

BD50

Conception des Bases de Données

Examen Final – 18 juin 2016

Département Informatique

Conditions de réalisation :

Support de cours et travaux pratiques de BD50 autorisés

Durée : **3 heures**

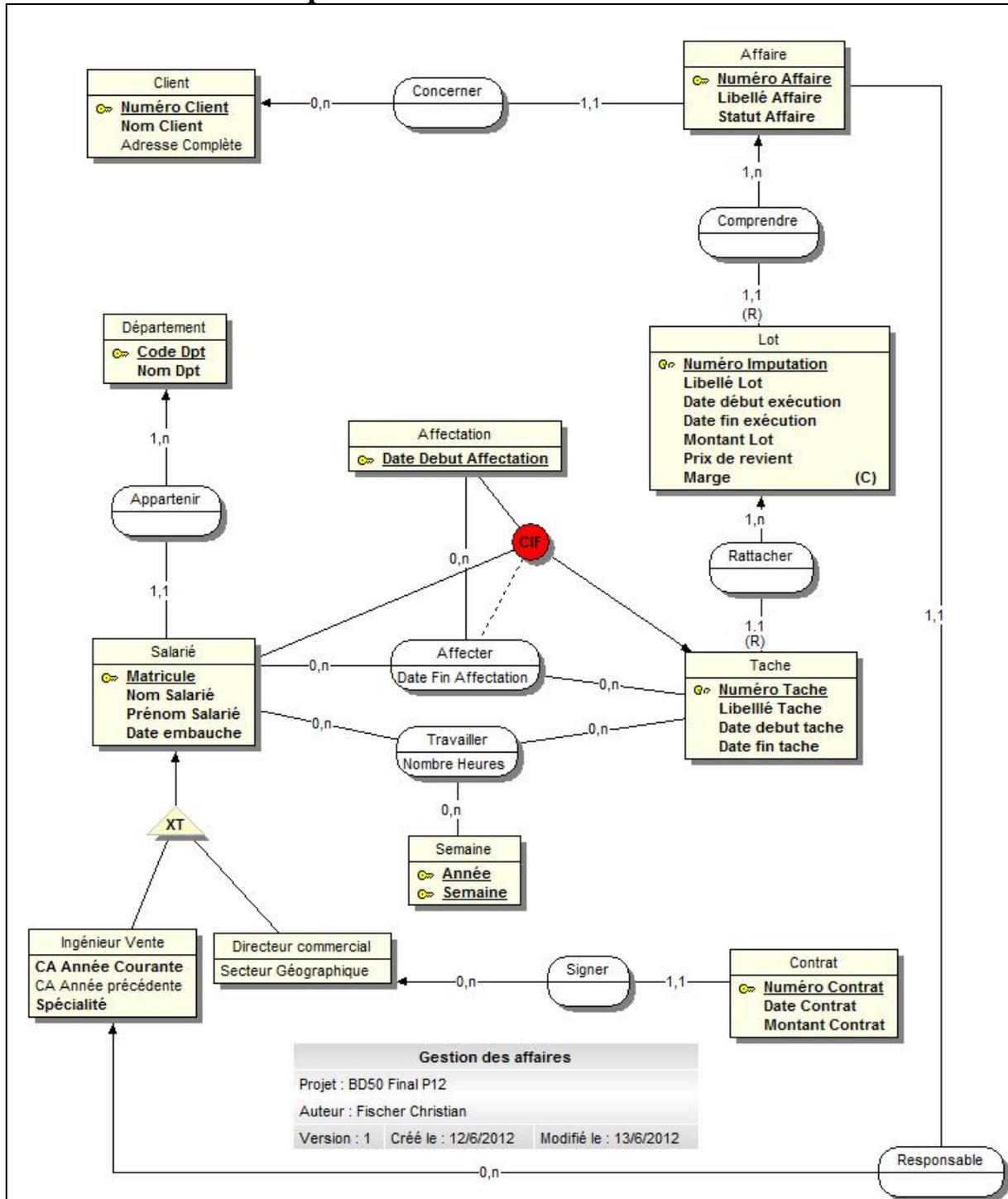
Barème

Gestion des affaires

Optimisation du Modèle Physique de données	5 pts
Optimisation Oracle	4 pts
Rédaction de 2 requêtes SQL	3 pts
Optimisation des requêtes	4 pts
Requête 1	1 pts
Requête 2	3 pts
Programmation PL/SQL	4 pts

Gestion des affaires : optimisations

1. Du MEA au MPD Optimisé



Travail à faire

1. A partir du MEA ci-dessus, fournir la représentation graphique du modèle du modèle physique de données optimisé (en justifiant vos choix).

2. Optimisation du modèle physique de données pour Oracle

Travail à faire

1. A partir du modèle physique de données optimisé indiquez les tables qui seront implantées en IOT (Index Organized Table).
2. Rédiger une instruction SQL de création d'une table organisée en index, de votre choix. Le tablespace de stockage de la clé primaire est nommé : AFF_IND.
3. Rédiger l'instruction de création de la table SALARIE qui sera partitionnée par liste.
Le choix des critères de partition est de votre responsabilité.

3. Requêtes SQL à faire sur votre modèle optimisé

Travail à faire

- La rédaction des requêtes doit être conforme aux règles de présentation mises en œuvre en TP.
- L'écriture des jointures doit être effectuée en **formulation ANSI avec la syntaxe : Inner join et Outer join avec la clause ON.**
- L'utilisation d'**alias de table** est **obligatoire.**

1. Écrire la requête SQL permettant de calculer le nombre d'heures total travaillés pour tous les salariés sur les 3 dernières années.

Afficher le matricule, le nom et le prénom du salarié, l'année et le nombre d'heures.

Pour chaque salarié : trois années doivent être affichées.

Le résultat sera trié par matricule et année.

Quel opérateur devez-vous utiliser pour afficher qu'une ligne par salarié et 3 colonnes avec le nombre d'heures par chaque année ?

Exemple de résultat attendu

Matricule	Nom	Prénom	2016	2015	2014
1	Dupont 1	Alain 1	1859	1839	1845
2	Dupont 2	Alain 2	1856	1834	
3	Dupont 3	Alain 3	1852	1839	1845
4	Dupont 4	Alain 4	1851		
5	Dupont 5	Alain 5	1855	1839	1845
etc ...					

Proposer une solution pour faire un calcul mensuel du nombre d'heures par salarié pour l'année en cours.

2. Ecrire la requête SQL permettant de calculer le nombre de jours d'affectation mensuel pour l'affaire numéro 1750123 pour les salariés affecté à une tâche de cette affaire sur l'année en cours.

Afficher le matricule et nom, le nom du mois et le nombre de jours d'affectation.

Le résultat sera trié par matricule et mois.

4. Optimisation des requêtes SQL

Requête 1 :

Liste des ingénieurs de vente (type salarié = IV).

```
select
    SAL.*
from
    salarie SAL
where
    type_salarie='IV'
;
```

Plan d'exécution associé :

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	155	2 (0)	00:00:01
* 1	TABLE ACCESS FULL	SALARIE	1	155	2 (0)	00:00:01

Predicate Information (identified by operation id):

1 - filter("TYPE_SALARIE"='IV')

Travail à faire

1. A partir du plan ci-dessus, analyser la méthode d'accès pour la table salarié.

Ce plan est-il optimal ?

Proposer une solution pour optimiser cette requête.

Requête 2 : pour calculer le montant d'une affaire vous disposez de la requête suivante :

```

select
    sum(LT.montant_lot)
from
    LOT LT inner join AFFAIRE AF
    on LT.NUM_AFFAIRE=AF.NUM_AFFAIRE
where
    AF.LIB_AFFAIRE = 'Optimisation Oracle UTBM' ;

```

Plan d'exécution associé :

```

SUM(LT.MONTANT_LOT)
-----
          11250

Plan d'exécution
-----
Plan hash value: 3795151851

-----
| Id | Operation                                | Name           | Rows  | Bytes | Cost (%CPU)| Time     |
-----|-----|-----|-----|-----|-----|-----|
|  0 | SELECT STATEMENT                          |                |      1 |    44 |       7 (0)| 00:00:01 |
|  1 |   SORT AGGREGATE                          |                |      1 |    44 |              |          |
|  2 |    NESTED LOOPS                            |                |      9 |   396 |       7 (0)| 00:00:01 |
| * 3 |     NESTED LOOPS                          |                |      9 |   396 |       7 (0)| 00:00:01 |
| * 4 |      TABLE ACCESS FULL                    | AFFAIRE        |      1 |    36 |       5 (0)| 00:00:01 |
| * 5 |       INDEX RANGE SCAN                     | I_FK_LOT_AFFAIRE |      9 |    90 |       1 (0)| 00:00:01 |
|  6 |        TABLE ACCESS BY INDEX ROWID      | LOT             |      9 |    72 |       2 (0)| 00:00:01 |
-----

Predicate Information (identified by operation id):
-----

   4 - filter("AF"."LIB_AFFAIRE"='Optimisation Oracle UTBM')
   5 - access("LT"."NUM_AFFAIRE"="AF"."NUM_AFFAIRE")

```

Travail à faire

1. A partir du plan ci-dessus, analyser les méthodes d'accès pour chaque table ainsi que l'algorithme de jointure.

Ce plan est-il optimal ?

Proposer une solution pour optimiser cette requête.

2. L'administrateur de base de données envisage d'optimiser cette requête en ajoutant une colonne montant_affaire dans la table affaire.

Rédiger les instructions SQL pour réaliser les actions suivantes :

2.1. Ajouter la colonne MONTANT_AFF de type numérique (13,2) dans la table AFFAIRE.

2.2. Rédiger l'instruction de mise à jour du montant des affaires existantes à partir du montant des lots de chaque affaire.

3. Pour les nouvelles insertions dans la table LOT, rédiger le trigger permettant de mettre à jour le montant de l'affaire concerné.

Ce trigger doit également gérer l'attribution du numéro d'imputation de chaque affaire.

5. Programmation PL/SQL

Dans le cadre de l'optimisation vous devez rédiger la procédure PL/SQL packagée permettant d'insérer 10000 lignes dans la table Affaire et 10 lots pour chaque affaire.

Le libellé de l'affaire sera générique : Optimisation Oracle affaire 1 à 10000

Le libellé du lot sera générique : Optimisation Oracle lot : 1 à 10

Le nom du package est : PA_AFFAIRE

L'exécution de la procédure sera effectuée par :

```
SQL> execute PA_AFFAIRE.ins_aff_lot(10000) ;
```

Travail à faire

Rédiger le **corps du package de gestion de la table PA_AFFAIRE** avec les procédures ou Fonctions :

- Ins_Aff_Lot,
- GetByPK,
- GetALL,
- Upd,
- Add,
- Remove