

# BD50

## Conception des Bases de Données

Examen Final – Juin 2015

Département Informatique

Conditions de réalisation :

Support de cours et travaux pratiques de BD50 autorisés

Durée : **3 heures**

Barème : Le sujet comprend 2 dossiers indépendants.

Dossier 1	Gestion des branchements électrique MPD Optimisé : 4 pts Optimisation Oracle : 3 pts Programmation PL/SQL : 4 pts	10 pts
Dossier 2	Gestion des rendez-vous LDD = 2 pts LMD = 5 pts Optimisation des requêtes : 3 pts	10 pts

## Dossier 1 : Gestion des branchements électrique

### Description du contexte :

L'activité du centre EDF de Douvres est essentiellement axée autour des branchements électriques dans le département du Calvados. Le centre sous-traite une partie de son activité en confiant à des entreprises extérieures la réalisation des branchements chez les clients.

Pour organiser les branchements, le département est découpé en **ZEI** (zones élémentaires d'intervention). Une ZEI correspond à un secteur autour d'une ville principale.

On planifie chaque opération à réaliser en tenant compte d'une durée théorique appelée **poïds**.

Le centre EDF signe chaque contrat avec un seul sous-traitant. Un sous-traitant peut signer plusieurs contrats avec le centre EDF.

On enregistre le code sous-traitant, le nom du sous-traitant et l'adresse complète (numéro, rue, code postal, ville) du sous-traitant.

Un contrat constitue un engagement formel du sous-traitant d'intervenir chez le client EDF pour faire les installations. Chaque contrat est identifié par un numéro de contrat et possède une date de signature.

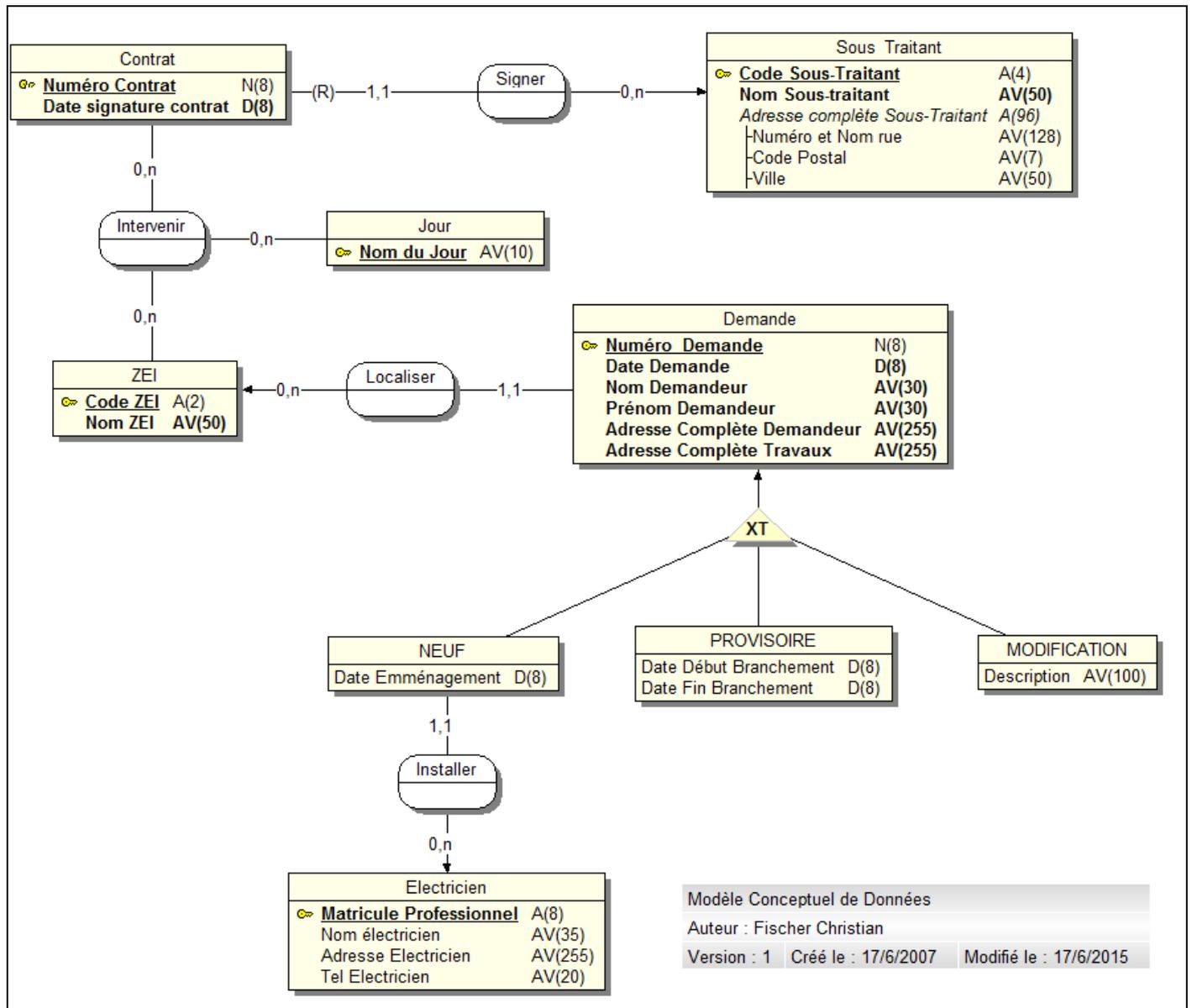
Un contrat précise dans quelles ZEI un sous-traitant est susceptible d'intervenir.

Pour chacune de ces ZEI, le contrat précise les jours d'intervention possibles. Par exemple, le sous-traitant « STEN », au titre du contrat n° 385, peut intervenir tous les mardis et mercredis chez les clients de la ZEI « Bayeux » et tous les mardis, jeudis et vendredis dans la ZEI « Isigny ».

Il existe trois types de demande de branchement électrique : branchement neuf, branchement provisoire ou modification du branchement actuel.

À chaque demande de branchement, le client doit compléter un dossier.

# MEA associé



## 1. Modèle de données physique optimisé

### Travail à faire

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

Note : un champ type de demande (nom du champ TYPE\_DEMANDE) doit être créé pour stocker le type de la demande avec la valeur (M, N, P).

## 2. Optimisation pour Oracle

### Travail à faire

1. Rédiger l'instruction SQL de création de la table DEMANDE du modèle optimisé en proposant une solution de partitionnement.

Note : 500 demandes sont réalisées par semaine. La première analyse a été réalisée le 1/4/2010. Toutes les analyses sont conservées dans la base.

2. Indiquer les tables candidates à une organisation en IOT.  
Rédiger l'ordre SQL de création d'une table organisée en index.

## 3. Programmation PL/SQL

### Travail à faire

1. Rédiger l'instruction de la création de la séquence SEQ\_DEMANDE sur la table DEMANDE.

Le dernier numéro attribué est 12207801.

2. Rédiger le trigger d'insertion dans la table DEMANDE qui doit affecter la colonne TYPE\_DEMANDE en fonction des dates renseignées avec la valeur N=Neuf, P=Provisoire, M=Modification et utiliser la séquence pour attribuer le numéro de la demande.

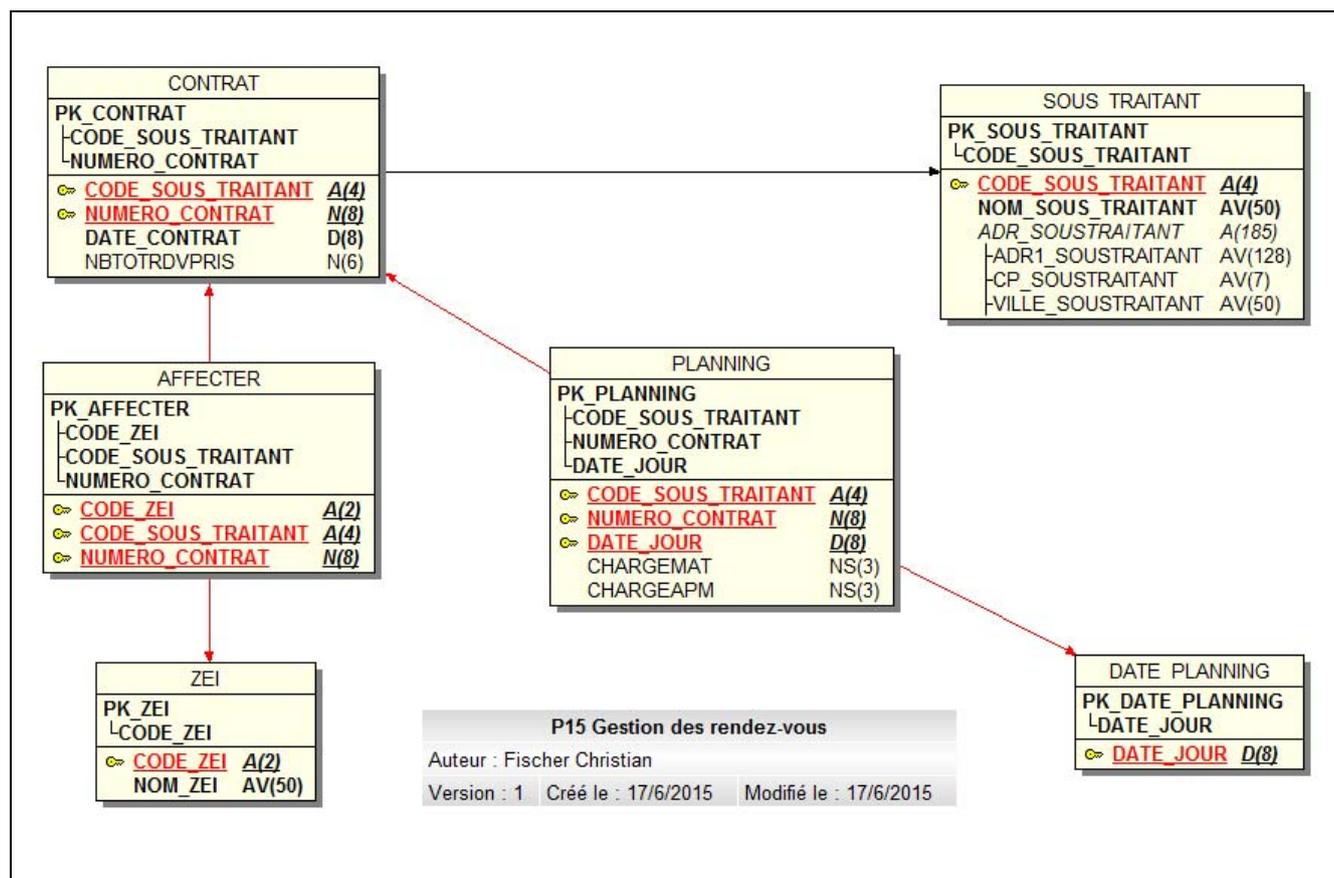
3. Rédiger le l'interface et le corps du package de gestion de la table PA\_DEMANDE avec les procédures ou Fonctions : GetByPK, GetALL, GetALLByTYpe, Upd, Add et Remove

## Dossier 2 : Gestion des rendez-vous

### Contexte

Pour effectuer les installations, le centre EDF de Douvres organise le planning des interventions des sous-traitants. Chaque contrat de sous-traitant couvre un certain nombre de ZEI et indique les jours d'intervention possibles.

### Modèle logique de données relationnel



### Commentaires

*nbTotRDVPris* : nombre total de rendez-vous pour l'année en cours

*Planning* : représente toutes les journées d'intervention planifiées à l'avance dans le cadre des contrats passés pour l'année en cours.

*chargeMAT* : correspond à la charge de travail affectée le matin ; elle est initialisée à zéro au moment de la création et ne peut dépasser 240 minutes.

*chargeAPM* : correspond à la charge de travail affectée l'après-midi ; elle est initialisée à zéro au moment de la création et ne peut dépasser 240 minutes.

*Exemple* : Si le mardi et le mercredi sont les jours d'intervention possibles dans le cadre du contrat N, la table *PLANNING* contient, pour ce contrat, autant de lignes que de mardis et de mercredis dans l'année en cours.

## **Travail à faire**

Remarques :

- 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**.
- L'utilisation des **alias de table** est recommandée.

### **Rédiger les commandes LDD**

1. Création de table Contrat avec les clés primaires et étrangères. Le tablespace pour les données est RDV\_DATA et le tablespace pour les index est RDV\_IND
2. Créer un index unique sur la colonne nom du sous-traitant de la table SOUS-TRAITANT.

### **Rédiger les commandes LMD**

3. Quelles sont les dates des journées entièrement pleines du contrat numéro 1632 ?
4. Quels sont les noms des sous-traitants qui travaillent dans la ZEI de Caen ?
4. Quelles sont les zones sans contrats ?
5. Quel est le (ou les) sous-traitant(s) ayant obtenu le plus grand nombre de rendez-vous (nom du sous-traitant et nombre total de rendez-vous pris)
6. Rechercher la journée ayant le plus de contrats affectés.
5. Compter le nombre de contrats souscrits par zone EI. Afficher le code ZEI, le nom de ville et le nombre souscrits depuis le début de l'année. Les zones n'ayant pas de contrats doivent toujours être affichées.
6. Mettre à jour la colonne nbTotRDVPris pour le sous-traitant STEN (nomSoustraitant) et le contrat numéro 3.  
Fournir une solution permettant d'automatiser cette mise à jour.

### **Travail à faire : Optimisation de requêtes SQL**

A partir du code SQL ci-dessous et du plan d'exécution associé, analyser le plan et proposer une optimisation pour cette requête.

**Note :** *la base de données Oracle ne contient que des index sur les clés primaires et clés étrangères du modèle relationnel présenté.*

#### **Requête 1**

```
select
  SST.*
from
  sous_traitant SST
where
  SST.NOM_SOUS_TRAITANT='STEN'
;
```

**Plan d'exécution** (colonne rows non significatif car test sur une base vide).

-----  
Plan hash value: 279794106

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	131	2 (0)	00:00:01
* 1	TABLE ACCESS FULL	SOUS_TRAITANT	1	131	2 (0)	00:00:01

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

1 - filter("SST"."NOM\_SOUS\_TRAITANT"='STEN')

## Requête 2

```
select
  PLA.NUMERO_CONTRAT as "Numero de contrat"
, sum(PLA.CHARGEMAT) as "Total Matin"
, sum(PLA.CHARGEAPM) as "Total Après Midi"
from
  Planning PLA
where
  PLA.DATE_JOUR between '15/6/2015' and '19/6/2015'
and
  PLA.CODE_SOUS_TRAITANT='S057'
group by
  PLA.NUMERO_CONTRAT
order by
  1;
```

### Plan d'exécution (colonne rows non significatif car test sur une base vide).

Plan hash value: 3856675224

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	54	1 (0)	00:00:01
1	SORT GROUP BY NOSORT		1	54	1 (0)	00:00:01
* 2	TABLE ACCESS BY INDEX ROWID	PLANNING	1	54	1 (0)	00:00:01
* 3	INDEX RANGE SCAN	I_FK_PLANNING_CONTRAT	1		1 (0)	00:00:01

Predicate Information (identified by operation id):

- ```
2 - filter("PLA"."DATE_JOUR">=TO_DATE(' 2015-06-15 00:00:00', 'yyyy-mm-dd hh24:mi:ss')
          AND "PLA"."DATE_JOUR"<=TO_DATE(' 2015-06-19 00:00:00', 'yyyy-mm-dd hh24:mi:ss'))
3 - access("PLA"."CODE_SOUS_TRAITANT"='S057')
```

### Requête 3

```

select
  SST.NOM_SOUS_TRAITANT      as "Nom Sous-Traitant"
  ,count(PLA.NUMERO_CONTRAT) as "Nombre de contrats"
  ,sum(PLA.CHARGEMAT)        as "Total Matin"
  , sum(PLA.CHARGEAPM)       as "Total Après Midi"
from
  Planning PLA inner join  sous_traitant SST
  on PLA.CODE_SOUS_TRAITANT = SST.CODE_SOUS_TRAITANT
where
  to_char(PLA.DATE_JOUR,'YYYY') ='2015'
and
  SST.VILLE_SOUSTRAITANT='Belfort'
group by
  SST.NOM_SOUS_TRAITANT
order by
  1;

```

#### Plan d'exécution *(colonne rows non significatif car test sur une base vide).*

Plan hash value: 448133056

| Id   | Operation                   | Name                  | Rows | Bytes | Cost (%CPU) | Time     |
|------|-----------------------------|-----------------------|------|-------|-------------|----------|
| 0    | SELECT STATEMENT            |                       | 1    | 105   | 3 (67)      | 00:00:01 |
| 1    | SORT ORDER BY               |                       | 1    | 105   | 3 (67)      | 00:00:01 |
| 2    | HASH GROUP BY               |                       | 1    | 105   | 3 (67)      | 00:00:01 |
| 3    | NESTED LOOPS                |                       |      |       |             |          |
| 4    | NESTED LOOPS                |                       | 1    | 105   | 1 (0)       | 00:00:01 |
| 5    | VIEW                        | VW_GBC_5              | 1    | 45    | 0 (0)       | 00:00:01 |
| 6    | HASH GROUP BY               |                       | 1    | 41    | 0 (0)       | 00:00:01 |
| * 7  | TABLE ACCESS BY INDEX ROWID | PLANNING              | 1    | 41    | 0 (0)       | 00:00:01 |
| 8    | INDEX FULL SCAN             | I_FK_PLANNING_CONTRAT | 1    |       | 0 (0)       | 00:00:01 |
| * 9  | INDEX UNIQUE SCAN           | PK_SOUS_TRAITANT      | 1    |       | 0 (0)       | 00:00:01 |
| * 10 | TABLE ACCESS BY INDEX ROWID | SOUS_TRAITANT         | 1    | 60    | 1 (0)       | 00:00:01 |

Predicate Information (identified by operation id):

```

7 - filter(TO_CHAR(INTERNAL_FUNCTION("PLA"."DATE_JOUR"),'YYYY')='2015')
9 - access("ITEM_1"="SST"."CODE_SOUS_TRAITANT")
10 - filter("SST"."VILLE_SOUSTRAITANT"='Belfort')

```