Compte-rendu de projet de base de données : Gestion des Stages

Youssef Sedrati ,

Mohamed-Said Mattiche ,

Romain d'Esparbès

7 décembre 2015

Introduction :

Ce projet prend place dans le cadre de l'apprentissage du cours de systèmes de gestion de bases de données. A l'issue de la formation du semestre 7, trois sujets ont été proposés. Nous nous sommes particulièrement sentis concernés par le sujet "Stages", et l'avons choisi en conséquence. Le but de ce projet et de mettre en place une base de données permettant de manipuler les offres de stages de l'école, et tout ce qui les entoure. Le but principal de ce projet est la création d'une base de données afin de gérer les stages de notre école . Cette dernière stockera toutes les informations nécessaires pour les stages proposés à l'ENSEIRB-MATMECA .

1 Modèle conceptuel :

Le modèle entité-association est utilisé principalement pour faire une description du sujet. Certains points ne sont pas clairs dans ce dernier, nous nous sommes ainsi mis d'accord pour faire certains choix. Pour se faire, nous avons suivi le plan suivant :

- Il a été nécessaire de modéliser les différents acteurs (au sens large) qui interviennent lors de l'organisation d'un stage
- Nous avons ensuite mis les entités en relation par des associations.
- La troisième phase consiste à étudier les différentes cardinalitées.
- La dernière étape s'intéresse à la normalisation du modèle.

Premiere et deuxieme version du diagramme conceptuel :

Une première étude du sujet a conduit au diagramme conceptuel de la figure page 3. Ce diagramme manquait de clareté notament dans la relation "Affecté" et présentait des aspects complexes au niveau de la cardinalité. Un ensemble d'observations de bon sens de ce type et une réflexion plus poussée nous ont donc conduit à une deuxième version du diagramme conceptuel.

Le modèle entité-association de la figure page 4 est ainsi obtenu.

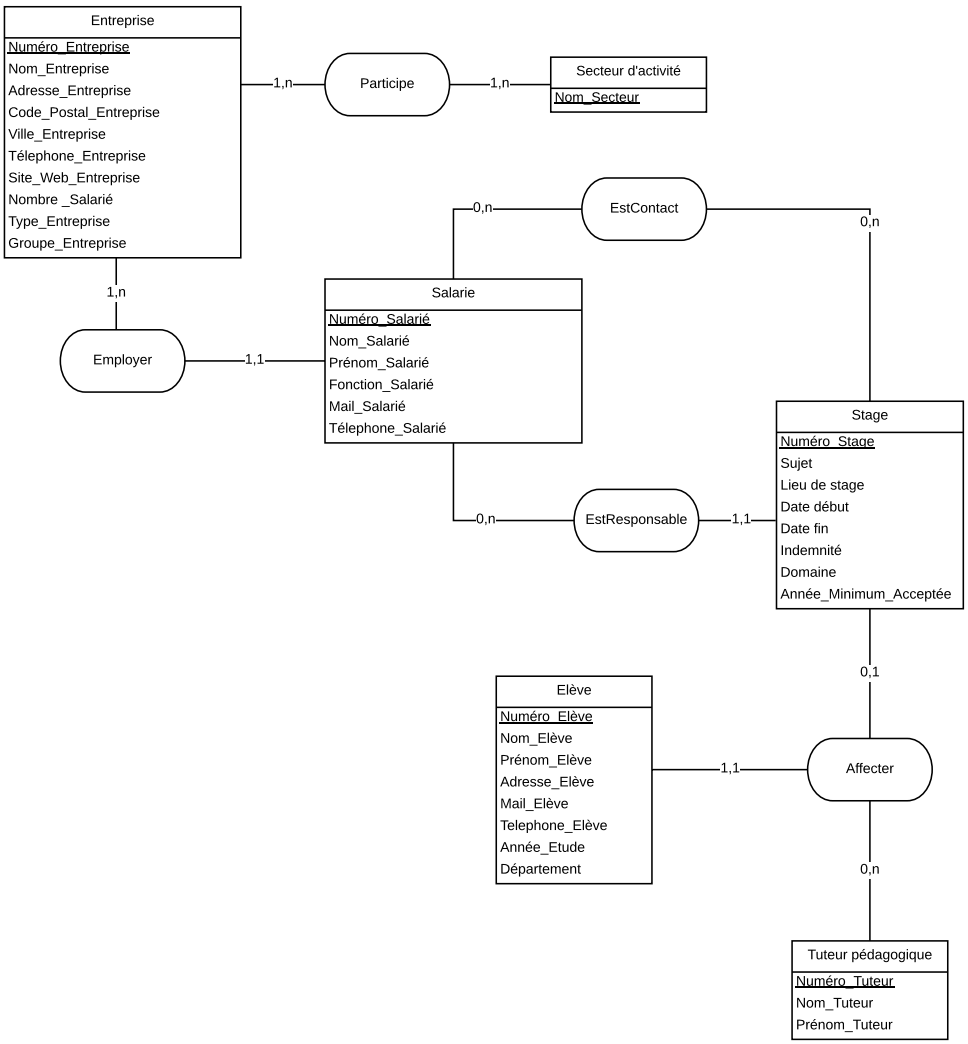

FIGURE 1 – Modèle entité-association de la première version

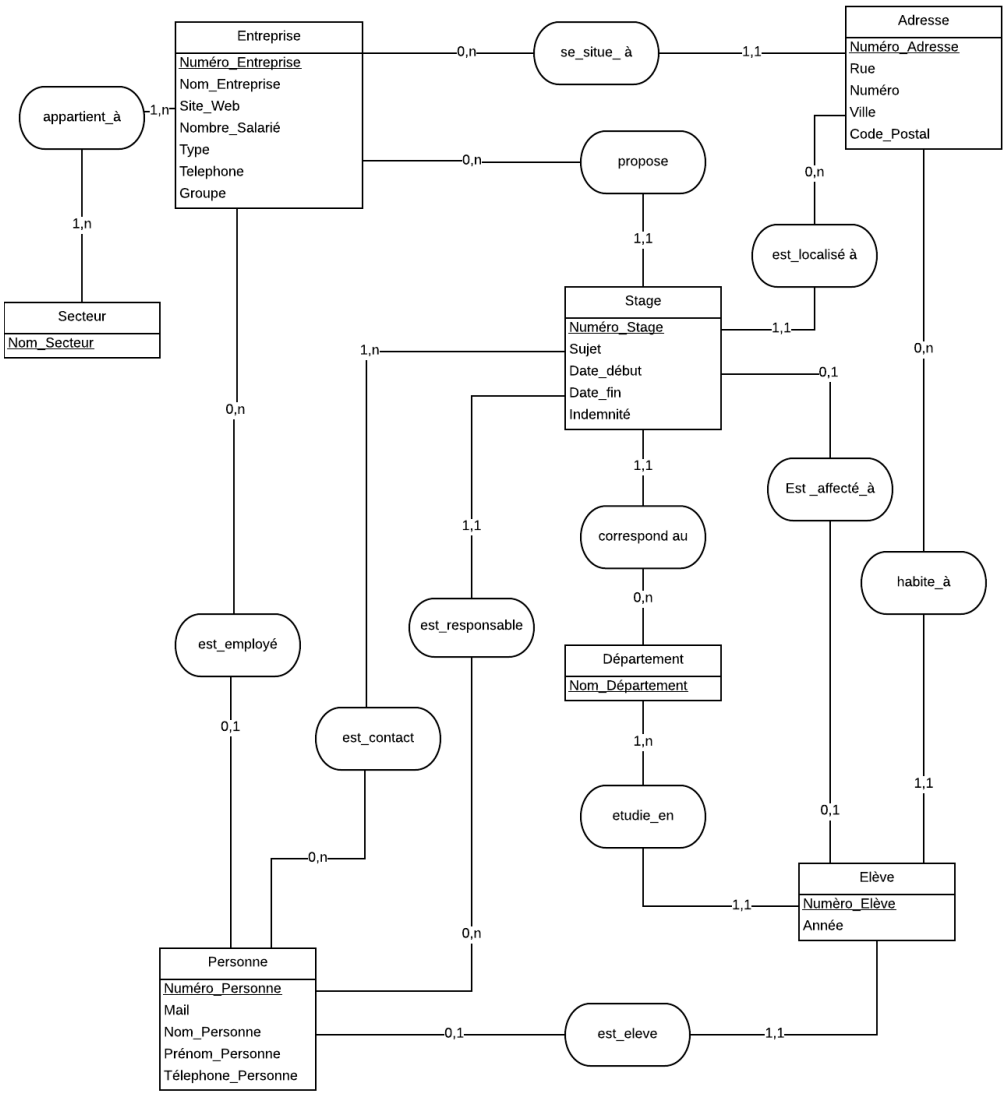

FIGURE 2 – Modèle entité-association de la deuxième version

2 Modèle relationnel :

Une fois le modèle conceptuel réalisé, il a été nécessaire de le traduire en modèle relationel en respectant les trois règles de transcription décrites ci-dessous :

— Dans le cas d'une relation avec cardinalités $(0,1),(1,1) - (0,n)(1,n)$, comme dans le cas de la relation "se_situe_à" entre "Entreprise" et "Adresse", on ne créé pas d'entité supplémentaire dans la base, mais on passe en clé étrangère le numéro d'entreprise dans l'entité "Adresse" de manière à garantir qu'une adresse ne contienne qu'une unique entreprise et qu'une entreprise peut être située à plusieurs adresses.

— Dans le cas d'une relation avec cardinalités $(0,n),(1,n) - (0,n)(1,n)$, comme dans le cas de la relation "est_contact" entre "Personne" et "Stage", on créé une entité dans la base qui représente la relation, et qui a pour clé primaire le couple des clés primaires des entités "Personne" et "Stage".

— Enfin, dans le cas d'une relation avec cardinalités $(0,1) - (1,1)$, comme dans le cas de la relation "est_elève" entre "Personne" et "Elève", on ne créé pas d'entité supplémentaire dans la base et on met la clé étrangère qui correspond à la clé primaire de "Personne" dans "Elève". On garantit de cette façon qu'un élève ne correspond qu'à une personne, et qu'une personne n'est pas forcément un élève.

Modèle relationnel pour la version 1 :

Entreprise(Numéro_Entreprise, Nom_Entreprise, Adresse_Entreprise,
Code_Postal_Entreprise, Ville_Entreprise, Téléphone_Entreprise,
Site_Web_Entreprise, Nombre_Salariés, Type_Entreprise, Groupe_Entreprise)
Secteur(Nom_Secteur)
Participe(#Numéro_Entreprise, #Nom_Secteur)
Salarié(Numéro_Salarié, Nom_Salarié, Prénom_Salarié, Fonction_Salarié,
Mail_Salarié, Téléphone_Salarié, #Numéro_Entreprise)
Stage(Numéro_Stage, Sujet, Lieu_Stage, Date_début, Date_Fin, Indemnité,
Domaine, Année_Minimum_Acceptée, #Numéro_Salarié)
Tuteur(Numéro_Tuteur, Nom_Tuteur, Prénom_Tuteur)
Est_Contact(#Numéro_Salarié, #Numéro_Stage)
Eleve(Numéro_Eleve, Nom_Elève, Prénom_Elève, Adresse_Elève, Mail_Elève,
Téléphone_Elève, Année_Etudes, Département, #Numéro_Stage,
#Numéro_Tuteur)
Affecter(#Numéro_Stage, #Numéro_Elève, #Numéro_Tuteur)

Modèle relationnel pour la version 2 :

Entreprise(Numéro_Entreprise, Nom_Entreprise, Site_Web, télephone,
nombre_salariés, type, groupe, Numéro_Adresse)
Eleve(Numéro_Eleve, #Numéro_Personne, #Numéro_Stage,
#Numéro_Adresse, #Nom_Département, année)
Personne(Numéro_Personne, mail, Nom_Personne, Prénom_Personne,
Télephone_Personne)

Stage(Numéro_Stage, #Numéro_Entreprise, #_Adresse, #Nom_Département, indemnités, sujet, Date_début, Date_Fin)
Secteur(Nom_Secteur)
Adresse(Numéro_Adresse, #Numéro_Entreprise, ville, rue, Numéro, Code_Postal)
Département(Nom_Département)
Est_employé(#Numéro_Entreprise, #Numéro_Personne)
Appartient(#Nom_Secteur, #Numéro_Entreprise)
Est_Contact(#Numéro_Stage, # Numéro_personne)
Est_responsable(#Numéro_Stage, #Numéro_Personne)

3 Implémentation en SQL :

Nous allons nous intéresser aux étapes de l'implémentation de notre base de données. Tout d'abord, le système de gestion de base de données et le langage de programmation choisi dans ce projet sont SQL*Plus et Java. Nous avons réalisé une interface permettant à l'utilisateur de notre application de consulter des données dans la base, de supprimer des données, insérer d'autres ou bien appliquer des modification sur des tables déjà existants dans la base.

3.1 Requêtes :

3.1.1 Requêtes de consultations et de statistiques :

Le choix le plus naturel a été de créer des vues pour les requêtes plus complexes. Par exemple, une vue a été construite pour le classement des meilleurs fournisseurs de stage. L'utilisateur choisi simplement dans un menu déroulant la requête qu'il souhaite effectue et celle-ci appelle la vue correspondante.

3.1.2 Requêtes de mise à jour

Chaque table possède des propriétés sur ses attributs lors de la suppression ou de la mise à jour. Par exemple, si l'on supprime une entreprise, on doit supprimer tous les stages qui étaient proposés par celle-ci sans pour autant supprimer l'élève qui suivait le dit-stage. C'est pour cette raison que plusieurs des entités contiennent une ligne "sentinelle", une valeur à laquelle pointerait par exemple l'élève ayant perdu son stage dans l'attente d'une réaffectation. Il a simplement suffit dans la spécification des clés étrangères d'utiliser les mots clés *on delete cascade* ou *on delete set default*. Il en va de même si l'on choisit de changer les identifiants des personnes, ou autres entités, on résout le problème avec un *on update cascade*.

4 Utilisation

Il a été choisi d'utiliser JDBC comme interface entre la base de données et l'utilisateur. Ce framework permet de gérer la base de données à travers du code Java. Un affichage en console a ainsi été produit, permettant de créer la base sur le serveur Oracle, de la supprimer, d'y ajouter des données d'exemple, de les

supprimer, et enfin de générer le résultat des requêtes sous la forme de tableaux.

Pour accéder à cette implémentation, il est necessaire de se connecter à la plateforme Oracle de l'ENSEIRB en ssh : `ssh oracle`. Depuis le répertoire de travail, on lance `source jdbc.sh` pour mettre à jour la variable d'environnement `CLASSPATH` en lui indiquant où se trouve le framework JDBC sur la machine. On peut ensuite lancer la compilation via un simple `make`, puis exécuter le programme avec `make launch`. Un menu s'affiche :

1. "Initialize tables" créé l'ensemble des tables de la base de données. Cette commande exécute le fichier `tables.sql`
2. "Delete tables" supprime toutes les tables de la base. Cette commande exécute le fichier `reset_tables.sql`
3. "Clean tables" vide le contenu de toutes les tables en exécutant le fichier `reset_data.sql`
4. "Add example data" remplit les tables avec des données d'exemple contenues dans `data.sql`
5. "Set display width" change la largeur d'une colonne dans l'affichage des tables.
6. "Execute query" permet de lancer une requete donnée à partir d'un chemin vers un fichier SQL. Un argument valide est par exemple "sta_departement.sql"
7. "Quit" arrête le programme.

```
MAIN MENU
1: Initialize tables
2: Delete tables
3: Clean tables
4: Add example data
5: Set display width
6: Execute query
7: Quit
6
Enter the name of the SQL file you want to execute:
test.sql
************************************************************************************************************************
NUMERO_SALARIE   *NOM_SALARIE      *PRENOM_SALARIE   *FONCTION_SALARIE *MAIL_SALARIE     *TELEPHONE_SALARIE*NUMERO_ENTREPRISE*
************************************************************************************************************************
1                |FRANCOIS         |CLAUDE           |TECHNICIEN       |CLAUDE.FRANCOIS@G|0101010101       |1                |
------------------------------------------------------------------------------------------------------------------------
2                |GABI2            |BERNARD2         |COMMUNICATION    |QSDKJ@GMAIL.FR   |321313123        |2                |
------------------------------------------------------------------------------------------------------------------------
3                |GABI3            |BERNARD3         |COMMUNICATION    |QSDKJ@GMAIL.FR   |321313123        |3                |
------------------------------------------------------------------------------------------------------------------------
4                |GABI4            |BERNARD4         |COMMUNICATION    |QSDKJ@GMAIL.FR   |321313123        |4                |
------------------------------------------------------------------------------------------------------------------------
5                |GABI5            |BERNARD5         |COMMUNICATION    |QSDKJ@GMAIL.FR   |321313123        |4                |
------------------------------------------------------------------------------------------------------------------------
MAIN MENU
1: Initialize tables
2: Delete tables
3: Clean tables
4: Add example data
5: Set display width
6: Execute query
7: Quit
```

FIGURE 3 – Exemple d'utilisation pour l'exécution d'une requête

Pour sélectionner une commande, il suffit d'entrer le numéro correspondant, puis d'appuyer sur la touche "Entrée". La figure 3 représente un cas simple d'utilisation où l'on souhaite exécuter un fichier "test.sql" qui contient la requête : `SELECT * FROM SALARIE`.

Conclusion générale

Ce projet nous a permis d'exploiter les différentes techniques enseignés dans le module SGBD . Il nous a donné l'opportunité d'assimiler les différentes étapes nécessaires à la création d'une base de données. Nous nous sommes convaincus que la création du modèle conceptuel est une phase très importante afin de réaliser le projet, et que ce modèle passe obligatoirement par l'étape de normalisation qui permet de déceler certains problèmes liés à l'intégrité de la base.
Par ailleurs, il permet de mettre en avant les difficultés que pose la modélisation de la base de données. Certes, certaines données sont facilement implémentable indépendament de la modélisation choisie, mais d'autres penvent soulever une discussion. Il faut alors bien définir les contraintes afin de maintenir une certaine cohérence.
Dans le but de garder cette cohérance, les opérations basiques (insertion, mise à jour et suppression) doivent être correctement réalisées.
Pour conclure, construire une base de données nécessite une bonne modélisation avec des choix pertinants et adéquats qui facilitent le travail lors de l'implémentation.