

PROJET DE SESSION – IFT723

<< Gestion des Paiements des Droits de Scolarités >>

SEM :

Document de Spécification des Exigences du Modèle

Émetteur :

Otham El Biyaali, elbo1901@usherbrooke.ca

Maxime Sourceaux, soum3004@usherbrooke.ca

Alseny Toumany Soumah, soua2604@usherbrooke.ca

Mohamed Boubacar Boureima, boum3688@usherbrooke.ca

Dernière modification :

2023 - 11 - 11

Statut :

Version 1.1.0

Table des Matières

Sommaire.....	3
1 Introduction	4
1.1 Objet et portée du document.....	5
1.2 Évolution du document.....	5
2 Présentation.....	5
2.1 Mise en contexte	5
2.2 Système existant.....	6
2.3 Système envisagé.....	6
3 Définition du problème.....	6
3.1 Présentation du problème	6
3.2 Modélisation du problème.....	7
4 Proposition d'une solution	7
4.1 Présentation de la solution.....	7
5 Spécifications	8
5.1 Scénarios de déploiement et d'exploitation.....	12
5.2 Exigences fonctionnelles	12
6 Conclusion	14

Sommaire

Notre projet consiste en la modélisation, la conception et la mise à l'essai de l'API de la BDT. Il s'agit d'une étape cruciale où nous adaptons, mettons en œuvre et, le cas échéant, migrons notre base de données initiale. Cette phase initiale requiert une compréhension approfondie des exigences de notre projet. En parallèle, il nécessite la rédaction de deux documents clés : la spécification des exigences du modèle (SEM) pour définir précisément les besoins du modèle de données, et la spécification de conception de la base de données (SCBD) qui détaille la structure et la conception de la base. L'objectif ultime est de garantir que la base de données réponde aux exigences du projet tout en maintenant la qualité et la cohérence des données.

1 Introduction

L'introduction de la Spécification des Exigences du Modèle (SEM) représente une étape cruciale dans le processus de développement de notre projet. Le présent document vise à définir de manière claire et précise les besoins du modèle de données.

La SEM constitue une étape fondamentale de notre cycle de développement, car elle servira de référence pour concevoir et implémenter la structure de la base de données. Elle permettra également de s'assurer que notre BDT répondra aux attentes et aux exigences définies par les parties prenantes du projet.

Au cours de cette phase, nous examinerons attentivement les exigences de notre, en les décomposant en éléments concrets et en identifiant les entités, les relations, les attributs et les contraintes nécessaires pour notre modèle de données. Nous tiendrons compte des besoins opérationnels, des sources de données disponibles, et des objectifs globaux du projet pour garantir une conception de modèle de données fonctionnelle et cohérente.

En fin de compte, la SEM servira de document de référence essentiel pour notre l'équipe de développement, en orientant nos efforts vers la création d'un nouveau schéma qui répondra aux besoins du projet tout en respectant les meilleures pratiques en matière de gestion des données. Ce document permettra à notre équipe de projet, une compréhension partagée des exigences et une base solide pour la suite du développement de notre BDT.

1.1 Objet et portée du document

Ce document a pour objectif de définir les exigences du modèle de données pour le projet de base de données temporalisée. Il servira de référence pour guider la modélisation, la conception et la mise à l'essai de l'API de la BDT.

La portée de ce document couvre l'analyse des exigences, l'identification des entités, relations, attributs et contraintes du modèle de données, ainsi que la documentation de ces éléments. Il est destiné à guider notre équipe de développement et à faciliter la compréhension partagée des besoins et objectifs du modèle de données. Ce document servira également de guide essentiel tout au long du processus de développement pour garantir la cohérence et la conformité avec les exigences du projet.

1.2 Évolution du document

L'évolution du document SEM pour le projet est un processus continu qui prendra en compte les modifications, les ajouts et les raffinements nécessaires au fur et à mesure de l'avancement du projet. Ce document sert de base, mais il évoluera pour refléter les besoins changeants, les nouvelles données et les objectifs du projet qui se développeront au cours des prochains jalons. Il demeurera un document dynamique, ajusté en fonction des retours d'expérience et des évolutions de l'environnement opérationnel, garantissant ainsi que le modèle de données reste en phase avec les exigences actuelles et nouvelles du projet.

2 Présentation

2.1 Mise en contexte

Le présent projet s'inscrit dans le cadre du cours IFT723 et vise à mettre en pratique l'ensemble des compétences acquises au cours de cette formation. L'objectif principal est de suivre le cycle de vie typique de l'élaboration d'une base de données temporalisée (BDT). Notre équipe démarre une démarche qui englobe la conception initiale d'une base de données non temporalisée (BD), le développement de son interface programmatique (API), sa transformation en une BDT intégrant la dimension temporelle, et enfin les tests de l'API de la BDT.

2.2 Système existant

Le système existant est une base de données non temporalisée conçue pour répondre aux besoins initiaux du projet. Cette base de données gère des informations. Cependant, avec l'évolution du projet vers une base de données temporalisée (BDT), des modifications sont nécessaires. Notre équipe doit adapter l'API au contexte bitemporel, impliquant une conception minutieuse et des tests approfondis. L'objectif est d'intégrer efficacement la dimension temporelle dans la structure existante, assurant ainsi la cohérence des données tout en répondant aux nouvelles exigences du projet.

2.3 Système envisagé

Le système envisagé représente une évolution significative de la base de données existante vers une architecture temporalisée plus complexe et sophistiquée. Cette nouvelle conception vise à intégrer pleinement la dimension temporelle dans la gestion des données, permettant une traçabilité précise des changements au fil du temps. Le système inclut une API adaptée au contexte bitemporel, offrant des fonctionnalités étendues pour interagir avec la base de données. L'objectif principal est d'améliorer la gestion des données en tenant compte des aspects temporels, tout en maintenant une interface programmatique robuste et une structure de base de données optimale.

3 Définition du problème

3.1 Présentation du problème

La présentation du problème met en évidence la nécessité d'améliorer le système en évoluant vers une base de données temporalisée. En analysant l'architecture actuelle, des défis liés à la traçabilité des changements et à la gestion temporelle ont été identifiés.

Cette observation motive l'optimisation du système en intégrant ces aspects. L'objectif du projet est de concevoir une base de données temporalisée, accompagnée d'une API adaptée, pour répondre de manière optimale aux exigences actuelles et futures de gestion des données.

3.2 Modélisation du problème

L'objectif de cette modélisation du problème implique la conception d'une base de données temporalisée visant à améliorer le système existant. L'approche vise à représenter de manière plus précise les données en tenant compte de la dimension temporelle, permettant ainsi une gestion efficace des changements et des historiques.

Ce processus nécessite une analyse approfondie des besoins spécifiques, la définition de structures de données adaptées, et l'élaboration d'une interface programmatique (API) capable de traiter les aspects temporels. L'objectif est d'aboutir à une modélisation optimale, offrant une solution robuste et cohérente pour la gestion temporelle des données.

4 Proposition d'une solution

4.1 Présentation de la solution

La solution proposée repose sur l'implémentation de triggers PostgreSQL (par exemple : IFT723_trig_etu.sql et IFT723_trig_adr.sql) pour la gestion des modifications sur plusieurs tables, entre autres Etudiant_Courante, Etudiant_Matricule_Validite, Adresse_Courante, Adresse_Code_Validite, etc.... Ces triggers enregistrent les opérations d'insertion, de mise à jour et de suppression dans des tables de log respectives, assurant ainsi la traçabilité des changements.

La mise en œuvre est accompagnée de scénarios de test (par exemple : IFT723_func_etu.sql et IFT723_func_adr.sql) visant à évaluer les fonctionnalités clés, telles que l'ajout, la modification et le retrait d'étudiants ainsi que la gestion des plages temporelles. Des scripts spécifiques illustrent ces scénarios pour chaque table, assurant une validation rigoureuse de la solution.

De plus, la structure des tables est décrite dans le script (par exemple : IFT723_cre_etu.sql et IFT723_cre_adr.sql), explicitant l'utilisation typique de chacune et soulignant l'importance des tables de log pour la préservation de l'historique des données.

Cette approche répond à la nécessité d'une base de données temporalisée tout en fournissant des procédures testées pour garantir la cohérence des données et la gestion efficace des périodes de validité.

5 Spécifications

Les exigences ou spécifications de notre projet de gestion des paiements des droits de scolarité sont essentielles pour garantir le succès de la mise en œuvre du système de base de données temporalisée.

Gestion des Informations des Étudiants et Accès Étudiant :

- Spécification 1 : Collecte des Informations Étudiantes

Le système doit permettre de collecter et stocker des informations personnelles, académiques et financières sur chaque étudiant, y compris des détails tels que les nom, prénom, adresse, coordonnées (adresse e-mail, numéro de téléphone), ainsi que des données académiques comme les cours suivis, les notes, les calendriers académiques, etc. Cela permet d'assurer une gestion centralisée des données étudiantes.

- Spécification 2 : Système de Contrôle d'Accès

Un système de contrôle d'accès basé sur les rôles doit être mis en place pour garantir la protection des données sensibles des étudiants. Les utilisateurs du système auront différents niveaux d'accès en fonction de leur rôle, assurant ainsi la confidentialité et la sécurité des informations.

- Spécification 3 : Accès en Ligne

Le système doit offrir un accès en ligne aux informations académiques, aux factures et aux données financières, permettant ainsi aux étudiants d'accéder facilement à leurs informations et de gérer leurs paiements.

- Spécification 4 : Personnalisation et Adéquation aux Règlements

Le système doit permettre la personnalisation pour répondre aux besoins spécifiques de l'institution éducative tout en restant conforme aux réglementations et normes de paiement locales et nationales.

- Spécification 4 : Intégration avec d'autres Systèmes

Le système doit être capable de s'intégrer avec d'autres systèmes utilisés par l'université, assurant une gestion financière cohérente et une synchronisation des données.

Facturation et Paiements :

- Spécification 5 : Création de Factures

Le système doit être en mesure de générer automatiquement des factures de droits de scolarité, incluant des informations détaillées sur les frais, les dates d'échéance, et les options de paiement. Les factures seront disponibles au format électronique, réduisant ainsi le besoin de documents papier.

- Spécification 6 : Paiements en Ligne

Le système doit prendre en charge des paiements en ligne sécurisés via différentes méthodes, y compris les cartes de crédit, les virements bancaires, etc. Il doit également être capable de s'intégrer avec des systèmes de paiement tiers si nécessaire.

- Spécification 7 : Plans de Paiement

Le système doit permettre la configuration de plans de paiement flexibles pour les étudiants, avec la possibilité de définir des échéances de paiement et des montants. Les étudiants doivent avoir la flexibilité nécessaire pour gérer leurs paiements.

- Spécification 8 : Notifications de Paiement

Le système doit envoyer des notifications automatiques aux étudiants pour rappeler les dates d'échéance des paiements, les confirmations de paiement réussi, ainsi que les rappels en cas de paiement en retard.

- Spécification 9 : Gestion des Remboursements

Le système doit permettre la gestion des demandes de remboursement et des processus associés, ainsi que le suivi des remboursements et des crédits.

- Spécification 10 : Gestion des Intérêts

Les intérêts doivent être calculés de manière quotidienne après la date limite, en prenant en compte la temporalité précise. Il ne doit pas y avoir d'intérêts négatifs.

Gestion Administrative :

- Spécification 11 : Gestion des Paiements et des Informations Étudiantes

Le système doit offrir un tableau de bord administratif convivial où le personnel administratif peut gérer les comptes étudiants de manière efficace. Ce tableau de bord doit permettre la recherche rapide d'informations sur les paiements, la mise à jour des données étudiantes, la création de factures, et le suivi des paiements en attente.

- Spécification 12 : Suivi des Paiements, des Dettes et des Remboursements

Le système doit permettre un suivi précis des paiements, des transactions en cours, des dettes et des remboursements pour une gestion financière efficace.

- Spécification 13 : Suivi des Modifications

Le système doit conserver un historique de toutes les modifications apportées aux données, y compris les changements d'adresse, les moyens de contact, les changements à la facture, les annulations de paiements, les changements à la fréquence des notifications, etc. Cette fonctionnalité permet une traçabilité complète.

- Spécification 14 : Évolutivité

Le système doit être évolutif pour répondre aux besoins croissants de l'institution éducative.

- Spécification 15 : Notifications Automatisées

Le système doit envoyer des notifications automatisées aux étudiants pour rappeler les échéances de paiement et les mises à jour académiques.

Gestion des Erreurs :

- Spécification 16 : Gestion des Erreurs Financières

Le système doit être en mesure de gérer les paiements en double, les paiements en retard et les erreurs de facturation. Il doit permettre aux utilisateurs de signaler et de résoudre ces erreurs de manière efficace.

- Spécification 17 : Tests et Qualité

Une phase de tests et de contrôle de la qualité doit être prévue pour garantir le bon fonctionnement du système, notamment en ce qui concerne les transactions financières et la gestion des données étudiantes.

- Spécification 18 : Formation et Support

Le système doit inclure des programmes de formation pour le personnel administratif et le personnel de support. De plus, un système de support en ligne via chat, e-mail ou téléphone doit être mis en place pour répondre aux questions des étudiants et résoudre les problèmes liés aux paiements.

- Spécification 19 : Gestion des Versions et des Mises à Jour

Le système doit permettre la gestion des versions et des mises à jour pour assurer son évolution et son amélioration continue en fonction des besoins changeants de l'institution éducative.

- Spécification 20 : Performances

Le système doit être conçu pour offrir de bonnes performances, même lors de périodes de charge élevée, assurant ainsi une expérience utilisateur fluide.

- Spécification 21 : Conformité Réglementaire

Le système doit être conforme aux réglementations en vigueur en matière de protection des données, de paiements en ligne et d'autres exigences pertinentes. Les mécanismes de sécurité mis en place doivent garantir la confidentialité et la conformité.

- Spécification 22 : Documentation Complète

Il est essentiel de créer une documentation complète, y compris des manuels d'utilisation, des spécifications techniques, des politiques de sécurité, et d'autres informations pertinentes, pour faciliter la gestion, la formation et le support du système.

Les spécifications, en relation avec nos visions et besoins antérieurs, permettent de mieux comprendre comment le système répondra aux exigences spécifiques. Cela nous donne donc une image complète des fonctionnalités et des objectifs du système de gestion des paiements des droits de scolarité.

5.1 Scénarios de déploiement et d'exploitation

5.2 Exigences fonctionnelles

Compte tenu du temps imparti, voici les exigences fonctionnelles spécifiques formulées auquel le système doit répondre :

Gestion des Étudiants :

- Le système doit permettre l'ajout d'étudiants avec des informations telles que le matricule, le nom, le prénom, les contacts, l'adresse, etc.
- Les opérations de mise à jour des informations des étudiants doivent être tracées et enregistrées dans des tables de log dédiées.

- La suppression d'un étudiant doit également être enregistrée pour assurer la traçabilité des changements.
- Les scénarios de test doivent couvrir les fonctionnalités clés telles que l'ajout, la modification et la suppression d'étudiants, ainsi que la gestion des plages temporelles.

Gestion des Adresses :

- Le système doit prendre en charge l'ajout, la modification et la suppression d'adresses avec des triggers spécifiques pour enregistrer ces opérations dans des tables de log associées.
- Des fonctions SQL doivent être disponibles pour regrouper les plages temporelles, ajouter des adresses dans différentes tables de validité, et modifier/retirer des adresses de la table courante.
- La création du schéma et des tables associées, avec des contraintes d'intégrité et des indexes, doit être réalisée par un script dédié.
- Des scénarios de test détaillés doivent être fournis pour chaque type d'opération liée aux adresses, accompagnés de scripts pour assurer la validation de la solution.

Ces exigences fonctionnelles définissent les fonctionnalités principales que le système doit fournir en termes de gestion des étudiants et des adresses, en mettant l'accent sur la traçabilité des changements, la cohérence des données et la gestion des périodes de validité.

6 Conclusion

En conclusion, la solution proposée offre une approche complète et bien documentée pour la gestion des étudiants et des adresses au sein du système ÉMIR. Les exigences fonctionnelles énoncées établissent un cadre robuste pour assurer la traçabilité, la cohérence des données et la validation des opérations.

L'utilisation de triggers PostgreSQL pour la gestion des modifications et la journalisation des opérations, associée à des scénarios de test détaillés, renforce la fiabilité de la solution. La conception du schéma, les contraintes d'intégrité, les indexes, ainsi que les fonctions SQL dédiées démontrent une approche méthodique pour garantir l'efficacité opérationnelle et les performances du système.

En somme, ce document établit les bases nécessaires pour le développement d'une solution temporalisée robuste répondant aux besoins spécifiques de la gestion des étudiants et des adresses dans le contexte de base de données temporalisée. La combinaison d'une architecture SQL bien pensée et de scénarios de test approfondis témoigne de l'engagement envers la qualité et la fonctionnalité du système à mettre en œuvre.