

Examen du 1^{er} semestre

Question de cours (04 pts)

Répondre par vrai ou faux et corriger les phrases incorrectes en gardant la première partie de la phrase.

- 1- On appelle Le passage de l'objet vers la Classe une instanciation.
- 2- La semi-jointure est une jointure suivie par une projection sur les colonnes de la première relation.
- 3- La projection est une fragmentation horizontale de la relation.
- 4- Les éléments d'un objet de type VARRAY peuvent être manipulés avec la requête SELECT.
- 5- L'estampillage est une approche optimiste.
- 6- Le « D » de l'acronyme « ACID » désigne la **Disponibilité**.

Exercice 1: (05 pts)

1. Dessiner le graphe de précédence, est-il sérialisable? Si oui quels sont les ordonnancements série équivalents.

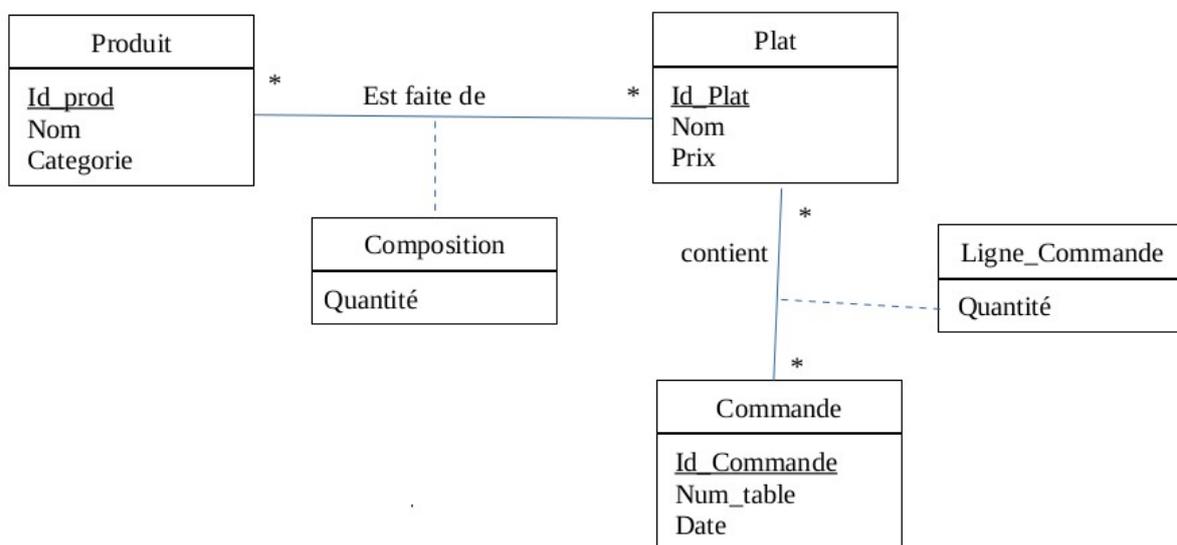
$r1(x), r3(y), w1(y), w4(x), w1(t), w5(x), r2(z), r3(z), w2(z), w5(z), r4(t), r5(t).$

2. Dérouler l'ordonnancement suivant avec le protocole 2PL:

$r1(x), w1(x), w3(x), r2(y), r3(y), w3(y), w1(y), r2(x).$

Exercice 2: (02+ 2,5 + 1,5 = 06 pts)

Soit le diagramme de classe suivant:



1. Donner le schéma d'une BD **relationnelle standard** qui correspond au diagramme ci-dessus.
2. On veut construire une base de données **objet-relationnel** à partir du diagramme ci-dessus. La base de données ne contient que trois tables suivantes: Produit, Plats, et Commandes.
 - ⊗ Donner en SQL3 les requêtes de création des types nécessaires pour cette conception.
 - ⊗ Répondre à la requête suivante en utilisant le langage SQL 3: Afficher les noms des plats qui contiennent le "fromage rouge".

Exercice 3 : (1,5+ 0,5 + 1,5 + 1,5 = 05 pts)

Soit le schéma de base de données relationnel suivant :

AGENCE (NumAg, NomAg, VilleAg, Actif)

CLIENT (NumCl, NomCl, VilleCl)

COMPTE (NumCpt, #NumAg, #NumCl, Solde)

1-Ecrire en Expression **algébrique** la requête suivante puis **optimiser** la requête :

Les noms et les soldes des clients de "Biskra" qui ont un compte à "El Oued" et leurs soldes est inférieur à 10000.

2- Ecrire les requêtes suivantes en SQL :

- ⊗ Verser 2000 aux clients qui n'ont pas de comptes dans leurs villes.
- ⊗ Afficher le nom du client qui a le solde le **plus grand**.

Corrigée type de l'examen
(Bases des données Avancées 1^{ère} année master informatique)

Question de cours (04 pts)

Répondre par vrai ou faux et corriger les phrases incorrectes en gardant la première partie de la phrase.

1- Faux. On appelle Le passage de l'objet vers la Classe une Abstraction.

2- Vrai.

3- Faux. La projection est une fragmentation verticale de la relation.

4- Faux. Les éléments d'un objet de type VARRAY peuvent être manipulés avec un bloc PL/SQL.

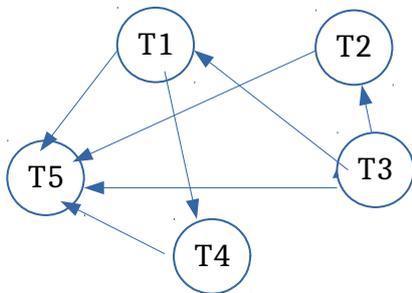
5- Vrai.

6- Faux. Le « D » de l'acronyme « ACID » désigne la **Durabilité**.

Exercice 1: (05 pts)

1. Dessiner le graphe de précedence, est-il sérialisable? Si oui quels sont les ordonnancements série équivalents.

$r1(x), r3(y), w1(y), w4(x), w1(t), w5(x), r2(z), r3(z), w2(z), w5(z), r4(t), r5(t).$



Graphe sans Cycle donc l'ordonnancement est sérialisable.

T3,T2,T1,T4,T5.

2. Dérouler l'ordonnancement suivant avec le protocole 2PL:

RL1(x),r1(x), WL1(x),w1(x), WL3(x) (Mise en attente de T3), RL2(y), R2(y), WL1(y) (Mise en attente de T1) RL2(x) (Mise en attente de T2) Interblocage entre T1 et T2. Il faut une transaction T1 ou T2.

Exercice 2: (06 pts)

- 1- Donner le schéma relationnel qui correspond au diagramme ci-dessus.

Produit (id_prod, nom, catégorie);

Plat (id_plat, nom, prix);

Commande (id_com, num_tab, datec);

Composition (id_plat, id_prod, quantité);

Lignecommande (id_plat, id_com, quantité)

2- En passant au modèle objet-relationnel, on veut construire une base de données de trois tables suivantes: Produits, Plats, et Commandes.

⊛ Donner en SQL3 les requêtes de création des types nécessaires pour cette conception.
 create type t_Produit as object(id_prod int, nom varchar(20), categorie varchar(20));
 create type t_composition as object(prod ref t_prod, quantite number(8,2));
 create type tabcompos as table of t_composition;
 create type t_Plat as object(id_plat int, nom varchar(20), prix number(10,2), compo tabcompos);
 create type t_lignecom as object(plat ref t_Plat, quantite number(8,2));
 create type lignecommandes as table of t_lignecom;
 create type t_commande as object(idcom int, num_tab int, datec date, lignes lignecommandes);

⊛ La requête : Afficher les noms des plats qui contiennent le "Fromage rouge".

```

Select p.nom
From Plat p, Table (P.compo) cmp
where cmp.prod.nom = 'Fromage rouge'
  
```

Exercice 03:

1- Ecrire en Expression **algébrique** la requête suivante puis **optimiser** la requête :

Les noms et les soldes des clients de "Biskra" qui ont un compte à "El Oued" et leurs soldes est inférieur à 10000.

$\Pi_{\text{nomcl,solde}}(\sigma_{\text{villecl}='Biskra' \text{ and } \text{villeag}='El Oued' \text{ and } \text{solde}<10000}(\text{Client} \bowtie \text{Agence} \bowtie \text{Compte}))$

Optimisation:

$\Pi_{\text{nomcl,solde}}(\Pi_{\text{numcl,nomcl}}(\sigma_{\text{villecl}='Biskra'} \text{Client}) \bowtie \Pi_{\text{Numag}}(\sigma_{\text{villeag}='El Oued'} \text{Agence}) \bowtie \sigma_{\text{solde}<10000} \text{Compte})$

2- Les requêtes SQL:

⊛ Verser 2000 aux clients qui n'ont pas de comptes dans leurs villes.

```
update compte set solde = solde + 2000
```

```
where num_client not in (select num_client from client c, compte cp, agence a
```

```
where c.num_client = cp.num_client and cp.num_agence = a.num_agence and a.ville = c.ville)
```

⊛ Afficher le nom du client qui a le solde le **plus grand**.

```
SELECT C.nom
```

```
FROM Client C, Compte cp
```

```
where c.Num_Client = cp.num_client and cp.solde = (select max(solde) from compte)
```