

Bases de données avancées

Projet : bourse d'échange

I) Modalités

Le projet est à faire en binômes. Une modélisation préliminaire (avec une brève explication) doit être rendue sur Moodle le 3 avril à 9h00. Elle fera l'objet d'une présoutenance les 4 et 9 avril. L'inscription aux présoutenances se fera sur Moodle et celle-ci est obligatoire. Il s'agit de s'assurer que vous avez commencé votre travail et ne suivrez pas une mauvaise direction. Elle fera partie de la notation. Le projet sera à rendre en fin de semestre, les modalités définitives vous seront communiquées sur Moodle.

II) Présentation du sujet

On veut ici modéliser la revente des billets de spectacle par l'intermédiaire d'un site du type "Bourse aux billets". Pour les besoins du projet on a simplifié certaines règles de gestion.

a) Fonctionnement

La bourse aux billets permet aux spectateurs de revendre des places de spectacle à un prix inférieur ou égal à leur prix d'origine. Les spectateurs peuvent mettre en vente une ou plusieurs places, ou en acheter à un revendeur, moyennant des frais de gestion. La transaction se fait en quatre étapes. Le vendeur propose à la vente des places. Le client achète le billet au site qui en conserve le montant en dépôt. Le vendeur transmet le billet à l'acheteur. Enfin le spectateur indique qu'il a reçu le billet. C'est lors de cette dernière étape que le vendeur est crédité du montant de la vente.

Le gestionnaire du site (il s'agit d'un théâtre, ou d'une salle de spectacles) fournit la liste des spectacles : le titre, la salle (s'il y en a plusieurs dans le théâtre), les horaires, et le prix des places par catégorie.

b) Inscription

Afin de pouvoir effectuer une vente ou un achat de place, l'utilisateur doit être inscrit. On peut cependant consulter les places à vendre sans être inscrit. L'utilisateur doit donner son nom, prénom, un nom de login, ainsi qu'un mot de passe. (Il n'est pas nécessaire de gérer les mots de passe dans le cadre de ce projet.)

Nous ne traiterons pas les transactions bancaires dans ce projet mais chaque utilisateur aura un solde associé à son compte. Vous pouvez supposer que chaque utilisateur dispose de 100 Euros sur son compte lorsqu'il s'inscrit.

c) Mise en vente d'un billet

Un utilisateur inscrit peut mettre en vente des places. Il doit préciser le spectacle, la date, la catégorie, les numéros de place, et le prix de revente. Les places proposées dans une annonce sont toutes de la même catégorie et sont consécutives (par exemple, 3 places au 2e balcon, côté impair, rangée T, places 21-25). Le prix de revente doit être inférieur ou égal au prix du billet. Lorsqu'il revend plusieurs places, il peut préciser si les places doivent toutes être achetées par le même acheteur (en bloc) ou si les places peuvent être rachetées individuellement.

Les billets mis en vente et non vendus sont retirés automatiquement une heure après le début de la représentation.

Un vendeur peut opter pour la mise en avant de son annonce, moyennant des frais de 10 Euros. Lorsqu'une annonce est mise en avant, elle est affichée avant les autres annonces. Un vendeur est limité à la revente de 10 places par annonce, et à 10 annonces dans une période de 1 mois.

Vous pouvez imaginer la mise en place d'un système de notation/satisfaction.

d) Achat

L'utilisateur peut obtenir la liste des places mises en vente pour un spectacle donné, ou pour un intervalle de dates données. L'utilisateur peut faire une proposition d'achat en précisant le numéro de l'annonce et le nombre de places sélectionnées. Le montant des billets, plus des frais de gestion de 5 Euros, est débité de son compte et placé provisoirement sur le compte de la bourse aux billets.

Lorsqu'un utilisateur effectue un achat, le vendeur est notifié par un message. Il a alors 24 heures pour lui envoyer le billet, sans quoi l'achat sera annulé et l'acheteur remboursé en totalité (y compris les frais de transaction).

Vous pouvez implémenter la messagerie avec une table de messages et une vue paramétrée sur le login de l'utilisateur de sorte que chaque utilisateur voit uniquement les messages qui lui sont adressés. Lorsque l'utilisateur reçoit le billet, il informe la bourse aux billets et le montant de l'achat (moins les frais de transaction) est transféré au vendeur.

e) Les alertes

Lorsqu'un utilisateur ne trouve pas de places pour le spectacle de son choix, il peut définir une alerte. Lorsqu'un billet qui satisfait aux critères de l'alerte est mis en vente, l'utilisateur reçoit un message l'informant qu'un billet est disponible. Une alerte porte sur un spectacle, une date, et un nombre de places.

III) Modélisation et tables

Pour donner un ordre de grandeur de ce qui est attendu, le nombre de tables résultantes ne devrait pas dépasser 12, mais peut être plus bas suivant les choix de modélisation et les simplifications entre le modèle et l'implémentation. Identifiez un maximum de contraintes d'intégrité au moment de la modélisation et précisez comment chacune sera gérée. Intégrez un maximum de contraintes lors de la création des tables, des triggers pour les contraintes complexes ainsi que des fonction psql pour simplifier les opérations de gestion et les tests.

Les tables devront être peuplées de données pertinentes permettant d’apprécier les résultats des requêtes — volume indicatif : le nombre total de lignes (somme sur toutes les tables) pourra être de l’ordre de 100. Portez une attention particulière aux cas particuliers.

IV) Fonctionnalités attendues

Vous devez implémenter les triggers nécessaires au maintien de la cohérence de la base et au respect des règles de gestion, conserver un historique des opérations pour permettre de traiter les réclamations.

Les situations ci-dessous devront donner lieu à une alerte automatique : l’abandon d’une transaction pour délai dépassé, l’accusé de réception donné par un client non concerné, les vendeurs qui atteignent leur limite mensuelle, l’apparition d’une vente attendue par un acheteur etc ...

Vous devrez pouvoir exécuter les opérations suivantes, soit par une instruction simple soit par une fonction quand c’est nécessaire, parfois un `PREPARE` suffit.

- inscrire un usager
- créer une annonce
- chercher un billet
- acheter un billet
- définir une alerte
- faire avancer le temps
- transférer les fonds
- ...

Dans tous les cas, vous penserez à utiliser une table qui contiendra une seule date, elle correspondra à une date courante fictive, que vous utiliserez à la place de `CURRENT_DATE`. Ceci vous permettra et nous permettra de tester vos fonctionnalités.

V) Présoutenance

D’une durée courte (une dizaine de minute) elle devra montrer que vous avez réfléchi en amont à tous les aspects du sujet, et que vous vous êtes organisés pour la phase de développement.

- La discussion aura pour support votre diagramme entité/relation qui devra déjà être soigné et quasiment définitif. Vous pourrez le faire sur papier et le scanner. Attention en effet aux outils que vous seriez tenté d’utiliser, ils ne respectent pas tous la convention fixée en cours.
- Si votre diagramme est très complet, vous devrez avoir fixé précisément la partie sur laquelle vous allez ensuite vous concentrer. Elle devra être suffisamment riche pour vous permettre de démontrer (lors de la soutenance finale) votre maîtrise des problématiques de cette année.
- Enfin vous devrez avoir établi la liste de toutes les contraintes, et décidé de comment elles seront gérées : (NOT NULL, CHECK, trigger, par construction, ...) ainsi que les grandes thématiques de vos tests.
- Ces éléments peuvent tenir sur environ 3 pages, C’est ce petit rapport qu’il vous faudra déposer sur Moodle. Le nom du fichier sera `rapport_XXX_YYY.pdf` en précisant vos noms.

VI) Soutenance Finale

La soutenance se déroulera par binôme, mais la notation et les questions pourront être individualisés. Vous devrez **chacun** maîtriser l'ensemble de ce qui est présenté, quelle que soit la façon dont vous vous êtes réparti le travail.

Vous pourrez être amené à modifier une partie de ce que vous avez fait pour que nous puissions apprécier votre réactivité.

Soyez prêt, c'est à dire ayez de quoi faire une démonstration complète, avec des scripts préparés à l'avance qui montrent chaque règle de gestion implémentée.

Nous devrions passer un minimum de temps au clavier, enchaîner rapidement d'une question à une autre, retrouver des lignes de codes très rapidement etc... Cela n'est possible que si vous avez pensé à tous les cas de figure, conservé et ordonné vos propres tests.

Pour vous aider, voici une liste de documents à rendre sur Moodle. Vous la déposerez sous la forme d'une archive dont le nom sera la concaténation de vos deux noms.

- Un rapport étendant celui de la présoutenance. Il contiendra à présent la description sql de votre schéma dans la base, et des considérations générale sur le développement.
- Une liste commentée des fonctions, des triggers, des règles de gestion ;
- La liste et la nature des index qui seront éventuellement ajoutés ainsi qu'une justification des bénéfices attendus ;
- Un fichier séparé permettant de distinguer clairement l'étapes de création des tables de la base appelé `create_All` ;
- Un fichier qui créera les triggers appelé `create_Trigger` ;
- Un fichier `insert_Data` qui permettra de peupler les tables
- Un fichier pour chaque scenario. Chacun permettra de tester comment se comporte votre base de données sur chacune des fonctionnalités, vérifier le déclenchement de chaque trigger, etc. Utilisez des commandes `prompt` et `echo` pour qu'on voie ce qu'il se passe. Chaque fichier s'appellera `test_XXX` ;
- Un fichier appelé README qui décrira brièvement le rôle de chacun des autres fichiers et éventuellement des petits détails permettant de démarrer.