

# Rennequinepolis (RQS)

Version du 24 septembre 2015 07:07

UE Techniques Avancées de Gestion des données  
AA Organisation et exploitation des données 4

3ème Bachelier en Informatique de Gestion

Laurence Herbiet <laurence.herbiet@hepl.be>  
Ludovic Kutty <ludovic.kutty@hepl.be>

2015 – 2016

## Table des matières

---

<b>1</b>	<b>Préambule</b>	<b>4</b>
<b>2</b>	<b>Contexte général</b>	<b>4</b>
2.1	Sites . . . . .	4
2.2	Intervenants . . . . .	5
2.3	Les films . . . . .	5
2.4	Les acteurs et réalisateurs . . . . .	7
2.5	Les images . . . . .	8
<b>3</b>	<b>Consignes générales de développement</b>	<b>8</b>
3.1	Personnalisation des fonctionnalités . . . . .	8
3.2	Gestion des erreurs . . . . .	9
3.3	Couche ORM ou pas ? . . . . .	9
3.4	Choix du langage de programmation . . . . .	10
3.5	Pré-requis aux évaluations . . . . .	10
<b>4</b>	<b>Réalisations attendues</b>	<b>10</b>
4.1	Partie I - Sauvegarde et restauration 1 . . . . .	10
4.1.1	Script SQL initial de création de CB et CBB - CreaCBLight . . . . .	10
4.1.2	Sauvegarde de CB 1 - BackupCBLight . . . . .	10
4.1.3	Restauration de CB 1 - RestoreCBLight . . . . .	11
4.2	Partie II - Création et alimentation de CB . . . . .	11
4.2.1	Script SQL de création de CB et CBB - CreaCB . . . . .	11
4.2.2	Alimentation de CB - AlimCB . . . . .	12
4.3	Partie III - Evaluations de films . . . . .	13
4.3.1	Recherche d'un film - RechFilm . . . . .	13
4.3.2	Cotes et avis manuels sur les films - EvalFilm . . . . .	14
4.3.3	Cotes et avis automatiques sur les films - EvalFilmAuto . . . . .	14
4.3.4	Affichage des meilleurs films - Palmares . . . . .	14
4.4	Partie IV - Sauvegarde et restauration 2 . . . . .	15
4.4.1	Simulation d'un plantage de CB - CrashCB . . . . .	15
4.4.2	Sauvegarde des informations de CB - BackupCB . . . . .	16
4.4.3	Restauration des informations de CB - RestoreCB . . . . .	16
4.5	Partie V - Création et alimentation de CC . . . . .	16
4.5.1	Script de création de CC - CreaCC . . . . .	16
4.5.2	Alimentation de CC - AlimCC . . . . .	16
4.5.3	Retour des Copies - RetourCopies . . . . .	17
4.6	Partie VI - Programmation et commande de places . . . . .	18
4.6.1	Programmation des films - ProgFilms . . . . .	18
4.6.2	Archivage des programmations - ArchProg . . . . .	18
4.6.3	Recherches de places libres - RechPlaces . . . . .	18
4.6.4	Commandes de places - CmdPlaces . . . . .	19
4.6.5	Génération de tickets - Tickets . . . . .	20
4.7	Partie VII - Les acteurs . . . . .	20
4.7.1	Détail des signalétiques d'acteurs - FicheActeurs . . . . .	20
4.8	Partie VIII - Service Marketing . . . . .	21
4.8.1	Création du modèle dimensionnel des données - CreaMDD . . . . .	21
4.8.2	Collecte d'indicateurs de production - CollectData . . . . .	21
4.8.3	Traitement des indicateurs - LoadData . . . . .	22

4.8.4	Consultation des tableaux de bord - ConsultData . . . . .	22
4.9	Partie IX - Administration de RQS . . . . .	22
4.9.1	Dossier Installation, mise en route, exploitation - AdminConfig . . . . .	22
4.9.2	Maintenance, mise à jour - AdminMaintenance . . . . .	23
4.9.3	Sauvegarde et récupération - AdminSvg . . . . .	23
<b>5</b>	<b>Organisation pratique et évaluations</b>	<b>23</b>
5.1	Téléversement du code avant évaluation . . . . .	23
5.2	Tableau synthétique . . . . .	24
5.3	Modalités d'évaluation du laboratoire . . . . .	24
5.3.1	Premier semestre et Evaluation continue . . . . .	24
5.3.2	Deuxième session . . . . .	25
5.4	Synthèse des modalités d'évaluation pour le cours d'Organisation et Exploitation des données (OED) . . . . .	26
5.4.1	Théorie . . . . .	26
5.4.2	Labo . . . . .	26
5.4.3	Cote globale . . . . .	26

## 1 Préambule

---

Vous trouverez dans ce document le contexte général du système informatique qui devra être réalisé. On y détaille ensuite les différentes fonctionnalités demandées. Ce document peut donc être considéré comme votre "contrat de travail". Il permettra de mieux appréhender l'organisation du travail de l'année et sa planification au travers des différentes semaines.

Les modalités d'évaluation se trouvent à la fin de cet énoncé reprenant les explications quant à la construction de la cote finale. Il convient d'en prendre connaissance et de signer cette prise de connaissance. Elles vous engagent pour toute la durée de l'année scolaire.

Toutes les ressources (énoncé, code source, fichiers textes, ...) sont disponibles sur la page web du cours à <https://cours.khi.be/sgbd3/> et dans le centre de ressources de L. Herbiet sur l'école virtuelle (EV). Donc lorsqu'il est fait référence dans le texte à un fichier précis en indiquant son nom mais pas son URL, vous pouvez aller le chercher sur la page web en question.

## 2 Contexte général

---

Rennequinopolis (RQS) est une société qui distribue des films et qui gère les complexes cinématographiques qui projettent ces films. L'objectif de ce travail est d'informatiser leur système actuel et de construire les différentes plate-formes web de consultation, d'achats et de gestion qu'ils souhaitent ainsi que de réaliser les transferts d'informations nécessaires.

Une première analyse a permis de distinguer un premier lot d'informations : les intervenants, les sites et les objets manipulés que sont les films et les copies. Sur base de cela un circuit fonctionnel et informationnel a pu être établi. Celui-ci est indiqué à la figure 1, page 27.

Nous insistons sur le fait que cette section permet d'établir le contexte dans lequel vous devrez réaliser les fonctionnalités qui vous sont demandées (et uniquement celles-là). Le contexte peut contenir plus d'informations que ce dont vous aurez pratiquement besoin pour réaliser les fonctionnalités demandées. Référez-vous au tableau des fonctionnalités demandée à la section 5.2, page 24, pour savoir ce que vous devez exactement réaliser et allez lire les fonctionnalités en question pour savoir ce dont vous avez besoin exactement.

### 2.1 Sites

Rennequinopolis est organisée en divers services, localisés à Bruxelles, en plus des complexes disséminés au travers du pays. Elle s'adresse à différents fournisseurs pour obtenir ses médias et en générer des copies qu'elle peut ensuite distribuer dans les différents cinémas.

- **La centrale internationale - CI** - Cette centrale est un entrepôt où sont disponibles toutes les oeuvres cinématographiques que pourraient souhaiter les entreprises comme RQS. Elle tient lieu de fournisseur principal de l'entreprise.
- **La centrale belge - CB** - Il s'agit de notre entreprise RQS. Située à Bruxelles, l'entreprise y dispose d'un entrepôt où elle peut stocker les copies avant qu'elles ne soient distribuées dans les cinémas. On y trouve également les serveurs principaux de RQS.
- **La centrale alternative belge - CBB** - Site de secours de RQS, il est capable de prendre le relais en cas de panne du site principal. Il propose les mêmes services que CB.
- **Le cinéma - CC** - Ce sont les différents complexes cinématographiques que gère RQS. Dans le cadre de cet énoncé il n'y aura qu'un cinéma : CC.
- **Les personnes - BP** - C'est un répertoire des acteurs et réalisateurs des films de CI reprenant des informations plus complètes à leur sujet.

- **Le fournisseur de Ticket - TCKT** - C'est un sous-traitant proposant des services de génération automatique de ticket aux complexes de cinémas
- **Le centre d'analyse - DW/MKT** - C'est là que sont prises toutes les décisions financières de l'entreprise. Elle dispose pour cela de rapports et d'interfaces lui permettant de prendre les décisions les plus judicieuses en fonction des indicateurs reçus.

## 2.2 Intervenants

Liste des intervenants provisoirement relevés :

- Les **opérateurs de RQS** - Divers gestionnaires ont la charge de la mise à jour du catalogue de films.
- Les **gestionnaires des cinémas** - Ils sont responsables des opérations courantes des complexes dont l'alimentation en nouveaux films.
- Les **fournisseurs** - Ils sont représentés par la centrale internationale (et pour certaines données par la banque de données des personnes). Il n'y a donc qu'un seul fournisseur qui est considéré comme implicite.
- Les **utilisateurs** - Les utilisateurs ont la possibilité de coter des films et de donner leur avis écrit sur un film.
- Les **directeurs de RQS** consultent régulièrement les données collectées auprès des différents sites de l'entreprise afin d'adapter la stratégie de celle-ci pour obtenir de meilleurs profits.

## 2.3 Les films

La signalétique d'un film comporte toutes les informations relatives à ce film. Par exemple, son titre, sa durée, ses genres, le public concerné, les acteurs principaux, une affiche, une ou plusieurs langues parlées, .... Chaque film est une ligne (un enregistrement) stocké dans un fichier texte appelé `movies.txt`<sup>1</sup>. On retrouve 21 champs par film expliqués au tableau 1.

Le format d'un enregistrement de la table externe est le suivant :

```
_id@|@title@|@original_title@|@release_date@|@status@|@vote_average@|@vote_count@|
@runtime@|@certification@|@poster_path@|@budget@|@revenue@|@homepage@|@tagline@|@overview@
|@genres[[id,,name]]@|@directors[[id,,name,,profile_path]]@|@actors[[id,,name,,cast_id,,
character,,profile_path]]@|@production_companies[[id,,name]]@|@production_countries[[
iso_3166_1,,name]]@|@spoken_languages[[iso_639_1,,name]]
```

Certains champs peuvent être absent ce qui est représenté par deux séparateurs consécutifs @|@@|@.

Les séparateurs utilisés pour chaque enregistrement (ou ligne du fichier) texte sont les suivants :

- @|@ est utilisé pour séparer les champs qui sont au niveau le plus haut de la hiérarchie (top-level).
- [[ et ]] sont utilisés comme délimiteurs d'un tableau. Le nom indiqué devant le délimiteur de début de tableau est le nom du champ contenant le tableau comme par exemple `genres`, `directors` ou `actors`.
- || est utilisé comme séparateur d'objets dans un tableau. Par objets, on entend les éléments du tableau. Comme ceux-ci contiennent des informations additionnelles, on les qualifie d'objets même si cela n'a aucun rapport avec la POO.
- ,, est utilisé pour séparer les champs d'un objet dans un tableau.

Pour le film *Inception* (id 27205)<sup>2</sup>, on a l'enregistrement ci-dessous dans le fichier texte `movies.txt` :

<sup>1</sup><https://cours.khi.be/sghd3/labo/movies.rar>

<sup>2</sup>L'enregistrement peut-être obtenu à l'aide d'un grep avec expression régulière, option -P pour choisir les expressions régulières PCRE à la Perl : `grep -P "^27205@" movies.txt`.

27205@|@Inception@|@Inception@|@2010-07-16@|@Released@|@7.5@|@5578@|@148@|@PG-13@|@/  
 tAXARVreJnWfoANIHASmgYk4SB0.jpg@|@1600000000@|@825532764@|@http://inceptionmovie.warnerbros.com/@|@Your mind is  
 the scene of the crime.@|@Cobb, a skilled thief who commits corporate espionage by infiltrating the  
 subconscious of his targets is offered a chance to regain his old life as payment for a task considered to be  
 impossible: "inception", the implantation of another person's idea into a target's subconscious.@|@[[28,,  
 Action|12,,Adventure|9648,,Mystery|878,,Science Fiction|53,,Thriller]]@|@[[525,,Christopher Nolan,,/7  
 OGmFDF4VHLLgbjxuEwTj3ga0uQ.jpg]]@|@[[6193,,Leonardo DiCaprio,,1,,Cobb,,/mNRMgj7K5ztvkSqrCdwEYNZIS1M.jpg  
 |24045,,Joseph Gordon-Levitt,,3,,Arthur,,/zSuXCR6xCKIgo0gWldp8moM1H3I.jpg|27578,,Ellen Page,,5,,Ariadne,,/  
 paexwBfm1VyZva7q4XgcBdqowmL.jpg|2524,,Tom Hardy,,7,,Eames,,/aYWIz7yWOFzyK7B1HHCQj2nq8s4.jpg|3899,,Ken  
 Watanabe,,2,,Saito,,/edm2ZInBtzjVB3p8AbkKTW2I1nt.jpg|2037,,Cillian Murphy,,8,,Robert Fischer,,/  
 tEERcmKFHXypcwKp28MxVfvf6Ph.jpg|8293,,Marion Cotillard,,4,,Mal,,/m1zdHPPhxCBCSDm8b9MRWonbyYrq.jpg|3895,,  
 Michael Caine,,6,,Miles,,/n1IjMLp9zvVM2eFm77Uhi7s1nW.jpg|95697,,Dileep Rao,,19,,Yusuf,,/  
 f9xQkvWw5guOePcCkxksoi09w12.jpg|13022,,Tom Berenger,,9,,Browning,,/ieHokBikODTVskOLwQ71Az88LYK.jpg|4935,,  
 Pete Postlethwaite,,22,,Maurice Fischer,,/o1W1V3jUKP0HvXbduvYjXM5tnaF.jpg|526,,Lukas Haas,,10,,Nash,,/1  
 J9ca3TFCzK1ywmacqMnVBdu7X2.jpg|66441,,Talulah Riley,,89,,Blonde,,/cmA8krd4hpPC9kWSF9wpSx6ffMt.jpg]]@|@  
 [[6194,,Warner Bros.||923,,Legendary Pictures||9996,,Syncopy]]@|@[[US,,United States of America]]@|@[[en,,  
 English||ja🇯🇵,,||fr,,Français]]

TAB. 1: Informations à propos d'un film

Nom	Explications
actors	Un tableau d'acteurs. Chaque acteur contient les informations suivantes: <b>id</b> , <b>name</b> , <b>cast_id</b> , <b>character</b> (le nom de leur personnage) et <b>profile_path</b> qui contient un chemin permettant de construire l'URL vers une image de la personne. Nous sommes exclusivement intéressés par <b>id</b> , <b>name</b> et <b>character</b> .
budget	Le coût du film en dollars.
certification	Le rating qui indique le public cible du film. Les abréviations sont tirées de la notation proposée par l'association MPAA <sup>3</sup> .
directors	Un tableau de réalisateurs. Chaque réalisateur contient les informations suivantes: <b>id</b> , <b>name</b> et <b>profile_path</b> qui contient un chemin permettant de construire l'URL vers une image de la personne. Nous sommes exclusivement intéressés par <b>id</b> et <b>name</b> .
genres	Les genres dont fait partie le film. Un tableau d'objets dont chacun possède les informations suivantes: <b>id</b> qui est l'identifiant du genre sous forme d'un nombre entier et <b>name</b> qui est le nom du genre comme "Action", "Comedy" ou "Thriller".
homepage	Page web du film.
id	L'identifiant TMDb du film.
original_title	Le titre original du film. Pour rappel, les données sont encodées en UTF-8. Typiquement il s'agit du titre dans la langue d'origine du film.

Suite sur la page suivante

<sup>3</sup><http://www.mpa.org/film-ratings/>

Tab. 1 – continuation de la page précédente

Nom	Explications
<code>overview</code>	Un résumé du film en UTF-8. Ce résumé peut être écrit dans une autre langue que l'anglais.
<code>poster_path</code>	Un chemin permettant de construire l'URL vers une image de type poster pour le film. La manière de procéder est indiquée à la section 2.5, page 8.
<code>production_companies</code>	Un tableau des entreprises ayant participé à la production du film. Pour chaque entreprise on a un objet avec les champs suivants : <code>id</code> et <code>name</code> .
<code>production_countries</code>	Un tableau des pays ayant participé à la production du film. Pour chaque pays on a un objet avec les champs suivants: <code>iso_3166_1</code> et <code>name</code> . Le premier pays indiqué est considéré comme étant le pays d'origine. Inutile de reprendre les autres.
<code>release_date</code>	La date de sortie du film au format YYYY-MM-DD.
<code>revenue</code>	Le revenu qu'a généré le film en dollars.
<code>runtime</code>	La durée du film en minutes.
<code>spoken_languages</code>	Un tableau d'objets représentant les langues parlées dans le film. Chaque objet possède les champs suivants : un code <code>iso_639_1</code> pour identifier la langue (comme "de" ou "en") et <code>name</code> qui est le nom de la langue (comme "German" ou "English").
<code>status</code>	L'état de production du film.
<code>tagline</code>	Le texte qui accompagne le film et qui est souvent écrit sur la pochette du film.
<code>title</code>	Le titre du film. Pour rappel, les données sont encodées en UTF-8.
<code>vote_average</code>	La cote qu'a obtenu le film suite aux votes des membres TMDb.
<code>vote_count</code>	Le nombre de votes TMDb qui a servi à établir le <code>vote_average</code> .

## 2.4 Les acteurs et réalisateurs

Une fiche signalétique d'acteur ou de réalisateur possède les informations utiles permettant d'identifier la personne mais également de le lier aux différents films auxquels il a participé. Les informations détaillées relatives à chaque acteur et réalisateur sont obtenues auprès de la banque des personnes (BP). En effet, la centrale internationale ne propose que l'identifiant et le nom de la personne en information visible. Nous n'avons pas la date de naissance ou d'autres informations utiles dans CI contrairement à BP.

BP est représenté par la base de données MongoDB `people`. Chaque personne est stockée sous forme d'un document dans BP. Vous pouvez la reprendre sur le site web du cours.

Les informations qui vont nous intéresser sont les éléments `id`, `name`, `birthday` qui contient la date de naissance au format `YYYY-MM-DD`, `birthplace` et sa filmographie.

## 2.5 Les images

Les images des films et des personnes sont accessibles par HTTP en vous basant sur les informations disponibles ci-dessous. La propriété `images` de l'objet JSON que l'on peut reprendre<sup>4</sup> en accédant à `http://api.themoviedb.org/3/configuration` contient un objet avec différentes propriétés :

- `base_url` qui est l'URL de base à utiliser pour reprendre l'image d'un film ou d'une personne.
- `poster_sizes` pour disposer des tailles possibles des images de type poster de film. Vous prendrez une image par film dont la taille est `w185`.
- `backdrop_sizes` que vous n'utiliserez pas.
- `profile_sizes` pour disposer des tailles possibles des images de type photo de personne. Vous prendrez une image par personne dont la taille est `w185`.

L'URL pour reprendre une image est construite en concaténant dans l'ordre :

1. La `base_url`
2. La taille
3. Le chemin renseigné dans le document

Par exemple, pour le film 348, "Alien", on concatènera les chaînes de caractères suivantes :

- `http://image.tmdb.org/t/p/`
- `w185`
- `/uU9R1byS3USozpWJ5oz7YAkXyk.jpg`

pour avoir `http://image.tmdb.org/t/p/w185/uU9R1byS3USozpWJ5oz7YAkXyk.jpg`

Listing 1: Configuration

```
{"images":{"base_url":"http://image.tmdb.org/t/p/","secure_base_url":"https://image.tmdb.org/t/p/","backdrop_sizes":["w300","w780","w1280","original"],"logo_sizes":["w45","w92","w154","w185","w300","w500","original"],"poster_sizes":["w92","w154","w185","w342","w500","w780","original"],"profile_sizes":["w45","w185","h632","original"],"still_sizes":["w92","w185","w300","original"]},"change_keys":[...]}
```

## 3 Consignes générales de développement

Cette section vous donne des indications générales sur ce que vous pouvez faire, ce que vous ne pouvez pas faire, ce à quoi il faut apporter une attention particulière, etc. lorsque vous réaliserez les différentes fonctionnalités demandées à la section 4, page 10.

### 3.1 Personnalisation des fonctionnalités

Un certain nombre de points ont été simplifiés par rapport à la réalité. Certains points ont également été volontairement maintenus dans le vague ou l'obscur. Dans tous les cas, l'imagination est la bienvenue pour donner à vos solutions un cachet plus personnel, plus réaliste, plus professionnel. Cependant, pour vous

<sup>4</sup>Il est nécessaire de disposer d'une clé pour utiliser leur API. Une fois qu'on a cette clé, il suffit d'ajouter le paramètre `api_key` dans l'URL.



éviter de vous fourvoyer dans une impasse ou perdre votre temps à réaliser des options présentant peu d'intérêt, il vous est vivement recommandé de prendre conseil auprès de l'un de des professeurs concernés par le projet.

### 3.2 Gestion des erreurs

On vous demande d'utiliser les mécanismes de gestions des erreurs/exceptions mis à votre disposition dans le ou les langages de programmation utilisés pour le développement de vos applications ainsi que les erreurs générées par la base de données quel que soit le contexte (PL/SQL, JDBC, ...).

Une idée intéressante est de produire deux types de message en cas d'erreur : un message généraliste mais clair destiné à l'utilisateur lambda et un message stocké précis destiné au développeur ou à l'administrateur de la base de données indiquant exactement quelle est l'erreur et à quel endroit elle a eu lieu.

Dans tous les cas de figure, toute erreur liée à la base de données doit obligatoirement être enregistrée de manière détaillée dans une table.

Il est vivement conseillé d'utiliser un package ou une procédure indépendante pour gérer ces insertions dans une table de log des erreurs et les différentes exceptions pouvant se produire. De même, tous les messages d'exception transmis au programme utilisateur ou au serveur web proviendront de messages prédéfinis, stockés dans la base de données et associés à un code d'exception.

On s'attend à ce que les données enregistrées relatives à l'erreur contiennent au minimum la date complète (date + heure), l'endroit précis où a eu lieu l'erreur (package / procédure / déclencheur / ...), le numéro de l'erreur et le message d'erreur correspondant. Ces informations sont destinées à l'administrateur de la base.

Attention, qu'il est nécessaire de différencier les messages informationnels non liés à une erreur des messages liés à une erreur. Vous pouvez utiliser une colonne pour cela ou même une table différente.

*Techniques attendues* : transactions autonomes, séquences.

### 3.3 Couche ORM ou pas ?

Quel que soit le langage de programmation choisi ou le framework web choisi, nous vous demandons de n'utiliser aucun intermédiaire de type ORM (Object Relational Mapping) entre votre code et les bases de données Oracle comme Java Data Objects (JDO), Java Persistence API (JPA) dont Hibernate, ActiveRecord (Ruby), DataMapper (Ruby), Linq, etc. Par conséquent, il est nécessaire que vous écriviez vous-même les requêtes SQL que vous soumettez à votre BD.

Néanmoins on demande à ce que vous encapsuliez proprement le code d'accès aux données dans vos programmes de manière à bien séparer la partie "modèle" (le M de MVC) du reste du code. Dans cet esprit, on vous demande de veiller à privilégier l'utilisation de procédures stockées.

Le but de ces restrictions est de favoriser l'utilisation du langage SQL au lieu du langage ou système d'interrogation de votre ORM dont l'expressivité est nettement moindre. De plus l'utilisation d'une couche aussi générique ajoute inutilement son lot de complexité dans le cadre de ce projet. Cela se justifie pleinement dans un cours de base de données et non de programmation orientée objet.

Vous pouvez lire le court article<sup>5</sup> intitulé "What ORMs have taught me: just learn SQL" si vous désirez un complément d'information.

A vous de voir ce que propose votre langage de programmation pour accéder à une base de données Oracle. Nous en discutons dans la section suivante.

<sup>5</sup><http://wozniak.ca/what-orms-have-taught-me-just-learn-sql>

### 3.4 Choix du langage de programmation

Le choix du langage de programmation à utiliser se pose lors de la réalisation de certaines fonctionnalités.

Ce choix est libre mais nous attirons votre attention sur les points suivants :

- Le choix supporté par le cours est d'utiliser JDBC pour accéder à votre BD Oracle. Cela implique d'utiliser la JVM. Le langage supporté par le cours est Java. Notez cependant qu'il existe d'autres langages qui tournent sur la JVM.
- Un autre choix est d'utiliser ODP.NET avec le langage C#. Cette solution n'est pas supportée par le cours, c'est-à-dire qu'en cas de problèmes nous ne pourrions vous apporter qu'une aide limitée.
- Les autres choix ne sont pas supportés par le cours. Attention aux problèmes liés au manque de mises à jour et de documentation des implémentations d'accès à Oracle. Parfois elles n'existent tout simplement pas et il faut passer par un pont ODBC. Bien que ce soit une très bonne idée d'explorer d'autres langages que ceux enseignés au sein de l'école, la perte de temps qui peut résulter de ces choix nous pousse fortement à déconseiller leur utilisation. Néanmoins, nous désirons laisser le choix du langage libre.

Attention, pour certaines fonctionnalités, nous imposons l'appel de procédures stockées. On vous demande de prêter attention à l'existence d'un support pour l'utilisation des BLOBs qui sont obligatoires et des types tableau et des variables de type curseur (REF CURSOR).

### 3.5 Pré-requis aux évaluations

Tout test de fonctionnalité nécessite des tuples en suffisance pour permettre l'évaluation du bon comportement de la fonction. Nous attendons de votre part que ces données soient présentes avant les évaluations. Voire mieux : que des jeux de test et des scénarios de test aient été préparés pour faciliter et accélérer l'évaluation.

## 4 Réalisations attendues

### 4.1 Partie I - Sauvegarde et restauration 1

#### 4.1.1 Script SQL initial de création de CB et CBB - CreaCBLight

Dans un premier temps, vous allez réaliser les schémas relationnels de CB et CBB sans disposer des tables nécessaires pour stocker les informations des films. On désire simplement pouvoir stocker les informations des utilisateurs et leurs cotes et avis sur les films. Il n'y aura donc pas de FK définie à ce stade vers la table contenant les films puisque celle-ci n'existe pas.

On souhaite disposer de la date-heure de la cote ou de l'avis qu'un utilisateur a donné pour un film.

Notez que le schéma de CBB est la copie conforme de celui de CB. On vous demande de produire un script SQL propre disposant d'un maximum de contraintes pour garantir sa cohérence.

#### 4.1.2 Sauvegarde de CB 1 - BackupCBLight

Vous allez réaliser deux répliquions de manière à sauvegarder les informations de CB dans CBB :

- Une répliquion synchrone (déclencheurs) pour copier les cotes des utilisateurs et leurs avis.
- Une répliquion asynchrone chaque nuit pour copier les nouveaux utilisateurs créés durant la journée ainsi que leurs cotes et avis (qui n'ont pas été copiés de manière synchrone). Lorsque vous complétez

plus tard cette fonctionnalité pour réaliser BackupCB (section 4.4.2, page 16), vous ajouterez la sauvegarde des informations liées aux films. Nous testerons cela en invoquant directement votre procédure sans utiliser `DBMS_SCHEDULER`. Veillez à gérer correctement votre transaction.

*Techniques attendues* : SQL, PL/SQL, déclencheurs, jobs (`DBMS_SCHEDULER`), transactions, séquences, logs.

### 4.1.3 Restauration de CB 1 - RestoreCBLight

Lorsque CB n'est plus accessible, le système informatique utilise CBB ce qui implique que des utilisateurs peuvent être ajoutés et des cotes et/ou avis ajoutés ou modifiés (écrasement des anciennes valeurs) sur CBB. Une fois que la connectivité vers CB est restaurée, il est nécessaire de recopier ces nouvelles informations sur CB. C'est l'objectif de cette fonctionnalité et de RestoreCB (section 4.4.3, page 16).

Il s'agit de transférer les ajouts (utilisateurs, cotes, avis) et mises à jour (cotes, avis) sur CB de manière à ce que les informations contenues dans les deux bases de données soient équivalentes. Par mise à jour on ne parle pas nécessairement d'un `UPDATE` de tuple mais de la modification d'une information dans votre BD. Cela pourrait être modélisé par un `DELETE` suivi d'un `INSERT` par exemple.

Nous testerons cela à l'aide d'un script PL/SQL qui est chargé de faire des ajouts/modifications/effacements ainsi que d'invoquer le sous-programme de restauration vers CB. Notez qu'on ne demande pas de tester la disponibilité de CB de manière automatique.

*Techniques attendues* : SQL, PL/SQL, déclencheurs, transactions, séquences, logs.

## 4.2 Partie II - Création et alimentation de CB

### 4.2.1 Script SQL de création de CB et CBB - CreaCB

Vous allez étendre le schéma de CB (et CBB) pour permettre le stockage des films et des informations qui leur sont associées. Cela demande d'analyser avec attention et de manière exhaustive les données contenues dans la table externe en utilisant une combinaison de SQL et PL/SQL.

En particulier, on estime qu'il est nécessaire de déterminer les informations de statistique descriptive suivantes pour la plupart des champs : la moyenne, l'écart-type, la médiane, le minimum, le maximum, le nombre total de valeurs, le nombre de valeurs non nulles, le nombre de valeurs nulles, le nombre de valeurs dite zéro (chaîne vide, entier nul, ...), le 99ème 100-quantile, le 9999ème 10000-quantile, et si cela a un sens le nombre de valeurs uniques et les valeurs en question (comme dans le cas des certifications par exemple).

Chacune de ces informations doit pouvoir être interprétée de manière à acquérir une connaissance plus fine du jeu de données utilisé. On vous demande de réaliser à l'aide du package `UTL_FILE` un rapport au format texte contenant les informations statistiques décrites ci-dessus pour chaque champ pertinent. Dans certains cas, ce n'est pas la valeur du champ qui nous intéresse mais plutôt la longueur de celle-ci comme dans le cas du titre par exemple.

Vous utiliserez des expressions régulières (regex) à l'aide du package PL/SQL `OWA_PATTERN` et/ou des fonctions SQL appropriées comme `REGEXP_SUBSTR` pour extraire les informations contenues dans les chaînes de caractères représentant les données multiples comme par exemple les acteurs.

Les champs à analyser sont : `_id`, `title`, `original_title`, `release_date`, `status`, `vote_average`, `vote_count`, `runtime`, `certification`, `poster_path`, `budget`, `revenue`, `homepage`, `tagline`, `overview`, `genres.id`, `genres.name`, `directors.id`, `directors.name`, `directors.profile_path`, `actors.id`, `actors.name`, `actors.cast_id`, `actors.character`, `actors.profile_path`, `production_companies.id`, `production_companies.name`, `production_countries.iso_3166_1`, `production_countries.name`, `spoken_languages.iso_639_1`, `spoken_languages.name`.

La table externe `movies_ext` vous donne accès aux données sans les stocker dans la base de données. On peut donc les analyser avec des requêtes SQL complexes avant de les insérer dans les tables de la base de données.

En particulier, ce mécanisme nous permet de découvrir à quoi ressemblent nos données et de pouvoir ainsi dimensionner correctement les champs de nos tables.

Pour pouvoir utiliser la table externe, il faut dans l'ordre :

1. Créer un directory dans le schéma de CB qui est mis en correspondance avec le répertoire qui contient le fichier `movies.txt`. Soit par exemple `e:\oracle`.

```
create or replace directory TMDb as 'e:\oracle';
```

2. Créer la table elle-même à l'aide du script SQL `movies_external_table.sql`<sup>6</sup>.

Parmi les 249877 films présents dans le fichier `movies.txt`, seuls deux d'entre eux sont exclus de la table externe :

- Le film 69224 car son overview a été placée dans le champ homepage dont la taille n'est pas suffisante. Le film a cependant été corrigé depuis chez TMDb.
- Le film 265358 car son overview a été placée dans le champ tagline dont la taille n'est pas suffisante. Le problème est présent chez TMDb. En date du 8 septembre 2015, le problème est toujours présent.

```
$ wc movies.txt
249877 14778400 186505767 movies.txt
$ rlrwrap sqlplus kuty@//192.168.0.1/orcl
...
kuty@ORCL> select count(*) from movies_ext;

COUNT(*)
-----
249875

1 row selected.
```

On vous demande de rédiger les scripts de création des tables *en veillant à placer un maximum de contraintes au niveau du schéma*.

Vous ne devez rédiger qu'un seul script SQL de création des tables puisque le contenu des deux bases de données CB et CBB est équivalent. Toutes les contraintes applicatives qui ne peuvent pas être définies au moyen du LDD et qui n'ont pas été définies à l'aide de déclencheurs devront apparaître sous la forme de commentaires dans les scripts.

Choisissez judicieusement vos types de données et la taille de ceux-ci de manière à ce que 99% des informations puissent être stockées telle quelles. Donc il s'agit des tailles inférieures ou égales au quantile 99%. Les autres informations seront tronquées ou arrondies de manière à être présentes dans la base de données<sup>7</sup>.

*Techniques attendues* : SQL, PL/SQL, procédures stockées.

#### 4.2.2 Alimentation de CB - AlimCB

Nous avons trois cas de figure selon la taille de la donnée considérée. Notez que les quantiles dépendent du champ considéré (par ex. le titre du film, la durée des films, ...) et sont tous différents.

1. On doit pouvoir stocker, sans aucune modification liée à la taille, les données dont la taille est inférieure ou égale au 99ème 100-quantile, que l'on appelle aussi le 99ème percentile ( $\leq 99\%$ ).
2. On doit pouvoir stocker après troncage ou arrondi, les données dont la taille est inférieure ou égale au 9999ème 10000-quantile ( $\leq 99.99\%$ ).

<sup>6</sup>[https://cours.khi.be/sghd3/labo/movies\\_external\\_table.sql](https://cours.khi.be/sghd3/labo/movies_external_table.sql)

<sup>7</sup>Sauf celles qui sont trop grandes. Cfr. AlimCB (section 4.2.2, page 12)

3. On ne stocke pas les données qui sont au-delà de ce quantile. Elles sont tellement grandes qu'on estime qu'elles contiennent une erreur. On utilisera la valeur NULL dans certains cas et l'absence de tuple dans d'autres cas. Il s'agit des informations dont la valeur est strictement supérieure au 9999ème 10000-quantile ( $> 99.99\%$ ).

On utilise le paramètre  $n$  pour déterminer le nombre de films qu'on désire ajouter dans la base de données CB. On vous demande de réaliser l'ajout à l'aide de SQL et PL/SQL en utilisant la table externe et en veillant à choisir les films de manière aléatoire. Vous utiliserez pour ce faire un `ORDER BY DBMS_RANDOM.VALUE`.

Pour chaque film ajouté, on détermine de manière aléatoire le nombre de copies à ajouter au film en se basant sur une distribution normale. On utilise pour cela une moyenne de 5 et un écart-type de 2. Bien entendu, le nombre de copies commandées pour chaque film doit être un nombre entier strictement supérieur à 0. Tous les films déjà présents dans CB ne devront pas être dupliqués ni re-transférés mais le stock des copies devra être mis à jour.

On vous demande d'ajouter les films et leurs informations associées (acteurs, réalisateurs, ...). Bien entendu, on n'ajoute pas et on ne modifie pas des données déjà présentes. Chaque fois qu'un film est ajouté dans la base de données, indiquez le dans une table de log en précisant quelles valeurs ont été altérées avant de les stocker en indiquant clairement le nom du champ, la modification effectuée (troncage, arrondi, NULL, mise à zéro, ...), les valeurs avant et après (sauf pour les champs de type texte) et dans le cas des champs texte les longueurs avant et après dans le cas d'un troncage. Si des données ne sont pas insérées (cas de l'absence de tuple), vous devez aussi le logger.

Attention, certaines données pourraient légèrement différer de ce qui est mentionné. Par exemple certaines valeurs pour le champ **certification** sont incorrectes car on a une chaîne de caractères ne contenant pas une certification valide mais plutôt une chaîne vide ou null ou un tiret ou encore autre chose. Donc lorsque vous devez insérer ces informations dans votre BD, veuillez à les vérifier et le cas échéant soit ne pas insérer le film soit remplacer la valeur incorrecte par NULL.

Dans le même esprit, vérifiez que les champs existent et si ils existent avec une valeur spéciale indiquant l'absence de valeur comme une chaîne vide (par ex. pour la **certification**) ou 0 (par ex. pour le **runtime**) alors il est assez logique des les représenter par le NULL du SQL dans CB. De même si elles n'existent pas bien entendu. De cette manière votre code d'importation sera robuste.

En ce qui concerne l'importation des données textuelles comme le titre, la tagline, etc. on vous demande de supprimer le ou les espaces blancs (`\t \n \r` et l'espace) en début et en fin de chaîne.

La qualité de votre code d'importation et donc des données importées présentes dans CB est un point important de l'évaluation de la fonctionnalité concernée.

L'instruction SQL **MERGE** peut s'avérer intéressante dans ce genre de cas de figure. Veuillez à utiliser les transactions de manière correcte et à logger vos erreurs.

*Techniques attendues* : SQL, PL/SQL, procédures stockées, transactions, log.

## 4.3 Partie III - Evaluations de films

### 4.3.1 Recherche d'un film - RechFilm

Ecrivez un programme dans le langage de votre choix pour rechercher un film donné de manière à pouvoir ensuite lui donner une cote ou un avis. Etant donné les critères utilisés pour la recherche, il est possible que celle-ci renvoie plusieurs films parmi lesquels il faudra en choisir un.

On désire disposer des critères de recherche suivants :

- Soit un identifiant précis correspondant ou non à un film.
- Soit un ou plusieurs acteur(s), le titre, un ou plusieurs réalisateur(s), les films sortis avant, après ou pendant une année donnée. Tous ces critères peuvent être combinés.

On doit pouvoir visualiser les détails du film, c'est-à-dire les informations précédentes (qui servent de critères de recherche), la moyenne des votes TMDb, le nombre de votes TMDb, la moyenne des votes et le nombre de votes des utilisateurs de RQS, l'image du film, le runtime et l'overview.

Les avis et les cotes des autres utilisateurs doivent être visibles avec un système de pagination qui permet de ne les afficher que 5 par 5. Vous utiliserez une requête SQL qui permet d'aller choisir la bonne "page" à l'aide de `OVER` ou `ROWNUM`<sup>8</sup>.

Votre code applicatif doit invoquer des procédures stockées PL/SQL.

Veillez à utiliser les transactions de manière correcte et à loguer vos erreurs.

*Techniques attendues* : SQL dynamique, PL/SQL, procédures stockées, transactions, log.

### 4.3.2 Cotes et avis manuels sur les films - EvalFilm

Ajoutez au programme développé dans le cadre de la fonctionnalité RechFilm (section 4.3.1, page 13), la possibilité de donner une cote au film sélectionné et éventuellement de lui adjoindre un avis textuel. On doit pouvoir visualiser les informations du film comme précédemment.

Si une cote ou un avis a déjà été donné par cet utilisateur pour le film en question, on met à jour la cote ou l'avis, c'est-à-dire qu'on remplace ce qui a été précédemment indiqué.

Au niveau du GUI, on doit pouvoir indiquer une cote entre 0 et 10 (bornes incluses).

Attention, cette fonctionnalité et la suivante (EvalFilmAuto) doivent être testées de différentes manières :

- Si CB est disponible, elles doivent agir sur CB et la fonctionnalité BackupCBLight (section 4.1.2, page 10) doit pouvoir être testée. Par après, on testera également la fonctionnalité BackupCB (section 4.4.2, page 16).
- Si CB n'est pas disponible, elles doivent agir sur CBB et la fonctionnalité RestoreCBLight (section 4.1.3, page 11) doit pouvoir être testée. Par après, on testera également la fonctionnalité RestoreCB (section 4.4.3, page 16).

*Techniques attendues* : SQL MERGE, PL/SQL, procédures stockées, transactions, log.

### 4.3.3 Cotes et avis automatiques sur les films - EvalFilmAuto

Cette fonctionnalité nous permet d'ajouter un grand nombre de cotes de manière automatique en générant aléatoirement des nombres entre 0 et 10 (bornes incluses) selon une distribution normale de moyenne 5 et d'écart-type 1.5. Pour chaque film on génère une cote de manière aléatoire.

On ajoute  $n$  films à l'utilisateur en cours. Celui-ci devra fournir le  $n$  dans le GUI et invoquer la fonctionnalité par exemple en appuyant sur un bouton. Un feedback visuel lui permettra de savoir quel film a reçu quelle cote.

Des informations à propos de la distribution normale se trouve dans les notes du cours théorique.

*Techniques attendues* : SQL, PL/SQL, procédures stockées, transactions, log.

### 4.3.4 Affichage des meilleurs films - Palmares

Ajoutez au programme développé dans le cadre de la fonctionnalité RechFilm (section 4.3.1, page 13), la recherche de films en se basant sur un des critères uniques suivants :

<sup>8</sup>On ROWNUM and Limiting Results, Thomas Kyte, <http://www.oracle.com/technetwork/issue-archive/2006/06-sep/o56asktom-086197.html>

- Les  $n$  meilleurs films ( $n$  premiers par ordre décroissant) selon la moyenne des cotes des utilisateurs de RQS.
- Les  $n$  meilleurs films ( $n$  premiers par ordre décroissant) selon le nombre de cotes des utilisateurs de RQS.

Le paramètre  $n$  est choisi par l'utilisateur dans le GUI.

*Techniques attendues* : SQL, PL/SQL, procédures stockées, transactions, log.

## 4.4 Partie IV - Sauvegarde et restauration 2

### 4.4.1 Simulation d'un plantage de CB - CrashCB

On peut simuler la perte de connectivité avec CB, comme s'il y avait un problème réseau ou que l'instance venait de crasher, à l'aide de trois étapes dans l'ordre :

1. On localise la session de l'utilisateur à l'aide de l'instruction SQL ci-dessous exécutée en tant que SYS. On parle de l'utilisateur CB qui a accès à son schéma nommé CB et qui correspond à la base de données de même nom dans le cadre de l'énoncé.

```
SELECT s.inst_id,
       s.sid,
       s.serial#,
       p.spid,
       s.username,
       s.program
FROM   gv$session s
       JOIN gv$process p ON p.addr = s.paddr AND p.inst_id = s.inst_id
WHERE  s.type != 'BACKGROUND';
```

L'exécution de l'instruction renverra une résultat qui ressemble au listing ci-dessous dans lequel on peut voir deux colonnes qui nous intéressent : SID et SERIAL#.

```
INST_ID SID SERIAL# SPID USERNAME PROGRAM
-----
      1 158      11 4632      SYS SQL Developer
      1  65     253 8044      KUTY SQL Developer
```

2. On tue la session précédemment identifiée par deux nombres dans l'ordre : SID et le SERIAL#. Pour ce faire on utilise l'instruction SQL ALTER SYSTEM DISCONNECT qui a comme effet de déconnecter la session en cours en détruisant le processus serveur dédié. L'option IMMEDIATE indique que l'effet est immédiat sans attendre la fin des transactions en cours et donc qu'un ROLLBACK est initié par la base de données. Il faut donc veiller à disposer d'une gestion robuste des transactions et de ne pas laisser le hasard se charger des COMMITs et ROLLBACKs.

```
ALTER SYSTEM DISCONNECT SESSION '65,253' IMMEDIATE;
```

3. On locke l'utilisateur de manière à ce qu'il ne puisse plus se connecter.

```
ALTER USER cb LOCK ACCOUNT;
```

Le programme qui accède à CB par l'intermédiaire de procédures stockées doit pouvoir recevoir une information (exception, timeout, ...) indiquant que la session et la connexion ont été perdues. Dès que cela est reçu, le programme va se connecter automatiquement à la base CBB et continuer ses opérations de manière transparente.

Attention, si on était au milieu de la réalisation d'une opération, celle-ci doit être annulée de manière propre et on doit le cas échéant prévenir l'utilisateur qu'il doit recommencer son opération. Cela vous demande de gérer vos transactions de manière propre et robuste.

Dans le cadre de l'évaluation de cette fonctionnalité, nous re-testerons les fonctionnalités RechFilm (section 4.3.1, page 13), EvalFilm (section 4.3.2, page 14), EvalFilmAuto (section 4.3.3, page 14) et Palmares (section 4.3.4, page 14).

*Techniques attendues* : SQL, PL/SQL.

#### 4.4.2 Sauvegarde des informations de CB - BackupCB

On étend la fonctionnalité BackupCBLight (section 4.1.2, page 10) de manière à tout sauvegarder (films et copies y compris) sur CBB de manière asynchrone. De manière synchrone, on ne sauvegarde que les copies, pas les films et informations associées.

*Techniques attendues* : SQL, PL/SQL, DB links, déclencheurs, jobs (DBMS\_SCHEDULER), transactions, log.

#### 4.4.3 Restauration des informations de CB - RestoreCB

Lors de l'utilisation des procédures stockées de CBB par votre application, il est nécessaire de vérifier au sein de ces procédures si CB est redevenue utilisable. Si c'est le cas, il faudra prévenir le code applicatif, par exemple à l'aide d'une exception utilisateur, pour que celui-ci puisse changer de base de données et recommencer à utiliser CB.

Dans le cadre du test de cette fonctionnalité, il est nécessaire de retester les fonctionnalités EvalFilm (section 4.3.2, page 14) et EvalFilmAuto (section 4.3.3, page 14).

*Techniques attendues* : SQL, PL/SQL, DB links, déclencheurs, jobs (DBMS\_SCHEDULER), transactions, log.

### 4.5 Partie V - Création et alimentation de CC

#### 4.5.1 Script de création de CC - CreaCC

La base de données contenant le cinéma est une base relationnelle ne contenant que des données de type XML structuré (XMLType). Vous devez étendre les modèles existants et rédiger le script de création des tables XMLType en plaçant un maximum de contraintes au niveau du schéma. On vous demande également les schémas XSD liés à chacune de ces tables. A défaut de script de création<sup>9</sup>, il faut disposer des procédures permettant la création de ces tables.

Choisissez judicieusement vos types de données et la taille de ceux-ci de manière à ce que toutes les informations puissent être stockées telle quelles. Cfr. AlimCB (section 4.2.2, page 12).

*Techniques attendues* : SQL, XML, XDB, PL/SQL, procédures stockées.

#### 4.5.2 Alimentation de CC - AlimCC

CC reçoit les nouveaux stocks de copies de CB de deux manières différentes. Si CB est indisponible, CBB doit pouvoir prendre le relais et effectuer le même traitement.

1. Chaque fois que des copies arrivent sur CB à partir de CI, un nombre aléatoire de celles-ci est livré automatiquement dans CC par le biais de documents XML. Ce transfert se fera au moyen d'un déclencheur.

---

<sup>9</sup>Dans le cas d'une création automatique des tables par Oracle.



2. Toutes les semaines, une routine envoie un nombre aléatoire de copies de chaque film présent dans CB. Ce transfert se fera au moyen d'un job et toujours par le biais de documents XML.

Dans les deux cas, le nombre aléatoire se base sur une distribution uniforme allant de 0 à  $\lfloor \frac{n}{2} \rfloor$  où  $n$  est le nombre total de copies réellement disponibles sur CB et  $\lfloor x \rfloor$  est l'entier le plus grand inférieur ou égal à  $x$ . On ne prend donc pas en compte uniquement le nouveau lot de copies d'un film mais l'ensemble des copies se trouvant sur CB.

Il est imposé qu'une copie ne puisse se trouver en même temps à la centrale CB et dans un complexe : il faut donc la supprimer de CB quand elle est transférée vers le complexe et vice et versa. De même, si le nombre de copies à transférer est 0, la fiche signalétique complète du film concerné et toutes ses informations annexes ne seront pas transférées. Outre les informations d'un film, si celui-ci a fait l'objet d'une note ou d'un avis par des utilisateurs de RQS, il y aura lieu de transférer ces avis et notes. Cependant, il est inutile d'en connaître la provenance et donc l'utilisateur.

Dans le cas présent, les documents XML sont intégrés dans une table des films et des copies de type XMLType et automatiquement restructurés. Toutes les données transférées sont au format XML.

*Techniques attendues* : PL/SQL, réplication, DB link, XMLType, distribution uniforme, transferts de BLOB.

#### 4.5.3 Retour des Copies - RetourCopies

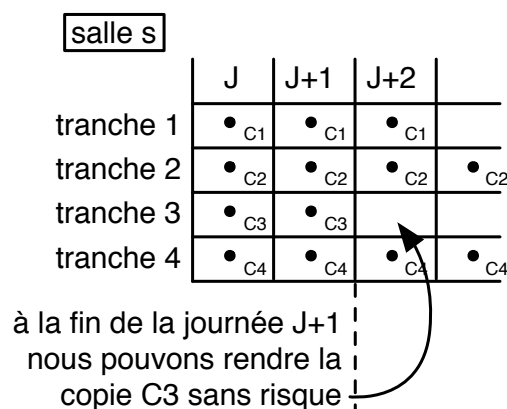
Quand une copie n'est plus utilisée dans une programmation, elle est renvoyée à la centrale CB. On peut voir sur la figure 2 que la copie  $C_3$  planifiée dans la salle  $s$  pour la tranche horaire 3 (18h) doit être retournée à CB à la fin de la journée  $J + 1$ . Pour optimiser les transferts, un retour vers CB ne pourra s'effectuer que lorsqu'une livraison de copies provenant de CB arrive au complexe.

Les copies en cours de programmation ou n'ayant pas encore fait l'objet d'une programmation depuis leur dernière arrivée dans le complexe ne doivent pas être retournées.

Même si le film n'a plus aucune copie, sa signalétique reste dans la base de données du complexe afin de garantir la cohérence de l'archivage.

Si CB n'est pas disponible, les copies doivent repartir vers CBB.

FIG. 2: RetourCopies



*Techniques attendues* : PL/SQL, réplication, DB link.

## 4.6 Partie VI - Programmation et commande de places

### 4.6.1 Programmation des films - ProgFilms

Chaque jour, des documents XML contenant des programmations sont déposés dans un DIRECTORY. Une routine en prend connaissance et effectue les programmations sollicitées. On désire disposer d'un autre document XML créé dans le même répertoire contenant le feedback de l'ajout de la programmation. Si une séance n'a pas pu être programmée, on indique la raison (slot déjà pris, copie indisponible, copie inexistante, ...). Le nom du fichier sera formé du nom original suffixé par `_feedback` avec l'extension `xml` dans les deux cas.

Ce rapport doit être agréable à utiliser. C'est pourquoi on vous demande de disposer d'une feuille de style CSS permettant de réaliser le rendu du document XML.

Les programmations ne concernent que des dates dans le futur. On doit y retrouver a minima un film, la salle et la plage horaire. La copie sera prise aléatoirement parmi celles existant pour le film s'il y en a une de disponible. Le nombre de jour de programmation sera déterminé aléatoirement sur la base d'une distribution normale. On utilise pour cela une moyenne de 8 et un écart-type de 3. Bien entendu, il ne peut y avoir 0 jours de programmation. Une programmation existante doit également pouvoir être étendue un certain nombre de jour selon la même base de calcul du nombre de jours.

Des informations à propos des distributions uniforme et normale se trouve dans les notes du cours théorique.

Outre cela, les plages horaires ne sont pas fixes et la durée du film doit également être prise en compte. On ne programme pas un film de 2h pour la dernière demi-heure d'ouverture du cinéma.

Pour faciliter le suivi de cette programmation, les différentes étapes de recherche et d'attribution seront enregistrées dans la table de log générale.

*Techniques attendues* : SQL, PL/SQL, DIRECTORY, UTL\_FILE, XML.

### 4.6.2 Archivage des programmations - ArchProg

Pour améliorer les projections et statistiques de RQS, il faut conserver la trace de différents indicateurs propre à la production interne de l'entreprise. Il faut donc quotidiennement archiver les programmations terminées de films en conservant au minimum pour un film sa pérennité (le nombre de jour au total qu'il a été à l'affiche) ainsi que le nombre de places vendues pour ce film. Un film présent dans plusieurs salles un même jour ne compte que pour une seule journée dans le calcul de pérennité. On souhaite également conserver le nombre global de copies différentes manipulées par films.

*Techniques attendues* : PL/SQL, XML.

### 4.6.3 Recherches de places libres - RechPlaces

Le complexe propose une interface de type GUI ou web permettant la commande en ligne de places de cinéma. Un module de recherche basé sur sa programmation est mis à la disposition des visiteurs. Plusieurs types de recherches sont possibles :

- La popularité des films disponibles (inférieure, supérieure ou égale à)
- La pérennité des films disponibles (inférieure, supérieure ou égale à)
- La position du film dans le palmarès des votes des utilisateurs de RQS (inférieure, supérieure ou égale à)
- Le titre ou une partie du titre
- Le(s) genre(s) du film

La popularité représente l'ensemble des places vendues toutes séances confondues et la pérennité le nombre de semaines de projection d'un film, c'est-à-dire qui est à l'affiche. Dans les deux cas, ces nombres sont calculés depuis qu'une des copies du film a été programmée dans le complexe. Le premier nombre est entier mais le second peut avoir une partie décimale. Notez qu'on utilise les archives pour faire cela.

Dans tous les cas, le résultat de la recherche consiste en la liste des titres de films rencontrant les critères demandés et faisant l'objet d'une programmation. Toute combinaison de ces critères doit être possible. Par exemple, une recherche pourrait être faite sur les films contenant "ocean" et ayant une popularité supérieure à 10 places. La casse importe peu.

Le résultat affiché contient au minimum l'affiche du film (en petit format de type icône), le titre du film, sa popularité et sa pérennité. Il sera trié sur base des critères de recherche retenus par ordre décroissant pour la popularité et la pérennité et par ordre alphabétique croissant pour les titres. Par ordre de priorité, on prend en considération la popularité, puis la pérennité, puis le titre. Si aucun critère n'est sélectionné, il faut afficher les titres des films programmés dans la journée uniquement.

A partir de cette liste de titres, on doit pouvoir afficher la fiche signalétique complète d'un film reprenant les informations pertinentes pouvant intéresser un spectateur potentiel. Au minimum l'affiche à la taille `w185` doit être présente, le titre, les acteurs, les réalisateurs, le résumé, les genres, les valeurs de popularité et pérennité, la position dans le palmarès et le public cible. Il doit également être possible à partir de la liste des films d'obtenir pour un des films l'ensemble des séances (journées + heures) où une copie de ce film est programmée dans le complexe. Cette liste des séances montre le titre du film, la séance et le nombre de places restantes.

L'affichage doit être conçu pour permettre ultérieurement la commande de places pour les séances affichées quels que soient les films.

Différentes solutions de codage sont possibles pour construire la requête correspondante comme par exemple une combinaison de `AND`, `OR` et `IS NULL`, ou l'utilisation de `LIKE %` ou encore l'utilisation de `EXECUTE IMMEDIATE`. A vous de voir la technique que vous choisirez. Il vous est demandé de la construire de la manière la plus générique possible afin que tout ajout ultérieur d'un ou plusieurs critères n'engendre pas une multitude de code supplémentaire. Aucun SQL direct n'est permis. De même, la construction de la requête ou du bloc PL/SQL contenant la requête ne peuvent pas apparaître du côté du client ou du serveur. Seul un appel de procédure ou de fonction PL/SQL est autorisé.

Le langage est libre de choix. Une interface fouillée n'est pas nécessaire, elle doit avant tout être pratique. Vous êtes libres de retourner un document JSON, XML, TXT ou autre qui nécessiterait un minimum de programmation pour être interprété ou encore de retourner du HTML nécessitant l'utilisation d'une page web minimale avec Ajax ou JS.

*Techniques attendues* : langage et technique au choix pour l'affichage, PL/SQL, manipulation de BLOBs.

*Contrainte obligatoire* : Aucun SQL direct ou bloc PL/SQL, passage par des appels et retours de fonctions PL/SQL.

#### 4.6.4 Commandes de places - CmdPlaces

A partir de RechPlaces (section 4.6.3, page 18), il doit être possible de commander des places de cinéma. Le visiteur doit pouvoir indiquer une quantité souhaitée face à la séance qui l'intéresse et l'insérer dans un "panier". Si des demandes de places pour cette séance se trouvent déjà dans le panier, elles seront cumulées. Il doit être possible de commander des places pour plusieurs séances différentes.

Le contenu du panier doit être consultable à tout moment. Outre la validation, on doit pouvoir modifier des quantités, supprimer une ou plusieurs lignes du panier ou le supprimer entièrement. La commande s'effectue sous la forme d'un document XML envoyé à la base de donnée lorsque le client valide sa commande. On demande de valider ce document par DTD ou XSD avant de l'envoyer.

Dans la base de données, le nombre de places disponibles pour la séance est mis à jour. Il n'y a pas d'overbooking et l'applicatif est bien sûr multi-utilisateurs. Dans un premier temps, le résumé général de la commande s'affichera avec les éventuels problèmes rencontrés. Dans un second temps, si l'utilisateur le confirme, elle engendrera la génération de tickets, cfr. Tickets (section 4.6.5, page 20).

Une attention particulière doit être portée à l'ergonomie et au suivi d'information : dans tous les cas de figures, l'utilisateur doit être informé correctement du statut de sa commande et des éventuels problèmes rencontrés dans le traitement de celle-ci. Il doit pouvoir être sollicité pour confirmer une opération si nécessaire. Toute erreur ou problème doit lui être signifiée dans un langage clair et compréhensible. Si une partie de la commande ne peut pas être réalisée, il vous revient de faire un choix quant à l'annulation de l'entièreté de celle-ci, la proposition d'alternative, etc.

*Techniques attendues* : Web, PL/SQL, réplication, XML, DB link.

*Contrainte obligatoire* : Aucun SQL direct, passage par des appels et retours de fonctions PL/SQL.

#### 4.6.5 Génération de tickets - Tickets

Quand une commande est réalisée, il s'agit de générer un ticket. Ce ticket prendra la forme d'un document PDF à plusieurs pages. Sur chaque page, on trouvera un QR code comportant les informations de base, à savoir le titre du film, la séance retenue et le nombre de places. Ces informations seront également recopiées sous le QR code. Il y aura autant de pages dans le PDF qu'il y a de séances réservées dans la commande. Un en-tête et un pied de page commun à tout le document seront également présents.

La génération de votre ticket sera intégré à votre application fournissant déjà les fonctionnalités RechPlaces et CmdPlaces. Celle-ci produira les QR codes et le document PDF. Le choix du langage est laissé à votre libre choix. Diverses librairies ou plug-ins existent pour la génération de QR Code en fonction du langage utilisé, par exemple ZXing combiné à QRGen en Java. Vous utiliserez l'outil Prince<sup>10</sup> qui vous permet de générer un document PDF à partir d'un document HTML et de la feuille de style CSS3 associée.

Dans tous les cas, il faudra que le PDF obtenu, le QR Code, les informations liées soient enregistrées dans la base TCKT qui est une base MongoDB, et donc fonctionne selon une approche clé-document. Dans le statut de la commande sur la base CC, il faudra indiquer si le ticket a bien été généré ou non. Un log des différentes étapes devra être fait .

Le retour de cette fonctionnalité consistera en un autre QR code, retourné à l'utilisateur, qui contiendra cette fois-ci une URL pointant vers une mini application Web permettant d'accéder à la MongoDB en lecture et de retourner le document PDF stocké dans celle-ci. Par exemple en Java vous pouvez vous contenter d'une servlet sans utiliser de JSP.

*Techniques attendues* : génération de QR Code, génération de Pdf, JSON/BSO, MongoDB .

### 4.7 Partie VII - Les acteurs

#### 4.7.1 Détail des signalétiques d'acteurs - FicheActeurs

Lorsque l'on affiche le détail de la signalétique d'un film, il doit maintenant être possible d'afficher également, à la demande, le détail d'un des acteurs du film. Pour cela il vous est demandé d'interroger directement la base de données BP. A vous d'intégrer cette réponse dans une fenêtre séparée. Attention, il se peut que certains acteurs n'existent pas dans BP.

Il est demandé d'afficher au minimum le nom, la date de naissance, le lieu de naissance, une photographie si elle existe et la filmographie (nom du film, nom du rôle, année du film).

*Techniques attendues* : JSON/BSO, MongoDB .

<sup>10</sup><http://www.princexml.com/>

## 4.8 Partie VIII - Service Marketing

RQS est très attentive au suivi de la qualité des services que les complexes fournissent. Pour ce faire, elle dispose d'un outil consolidant diverses informations et proposant surtout des indicateurs lui permettant d'évaluer ces différents points. Pour le moment, ces indicateurs sont purement analytiques mais il est probable que les choses évoluent.

Les indicateurs utilisés ou questions que se posent les gestionnaires et que les indicateurs doivent illustrer sont :

1. La nationalité de l'acteur influe-t-elle sur le nombre de jours qu'un film reste à l'affiche ? Et sur la moyenne des notes que donnent les utilisateurs à un film ?
2. Quelles sont les popularités des films projetés tenant compte ou non des genres du film, de la salle, de la période de la journée (matin, après midi, soir) ou encore du jour, de la semaine ou du mois de projection ?
3. La cote d'un film est-elle liée au nombre de jours qu'un film est projeté ?
4. Quelles sont les proportions de places vendues par salle ou complexe, par mois ou par année ?
5. Quel est le turn over (pourcentage moyen se trouvant dans CC) des copies au sein des installations tenant compte du genre des films, de la semaine ou de l'année et du pays de production du film ?

Comme on le constate, ces différents points nécessitent une certaine préparation des données. C'est ce qui concerne les activités suivantes.

### 4.8.1 Création du modèle dimensionnel des données - CreaMDD

Deux nouvelles bases de données sont nécessaires pour effectuer la collecte et la présentation des données destinées au DataWarehouse et son outil de consultation.

La première base, MKT, hébergera le schéma d'intégration des données. C'est à dire un sous-ensemble des données issues des différents sites où se trouvent les données sources pour répondre aux questionnements mentionnés ci-dessus. Seules les données réellement utiles seront rapatriée. Il s'agit de fournir le modèle conceptuel et le modèle relationnel correspondant à ce schéma.

La seconde base, DW, hébergera le schéma de diffusion des données. C'est à dire les données restructurées issues du modèle d'intégration et destinées à construire un cube OLAP ou être exploitées directement dans une interface de consultation. Ce schéma se construit sur la base d'un modèle dimensionnel de données. Il s'agit de fournir ce modèle dimensionnel et le modèle relationnel qui en découle, puisque le stockage se basera sur un cube ROLAP. Au final 4 modèles sont donc attendus. Ces modèles validés, il conviendra de générer les scripts et tables liés. Si vous utilisez des vues matérialisées, vous devez en fournir l'instruction complète et les ajouts de privilèges liés. Il est demandé d'avoir en tout temps au moins 30 tuples dans chaque objet, que ce soit une table ou une vue matérialisée, représentant une dimension ou un fait.

### 4.8.2 Collecte d'indicateurs de production - CollectData

Cette procédure sera responsable de la collecte des différentes informations utiles dans les bases du systèmes. Cette collecte se fera au moyen diverses techniques de communications et d'insertions dans un modèle d'intégration. Pour rappel, un modèle d'intégration est toujours un modèle relationnel rassemblant toutes les informations utiles mais non encore organisées en dimensions. Cette collecte a donc pour objectif d'alimenter les tables relationnelles de la base de donnée BD\_MK qui est une base Oracle. Les techniques à employer sont laissées au libre choix mais devront être argumentées et justifiées dans le dossier à remettre.

### 4.8.3 Traitement des indicateurs - LoadData

Cette fonction alimentera le data warehouse en tant que tel, c'est à dire le modèle de diffusion basé sur un modèle dimensionnel qui aura été établi au préalable. Les données du modèle d'intégration seront donc traduites en termes de dimensions et de faits dans un nouveau schéma et insérées dans celui-ci. Il est demandé d'utiliser une vue matérialisée pour au moins deux faits.

### 4.8.4 Consultation des tableaux de bord - ConsultData

À l'aide d'un logiciel (Qlikview / Excel / ...), il s'agit de présenter les différents faits et indicateurs. L'écran de visualisation devra permettre de pouvoir effectuer aisément du slicing, du scoping, du roll up et du drill down. C'est-à-dire que certaines données devront pouvoir être figées, de même que d'autres pourront faire l'objet de regroupement ou de forage. Les visualisations se présenteront sous une forme graphique. Il convient que les choix de graphiques soient effectués judicieusement vis-à-vis des informations à représenter. Les différentes étapes de réalisations de ces écrans sont à présenter dans un dossier à remettre.

## 4.9 Partie IX - Administration de RQS

La maintenance et la réutilisabilité du produit ainsi développé sont un souci important de nos jours. Si toutes les fonctionnalités développées jusqu'ici pouvaient n'être testées que dans un milieu de développement, il faudra néanmoins envisager leur déploiement sur d'autres bases et donc ne pas être trop fortement tenu par les caractéristiques matérielles et systémiques de l'environnement de développement.

Un dossier est donc attendu. Ce dossier s'adresse à des personnes qui ont le même niveau de connaissance en bases de données que vous-même lorsque vous êtes arrivés en 3e année. Il doit remplir 3 objectifs :

1. L'installation, la configuration, la mise en route et l'utilisation de votre application;
2. La maintenance/mise à jour de votre application;
3. Le plan de sauvegarde et récupération de votre système d'information.

### 4.9.1 Dossier Installation, mise en route, exploitation - AdminConfig

L'utilisateur doit pouvoir facilement configurer les outils nécessaires au bon fonctionnement de l'application et s'y retrouver dans le code/les packages que vous fournissez. Sans rien connaître de l'application, de vos packages, il doit pouvoir installer le tout et comprendre l'articulation de votre code et de vos données.

Il convient donc de fournir l'intégralité du code (XML et DataWarehouse sont également pris en compte) mais aussi les éventuels documents explicatifs comme les MCD/MRD ou rapports décrivant les packages. Un minimum de documentation à l'intérieur même du code est également attendu pour que lorsqu'on ouvre celui-ci on comprenne ce qu'il fait.

Un mini mode d'emploi de l'application est également nécessaire. N'oubliez pas que l'application doit se déployer sur plusieurs sites et non pas uniquement sur une seule base ou un seul pc. Les configurations propres à ce déploiement doivent être explicitement expliquées. De même, les tables et autres n'existent pas à l'avance, tout doit pouvoir être créé sur base des documents fournis.

Il est donc attendu :

1. Une présentation générale de l'application et la logique générale suivie pour l'organisation des fichiers utiles à la mise en place des applications
2. Toutes les informations nécessaires à l'installation et la configuration de l'application. Au terme de cette explication et du respect des consignes qui y sont données, l'utilisateur doit pouvoir démarrer les applications
3. Un mode d'emploi des différentes applications

#### 4. Les codes et sources liés.

##### 4.9.2 Maintenance, mise à jour - AdminMaintenance

L'utilisateur doit pouvoir faire n'importe quelle modification dans votre code sans devoir pour autant l'étudier ligne par ligne en tentant de comprendre ce que fait le package / la fonction ou procédure / le bloc de code. Il faut que l'on puisse comprendre facilement votre logique, dès lors une documentation adéquate dans le code voir à côté du code est nécessaire.

Vous avez du apprendre en première année ou en deuxième à indiquer ce que faisait une procédure ou fonction (IN / OUT / PROCESS), idem pour les blocs de code où vous appliquez une logique particulière. Il faut que la personne qui doit reprendre votre code puisse le faire aisément.

N'hésitez pas à fournir éventuellement un schéma fonctionnel ("tel package fait appel à tel package pour telle fonctionnalité" à l'aide de flèches, ou autres méthodes de représentations).

Il est donc attendu :

1. Une présentation générale des différents fichiers et packages de codes ainsi que leur organisation
2. Le code commenté

##### 4.9.3 Sauvegarde et récupération - AdminSvg

Il s'agit ici de discuter de la politique et de la stratégie de sauvegarde que vous mettriez en place pour chaque site ou partie de site selon les traitements qui y sont effectués. Quel type de sauvegarde ? La sauvegarde est-elle nécessaire ? Oui ? non ? Pourquoi ? De justifier vos choix sur base de ce qui a été présenté au cours ou se trouve dans les documents de référence. Ceci outre le fait que CBB est déjà une sauvegarde de CB.

Il est donc attendu :

1. Une présentation générale de ce qu'est un plan de sauvegarde et/ou récupération.
2. Une présentation des différentes techniques exploitables dans le cadre de RQS.
3. Une justification de votre plan de sauvegarde et des choix techniques retenus pour la mise en oeuvre de ce plan.

## 5 Organisation pratique et évaluations

### 5.1 Téléversement du code avant évaluation

Tout le code utilisé dans votre projet (scripts SQL, code PL/SQL, code source de votre langage, ...) concerné par l'évaluation que vous allez présenter doit être téléversé avant le début de la séance où a lieu l'évaluation. On vous demande de respecter le format suivant :

- **groupe** est une séquence de quatre chiffres comme 2321.
- les noms **nom1** et **nom2** sont indiqués en minuscules par ordre alphabétique en ayant supprimé les espaces et transformé les lettres accentuées en leur équivalent non accentué (par exemple é → e).
- la fonctionnalité présentée est indiquée en minuscules.

L'archive est de type rar. Notez que si l'évaluation est réalisée seul alors il n'y aura pas de **nom2**.

Le format est le suivant :

**groupe-nom1-nom2-fct.rar**

Par exemple, si Ludovic Kutty et Laurence Herbiet du groupe 2327 présentent les fonctionnalités **BackupCBLight** et **RestoreCBLight**, on aura :

`2327-herbiet-kuty-backupcblight-restorecblight.rar`

L'archive devra être téléversé par FTP (port 21) sur **nemo** dont l'IP est 10.59.26.135 avec l'utilisateur **anonymous** sans mot de passe et dans le répertoire `/upload`. Notez qu'il n'est pas possible de télécharger ou d'effacer des fichiers déjà présents. Vous pouvez juste téléverser (upload) votre archive.

## 5.2 Tableau synthétique

TAB. 2: Organigramme d'évaluation

Fonctionnalité	Pondération	Groupe	Intégration
BackupCBLight	10	1	OED4
RestoreCBLight	10	1	OED4
CreaCB	30	solo	OED4
AlimCB	20	solo	OED4
RechFilm	20	solo	OED4
EvalFilm	15	2	OED4
EvalFilmAuto	10	2	OED4
Palmares	10	3	OED4
CrashCB	20	3,5	OED4
BackupCB	15	4,5	OED4
RestoreCB	15	4	OED4
CreaCC	10	solo	I
AlimCC	15	6	I
RetourCopies	10	6	I
ProgFilms	20	7	I
ArchProg	10	7	I
RechPlaces	20	8	I
CmdPlaces	10	9	I
Tickets	15	9	I
FicheActeurs	10	8	I
CreaMDD	10	solo	I
CollectData	15	10	I
LoadData	15	10	I
ConsultData	20	solo	I
AdminCfg	15	11	I
AdminMaintenance	20	11	I
AdminSvg	20	11	I

## 5.3 Modalités d'évaluation du laboratoire

### 5.3.1 Premier semestre et Evaluation continue

- L'évaluation de laboratoire de l'unité Techniques Avancées de Gestion des données est une évaluation continue
- La cote de laboratoire constitue 50% de la cote globale de celui-ci.
- Le travail est prévu pour deux mais chaque étudiant doit être capable de répondre à toute question du titulaire portant sur l'application présentée ou la matière théorique et pratique liée à l'application.
- Au delà de la dernière séance de laboratoire de la semaine indiquée, toute fonctionnalité non présentée aura la cote de zéro et ne pourra faire l'objet d'une présentation d'amélioration en janvier.



- **ATTENTION** : une seule fonctionnalité peut être présentée lors d'une séance de laboratoire. Excepté si elle fait l'objet d'un groupe. En ce cas, elle peut être présentée seule ou avec le reste du groupe portant le même numéro.
- Le code de la fonctionnalité présentée au cours d'une séance de laboratoire doit être envoyé **AVANT LA SEANCE** sur le serveur FTP (cfr. section 5.1). C'est sur ce code que sera effectuée l'évaluation. Sans code déposé sur le serveur, il ne peut y avoir de présentation. De même, si le format des noms de fichier n'est pas respecté. La date de dépôt sur le serveur FTP est celle qui constitue la date de remise du code de vos fonctionnalités.
- La cote finale représente la somme des points des différentes fonctionnalités réalisées au cours du semestre. Les points attribués à chaque fonctionnalités sont indiqués dans le tableau 2 de la section 5.2.
- En janvier, il est possible, à toute équipe le souhaitant, de proposer une amélioration de fonctionnalité(s) présentée(s) précédemment. Dans ce cas, la cote finale de la fonctionnalité sera la moyenne arithmétique des cotes obtenues aux deux présentations.
- Les étudiants sont libres d'organiser et répartir leur travail comme ils le souhaitent en tenant compte des règles indiquées ci-dessus.
- les cotes pour les fonctionnalités indiquées d'un "I" font l'objet d'une évaluation intégrée entre les deux activités d'apprentissage de l'unité. Les cotes obtenues sont cumulées respectivement avec celles d'OED ou d'XML pour former la cote finale de laboratoire de chaque activité.

On peut vous demander de modifier certaines fonctionnalités lors de l'évaluation.

### 5.3.2 Deuxième session

L'examen de laboratoire de seconde session sera pondéré à 100% de la partie laboratoire. Il s'agit de compléter les fonctionnalités du premier semestre. L'évaluation sera reprise de zéro et compte tenu du délai supplémentaire, des commentaires et corrections donnés lors de l'évaluation continue de première session, il est possible de ne pas obtenir autant de points pour certaines réalisations qu'en janvier. La cote de première session n'est pas conservée si celle de 2ème session est moins bonne.

Vous n'êtes pas obligé de représenter une fonctionnalité et donc de garder la cote de première session selon les modalités indiquées dans le tableau à la section 5.4.

On peut vous demander de modifier certaines fonctionnalités lors de l'évaluation.

## 5.4 Synthèse des modalités d'évaluation pour le cours d'Organisation et Exploitation des données (OED)

### 5.4.1 Théorie

Evaluation	Modalites	Cotes	Report de parties
1ère session	1 Questionnaire (3 ou 4 parties) + EC	moyenne pondérée	partiel si $\geq 10$ pour une partie
2ème session	1 Questionnaire (3 ou 4 parties)	total pondéré	-

### 5.4.2 Labo

Evaluation	Cotes	Report
1ère session	somme des fonctionnalités (avec moyenne si améliorations)	si $\geq 50$ % à la fcté
2ème session	somme des fonctionnalités (repris à 0)	-

### 5.4.3 Cote globale

Évaluation	Cotes	Report
1ère session	50% théorie/50% labo	partiel si $\geq 10$ pour la partie
2ème session	50% théorie/50% labo	-

FIG. 1: Schéma global

