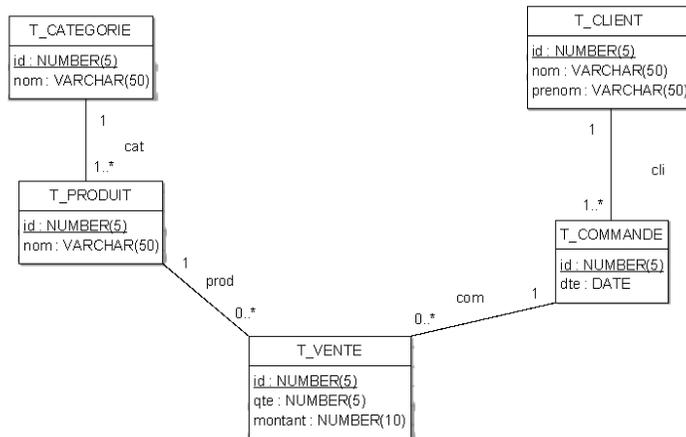


Soit l'entrepôt de données dont le schéma conceptuel est représenté ci-dessous.



1. Créer les types Oracle correspondant à ce schéma conceptuel. Représenter les associations par des références (ex. dans T_PRODUIIT, cat est une référence vers un objet de type T_CATEGORIE).

2. Créer les tables objets indiquées dans le tableau ci-dessous.

Table objet	Type associé
D_CATEGORIE	T_CATEGORIE
D_PRODUIIT	T_PRODUIIT
D_CLIENT	T_CLIENT
D_COMMANDE	T_COMMANDE
F_VENTE	T_VENTE

3. Alimenter la table D_CATEGORIE à l'aide d'un bloc PL/SQL qui associe un identifiant numérique à chaque catégorie distincte de la table DARMONT.DEMO_PRODUCT_INFO. Consulter la table D_CATEGORIE.

4. Alimenter la table D_PRODUIIT à l'aide d'une requête SQL qui sélectionne les attributs PRODUCT_ID et PRODUCT_NAME de la table DARMONT.DEMO_PRODUCT_INFO et la référence de la catégorie CATEGORY correspondante dans la table D_CATEGORIE. Consulter la table D_PRODUIIT.

5. Alimenter la table D_CLIENT à l'aide d'une requête SQL qui sélectionne les attributs CUSTOMER_ID, CUST_LAST_NAME et CUST_FIRST_NAME de la table DARMONT.DEMO_CUSTOMERS. Consulter la table D_CLIENT.

6. Alimenter la table D_COMMANDE à l'aide d'une requête SQL qui sélectionne les attributs ORDER_ID et ORDER_TIMESTAMP de la table DARMONT.DEMO_ORDERS et la référence du client CUSTOMER_ID correspondant dans la table D_CLIENT. Consulter la table D_COMMANDE.

7. Alimenter la table F_VENTE à l'aide d'une requête SQL qui sélectionne les attributs ORDER_ITEM_ID, QUANTITY et QUANTITY * LIST_PRICE (champ calculé) des tables DARMONT.DEMO_ORDER_ITEMS et DARMONT.DEMO_PRODUCT_INFO (jointure) et les références de la commande ORDER_ID correspondante dans la table D_COMMANDE et du produit PRODUCT_ID correspondant dans la table D_PRODUIIT. Consulter la table F_VENTE.

8. À partir de la table F_VENTE, calculer le cube permettant d'obtenir la somme des quantités et des montants par produit (indiquer le nom du produit) à l'aide d'une requête SQL utilisant la clause GROUP BY ROLLUP. Exploiter la jointure implicite permise par les références d'objets et utiliser les fonctions GROUPING et DECODE pour identifier les agrégats ALL.

9. Même question que la précédente, mais uniquement pour les produits de catégorie « Computer ».

10. Activer l'affichage du plan d'exécution des requêtes.

11. À partir de la table objet F_VENTE, calculer le cube permettant d'obtenir la somme des quantités et des montants par catégorie de produit (indiquer le nom de la catégorie) et par client (indiquer le nom du client) à l'aide d'une requête SQL utilisant la clause GROUP BY ROLLUP. Exploiter la jointure implicite permise par les références d'objets et utiliser les fonctions GROUPING et DECODE pour identifier les agrégats ALL.

12. Même question que la précédente, mais à partir des tables sources relationnelles DARMONT.DEMO_ORDER_ITEMS, DARMONT.DEMO_ORDERS, DARMONT.DEMO_CUSTOMERS et DARMONT.DEMO_PRODUCT_INFO.

13. Conclusion (facilité d'écriture de la requête, efficacité) ?

Correction

```
-- Schéma de classes

create type t_categorie as object(
  id number(5),
  nom varchar(50)
)
/

create type t_produit as object(
  id number(5),
  nom varchar(50),
  cat ref t_categorie
)
/

create type t_client as object(
  id number(5),
  nom varchar(50),
  prenom varchar(50)
)
/

create type t_commande as object(
  id number(5),
  dte date,
  cli ref t_client
)
/

create type t_vente as object(
  id number(5),
  qte number(5),
  montant number(10),
  com ref t_commande,
  prod ref t_produit
)
/

-- Tables objets

create table d_categorie of t_categorie (
  constraint cat_pk primary key(id)
);
create table d_produit of t_produit (
  constraint prod_pk primary key(id),
  constraint cat_ref cat references d_categorie
);
create table d_client of t_client (
  constraint cli_pk primary key(id)
);
create table d_commande of t_commande (
  constraint com_pk primary key(id),
  constraint cli_ref cli references d_client
);
create table f_vente of t_vente (
  constraint vente_pk primary key(id),
  constraint com_ref com references d_commande,
  constraint prod_ref prod references d_produit
);
```

```
-- ETL

declare
  cursor les_cat is select distinct category from darmont.demo_product_info
  order by category;
  c les_cat%ROWTYPE;
  id integer;
begin
  for c in les_cat loop
    id := les_cat%ROWCOUNT;
    insert into d_categorie values(id, c.category);
  end loop;
end;
/
select * from d_categorie;

insert into d_produit
  select product_id, product_name,
  (select ref(c) from d_categorie c where c.nom = category)
  from darmont.demo_product_info;
select * from d_produit;

insert into d_client
  select customer_id, cust_last_name, cust_first_name
  from darmont.demo_customers;
select * from d_client;

insert into d_commande
  select order_id, order_timestamp,
  (select ref(c) from d_client c where c.id = customer_id)
  from darmont.demo_orders;
select * from d_commande;

insert into f_vente
  select order_item_id, quantity, quantity * list_price,
  (select ref(c) from d_commande c where c.id = order_id),
  (select ref(p) from d_produit p where p.id = oi.product_id)
  from darmont.demo_order_items oi, darmont.demo_product_info pi
  where oi.product_id = pi.product_id;
select * from f_vente;

commit;

-- Requêtes décisionnelles par jointure implicite

select
  decode(grouping(v.prod.nom), 1, 'ALL', v.prod.nom) produit,
  sum(qte), sum(montant)
from f_vente v
group by rollup(v.prod.nom);

select
  decode(grouping(v.prod.nom), 1, 'ALL', v.prod.nom) produit,
  sum(qte), sum(montant)
from f_vente v
where v.prod.cat.nom = 'Computer'
group by rollup(v.prod.nom);

-- Comparaison jointure implicite sur table objet vs. Jointure
-- sur tables relationnelles

set autotrace on explain
```

```
select
  decode(grouping(v.prod.cat.nom), 1, 'ALL', v.prod.cat.nom) categorie,
  decode(grouping(v.com.cli.nom), 1, 'ALL', v.com.cli.nom) client,
  sum(qte), sum(montant)
from f_vente v
group by rollup(v.prod.cat.nom, v.com.cli.nom);
```

```
select
  decode(grouping(category), 1, 'ALL', category) categorie,
  decode(grouping(cust_last_name), 1, 'ALL', cust_last_name) client,
  sum(quantity) qte, sum(quantity*list_price) montant
from darmont.demo_order_items doi, darmont.demo_orders do,
  darmont.demo_customers dc, darmont.demo_product_info dpi
where doi.order_id = do.order_id
and do.customer_id = dc.customer_id
and doi.product_id = dpi.product_id
group by rollup(category, cust_last_name);
```