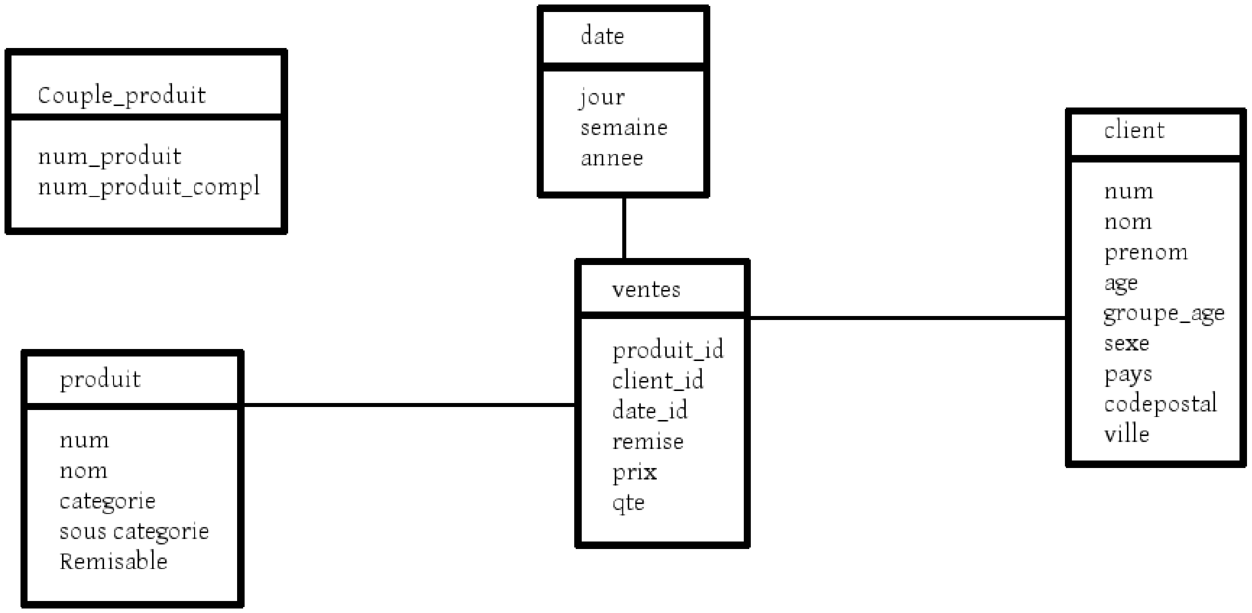
Thibault Rosa

# BI TP2



Le couple_produit permet de faire un lien entre les divers produits, afin de pouvoir conseiller ou ranger ensemble des produits qui ce complémentent.

1.View Produit ,

```
drop MATERIALIZED view v_produit;

column designation heading designation Format a30
column num heading num Format 9999
column categorie Format a20
column souscategorie Format a20
column remisable Format 9
Create materialized view  v_produit refresh force with primary key as
(SELECT num , substr(designation,1,instr(designation,'.',1,1)-1) as nom,
CASE
        WHEN instr(designation,'.',1,2) = 0 THEN
substr(designation,instr(designation,'.',1,1)+1,length(designation)-instr(designation,'.',1,1))
        ELSE substr(designation,instr(designation,'.',1,1)+1,instr(designation,'.',1,2)-
instr(designation,'.',1,1)-1)
END categorie,
CASE
        WHEN instr(designation,'.',1,2) = 0 THEN 'Pas de sc'
        ELSE substr(designation,instr(designation,'.',1,2)+1, length(designation) -
instr(designation,'.',1,2)+1)
END souscategorie,
CASE
        WHEN 1= 1 THEN 1
        ELSE 1
END remisable
FROM PRODUIT
where PRODUIT.NUM in (select PRIX_DATE.PRODUIT as num from  PRIX_DATE
                                where PRIX_DATE.remise > 0)
UNION
SELECT num , substr(designation,1,instr(designation,'.',1,1)-1) as nom,
CASE
        WHEN instr(designation,'.',1,2) = 0 THEN
substr(designation,instr(designation,'.',1,1)+1,length(designation)-instr(designation,'.',1,1))
        ELSE substr(designation,instr(designation,'.',1,1)+1,instr(designation,'.',1,2)-
instr(designation,'.',1,1)-1)
END categorie,
CASE
        WHEN instr(designation,'.',1,2) = 0 THEN 'Pas de sc'
        ELSE substr(designation,instr(designation,'.',1,2)+1, length(designation) -
instr(designation,'.',1,2)+1)
END souscategorie,
CASE
        WHEN 1= 0 THEN 0
        ELSE 0
END remisable
FROM PRODUIT
where not exists (select PRIX_DATE.PRODUIT as num from  PRIX_DATE
                                where PRIX_DATE.remise < 0
                                and     PRIX_DATE.PRODUIT = PRODUIT.NUM)
);
```

Cette requette créer une vue qui met le champs « remisable » à 1 si le produit à été une fois remisé.
View Client,

```
drop MATERIALIZED view v_client;
CREATE MATERIALIZED VIEW v_client refresh force with primary key AS
SELECT num, nom, prenom, to_number(EXTRACT(year FROM current_date))-
to_number(EXTRACT(year FROM date_nais)) as age,
    case
        when to_number(EXTRACT(year FROM current_date))-
to_number(EXTRACT(year FROM date_nais))<30 then '<30 ans'
        when to_number(EXTRACT(year FROM current_date))-
to_number(EXTRACT(year FROM date_nais))<=45 then '30-45 ans'
        when to_number(EXTRACT(year FROM current_date))-
to_number(EXTRACT(year FROM date_nais))<=60 then '45-60 ans'
        when to_number(EXTRACT(year FROM current_date))-
to_number(EXTRACT(year FROM date_nais))>60 then '>60 ans'
    end as groupe_age,
    sexe,
    substr(adresse,
        instr(adresse, ',',-1)+1,
        length(adresse)
        ) as pays,
    substr(
        substr(adresse,0,instr(adresse, ',',-1)-1),
        instr(substr(adresse,0,instr(adresse, ',',-1)-1), ',',-1)+1,
        length(substr(adresse,0,instr(adresse, ',',-1)-1))
        ) as code_postal,
    substr(
        substr(
            substr(adresse,0,instr(adresse, ',',-1)-1),
            0,
            instr(substr(adresse,0,instr(adresse, ',',-1)-1), ',',-1)-1
        ),
        instr(
            substr(
                substr(adresse,0,instr(adresse, ',',-1)-1),
                0,
                instr(substr(adresse,0,instr(adresse, ',',-1)-1), ',',-1)-1
            ),
             ',',
            -1)+1,
        length(
            substr(
                substr(adresse,0,instr(adresse, ',',-1)-1),
                0,
                instr(substr(adresse,0,instr(adresse, ',',-1)-1), ',',-1)-1
            )
        )
    )as ville
FROM client;
```

View date,

```
drop MATERIALIZED view v_date;
CREATE MATERIALIZED VIEW v_date refresh force with primary key AS
Select  ROWNUM as num,
EXTRACT(day FROM (level + TO_DATE('2000/01/01', 'yyyy/mm/dd')-1)) as  jour,
EXTRACT(month FROM (level + TO_DATE('2000/01/01', 'yyyy/mm/dd')-1)) as  mois,
EXTRACT(year FROM (level + TO_DATE('2000/01/01', 'yyyy/mm/dd')-1)) as  annee
from  dual
connect  by  level  <  current_date -TO_DATE('2000/01/01', 'yyyy/mm/dd')  + 2 ;
```

View vente

```
drop MATERIALIZED view v_vente;
CREATE MATERIALIZED VIEW v_vente refresh force with primary key AS
select ligne_facture.produit as produit_id,
facture.client as client_id,
v_date.num as date_id,
prix_date.remise as remise,
prix_date.prix as prix,
ligne_facture.qte as qte
from facture, ligne_facture, v_date, prix_date
where facture.num=ligne_facture.facture
and ligne_facture.id_prix=prix_date.num
and to_date(to_char(v_date.annee,'9999')||to_char(v_date.mois,'99')||
to_char(v_date.jour,'99'),'yyyymmdd') = date_etabli;
```

```
drop MATERIALIZED view v_compl;
CREATE MATERIALIZED VIEW v_compl refresh force with primary key AS
select  PRODUIT.NUM as num_produit,
v_vente.PRODUIT_ID as num_produit_compl
from PRODUIT,v_vente
where v_vente.CLIENT_ID in ( select CLIENT_ID from v_vente
                                          where PRODUIT_ID=PRODUIT.NUM);
```

La vue des produits complémentaires est basé sur le principe du : « D'autre utilisateurs qui ont acheté le même produit, on aussi acheté ces produits ».

Pour recharger les nouvelles lignes sans recharger toute la table, il faut faire un refresh forcé pour chaque table.
Cela doit être fait la nuit pour éviter de surcharger le système.

```
CREATE MATERIALIZED VIEW LOG ON nom_vue;
```

## Question 2

```
alter MATERIALIZED VIEW v_produit modify (num primary key);
alter MATERIALIZED VIEW v_client modify (num primary key);
ALTER MATERIALIZED VIEW v_vente  ADD PRIMARY KEY(PRODUIT_ID, CLIENT_ID,
DATE_ID);
alter MATERIALIZED VIEW v_date modify (num primary key);
```

## Question 3

```
INSERT INTO CLIENT VALUES(client_seq.nextval,'PETIT','JEaAN','144 rue qui
danse,59650,Villeneuve d-asc,France',
TO_DATE(TRUNC(DBMS_RANDOM.VALUE(2433283,2433283+40*364)),'J')   ,'homme');
```

```
exec DBMS_SNAPSHOT.REFRESH( 'v_client');
```

Et je retrouve mon JEAN PETIT Qui danse !

## Question 4

```
create index idx_sex on v_client(sexe);
create index idx_cat on v_produit(category);
create index idx_an on v_date(annee);
```

## Question 5

```
CREATE DIMENSION produit_dim
LEVEL produit IS (v_produit.num)
LEVEL nom IS (v_produit.nom)
LEVEL souscategorie IS (v_produit.souscategorie)
LEVEL categorie IS (v_produit.categorie)
HIERARCHY prod_rollup (
Produit CHILD OF nom CHILD OF souscategorie CHILD OF categorie) ;
```

```
CREATE DIMENSION date_dim
LEVEL date_v IS (v_date.num)
LEVEL jour IS (v_date.jour)
LEVEL mois IS (v_date.mois)
LEVEL annee IS (v_date.annee)
HIERARCHY date_rollup (
date_v CHILD OF jour CHILD OF mois CHILD OF annee);
```

```
CREATE DIMENSION client_dim
LEVEL client IS (v_client.num)
LEVEL nom IS (v_client.nom)
LEVEL prenom IS (v_client.prenom)
LEVEL age IS (v_client.age)
LEVEL groupe_age IS (v_client.groupe_age)
LEVEL sexe IS (v_client.sexe)
LEVEL pays IS (v_client.pays)
LEVEL code_postal IS (v_client.code_postal)
LEVEL ville IS (v_client.ville)
HIERARCHY client_rollup (
client CHILD OF ville CHILD OF code_postal CHILD OF pays);
```

Questions d'implementation

1.

```
select nom, sum(qte*prix) as ca
from v_produit, v_vente
where v_produit.num = v_vente.produit_id
group by num,nom;
```

2.

```
SELECT categorie, mois, sum(qte*prix) as ca
from v_produit, v_date, v_vente
where v_produit.NUM=v_vente.PRODUIT_ID
and v_vente.DATE_ID=v_date.NUM
group by rollup(categorie,mois);
```

3.

```
SELECT groupe_age, sum(qte*prix) as ca, rank() OVER (ORDER BY  sum(qte*prix)  DESC) as rank
from v_client, v_vente
where v_client.NUM = v_vente.CLIENT_ID
group by v_client.GROUPE_AGE
order by ca desc;
```

4.

```
SELECT * FROM
(select nom, sum(qte) as quanti, ROW_NUMBER() OVER (ORDER BY  sum(qte) desc) AS rownumber
from v_produit, v_vente
where v_produit.num = v_vente.produit_id
group by num,nom
order by quanti desc)
WHERE rownumber <= 3;
```