMPS, CODE_CLIENT, REF_PRODUIT, CODE_VENDEUR,
NUM_COMMANDE)

FAIT_ACHAT (REF_PRODUIT, ID_TEMPS, CODE_FOURNISSEUR)

Question 4 : voir /TP_ID/Exercice1/q4

Question 5 : voir /TP_ID/Exercice1/q5

Exercice 2 : Rappels SQL et requêtes de classement sous Oracle

Il existe un script SQL dans /TP_ID/Exercice2/reponses.sql

DEPT(DEPTNO, DNAME , LOC)

EMP(EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)

ALTER TABLE Emp
ADD CONSTRAINT fk_deptno
FOREIGN KEY (DEPTNO)
REFERENCES Dept(DEPTNO);

ALTER TABLE Emp
ADD CONSTRAINT pk_empno
PRIMARY KEY (EMPNO);

Question : a

SELECT sal **FROM** emp **WHERE** deptno=10 **OR** deptno=30 **ORDER BY** sal;

Question : c

SELECT **ROUND**(**AVG**(sal)) **FROM** emp **GROUP BY** deptno ;

Question : d

SELECT **ROW_NUMBER**() **OVER**(**ORDER BY** sal **DESC**) **AS** "numero", deptno, sal
FROM emp
WHERE deptno=10 **OR** deptno=20
GROUP BY deptno,sal
ORDER BY deptno;

Question : e

SELECT job, **SUM**(sal) **as** salaire_total **FROM** emp **GROUP BY** job;
SELECT **DISTINCT** job , **SUM**(sal) **OVER**(**partition by** job) **AS** salaire_total **FROM** emp;

Question : f

Partition by indique un découpage logique, la fonction analytique utilisée ne calcule qu'à l'intérieur d'une partition. C'est tout à fait similaire au group by dans une requête traditionnelle.

(http://docs.oracle.com/cd/B19306_01/server.102/b14231/partiti.htm#sthref2652)

Question : g

-montant total des salaires versés tout département et job confondus:

SELECT **sum**(sal) **as** montant_total_salaire_versee **from** emp;

-montant total des salaires versés par département:

SELECT deptno,**sum**(sal) **from** emp **group by** deptno;

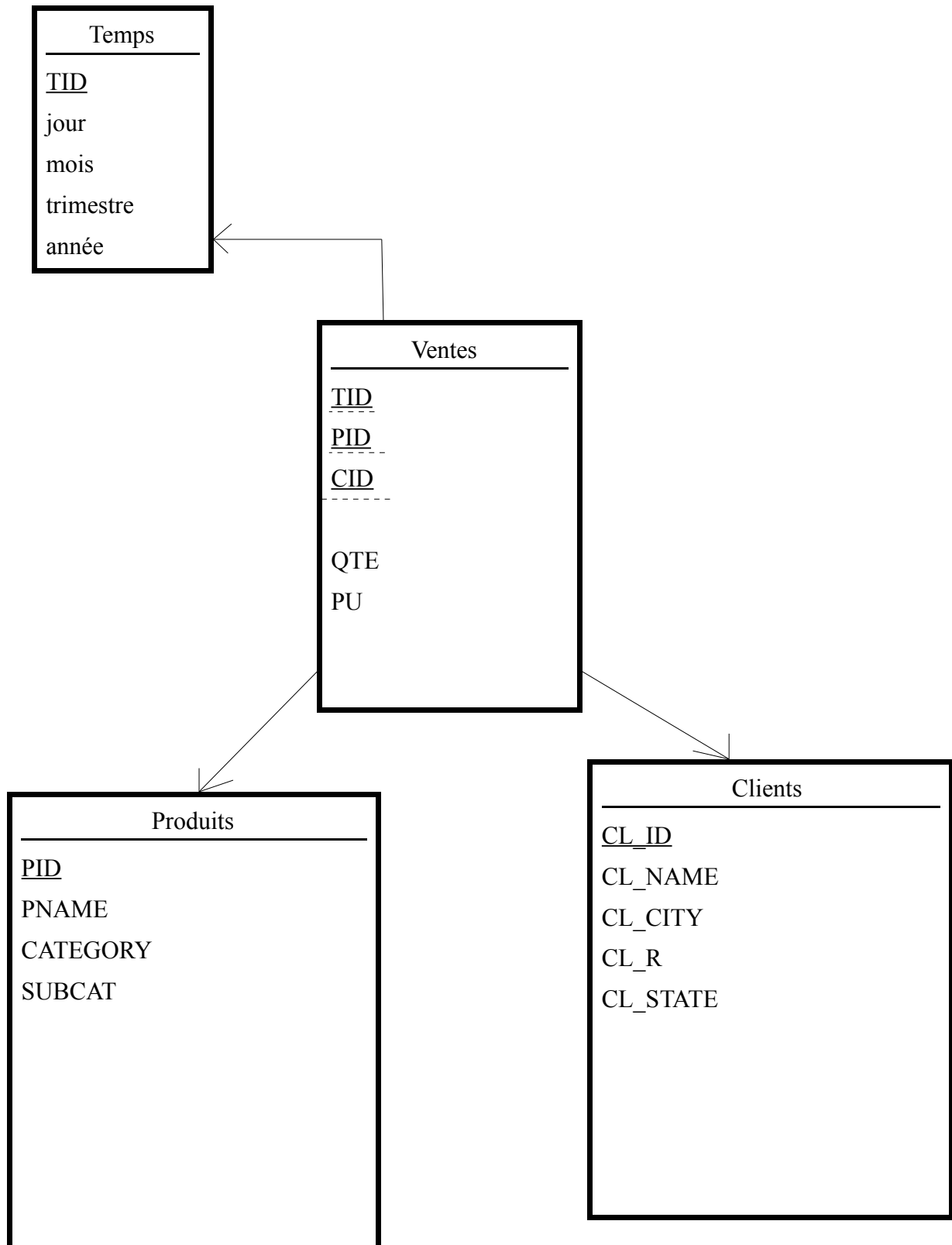
-montant total des salaires versés par département et job:

SELECT job,deptno,**sum**(sal) **from** emp **group by** job,deptno ;

Les questions b et h ne sont pas traitées comme l'a indiquée le professeur..

Exercice 3 : OLAP

Question 1 : MCD du schema en étoile pour le fait de ventes :



Question 2 :

Création et modification du schéma de la base de données:

TEMPS (TID, JOUR, MOIS, TRIMESTRE, ANNEE)

PRODUIT (PID, PNAME, CATEGORY, SUBCAT)

CLIENT (CL_ID, CL_NAME, CL_CITY, CL_R, CL_STATE)

VENTES (TID, PID, CID, QTE, PU)

Libellé :

TEMPS

TID : identifiant d'une date

JOUR : le jour de la semaine

MOIS : le mois

TRIMESTRE : le trimestre de l'année

ANNEE : l'année

(tous sont des numéros)

PRODUIT

PID : identifiant d'un produit

PNAME : son nom

CATEGORY : sa catégorie

SUBCAT : sa sous-catégorie

CLIENT

CL_ID : identifiant d'un client

CL_NAME : son nom

CL_CITY : sa ville

CL_R : sa région (divers suivant les pays)

CL_STATE : son pays

Il existe un script SQL dans TP_ID/Exercice3/q2/question2.sql

Question 3 :

i) Moyenne des ventes

- La moyenne des ventes par année, région, catégorie:

```
SELECT t.annee, c.cl_r, p.category , AVG(v.qte * v.pu) as moyenne_ventes
FROM ventes v, produits p , temps t, clients c
WHERE v.tid = t.tid
AND v.cid = c.cl_id
AND v.pid = p.pid
GROUP BY (t.annee, c.cl_r, p.category);
```

- La moyenne des ventes par année, région :

```
SELECT t.annee, c.cl_r , AVG(v.qte * v.pu) as moyenne_ventes
FROM ventes v, temps t , clients c
WHERE v.tid = t.tid
AND v.cid = c.cl_id
GROUP BY (t.annee, c.cl_r) ;
```

- La moyenne des ventes par année pour les années 2009 et 2010:

```
SELECT t.annee, AVG(v.qte * v.pu)
FROM ventes v, temps t
WHERE v.tid = t.tid
AND t.annee IN (2009,2010)
GROUP BY rollup(t.annee);
```

ii) La moyenne des ventes par année, selon les dimensions région et catégorie pour les années 2009 et 2010:

```
SELECT t.annee, c.cl_r, p.category, AVG(v.qte * v.pu)
FROM ventes v, produits p , temps t, clients c
WHERE v.tid = t.tid
AND v.cid = c.cl_id
AND v.pid = p.pid
AND t.annee in (2009,2010)
GROUP BY rollup(t.annee,c.cl_r, p.category);
```

iii) Le produit le plus vendu par année et par catégorie (cf. RANK):

```
SELECT * FROM (
  SELECT t.annee , p.category , p.pname , v.qte as quantiteVendue,
    rank() over (partition by annee, category order by v.qte desc) as rangProduit
  FROM ventes v, produits p , temps t, clients c
  WHERE v.tid = t.tid
    AND v.cid = c.cl_id
    AND v.pid = p.pid
  GROUP BY rollup (t.annee , p.category ) , p.pname , v.qte
  ORDER BY annee
)
WHERE rangProduit = 1;
```


iv)

Pour chaque année, donner le total des ventes ainsi que le total des ventes par catégorie. On ne veut pas le résultat pour le total des années (utiliser la fonction GROUPING_ID):

```
SELECT t.annee, SUM(v.qte) as total_ventes
FROM ventes v, produits p , temps t, clients c
WHERE v.tid = t.tid
AND v.cid = c.cl_id
AND v.pid = p.pid
GROUP BY grouping sets ( rollup (t.annee) )
;
```

Pour chaque année, donner le total des ventes par catégorie (utiliser la fonction GROUPING_ID):

```
SELECT t.annee, p.category ,
grouping_id ( t.annee , p.category ) groupingid ,
sum(v.qte) as total_ventes_par_categorie
FROM ventes v, produits p , temps t, clients c
WHERE v.tid = t.tid
AND v.cid = c.cl_id
AND v.pid = p.pid
GROUP BY grouping sets ( rollup (t.annee, p.category) );
```

v)

Donner tous les totaux de ventes par année selon la dimension catégorie, Utilisez la commande GROUPING_SETS. :

```
SELECT t.annee, p.category , SUM(v.qte) as totaux_vente
FROM ventes v, produits p , temps t, clients c
WHERE v.tid = t.tid
AND v.cid = c.cl_id
AND v.pid = p.pid
GROUP BY grouping sets (rollup (t.annee,p.category)) ;
```

Donner tous les totaux de ventes selon la dimension cl_name (nom du client). Utilisez la commande GROUPING_SETS. :

```
SELECT c.cl_name , SUM(v.qte) AS totaux_vente
FROM ventes v, produits p , temps t, clients c
WHERE v.tid = t.tid
AND v.cid = c.cl_id
AND v.pid = p.pid
GROUP BY grouping sets (rollup (c.cl_name ));
```

vi) *Quel est le meilleur mois de vente du produit "siropDerable" pour chacune des années ?*

```
SELECT * FROM (  
SELECT t.annee, t.mois ,  
RANK() OVER (PARTITION BY t.annee ORDER BY SUM(v.qte) DESC) AS quantite_rang  
FROM ventes v, produits p , temps t  
WHERE v.tid = t.tid  
AND v.pid = p.pid  
AND p.pname = 'siropDerable'  
GROUP BY rollup( t.mois , t.annee )  
)  
WHERE quantite_rang = 1 ;
```

vii) *Quelle est la quantité de produits vendus pour chaque catégorie, les 5 premiers jours de chaque mois de 2010 ? (NTILE)*

```
SELECT * FROM  
(SELECT DISTINCT t.mois, p.category , SUM(v.qte) as quantite_vendue ,  
ntile(5) over (order by t.jour) as cinqpremierjour  
FROM ventes v, produits p , temps t, clients c  
WHERE v.tid = t.tid  
AND v.cid = c.cl_id  
AND v.pid = p.pid  
AND t.annee=2010  
GROUP BY rollup ( p.category , t.mois ) , t.jour  
ORDER BY p.category,t.mois  
)  
WHERE cinqpremierjour = 1 ;
```

viii) *Quelle est la répartition par tiers des catégories selon leurs quantités totales vendues en 2010 ?*

```
SELECT p.category,  
sum(v.qte) as quantite_vendue ,  
ntile(3) over (order by sum(v.qte)) as repartitioncategory  
FROM ventes v, produits p , temps t, clients c  
WHERE v.tid = t.tid  
AND v.cid = c.cl_id  
AND v.pid = p.pid  
AND t.annee=2010  
GROUP BY p.category ;
```

Il existe un script SQL dans TP_ID/Exercice3/q3/question3.sql