

BD50

Conception des Bases de Données

Examen Final – 23 juin 2018

Département Informatique

Conditions de réalisation :

Support de cours et travaux pratiques de BD50 autorisés

Durée : **3 heures**

Barème

		Barème	Durée Conseillée
Dossier 1	Gestion des ventes sur offre <ul style="list-style-type: none">• Optimisation du MPD : 5 pts• Optimisation Oracle : 5 pts	10 pts	90 minutes
Dossier 2	Gestion des ventes sur offre <ul style="list-style-type: none">• SQL : 4 pts• PL/SQL : 4 pts	8 pts	70 minutes
Dossier 3	Client-Serveur (à compléter et à rendre avec votre copie)	2 pts	20 minutes

Dossier 1 : Gestion des ventes sur offre

Présentation du contexte :

Une maison philatélique fondée en 1902, Roulet SA, achète et vend des timbres et des cartes postales.

Tous les ans la société effectue entre 5 et 8 ventes sur offres.

Pour chaque vente, un catalogue établi à partir d'un ensemble de timbres achetés est adressé à plus de 5000 clients.

Entre la date de début et la date de clôture de la vente, chaque client renvoie à la société la liste des références des lots qu'il désire acquérir, reporté sur un ordre d'achat, ainsi que le montant qu'il propose. La vente s'effectue au plus offrant.

Le prix offert doit être obligatoirement supérieur ou égal au prix plancher.

En fin de période, il faut traiter les ordres d'achat de tous les lots, et donc effectuer la vente proprement dite en sélectionnant les ordres les mieux disant.

On édite pour chaque client une facture, spécifiant les lots qu'il a réussi à acquérir lors de cette vente.

La facture indiquera un montant total hors taxes, un montant de TVA (à 20%) et un montant TTC.

Par convention, le timbre est vendu, non pas au prix proposé par le client le mieux offrant, mais au second prix proposé augmenté de 5% (la vente est dite à « l'américaine », ou au prix minimum si l'acheteur est le seul candidat.

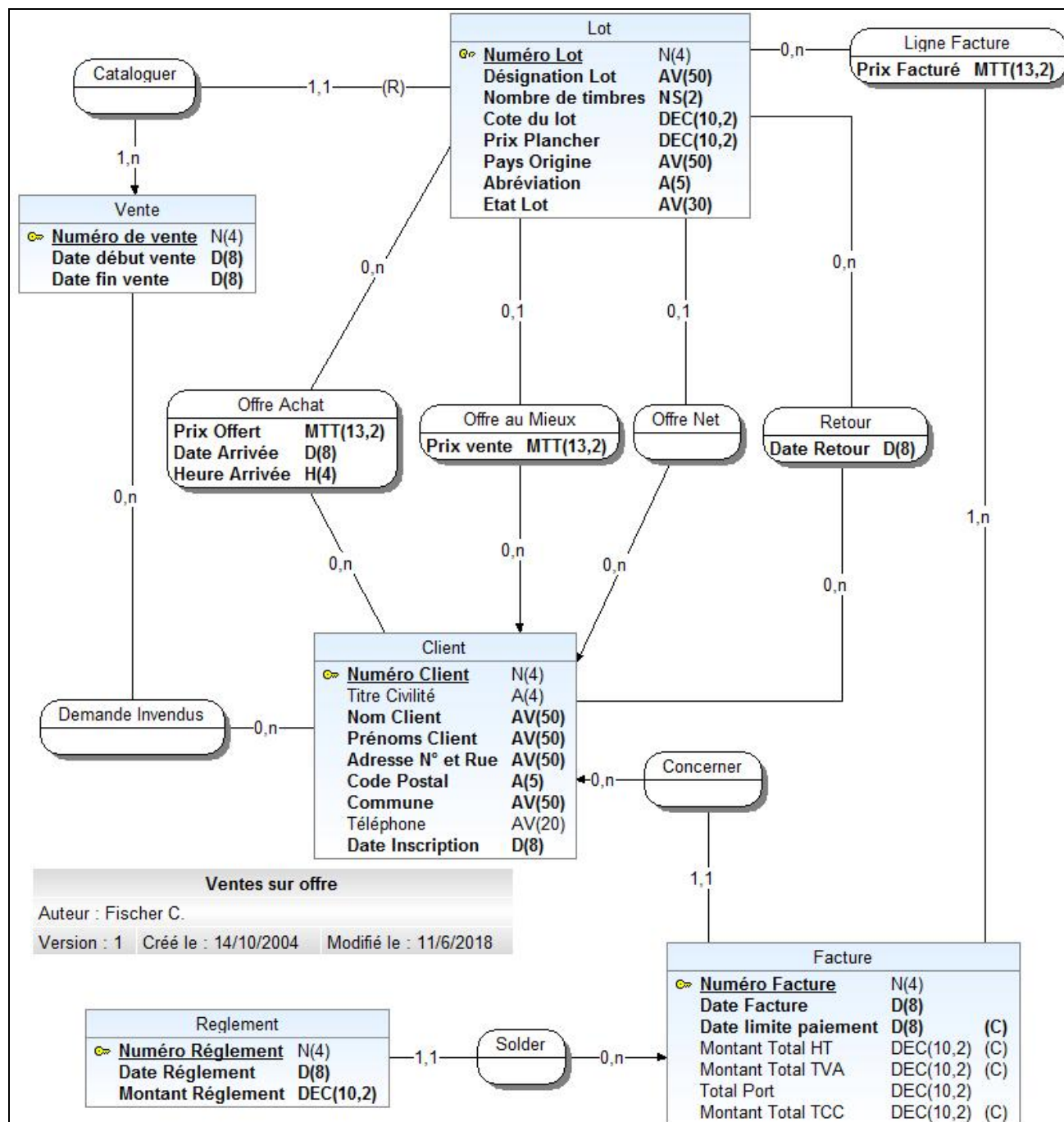
Les clients règlent dans un délai de 60 jours après facturation et disposent de 48 heures pour retourner un lot non désiré.

Après clôture de la vente, les lots invendus sont reportés sur une liste d'invendus, adressée aux clients qui, le désirent. Ceux-ci peuvent les acquérir au prix plancher.

La vente se fait à prix net (premier arrivé, premier servi).

La facture est établie dès l'arrivée de la première offre.

Le modèle entité –association étendu est présenté ci-dessous :



Travail à faire

- À partir du MEA ci-dessus, fournir la représentation graphique du modèle relationnel normalisé puis du modèle physique de données optimisé (en justifiant vos choix).

2. Optimisation du modèle physique de données pour Oracle

Travail à faire

1. À 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 : OFFRE_IND.

3. Rédiger l'instruction de création de la table VENTE partitionnée par année sur la date de début de vente.

La première partition contient les données avant le 1/1/2015, les autres partitions contiennent toutes données d'une année, la dernière partition contient toutes les données après le 1/1/2020.

La table des LOTS devra également être partitionnée.

Le choix de la méthode vous appartient.

Rédiger l'instruction de création de la table LOTS.

Dossier 2 : Gestion des ventes sur offre (SQL et PL/SQL)

1. Requête 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 chiffre d'affaires par client pour toutes les ventes sur les 3 dernières années (2017, 2016 et 2015).

Afficher le numéro et le nom du client, l'année de ventes et le chiffre d'affaires.

Le résultat sera trié par numéro de client.

L'extraction de l'année est effectuée avec la fonction to_char()

Exemple de résultat attendu

Numéro Client	Nom client	Année	CA
1	ACAR	2017	200,00 €
1	ACAR	2016	150,00 €
1	ACAR	2015	120,00 €
2	AFIR	2017	200,00 €
2	AFIR	2016	100,00 €
3	ALAO	2017	400,00 €
4	ARI GONIMI	2016	352,50 €
4	ARI GONIMI	2015	120,00 €
5	BAGOUDOU	2017	80,00 €
5	BAGOUDOU	2016	150,00 €
5	BAGOUDOU	2015	54,00 €
6	BAO	2015	100,00 €
...			

2. À partir de la requête précédente on désire obtenir le résultat suivant :

Numéro Client	Nom client	2017	2016	2015
1	ACAR	200,00 €	150,00 €	120,00 €
2	AFIR	200,00 €	100,00 €	- €
3	ALAO	400,00 €	- €	- €
4	ARI GONIMI	- €	352,50 €	120,00 €
5	BAGOUDOU	80,00 €	150,00 €	54,00 €
6	BAO	- €	- €	100,00 €
7	BARETH-DESGRANGES	100,00 €	100,00 €	100,00 €
8	BELCAID	600,00 €	200,00 €	- €
9	BENZAGMOUT	200,00 €	150,00 €	200,00 €
10	BOUAALAM	200,00 €	180,00 €	190,00 €

Écrire la requête SQL permettant d'obtenir une seule ligne par client avec le CA sur les 3 dernières années en colonnes.

2. Optimisation des requêtes SQL

Requête 1 : Calcul du montant des règlements par mois.

```
select
  to_char(reg.datreg, 'YYYY-MM') as periode
, sum(reg.mtreg)                as montant
from
  reglement REG
where
  reg.datreg between '1/1/2018' and sysdate
group by
  to_char(reg.datreg, 'YYYY-MM')
order by
  to_char(reg.datreg, 'YYYY-MM')
;
```

Plan d'exécution associé :

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	22	4 (50)	00:00:01
1	SORT ORDER BY		1	22	4 (50)	00:00:01
2	HASH GROUP BY		1	22	4 (50)	00:00:01
* 3	FILTER					
* 4	TABLE ACCESS FULL	REGLEMENT	1	22	2 (0)	00:00:01

Predicate Information (identified by operation id):

```
3 - filter(TO_DATE(' 2018-01-01 00:00:00', 'syyy-mm-dd
hh24:mi:ss')<=SYSDATE@!)
4 - filter("REG"."DATREG">=TO_DATE(' 2018-01-01 00:00:00',
'syyy-mm-dd hh24:mi:ss') AND "REG"."DATREG"<=SYSDATE@!)
```

Travail à faire

2. À partir du plan ci-dessus, analyser les méthodes d'accès pour la table REGLEMENT

Ce plan est-il optimal ?

Proposer une solution pour optimiser cette requête.

Requête 2 : Liste des offres d'achat pour la vente en cours.

```
select
    OFA.NUMVENTE
, OFA.NUMLOT
, OFA.PRIXOFFERT
, OFA.NUMCLIENT
from
    VENTE VEN INNER JOIN OFFRE_ACHAT OFA
    on VEN.NUMVENTE=OFA.NUMVENTE
where
    sysdate between VEN.DTDEBVENTE and DTFINVENTE
order by
    OFA.NUMLOT      asc
, OFA.PRIXOFFERT  desc
;
```

Plan d'exécution associé :

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	83	4 (25)	00:00:01
1	SORT ORDER BY		1	83	4 (25)	00:00:01
2	NESTED LOOPS					
3	NESTED LOOPS		1	83	3 (0)	00:00:01
* 4	TABLE ACCESS FULL	VENTE	1	31	2 (0)	00:00:01
* 5	INDEX RANGE SCAN	I_FK_OFFRE_ACHAT_LOT	1		1 (0)	00:00:01
6	TABLE ACCESS BY INDEX ROWID	OFFRE_ACHAT	1	52	1 (0)	00:00:01

Predicate Information (identified by operation id):

- 4 - filter("VEN"."DTDEBVENTE"<=SYSDATE@! AND "VEN"."DTFINVENTE">=SYSDATE@!)
- 5 - access("VEN"."NUMVENTE"="OFA"."NUMVENTE")

Travail à faire

1. A partir du plan ci-dessus, analyser la méthode d'accès pour la table VENTE. Ce plan est-il optimal ?
Proposer une solution pour optimiser cette requête.

Comment déterminer la dernière vente clôturée lorsque la date du jour est supérieure à la date de clôture ?

3. Programmation PL/SQL

Travail à faire

1. Rédiger la **procédure ou fonction** PL/SQL **SET_LOT** permettant d'attribuer un lot d'une vente à la meilleure offre et de calculer le montant à payer pour le lot par le client.

Paramètres en entrée : Numéro de vente et numéro de lot

Paramètre de retour : Prix_A_Payer

2. Rédiger le **corps du package de gestion de la table PA_CLIENT** avec les procédures ou fonctions :

- Add : pour créer un nouveau client
- GetByPK : pour afficher un client à partir de son nom (PK)
- GetALLByCom : pour afficher tous les clients d'une commune
- Upd : pour mettre à jour toutes les informations client
- Remove : pour supprimer un client

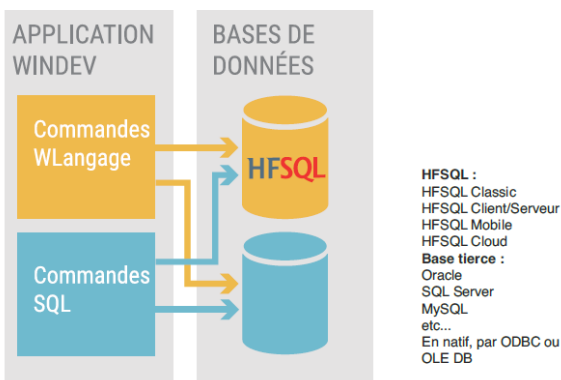
Dossier 3 : CLIENT-SERVEUR

La société de services LOGI_ETUDE a développé des applications client-serveur pour le compte des mairies.

Les logiciels développés fonctionnent avec des postes de travail, sous Windows 7 et 10, reliés au réseau local qui dispose d'un serveur Windows 2012 Server utilisé comme serveur de fichiers et d'impression.

L'application client-serveur de gestion de l'état civil a été développée avec **Windev 23**.

LA PROGRAMMATION EST IDENTIQUE QUELLE QUE SOIT LA BASE DE DONNÉES



Une base de données Oracle 11G ou 12C (toutes les éditions sont supportées), sous Linux (distribution REDHAT) ou sur Windows Server 2012 (mini), le choix dépend du site client. La base de données au format HFSQL (HyperFile SQL) n'est pas utilisée par la société LOGI_ETUDE.

Certaines requêtes sont stockées dans des procédures PL/SQL Oracle de manière à être exécutées plus rapidement sans transiter par le réseau et sans nécessiter de traduction ou de compilation. Seules les requêtes qui changent en fonction des besoins de l'utilisateur sont construites dynamiquement en utilisant les fonctionnalités Native Dynamique Sql de PL/SQL.

Une application Intranet de demande des actes d'état civil a été développée en PHP et MYSQL version 8 par un stagiaire au service informatique. Cette application est accessible aux administrés de la commune.

Toutes les nuits les données de la base MYSQL sont remontées dans la base Oracle à l'aide d'une procédure externe.

L'application de gestion de la police municipale a été développée en ASP .Net avec une base SQL Server 2016. Elle est uniquement accessible aux membres de la police municipale.

Le Directeur des services informatiques a décidé une migration de toutes les bases de données vers un nouveau serveur utilisant le serveur Oracle. Les applications Windows seront toutes accessibles via un serveur de présentation de type Metaframe de Citrix.

Le protocole réseau est TCP/IP sur tous les postes. Un serveur DHCP permet d'utiliser un adressage dynamique des postes.

A1. Pour mettre en œuvre l'application de gestion de l'état civil, quels éléments parmi les suivants sont indispensables.

- Le SGBDR SQL Server 2016
- Le SGBDR MYSQL 8
- Le SGBDR Oracle 11G
- Le SGBDR Oracle 12C
- Le logiciel Windev 23
- L'application développée avec Windev 23
- L'application développée avec PHP
- Le logiciel d'interrogation de base de données Oracle SQL*Plus
- Le logiciel d'interrogation de base de données MySQL Workbench
- Le logiciel d'interrogation de base de données SQL Server Management Studio
- Le système d'exploitation Linux Mandrake
- Le système d'exploitation Linux Redhat
- Le système d'exploitation Windows 2008 Server
- Le système d'exploitation Windows 2012 Server
- Le système d'exploitation Windows 2016 Server
- Les tables stables de la base de données Access au format Jet
- Les tables stables de la base de données HFSQL au format Hyperfile
- Le middleware Net Library et SQL Native pour SQL Server
- Le middleware Oracle Net Client et Serveur
- Un pilote ODBC pour SQL Server 2016
- Un pilote ODBC pour Oracle 11G
- Un pilote ODBC pour Oracle 12C
- Un pilote ODBC pour MYSQL 8

A2. Dans le contexte de l'application client-serveur de gestion de l'état-civil, placez chaque composant ci-dessous côté serveur ou côté client selon qu'il est utilisé sur un poste serveur ou sur un poste client. Cochez les deux options quand le composant est utilisé à la fois côté client et côté serveur. *Pour chaque composant, cochez l'une, l'autre, aucune ou les deux options.*

Composant	Client	Serveur
Une table contenant des données peu stables	<input type="checkbox"/>	<input type="checkbox"/>
Le système d'exploitation réseau serveur	<input type="checkbox"/>	<input type="checkbox"/>
Une table contenant des données stables	<input type="checkbox"/>	<input type="checkbox"/>
Le système d'exploitation pour un poste de travail	<input type="checkbox"/>	<input type="checkbox"/>
Le SGBDR en partage de fichier	<input type="checkbox"/>	<input type="checkbox"/>
Le SGBDR Oracle 11G ou 12C	<input type="checkbox"/>	<input type="checkbox"/>
Le SGBDR MYSQL	<input type="checkbox"/>	<input type="checkbox"/>
Le SGBDR SQL Server	<input type="checkbox"/>	<input type="checkbox"/>
Le serveur Metaframe Citrix	<input type="checkbox"/>	<input type="checkbox"/>
Les requêtes stables	<input type="checkbox"/>	<input type="checkbox"/>
Le composant logiciel qui permet de saisir une date en simulant le fonctionnement d'un calendrier	<input type="checkbox"/>	<input type="checkbox"/>
La base de données au format Access (Jet)	<input type="checkbox"/>	<input type="checkbox"/>
L'application Windev	<input type="checkbox"/>	<input type="checkbox"/>
Le protocole TCP/IP	<input type="checkbox"/>	<input type="checkbox"/>
Le middleware Oracle NET	<input type="checkbox"/>	<input type="checkbox"/>
Le pilote ODBC pour Oracle 11G/12C fourni par Oracle	<input type="checkbox"/>	<input type="checkbox"/>
Le pilote ODBC pour Oracle 11G/12C fourni par Microsoft	<input type="checkbox"/>	<input type="checkbox"/>