Mise en œuvre d'une application de gestion de contrôle d'accès

1 Organisation du projet

Le projet "administration de bases de données" a lieu du 10/01 au 1/02 2013. Il doit être réalisé par des groupes de quatre personnes. Les salles affectées à cette activité sont indiquées sur le planning ADE.

Des permanences sont organisées par les enseignants pour chacune des demi-journées consacrées au projet: en général de 9H à 11H le matin et de 15H à 17H l'après-midi (variable selon les jours). Un pointage des gens présents sera réalisé pour chaque demi-journée.
Des ressources et un forum sont disponibles sur Moodle IM^{2}AG. Les différents rapports intermédiaires et l'archive contenant le rapport final et les sources de l'application prêtes à l'emploi (scripts SQL, déclencheurs et programmes Java, scripts de lancement, tests automatisés) devront être déposés sur Moodle.

L'évaluation portera sur le travail réalisé et sur les rapports de projet: la présentation des documents (clarté, expression, orthographe), l'état d'avancement de l'application réalisée, la qualité de programmation (lisibilité, justification, pertinence et couverture des tests), ainsi que la soutenance (présentation et réponses aux questions). En outre, la prise de recul de chaque groupe par rapport à ses réussites et à ses lacunes sera également appréciée par les enseignants.

Les différents rapports devront fournir des réponses claires et synthétiques aux questions de l'énoncé, une présentation de la solution aux problèmes à traiter avec des jeux de tests complets. Le rapport final devra en plus présenter la mise en œuvre des solutions proposées, un bilan sur le projet et le travail effectué, ainsi que le code Java commenté.

2 Objectifs du projet

Le projet consiste en la réalisation d'une application cliente Java accédant via le protocole JDBC à une base de données sur un serveur distant. Plusieurs clients peuvent accéder à la base de manière concurrente. L'accent est mis sur le maintien de la cohérence des données et en aucun cas sur l'interface homme-machine: une interface en mode texte offrant un minimum d'interactivité (menus, affichage des résultats, trace de la gestion des transactions concurrentes, etc.) est demandée. L'évaluation ne tiendra pas compte de la présence d'une interface graphique avancée mais uniquement de la facilité à suivre les tâches effectuées pour mener à bien une transaction et à vérifier le bon fonctionnement de l'application.

Les technologies utilisées sont: Java et JDBC pour la partie applicative, Oracle (SQL, PL/SQL) pour la partie base de données.

L'objectif visé est une meilleure compréhension des rôles respectifs de ces technologies lors de leur utilisation dans la perspective de l'implantation d'une application de gestion de données.

3 Position du problème

On désire mettre en place un **système de contrôle d'accès** sur un ensemble de bâtiments pour en assurer la sécurité.
Les **utilisateurs** ont des statuts divers et peuvent y accéder selon des règles différentes; ce sont des étudiants, des personnels administratifs, des techniciens, des enseignants et des chercheurs et aussi, plus rarement, des visiteurs.
Les étudiants peuvent accéder librement aux salles de cours et de TD entre 7h30 et 20h30 et ont un accès contrôlé aux salles de TP. Les personnels administratifs, les techniciens accèdent selon des règles diverses (heures et groupe de bâtiments) en fonction de leur rattachement (laboratoire, UFR...) et de leurs responsabilités. Il en est de même pour les enseignants et les chercheurs.
En fait les **autorisations** sont données sous contrôle du superviseur, en définissant des **groupes de personnes** dont les autorisations (périodes de l'année, jours et heures) sont homogènes pour le groupe par rapport à un ensemble des bâtiments du site.
Enfin les visiteurs occasionnels soit accèdent librement aux salles de cours et de TD dans leurs plages d'ouverture publiques, soit reçoivent un badge provisoire d'accès à un groupe pour la journée en échange d'une pièce d'identité.
L'ensemble du site dont les accès sont contrôlés est découpé en une série de "**bâtiments**"; chaque bâtiment est géré suivant ses propres règles d'accès en fonction des règles d'utilisation définies. Chaque bâtiment a une vocation unique (recherche pour un laboratoire, cours, td, tp, administration, publique) et les règles sont établies par un superviseur qui administre l'ensemble des autorisations.

Chaque bâtiment est identifié par un code de 3 ou 7 caractères alphabétiques reflétant la structure à laquelle il appartient (IM2AG, IMA, ISTG....), complété d'une lettre indiquant un sous-ensemble connexe (IM2AG-A, IM2AG-B ....) et (ENSIMAG-D3), éventuellement un chiffre indiquant un étage. Chaque bâtiment, au sens d'une unité d'accès homogène, comporte un ou plusieurs **points d'accès.** Un point d'accès permet d'accéder à un bâtiment ou d'en sortir, en ayant à cet endroit un point de contrôle qui permet de vérifier que seules entrent ou sortent les personnes autorisées. Certains points d'accès sont seulement des entrées ou des sorties (sens de passage unique), d'autres sont à double sens.
Aux points d'accès un **lecteur de badge,** avant la porte permet de valider l'accès "par badge" et de débloquer la porte, deux capteurs permettent de vérifier le sens de passage. Un mécanisme adéquat permet de vérifier qu'une seule personne passe à la fois, que la porte se referme dans un délai raisonnable (3 secondes après passage et rebloquage 10 secondes après une demande d'accès non suivie d'un passage). **Un afficheur** permet de faire connaître aux utilisateurs, si le point d'accès est actif ou fermé, si leur badge a été validé ou pas, si la porte a été rebloquée (jeux de couleurs ou son).

La cohérence du système exige qu'un utilisateur ne puisse évidemment pas être enregistré comme étant sorti d'un bâtiment sans y être entré, ni entré dans un bâtiment alors qu'il est censé être déjà entré dans un autre, ou ce même, bâtiment sans en être sorti. De même, il ne peut "badger" en sortie d'un bâtiment, que s'il y est entré. S'il cherche à violer ces règles, une alerte est déclenchée, et un incident est enregistré. En cas de passages multiples ou en sens contraire du sens défini au point utilisé, une alarme est transmise au poste de gardiennage pour l'alerter sur la situation et un incident est enregistré.
Un point d'accès peut aussi être à l'interface entre deux bâtiments (dans un sens de passage ou dans les deux sens). L'accès doit alors être conforme à la logique d'autorisation de chacun des deux bâtiments. Un point d'accès est identifié par le bâtiment auquel il appartient et un numéro dit "numéro de porte". Pour un point interface, ce numéro est associé à un des deux bâtiments et suivi de la lettre I, et le deuxième bâtiment lié au point d'accès étant connu par la gestion de la relation de la porte entre les bâtiments.

Les personnes concernées par le système de contrôle sont de 3 types: superviseur, administrateur de groupes et gardien.
Le **superviseur** a pour rôle de gérer l'ensemble de la sécurité en définissant son fonctionnement. Il ***décrit donc l'ensemble des bâtiments*** et de leurs points d'accès, ***enregistre les différents groupes de personnes*** et ***établit les autorisations qui sont associées*** aux groupes de personnes après discussion avec les responsables des différents secteurs.
Les personnes peuvent ***être enregistrées dans le système par l'administrateur*** et associées à divers groupes. **L'administrateur de groupe,** autorisé par le superviseur, assure la gestion des groupes d'un secteur organisationnel; c'est par exemple, le rôle du service scolarité pour les étudiants lors des inscriptions, du personnel d'accueil général, du secrétariat d'un laboratoire ou d'un bâtiment pour certains visiteurs et pour les enseignants ou chercheurs du labo.
Le (ou les) gardien(s) ***sur)veille(nt) le fonctionnement continu du système, enregistre(nt) les alarmes en les commente(nt***, ***les traite(nt), alerte(nt) le superviseur*** pour les cas graves. Ils font appel aux services compétents lors d'incidents techniques ou de sécurité prévus. Si une trace vidéo des événements est conservée pour certains points d'accès, ils ont la possibilité d'en provoquer la sauvegarde en cas d'incident suite au déclenchement d'une alarme.

L'Accès à un bâtiment suit la procédure suivante:

- un utilisateur présente son badge au lecteur, puis quand il y est autorisé, passe en ouvrant la porte, à moins qu'elle ne s'ouvre automatiquement,
- le lecteur affiche son état d'activité, lit le badge, transmet les infos au système de contrôle, affiche le résultat de la validation suivant les infos retransmises par le système de contrôle
- le système de contrôle analyse et valide l'autorisation, envoie les signaux adéquats pour l'affichage, pour débloquer la porte, la fermer et la rebloquer, ou envoyer les alarmes significatives dans les diverses situations d'erreur.

Une description factuelle des différentes entités a déjà été validée:

- Les utilisateurs sont des **personnes** connues au minimum par leur nom, leur prénom et **identifiées par un numéro de code** lié au système d'accès. Pour les personnels permanents, on enregistre en outre, le bâtiment et numéro de leur bureau, leur numéro de téléphone professionnel ainsi que leur adresse électronique. Pour les étudiants, on enregistre leur numéro de carte d'étudiant, ainsi que leur filière et leur année et leur adresse électronique. Pour les visiteurs, on enregistre le type et le numéro d'un papier d'identité, ses dates et lieux de délivrance.
- Un **badge** comporte **un numéro** qui l'identifie. On enregistre lorsqu'on le donne à quelqu'un, le propriétaire et la date-heure de remise. Un badge peut être de type permanent ou provisoire. Si le badge est permanent, il n'y a pas de date de fin de validité; si il est provisoire, il comporte une date-heure de fin de validité. Les badges remis aux étudiants sont considérés comme des badges permanents, les autorisations évolueront avec les groupes d'appartenance (annulation d'autorisation durant l'été par exemple).
- Les **bâtiments** ont un code constitué de 3 à 7 lettres majuscules indiquant le secteur d'appartenance (ISTG, IM2AG, ENSIMAG, ADMUJF....) suivi d'une lettre précisant un bâtiment dans un ensemble (IM2AG-A,B,...,F; PHYS- A, B,...E; DLST-A, B,...F, etc.) et de l'étage éventuel.
- Un **point d'accès** est rattaché à un bâtiment et porte un numéro de 3 chiffres, le premier indiquant l'étage, les deux suivants, le numéro à cet étage. Chaque **lecteur** est identifié par le point d'accès qu'il dessert, suivi d'un rôle, E pour externe et I pour interne. Un point d'accès entre deux bâtiments est identifié de manière privilégiée par rapport à l'un des deux bâtiments ("accès principal"), le deuxième étant lié comme "accès secondaire". Chaque **point d'accès** possède un type d'accès autorisé : entrée, sortie, entrée-sortie, interface à double sens, à sens unique (entrée dans le bâtiment principal ou sortie du

bâtiment principal). Les portes peuvent être dans divers états (actif, en panne, en attente, bloquée, débloquée, ouverte, fermée) mais sont indissociables du point d'accès dont elles constituent des interfaces.

- **Une autorisation d'accès** est donnée pour **un groupe de personnes** à un **groupe de bâtiment** sur des périodes de plusieurs semaines consécutives pour lesquelles on définit un régime pour les jours ouvrés (avec éventuellement une heure de début de fin, valable pour tous les jours) et un autre pour les jours fériés, c'est la notion de **période**. Certaines autorisations peuvent ne pas avoir de limites horaires ou de jours fériés. Il peut y avoir des périodes de fermeture complète des bâtiments.
- Chaque personne est rattachée à un ou plusieurs groupes, son **droit d'accès** étant alors la configuration la plus souple c'est-à-dire celle correspondant à l'union des autorisations des groupes auxquels elle appartient. Les droits de la personne sont liés à son badge. Une personne possède un unique badge actif pour le système. Les références et les journaux d'activité des cartes qu'elle peut avoir perdues dans le passé, ou qui peuvent lui avoir été volés, sont archivés dans le système mais les autorisations associées la concernant sont annulées pour ces badges.

Le fonctionnement du système suit **les contraintes** suivantes sur les parties de bâtiment contrôlées :

- une personne présente dans un bâtiment pour y être entrée après lecture de son badge, ne peut demander d'accès ici ou ailleurs, et peut seulement sortir du bâtiment par une porte autorisée.
- une personne ne peut sortir si son entrée n'a pas été enregistrée, sauf dans les bâtiments qui passent en utilisation contrôlée après avoir été publics. Dans ce cas, si la personne possède l'autorisation d'accès pour la plage horaire correspondante, elle est autorisée à sortir. Sinon le gardien sera chargé de contrôler la sortie et débloquera la porte.
- Un point d'accès doit être utilisé dans le sens associé à l'autorisation demandée par badge et ne concerne qu'une personne à la fois.

Le système permet aussi **la réservation de salle d'un bâtiment** donné par l'administrateur de groupe. En cas de demande de réservation d'une salle en cours de réservation au même moment pour un autre groupe, le second administrateur devra être prévenu que sa réservation peut échouer si la première réservation est validée.

4 Schémas des relations

Après analyse des besoins, un modèle logique relationnel du schéma de cette base de données a été proposé. Ce modèle décrit formellement chaque relation, le domaine des attributs et les contraintes d'intégrité référentielle.

Ce schéma « métier » ne doit pas être modifié ni étendu. Cependant, il est envisageable d'utiliser, en complément, quelques tables « outils », par exemple pour la gestion de files d'attente.

ACCES (NUMACCES, ID_PERSONNE)
{<n, p> ∈ ACCES ⇔ la personne identifiée par le numéro *p* est associée à l'accès identifié par le numéro *n.}*

ADMINISTRATEUR (ID_GROUPEPERS, ID_PERSONNE)
{<g, p< ∈ ADMINISTRATEUR ⇔ La personne identifiée par le numéro *p* a un rôle d'administrateur pour le groupe identifié par le numéro *g.}*

AFFECTATION (ID_PERSONNE, NUMBADGE, DATE_AFFECTATION, DATE_FIN_AFFECTATION)
{<p, b, d,df> ∈ AFFECTATION ⇔ la personne identifiée par le numéro *p* possède le badge de numéro *b* de la date *d* à *df* si ce badge n'est pas permanent*.}*

ALARME (NUMALARME, ETAT_ALARME, TYPE_ALARM, REF_VIDEO)
{<a, e, t, v> ∈ ALARME ⇔ l'alarme de numéro *a* est dans l'état *e*, de type *t* et la vidéo associée est *v.}*

ALARME_ACCES (NUMALARME, NUMACCES)
{<a, n> ∈ ALARME_ACCES ⇔ l'alarme de numéro *a* est déclenchée par l'accès de numéro *n.}*

AUTORISATION (ID_GROUPEBAT, ID_GROUPEPERS, LIBELLE_PLAGE_ACCES)
{<b, g, a> ∈ AUTORISATION ⇔ le groupe de numéro *g* est autorisé à accéder au groupe de bâtiments de numéro *b* dans la plage d'accès *a.}*

BADGE (NUMBADGE, ETATBADGE)
{<b, e> ∈ BADGE ⇔ l'état du badge de numéro *b* est *e*.}

BATIMENT (CODE_BATIMENT, ID_GROUPEBAT, ADRESSE)
{<g, b, a> ∈ BATIMENT ⇔ le bâtiment de code *b* dont l'adresse est *a* est attaché au groupe de bâtiments de numéro *g*.}

ENTREE (NUMACCES, CODE_BATIMENT, CODEPOINTACCES, ETAT_ENTREE, DATE_ENTREE)
{<a, b, c, e, d> ∈ ENTREE ⇔ Un accès de numéro *a* s'est produit en entrant dans le bâtiment *b* au point d'accès *c* à la date *d*. L'état de l'entrée est *e*.}

GROUPE (ID GROUPEPERS, NOMGROUPEPERS)
{<n, g> ∈ GROUPE ⇔ le groupe de personnes de numéro *n* est libellé *g*.}

GROUPE_BATIMENTS (ID GROUPEBAT, NOMGROUPEBAT)
{<n, g> ∈ GROUPE_BATIMENT ⇔ le groupe de bâtiments de numéro *n* est libellé *g*.}

JOUR_FERIE (DATE_FERIE)
{<d> ∈ JOUR_FERIE ⇔ *d* est un jour férié pendant lequel tout bâtiment est inaccessible, à moins de disposer de droits spécifiques.}

MEMBRE (ID GROUPEPERS, ID PERSONNE)
{<g, p> ∈ MEMBRE ⇔ la personne de numéro *p* appartient au groupe *g*.}

PERIODE_ACCES (LIBELLE_PLAGE_ACCES, LIBELLE_PLAGE_HORAIRE, FERIE, OUVRE)
{<p, h, f, o> ∈ PERIODE ACCES ⇔ La plage d'accès *p* est définie par la plage horaire *h*. Elle peut être définie pour un jour ouvrable et/ou un jour férié.}

PERSONNE (ID_PERSONNE, NOM, PRENOM, DATE_NAISSANCE, LIEU_NAISSANCE, BUREAU)
{<p, n, np, d, n, b> ∈ PERSONNE ⇔ la personne de numéro *p* porte le nom *n*, le prénom principal *np*. Elle est née le *d* à *n*. Elle est située dans le bureau *b* si elle fait partie du personnel.}

PLAGE (LIBELLE_PLAGE_SEMAINE, LIBELLE_PLAGE_ACCES)
{<s, p> ∈ PLAGE ⇔ la plage d'accès *p* est attachée à la plage de semaines *s*.}

PLAGE_HORAIRE (LIBELLE_PLAGE_HORAIRE, HORAIRE_DEBUT, HORAIRE_FIN)
{<h, d, f> ∈ PLAGE_HORAIRE ⇔ la plage horaire *h* est définie de l'horaire *d* à *f*.}

PLAGE_SEMAINE (LIBELLE_PLAGE SEMAINE, SEMAINE_DEBUT, SEMAINE_FIN)
{<p, d, f> ∈ PLAGE_SEMAINE ⇔ la plage de semaines *s* débute la semaine *s* et se termine la semaine *f*.}

POINT_ACCES (CODEPOINTACCES, TYPE_ACCES)
{<c, t> ∈ POINT_ACCES ⇔ le point d'accès *c* est de type *t*.}

POINT_ACCES_BATIMENT (CODE_BATIMENT, CODEPOINTACCES)
{<g, b, c> ∈ POINT_ACCES BATIMENT ⇔ le bâtiment *b* possède le point d'accès *c*.}

RESERVATION (ID_GROUPEPERS, CODE_BATIMENT, NUMERO_SALLE, LIBELLE_PLAGE_SEMAINE, LIBELLE_PLAGE_HORAIRE, DATE_RESA, JOUR_SEMAINE)
{<gp, b, s, ps, ph, r, j> ∈ RESERVATION ⇔ la salle *s* dans le bâtiment *b* est réservée pour le groupe de personnes *gp* dans la plage de semaine *ps* à l'horaire *ph* le jour de la semaine *j*. Cette réservation a été exécutée à la date *r*.}

SALLE (CODE_BATIMENT, NUMERO_SALLE, TYPE_SALLE, CAPACITE)
{<b, s, t, c> ∈ SALLE ⇔ la salle *s* dans le bâtiment *b* est de type *t* et a une capacité de *c* places.}

SORTIE (NUMACCES, CODEPOINTACCES, CODE_BATIMENT, ETAT_SORTIE, DATE_SORTIE)
{<a, c, b, e, d> *2* SORTIE ⇔ un accès de numéro *a* s'est produit en sortant du bâtiment *b* au point d'accès *c* à la date *d*. L'état de la sortie est *e*.}

Les contraintes de domaine suivantes sont indicatives et peuvent être compléter/modifier au besoin.

dom(ID_PERSONNE,ID_GROUPEPERS,ID_GROUPEBAT)=entier
dom(NUMACCES,NUMBADGE,NUMALARME) = entier
dom(DATE_NAISSANCE,DATE_AFFECTATION,DATE_FIN_AFFECTATION)= date
dom(NOM,PRENOM,LIEU NAISSANCE)= chaine de caractères
dom(NOMGROUPEPERS,TYPE,NOMGROUPEBAT,NOMGROUPEPERS)= chaine de caractères
dom(LIBELLE_PLAGE_ACCES)= chaine de caractères
dom(ETATBADGE, ETAT_ALARME, ETAT_ENTREE, ETAT_SORTIE)=*{* 'ENABLE', 'DISABLE' *}*

dom(REF VIDEO)= chaine de 10 caractères
dom(NOMGROUPEBAT,CODE_BATIMENT,ADRESSE)= chaine de caractères
dom(CODEPOINTACCES,BUREAU)= chaine de caractères
dom(DATE_ENTREE, DATE_SORTIE, DATE_FERIE,DATE_RESA)= date
dom(LIBELLE_PLAGE_ACCES,LIBELLE_PLAGE_HORAIRE)= chaine de caractères
dom(HORAIRE_DEBUT,HORAIRE_FIN,SEMAINE_DEBUT,SEMAINE_FIN)= entier
dom(FERIE,OUVRE)= booleen
dom(LIBELLE_PLAGE_SEMAINE)= chaine de caractères
dom(NUMERO_SALLE,JOUR SEMAINE,CAPACITE)= entier
dom(TYPE_SALLE)={ 'CM','TD','TP','REUNION', 'BUREAU', 'AUTRE' *}*
dom(TYPE_ACCES)={ 'ENTER', 'EXIT', 'ENTEREXIT' *}*
dom(TYPE_ALARM)=*{* 'Passage Multiple', 'Pas sorti', 'Pas entré', 'Ouverture Porte trop long', ... *}*

ACCES[ID_PERSONNE] ⊂ PERSONNE[ID_PERSONNE]
ADMINISTRATEUR[ID_GROUPEPERS] ⊂ GROUPE[ID_GROUPEPERS]
ADMINISTRATEUR[ID_PERSONNE] ⊂ PERSONNE[ID_PERSONNE]
AFFECTATION_BADGE[ID_PERSONNE] ⊂ PERSONNE[ID_PERSONNE]
AFFECTATION_BADGE[NUMBADGE] ⊂ BADGE[NUMBADGE]
ALARME_ACCES[NUMALARME] ⊂ ALARME[NUMALARME]
ALARME_ACCES[NUMACCES] ⊂ ACCES[NUMACCES]
AUTORISATION[ID_GROUPEBAT] ⊂ GROUPE BATIMENTS[ID_GROUPEBAT]
AUTORISATION[ID_GROUPEPERS] ⊂ GROUPE[ID_GROUPEPERS]
AUTORISATION[LIBELLE PLAGE ACCES] ⊂ PERIODE ACCES[LIBELLE PLAGE ACCES]
BATIMENT[ID_GROUPEBAT] ⊂ GROUPE_BATIMENTS[ID_GROUPEBAT]
ENTREE[NUMACCES] ⊂ ACCES[NUMACCES]
ENTREE[CODE_BATIMENT] ⊂ BATIMENT[CODE_BATIMENT]
ENTREE[CODEPOINTACCES] ⊂ POINT_ACCES[CODEPOINTACCES]
MEMBRE[ID_GROUPEPERS] ⊂ GROUPE[ID_GROUPEPERS]
MEMBRE[ID_PERSONNE] ⊂ PERSONNE[ID_PERSONNE]
PERIODE_ACCES[LIBELLE_PLAGE_HORAIRE] ⊂ PLAGE_HORAIRE[LIBELLE_PLAGE_HORAIRE]
PLAGE[LIBELLE PLAGE_SEMAINE] ⊂ PLAGE_SEMAINE[LIBELLE_PLAGE_SEMAINE]
PLAGE[LIBELLE PLAGE_ACCES] ⊂ PERIODE_ACCES[LIBELLE_PLAGE_ACCES]
POINT_ACCES BATIMENT[CODE BATIMENT] ⊂ BATIMENT[CODE BATIMENT]
POINT_ACCES BATIMENT[CODEPOINTACCES] ⊂ POINT_ACCES[CODEPOINTACCES]
RESERVATION[ID_GROUPEPERS] ⊂ GROUPE[ID_GROUPEPERS]
RESERVATION[CODE_BATIMENT, NUMERO_SALLE] ⊂ SALLE[CODE_BATIMENT, NUMERO_SALLE]
RESERVATION[LIBELLE_PLAGE_SEMAINE] ⊂ PLAGE_SEMAINE[LIBELLE_PLAGE_SEMAINE]
RESERVATION[LIBELLE_PLAGE_HORAIRE] ⊂ PLAGE_HORAIRE[LIBELLE_PLAGE_HORAIRE]
SALLE[CODE_BATIMENT] ⊂ BATIMENT[CODE_BATIMENT]
SORTIE[NUMACCES] ⊂ ACCES[NUMACCES]
SORTIE[CODEPOINTACCES] ⊂ POINT ACCES[CODEPOINTACCES]
SORTIE[CODE_BATIMENT] ⊂ BATIMENT[CODE_BATIMENT]

5 Cahier des charges de l'application

La liste ci-dessous énumère les fonctionnalités essentielles que doit proposer l'application.

- **Administration locale du système** (rôle administrateur). Les opérations des administrateurs doivent répondre à des contraintes très fortes pour éviter les incohérences dans les droits d'accès. Plusieurs administrateurs d'un même groupe peuvent effectuer des accès concurrents à la base de données.
    1. **Gestion des personnes et des badges**. Des badges sont créés et affectés à des personnes. Lors de la création d'une personne, un badge lui est obligatoirement affecté. Un nouveau badge peut être affecté à une personne (en cas de perte). Une personne ne peut posséder qu'un seul badge actif. Un badge peut être désactivé. Une personne sans badge actif est supprimée. La suppression d'une personne désactive son badge actif.
    2. **Gestion des plages d'accès d'un groupe de bâtiments**. Chaque groupe de bâtiments est associé à un groupe qui porte le même nom que le groupe de bâtiments. Ce groupe un peu particulier (ce n'est pas un groupe de personne même s'il est représenté comme tel) permet d'affecter une plage horaire d'ouverture générale pour un groupe de bâtiments. Cela impose donc des contraintes sur l'affectation des plages d'accès aux groupes d'utilisateurs pour ce groupe de bâtiments. En effet, la plage horaire autorisée pour un groupe d'utilisateurs au sein d'un bâtiment ne doit pas être plus permissive que celle définie par ce "méta-groupe" correspondant.

3. **Gestion des plages d'accès pour les groupes**. Il s'agit d'affecter des plages d'accès à certains groupes de bâtiments pour un groupe d'utilisateurs (ces plages d'accès doivent être compatibles avec les contraintes définies pour les groupes de bâtiments concernés). Si la définition d'autorisation d'un groupe englobe des jours fériés, un message d'alerte (au niveau applicatif) doit résumer ces jours fériés. Par ailleurs, un message d'alerte doit également être affiché si le groupe de bâtiments n'est pas accessible pendant l'intégralité de la plage d'accès choisie.

- **Réservation de salles** (rôle administrateur). Les administrateurs peuvent réserver des salles dans les groupes de bâtiments dont ils ont la gestion. Une réservation est faite pour un groupe. Différentes réservations ne peuvent coexister que si elles sont cohérentes entre elles.
  1. La salle doit être réservable (contrainte d'ouverture du groupe de bâtiments, d'autorisation du groupe). De nouvelles réservations de salles sont impossibles un jour férié.
  2. La salle à réserver ne doit pas être déjà réservée sur la même plage.
  3. Il faut toujours conserver une salle libre de chaque type pour les semaines futures (contrainte levée pour la semaine courante).
- **Administration globale du système** (rôle superviseur). Un superviseur possède les mêmes droits qu'un administrateur. Il possède en plus le contrôle sur les objets de la base (bâtiment, groupe de bâtiments, groupe, salle)[1].
  1. **Ajout, modification, suppression de bâtiment, de groupe de bâtiments, de groupe, de salles**. Chacune de ces opérations peut avoir un impact très important sur les autorisations actives. La stratégie à dérouler sera différente selon les cas. Au niveau d'un bâtiment, sa suppression entraine la suppression de ses salles et de leurs réservations. Pour un groupe de bâtiments, sa suppression annulera toutes les autorisations et toutes les réservations, mais on laissera les bâtiments intacts (au superviseur de les réaffecter à un autre groupe de bâtiment). Pour un groupe d'utilisateurs, on autorisera sa suppression uniquement s'il ne possède plus aucun membre. Pour une salle, sa destruction entrainera une opération de relogement des réservations sur d'autres salles libres et compatibles dans la mesure du possible (sinon suppression des réservations impossibles). Un bâtiment ne peut changer de groupe de bâtiments que si les réservations futures de salle dans ce bâtiment sont compatibles avec les droits d'accès des demandeurs. Si des conflits entre ces opérations apparaissent, l'annulation de l'opération sera toujours préférée à un autre choix.
  2. **Ajout, modification, suppression de jours fériés**. Seul un superviseur a la possibilité de modifier les jours fériés. L'apparition d'un jour férié à une date où se trouvent des réservations n'a pas d'implication particulière (les réservations restent mais les salles seront inaccessibles pour certaines personnes).
- **Passage des points d'accès** (rôle utilisateur). Le passage d'un badge dans un point d'accès produit différentes réactions selon la situation.
  1. **Passage normal**. La porte s'ouvre si la personne est autorisée (en tenant compte de toutes les contraintes d'accès).
  2. **Archivage d'un passage**. Tout passage (entrée ou sortie) est archivé pendant 48h, puis supprimé si aucune alarme n'est associée à ce passage. **Le système global étant simulé, la notion d'avancement dans le temps sera simulée par la modification d'une variable au niveau applicatif.**
  3. **Génération d'alarme**. Un passage peut déclencher une alarme dans les conditions énoncées par le cahier des charges. Le blocage du point d'accès génère un message au concierge du bâtiment (sous la forme d'affichage textuel dans notre application). Une fonction de l'application devra simuler le déblocage du point d'accès par le concierge.
- **Archivages** (rôle système). Les passages à chaque point d'accès sont archivés (et conservés pendant 48h). Les alarmes sont archivées de manière définitive.

[1] Dans le cadre du projet, il n'est pas demandé de contrôler les droit d'accès au SGBD pour les différents rôles. On supposera notamment qu'un administrateur ne manipule que les informations qu'il est habilité à gérer.

6 Analyse des contraintes d'intégrité

Les contraintes énoncées dans les sections précédentes se décomposent en trois catégories :

1. les contraintes simples sur les données qui sont exprimables dans la description du schéma de la base de données (schéma SQL),
2. les contraintes complexes sur les données qui ne sont pas exprimables directement dans le schéma de la base mais à l'aide de déclencheurs disponibles dans une BD dynamique,
3. les contraintes applicatives qui sont gérées au sein de l'application.

Question 1: Cette question donnera lieu à un rapport intermédiaire.
Donner la liste des contraintes d'intégrité, spécifiées et déduites des sections 3 et 4, en précisant pour chaque contrainte si elle sera gérée au niveau du SGBD ou de l'application.

Question 2: Cette question donnera lieu à un rapport intermédiaire.

Donner le script SQL (Oracle) qui implémente le schéma de la base de données énoncé en section 4. Vous peuplerez la base à l'aide d'un jeu d'essai judicieusement choisi (et commenté). Vous prendrez un grand soin dans le choix des types pour les dates et les durées.

7 Gestion de l'aspect transactionnel

Question 3: Cette question donnera lieu à un rapport intermédiaire.
La mise en œuvre des fonctionnalités énoncées dans la section 5 demande une décomposition de ces fonctionnalités en unités transactionnelles. Il faut pour cela:

1. identifier correctement les transactions,
2. choisir le niveau d'isolation pertinent pour chaque transaction,
3. développer un ensemble de scénarios de tests pour valider le bon fonctionnement d'une fonctionnalité en mode d'accès concurrent.

L'énoncé des transactions pourra se faire dans un pseudo langage mêlant algorithme (simplifiée) et SQL. L'objectif est de donner l'idée des tâches réalisées par chaque transaction.
L'intégralité des transactions devra apparaître dans ce document. Seules des corrections mineures pourront être acceptées dans le rapport final.

8 Réalisation de l'application

Cette partie donnera lieu à un rapport final qui fera la synthèse de votre travail sur ce projet.

Vous devrez réaliser une application Java qui met en œuvre les fonctionnalités énoncées dans la section 5.

L'exécution de l'application doit faire apparaître des traces démontrant la prise en compte des différents types de contraintes d'intégrité et la gestion correcte des transactions (en mode d'accès concurrent), et cela en limitant les saisies manuelles : le jour de la démo on évitera, dans la mesure du possible, de perdre du temps à devoir saisir bêtement de nombreuses données pour montrer un point important.

Chaque opération devra réaliser une trace permettant de faciliter au maximum la visualisation de la bonne marche de l'opération (contraintes prises en compte, transactions exécutées). Au contraire, le développement d'une interface utilisateur riche n'est absolument pas demandé, l'application doit juste proposer le minimum pour faciliter la saisie des informations et fluidifier le déroulement de vos scénarios lors de la soutenance.

Evidemment, plusieurs opérations concurrentes doivent être possibles sans provoquer d'incohérence et en favorisant la concurrence d'accès.

Ayant identifié les contraintes d'intégrité à la charge du SGBD, vous réaliserez les déclencheurs correspondants sous la forme de triggers Oracle. Des jeux d'essais complets devront valider la gestion des contraintes par le SGBD.

Réaliser en Java les opérations énoncées dans la section 5 en considérant les déclencheurs actifs au niveau de la base Oracle.Vous mettrez en évidence la bonne gestion des transactions dans un contexte d'accès concurrents.

9 Rapport final et soutenance

Le document (au format PDF) doit répondre aux différentes questions dans un français ou anglais correct.

Question 4: Rapport final. Vous devrez faire l'effort d'être à la fois précis et concis. Le contenu du rapport doit reprendre les points suivants:

- Rappeler les objectifs du projet ainsi que l'organisation du document dans une brève introduction. Décrire les opérations de l'application (explications, code commenté, protocoles de tests).
- Faire le lien avec le contenu des rapports intermédiaires afin de mettre en valeur l'implémentation des contraintes et de la gestion concurrente. Faire notamment apparaître les éventuelles corrections apportées par rapport aux spécifications des rapports intermédiaires (gestion des contraintes et des transactions)
- Décrire les opérations de l'application (explications, code commenté, protocoles de tests).

Soutenance. Vous aurez environ 15 minutes (sur 30) à votre disposition pour nous faire une démonstration de votre application. Tous les membres d'un groupe sont conviés à la soutenance (sous réserve d'absence justifiée). Vous emploierez ce temps à:

- Réaliser la démonstration de votre application en laissant libre cours à vos talents de présentation. Le jury se laisse le droit d'interrompre la démonstration pour utiliser son propre jeu de tests et poser des questions. Il est très important de préparer des scripts pour éviter la saisie manuelle des données pour illustrer des cas complexes (l'interaction utilisateur n'est pas au centre de ce projet).
- Répondre aux questions posées à chacun des membres du groupe. Une pondération de la note globale pourra être envisagée à cette occasion.

10 Dates importantes

Afin de contrôler l'avancement du travail au fur et à mesure du projet, des documents intermédiaires devront être fournis aux enseignants à différentes étapes du projet.
Les dates clés du planning sont les suivantes :

- **Jeudi 10 janvier :** lancement du projet
- **Mercredi 15 janvier à 20h :** remise d'un rapport (d'environ 5 pages) résumant l'analyse de toutes les contraintes d'intégrité du cahier des charges. Pour chaque contrainte, indiquer à quel(s) niveau(x) elle sera gérée : schéma, déclencheur(s), application.
- **jeudi 16 janvier à 20h :** remise du script SQL implémentant la création du schéma de la base de données, avec prise en compte de toutes les contraintes gérées à ce niveau.

- **Mardi 23 janvier à 20h :** remise d'un rapport (d'environ 5 pages) décrivant les différentes séquences transactionnelles à mettre en œuvre au niveau de l'application. Pour chaque séquence, décrire :
    - le niveau d'isolation SQL correspondant
    - les conditions de validation (commit), d'annulation (rollback) et éventuellement d'annulation partielle (rollback to savepoint), en fonction des contraintes concernées.
    - des scénarios de concurrence d'accès (entrelacement de transactions) représentatifs du bon fonctionnement de l'application.

- **Jeudi 31 janvier à 20h :** remise du rapport final de projet (environ 20 pages). Il n'est pas nécessaire d'inclure les listings du code du projet en annexe (cependant, des courts extraits représentatifs de code peuvent éventuellement être intégrés dans le corps du rapport si cela facilite certaines explications). Le rapport doit contenir le descriptif des scénarios de test/démonstration qui seront présentés lors de la soutenance.
- **Vendredi 1er février à 20h :**
    - Remise du code complet du projet (fichiers Java et SQL). Ce code devra être suffisamment commenté pour être compréhensible par des personnes extérieures à votre groupe de projet. Le code sera accompagné d'un fichier README précisant (i) les scripts à lancer pour créer la base, la peupler et initialiser les déclencheurs et (ii) les commandes de compilation et de lancement pour l'application java.
    - Remise des jeux de tests automatisés (en SQL et en Java).
- **Lundi 4er février**: Soutenances