

# Projet Bases de Données et Conception Objet

---

## 1 – Introduction

Ce projet a pour but de mettre en œuvre vos compétences en systèmes de gestion de bases de données relationnelles et en conception objet. Ce projet vient en application du cours de « Principes des systèmes de gestion de bases de données » (PSGBD) et du cours d' « Analyse et conception objet de logiciels embarqués » (ACLE). Vous devez réaliser l'analyse, la conception et la programmation complète d'une application de gestion de tournois d'échecs utilisant une base de données. Vous devez notamment concevoir et implanter un schéma relationnel. Le programme doit être écrit en *Java*, et les accès à la base de données effectués grâce l'API *JDBC*.

Le projet est à faire en équipe de 6 et donnera lieu à la remise d'un rapport, ainsi qu'à une soutenance comprenant, entre autres, une démonstration. L'évaluation concerne l'analyse, la conception et l'implémentation, à la fois sur les aspects bases de données et sur la partie programmation orientée objets.

## 2 – Description de l'application

L'application que vous devez développer doit permettre la gestion d'un tournoi d'échecs. Elle se présente sous la forme de deux exécutables :

- Le premier exécutable sert à l'organisation d'un tournoi d'échecs. Un tournoi se compose de plusieurs rencontres organisées en différents tours entre les joueurs qui se sont inscrits. Cet exécutable permet d'inscrire les participants et d'organiser chacun des tours du tournoi (qualifications, quart de finale, demi-finale et finale).
- Le second exécutable permet de jouer ou rejouer une rencontre. Une nouvelle rencontre peut avoir lieu parmi celles du tour actif, les joueurs accédant alternativement à l'application pour jouer leurs coups. Plusieurs parties d'un même tour pourront être jouées en même temps en lançant l'application plusieurs fois. Tous les coups d'une partie sont enregistrés de façon à pouvoir rejouer la partie pour analyse *a posteriori*.

Les deux exécutables disposeront d'une *interface graphique*. Par contre, il n'est pas nécessaire pour cette version du logiciel que l'accès des personnes l'utilisant soit géré par un système d'identification et de mot de passe. *Il vous est explicitement demandé de ne pas réaliser cela* : la personne utilisant le logiciel sera simplement sélectionnée parmi la liste des personnes enregistrées.

### a) Organisation d'un tournoi

Les rencontres d'un tournoi se conforment au programme suivant :

- Un *tour de qualifications*, lors duquel chaque participant doit rencontrer chacun des autres ;
- Un *tour de quart de finale*, lors duquel les 8 joueurs ayant remportés le plus de victoires au tour de qualification se rencontrent deux à deux (1<sup>er</sup> et 8<sup>ème</sup>, 2<sup>nd</sup> et 7<sup>ème</sup>, 3<sup>ème</sup> et 6<sup>ème</sup>, 4<sup>ème</sup> et 5<sup>ème</sup> dans le classement) ;
- Un *tour de demi-finale*, où les vainqueurs des deux premiers quarts de finale se rencontrent et les vainqueurs des deux derniers quarts de finale se rencontrent également ;
- Une *finale* où s'affrontent les deux vainqueurs des demi-finales.

Un joueur est identifié par un numéro unique pour le tournoi et est décrit par son nom, son prénom, son adresse postale et sa date de naissance. Une rencontre entre deux joueurs est identifiée par le tour et le numéro de partie dans le tour (ex : 1er quart de finale). La couleur des pièces d'un joueur (blanc ou noir) pour une partie est tirée aléatoirement. Le vainqueur de la partie doit être mémorisé afin de faciliter la préparation du tour suivant.

### b) Jouer une partie

L'échiquier sera représenté par l'ensemble des pièces présentes (32 pièces en début de partie) et leurs positions. Précisément, une pièce est identifiée par un numéro unique pour la partie en cours et décrite par sa couleur, sa position courante sur une grille de 8x8 cases, ainsi que sa position précédente. Cette dernière information servira à valider le déplacement d'une pièce en fonction de son type et de sa couleur.

Nous allons considérer ici des règles du jeu d'échecs simplifiées par rapport aux règles officielles<sup>1</sup> :

- Les blancs sont les premiers à jouer ;
- Les pions ne se déplacent que d'une seule case, quelles que soient leurs positions ;
- Pas de roque ;
- Pas de prise en passant ;
- Pas de promotion ;
- Un joueur peut décider à tout moment d'abandonner la partie ;
- Si les deux joueurs sont d'accord, une partie peut être annulée et rejouée depuis le début (exemple : cas du *pat*).

Avant qu'un joueur ne joue, il faut vérifier sa situation :

- Vérifier s'il est en *échec* ;
- Le cas échéant, vérifier s'il est *mat* :
  - Vérifier les déplacements possibles du roi,
  - Vérifier s'il est possible ou non de prendre la pièce qui le met en échec sans se démasquer (et se retrouver de nouveau en échec),
  - Vérifier s'il est possible ou non de mettre un obstacle entre le roi et la pièce qui le met en échec sans se démasquer ;
- Il n'est pas nécessaire de vérifier une situation de *pat*.

Chaque coup joué par un joueur donnera lieu à un certain nombre de vérifications :

---

1 Cf. [http://fr.wikipedia.org/wiki/%C3%89checs#R.C3.A8gles\\_du\\_jeu](http://fr.wikipedia.org/wiki/%C3%89checs#R.C3.A8gles_du_jeu)

- Vérifier la validité de la nouvelle position par rapport à l'ancienne (**DOIT ETRE REALISE A L'AIDE DE CONTRAINTES SUR LE SCHEMA RELATIONNEL**) ;
- Vérifier l'absence d'obstacle sur le déplacement, sinon le coup est invalide ;
- Vérifier si le coup entraîne la prise d'une pièce adverse ;
- Vérifier si le joueur ne se met pas en échec en jouant ce coup (démasquage d'une pièce adverse), sinon le coup est invalide ;

Ne pas oublier d'enregistrer le coup dans l'historique de la partie (coups numérotés dans l'ordre chronologique pour une partie et comprenant la position initiale et la position finale de la pièce manipulée par le coup).

### c) Rejouer une partie

Une partie jouée et terminée lors d'un tournoi doit pouvoir être rejouée pas à pas dans son intégralité. Il faut pour cela, conserver l'historique des coups au fur et à mesure qu'une partie se joue. On ne demande pas de pouvoir reculer dans l'historique d'une partie, ni de permettre de jouer une variante à partir d'un certain coup, mais juste de rejouer la partie pour l'analyser.

## 3 – Travail à réaliser (12h encadrées + temps personnel)

Les différentes tâches à réaliser sont présentées dans le schéma de la page 6. Chaque rectangle du schéma correspond à une tâche à réaliser par une sous-équipe. Hormis les dates de rendu (intermédiaire et final), le temps à passer sur chaque tâche est indicatif et la répartition des membres sur les tâches (combien de personnes et qui) est laissée à votre appréciation.

### Journal

Il vous est demandé de maintenir **un journal** de projet dans lequel vous préciserez (en quelques lignes) ce qui a été réalisé au cours de chaque séance de travail sur le projet. Ce journal devra être déposé sur l'application *Teide* après chaque mise à jour (déposé, mais non validé) et pourra être consulté à tout moment par vos encadrants tout au long du projet.

### Liste des différents points à aborder :

- 1) **Analyse** : analyse fonctionnelle, analyse statique
- 2) **Conception de la base de données** : élaboration du schéma conceptuel, conception de la base de données, analyse des accès à la base de données
- 3) **Conception de l'application** : architecture de l'application, conception objets
- 4) **Implémentation** : instanciation de la base de données, programmation de l'application.

Réaliser le projet revient à répondre à l'ensemble de ces points (ils sont détaillés dans la suite de l'énoncé). Pourtant, cette liste ne doit pas être prise chronologiquement. C'est une structuration logique du travail, qui pourra être reprise dans la documentation finale. Nous vous proposons de réaliser l'ensemble de ces points avec la répartition (thématique et chronologique) suivante.

### Première semaine

Nous vous conseillons de vous répartir en deux sous-équipes, pour réaliser les deux tâches suivantes :

#### Tâche A : analyse statique et schéma conceptuel de la base de données

- a) **Analyse statique**. Inventorier les données élémentaires, identifier les dépendances fonctionnelles et les contraintes (invariants) entre les données persistantes.

- b) **Élaboration du schéma conceptuel.** Construire, à partir de l'analyse statique un schéma Entités/Associations (ce schéma correspond à un diagramme de classe d'analyse). Expliquer ce schéma (notamment les points difficiles de sa construction). Préciser les contraintes d'intégrité non représentées dans le schéma Entités/ Associations.

### Tâche B : analyse fonctionnelle et architecture de l'application

- a) **Analyse fonctionnelle.** Identifier les acteurs et les cas d'utilisation, à une granularité très grosse et faire un diagramme de cas d'utilisation. Préciser le fonctionnement global du système (en vous aidant si nécessaire d'un ou plusieurs diagrammes états/transitions pertinents). *Le but de cette analyse est principalement de faire en sorte que l'ensemble de l'équipe ait la même lecture du cahier des charges.*
- b) **Architecture de l'application.** Proposer une architecture pour l'application. Préciser en particulier l'architecture permettant d'intégrer les interfaces utilisateur et la base de données à votre logiciel. Pour vous aider, vous pourrez considérer des scénarios typiques pour :
- une action de l'utilisateur sur l'interface.
  - le déclenchement et l'exécution d'une requête depuis l'application.

Chacun de ces scénarios pourront être illustrés par un diagramme de séquence ou de collaboration.

Bien entendu, les sous équipes des tâches A et B devront communiquer très régulièrement car au début du projet, il y a beaucoup d'éléments à déterminer !

### Rendu intermédiaire : avant le 3 avril à 13h.

Vous rendrez un résumé des résultats des tâches A et B. Précisément, nous vous demandons de rendre sur Teide :

- le schéma entité association de votre base de données, sous la forme d'un pdf, format A4
- le schéma ou les schémas de l'architecture de votre logiciel, sous la forme d'un pdf, format A4.

Votre journal de bord doit être à jour !

### Deuxième semaine

Nous vous conseillons de vous répartir en trois sous-équipes, pour réaliser les trois tâches suivantes :

#### Tâche C : schéma relationnel et implémentation de la base de données

Traduire le schéma Entités/Associations en **relationnel** ; écrire les **tables et les contraintes**. Préciser et justifier la **forme normale** des relations, les **contraintes d'intégrité** implantées, ainsi que les **autres contraintes** ne pouvant être implantées en SQL2 (et qui sont donc à vérifier dans le code applicatif Java). **Instancier la base de données** ; elle doit être créée sur le serveur Oracle 12c disponible sur le serveur *ensioracle1*.

#### Tâche D : analyse des accès à la base de données

Identifier les requêtes nécessaires aux différentes fonctionnalités à l'aide de l'analyse fonctionnelle (que vous pourrez compléter si besoin) et les écrire. Délimiter les transactions permettant d'implanter les fonctionnalités tout en préservant la cohérence globale des données.

#### Tâche E : conception objet et implémentation du cœur de l'application

Faire un diagramme de classes logicielles et implémenter en Java les classes du cœur de l'application. Elles sont définies en priorité sur toutes les autres car le reste du logiciel va en dépendre.

## Troisième et quatrième semaines

Nous vous conseillons de vous répartir en trois sous-équipes, pour réaliser les trois tâches suivantes :

### Tâche F : interface graphique

Dessiner et programmer, en Java, l'interface graphique du logiciel. Cette étape requiert de savoir programmer une interface graphique en Java : attention à prévoir un temps d'apprentissage, si nécessaire dans votre organisation. Ne pas oublier d'intégrer et de tester régulièrement !

### Tâche G : implémentation des accès à la base de données

Implémenter, en Java, le module applicatif des accès à la base de données. L'accès à la base de données se fera avec l'API JDBC : attention à prévoir un temps d'apprentissage dans votre organisation. Ne pas oublier d'intégrer et de tester régulièrement !

### Tâche H : interface textuelle

Programmer, en Java, l'interface textuelle et les exécutables les utilisant. Ne pas oublier d'intégrer et de tester régulièrement !

## 4 – Documentation

### Bilan

Rédiger un ***bilan*** de ce projet (répartition des tâches, difficultés rencontrées, leçons retenues, etc.). Pour vous aider dans cette tâche, vous pourrez vous servir du journal du projet.

### Documentation finale

Vous rendrez une documentation du travail réalisé pendant le projet. Elle devra contenir quatre parties correspondant à l'analyse, la conception et l'implantation de la base de données, la conception de l'application, et le bilan.

## 4 – Soutenance (7 et 9 mai)

Votre projet donne lieu à une soutenance de **20 à 25 minutes** (inscription sur les créneaux horaires *via* l'application *Teide*), suivie de 10 à 15 minutes de questions, durant laquelle vous devrez :

- présenter votre modélisation entités-associations et son implantation en relationnel
- présenter votre architecture
- présenter le diagramme de classes logicielles
- faire une démonstration des fonctionnalités ; prévoir un scénario montrant le bon fonctionnement des transactions
- faire un bilan du déroulement du projet

L'objectif est de vendre **votre produit** (en montrant qu'il fait bien ce qu'on attend) **et votre équipe de projet** (en montrant que le produit a été bien construit).

**Note :** Il est très fortement recommandé de consulter régulièrement les pages consacrées à ce projet.

