



Formation continue et services aux entreprises

Programmation, bases de données et serveurs –AEC (LEA.D4)

Techniques de l'informatique DEC accéléré (420.B0)

TP II

Bases de données relationnelles

420-W23-SF

Pondération : 2-3-3

Prof. responsable du TP : Ali Awdé
Bureau : P-416
Courriel : aawde@csfoy.ca

Évaluation : 20% de la session.

Réalisation : En équipe.

Date limite de remise : tel qu'indiqué sur LÉA.

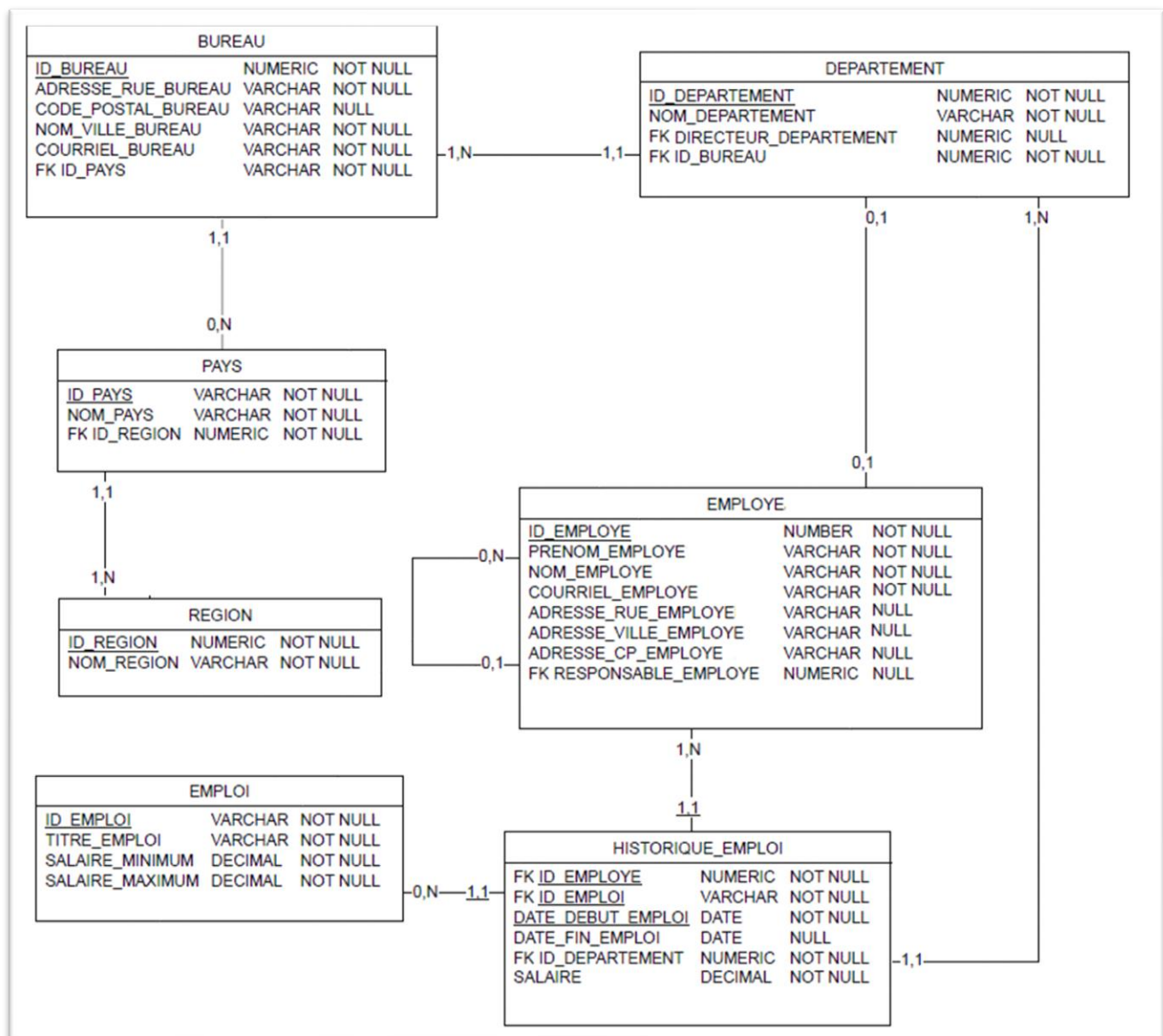
Environnement : MySQL Serveur 8.0

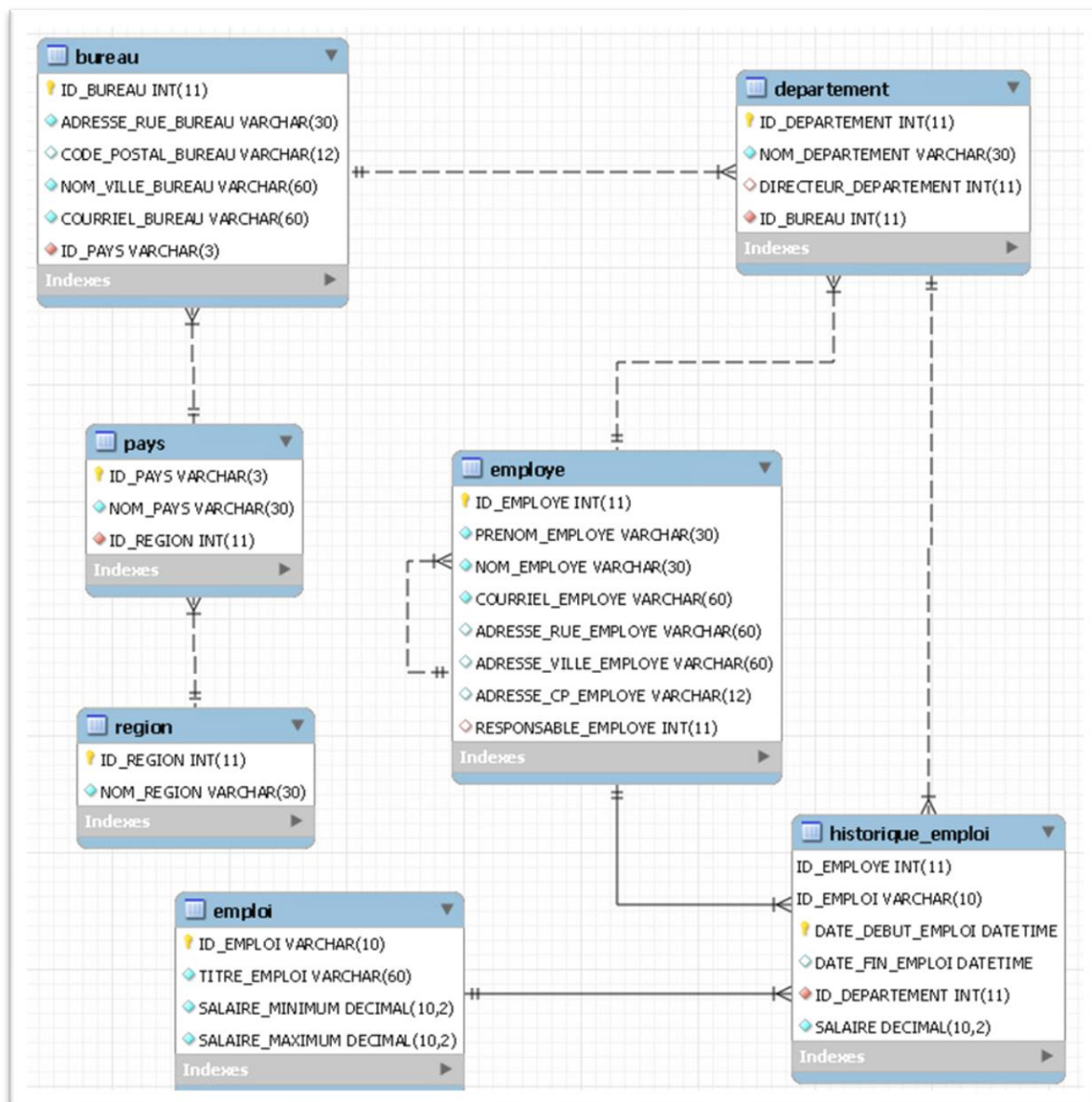
SPÉCIFICATIONS DU TRAVAIL

Contexte

Vous devez partir de ces diagrammes pour construire votre base de données (HR). Lisez bien les deux diagrammes pour avoir les informations dont vous avez besoin pour créer votre propre BD HR.

Vous ne pouvez pas modifier le MRD.





Partie 1 :

À partir du modèle donné, créer la base de données avec les tables et toutes les contraintes. Il faudra tenir compte des contraintes suivantes :

- SALAIRE_MINIMUM est entre 100 et 100000.
- SALAIRE_MAXIMUM est entre 250 et 250000.
- SALAIRE_MINIMUM < SALAIRE_MAXIMUM.
- DIRECTEUR_DEPARTEMENT est une clé étrangère (ID_EMPLOYE).
- DATE_DEBUT < DATE_FIN.
- (ID_EMPLOYE, ID_EMPLOI, DATEDEBUT_EMPLOI) forme la clé primaire de la table historique_emploi.

- Les clés primaires et les clés étrangères.
- Vous devez respecter les noms des champs et le type de données des champs.
- Valeur par défaut de la DATE_FIN est 10 ans après la date d'aujourd'hui.
- SALAIRE_MINIMUM est par défaut à 100.
- N'oubliez pas que certains champs ne peuvent pas être NULL (voir le diagramme).

Il faut écrire un script SQL qui permet de faire ce travail. On devra être capable de réexécuter ce script sans problème si la BD existe déjà on la supprime et on crée une nouvelle BD.

Partie 2 :

Écrire un script qui permet d'insérer les données dans les tables créées dans la partie 1. On devra être capable de ré exécuter ce script sans problème si on veut ramener le contenu de la BD à l'état initial.

Pour insérer les données, consulter les fichiers csv remis avec l'énoncé (*Bureaux.csv, Departements.csv, Emplois.csv, Employes.csv, Historique_Emplois.csv, pays.csv et regions.csv*)

Vous ne pouvez pas modifier les structures ni les données de ces fichiers csv.

Partie 3 :

Écrire les requêtes SQL pour répondre aux besoins exprimés ci-dessous :

- 1) Retrouver et afficher les informations sur les bureaux d'un pays donné si on connaît son ID (ex. JP)
- 2) Retrouver et afficher les informations sur les bureaux triés selon le ID_PAYS.
- 3) Afficher le ID_PAYS, le nom du pays et le nombre de bureaux de chaque pays.
- 4) Afficher le ID_EMPLOI et le TITRE_EMPLOI de l'emploi qui a le plus cher salaire selon salaire_maximum.
- 5) Afficher le salaire minimum moyen et le salaire maximum moyen des emplois de la compagnie.
- 6) Afficher le ID_REGION et le nombre de pays de chaque région.
- 7) Afficher le nom de la région et le nombre de pays de chaque région.
- 8) Afficher le nom du pays et nombre de bureaux de chaque pays.

Table temporaire

1. Créer une table temporaire (tbl_tmp_burpays) qui contient le nom du pays ainsi que les informations des bureaux de chaque pays.
2. Remplir cette table via une requête à partir de vos tables de la BD HR.

VUES

Créer des vues qui permettent de :

1. Afficher le nom de la région et le nom des pays de chaque région.
2. Afficher le nom des régions qui possède plus que 3 pays.
3. Afficher pour chaque employé, son nom, prénom, son titre d'emploi actuel, le nom de son département, son salaire et la date de début de son emploi actuel. Si la date de fin est NULL ou il est une date dans le futur, on considère que l'employé occupe encore cet emploi.
4. Afficher le nom et le salaire moyen des emplois actuels de chaque département.
5. Afficher le nom et la masse salariale (c'est la somme des salaires) de chaque département (les emplois actuels)
6. Afficher le nom et la masse salariale de chaque bureau (les emplois actuels).
7. Afficher le nom des employés qui ont occupé 3 emplois incluant celui qu'ils occupent encore. Si la date de fin est NULL ou il est une date dans le futur, on considère que l'employé occupe encore cet emploi.
8. Afficher le nom et le prénom de l'employé qui a changé le plus d'emploi.
9. Afficher le nom (sous forme Nom, Prénom) de chaque employé ainsi que le nom (sous forme Nom, prénom) de son responsable.

Nommer les vues comme suit : Vue1, Vue2 ...Vue9

À remettre et répartition des points

1. Vous devez déposer un seul fichier SQL où les trois (3) sections suivantes sont bien identifiées :
 - DDL : création de la base de données avec ses contraintes ;
 - DML : insertions des données ; (les données des fichiers csv sans aucune modification des données).
 - DML : requêtes demandées.
2. Section des requêtes demandées (50 % en correction individuelle) :
 - i. Vous devez indiquer pour chaque demande, qui a réalisé le code.
 - ii. Chaque coéquipier est responsable d'une partie (même pourcentage) des requêtes.
 - iii. La qualité de votre code sera notée.
 - iv. Le barème de correction de cette section est le suivant :
 - A : excellent : 45 à 50 – dépasse les attentes.

- B : très bon : 40 à 44 – réponds à la demande, la qualité du code est bonne.
- C : bon : 30 à 39 – réponds à la demande, mais la qualité du code est passable.
- D : passable 25 à 30 – fonctionne, mais comporte des erreurs.
- E : insuffisant 0 à 25 – ne s'exécute pas

3. Qualité du document (10 % en correction d'équipe).
4. Le non-respect des consignes (nom des entités, modifications des données, etc.) entraîne des pertes des points selon la gravité du problème (-20%)

Sommaire de la répartition des points :

Item	Points	Équipe ou individuelle
DDL	25	Équipe
DML	15	Équipe
Requêtes	50	Individuelle
Qualité du document	10	Équipe
Consignes	-20	Équipe/individuelle
Total	100 points	