



GÉNIE LOGICIEL : BASE DE DONNÉES

Christian INGOUFF
Pierre-Alexandre TYNDAL
Hugo DOS

EISTI

2013/2014
Semestre 1

Table des matières

1	Introduction	2
2	Modèle conceptuel de données	3
2.1	Les entités	5
2.2	Les associations	5
2.3	Précisions	6
3	Modèle logique de données	7
4	Opérations SQL	8
4.1	Utilisation	8
4.2	Création et suppression de tables	9
4.3	Jeu d'essai	9
5	Conclusion	13

Chapitre 1

Introduction

Ce projet de gestion de questionnaires consiste, grâce à une base de données et de scripts SQL, à récupérer les différents questionnaires et les différents calculs possibles, propres à chaque type de questionnaire.

Sachant qu'un QCM est défini dans un module et peut être utilisé dans toute activité (TD, TP, Projet,...) du module, qu'une fiche d'évaluation est définie dans un programme (CPI1, CPI2, ING1,...) et peut être utilisée dans les modules du programme, qu'une fiche de catégorisation permet de catégoriser les interviewés et que tous ces questionnaires peuvent être réutilisés chaque année scolaire, nous avons donc, en suivant ces contraintes, créé un modèle conceptuel de données (MCD) afin d'illustrer au mieux la structuration de notre base de données. Grâce à la création d'un jeu d'essai par le biais d'Excel permettant de réaliser une première approche du projet, et la réalisation de ce MCD, nous pouvons déjà visualiser comment fonctionnera et s'établira la base de données.

Suite à un entretien avec le professeur référent (Besma ZEDDINI), nous avons choisi de revoir notre MCD. La poursuite du projet se basera donc maintenant sur le MCD noté "Modèle conceptuel de données 4" sur le précédent livrable. Nous allons par ce rapport retranscrire le modèle conceptuel des données sous forme de modèle logique des données (MLD) et expliquer le script de création des tables et des contraintes associées (primary key, not null, foreign key, default, check, unique) ainsi que le jeu de test qui l'accompagne, basé sur la maquette au préalable réalisée sous Microsoft Excel.

Chapitre 2

Modèle conceptuel de données

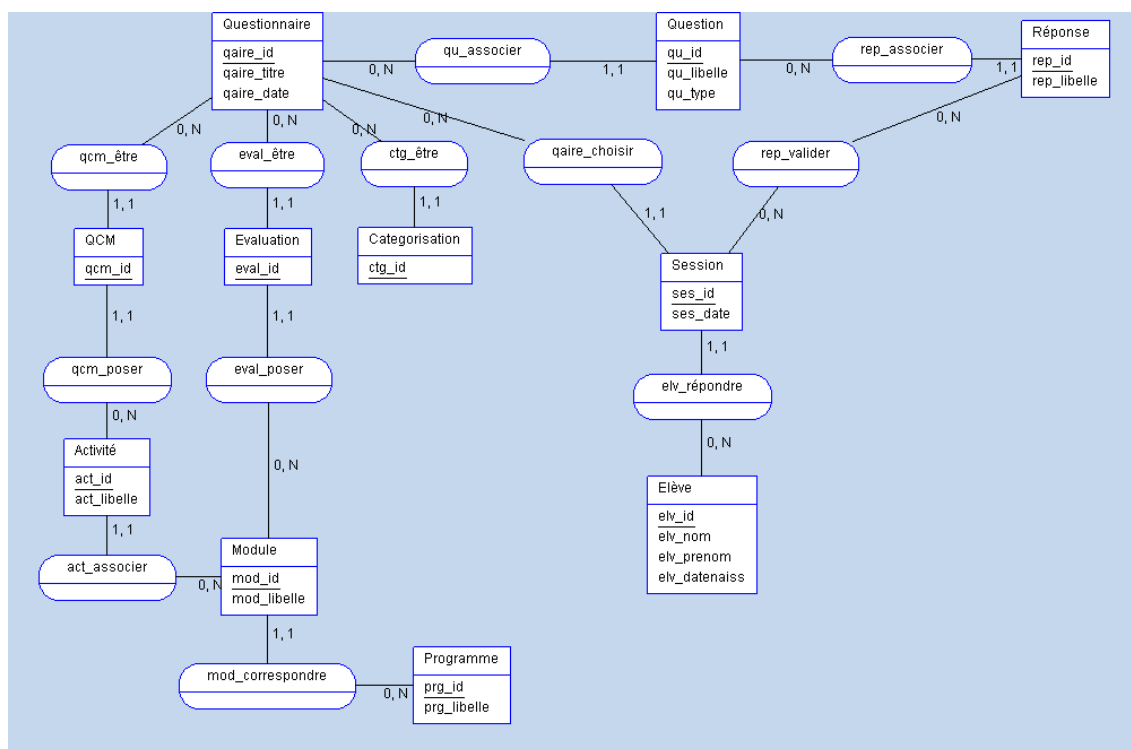


FIGURE 2.1 – Modèle conceptuel de données

Nom ▲	ID	
act_id	act_id	AUTO_INCREMENT
act_libelle	act_libelle	CHAR
ctg_id	ctg_id	AUTO_INCREMENT
elv_datenaiss	elv_datenaiss	DATE
elv_id	elv_id	AUTO_INCREMENT
elv_nom	elv_nom	CHAR
elv_prenom	elv_prenom	CHAR
eval_id	eval_id	AUTO_INCREMENT
mod_id	mod_id	AUTO_INCREMENT
mod_libelle	mod_libelle	CHAR
prg_id	prg_id	AUTO_INCREMENT
prg_libelle	prg_libelle	CHAR
qaire_date	qaire_date	DATETIME
qaire_id	qaire_id	AUTO_INCREMENT
qaire_titre	qaire_titre	CHAR
qcm_id	qcm_id	AUTO_INCREMENT
qu_id	qu_id	AUTO_INCREMENT
qu_libelle	qu_libelle	CHAR
qu_type	qu_type	CHAR
rep_id	rep_id	AUTO_INCREMENT
rep_libelle	rep_libelle	CHAR
ses_date	ses_date	DATETIME
ses_id	ses_id	AUTO_INCREMENT

FIGURE 2.2 – Dictionnaire de données

2.1 Les entités

- **Questionnaire** : Ensemble des questionnaires, contient un id unique, un titre et une date de création
- **QCM, Catégorisation, Evaluation** : types de questionnaires, contient un id unique.
- **Question** : Ensemble des questions. Ces questions appartiennent à un questionnaire. Contient un id unique, un libellé (le texte de la question) et le type de question (fermée, ouverte, etc...).
- **Réponse** : Ensemble des réponses. Ces réponses sont sous-jacentes à une question. Contient un id unique et le texte correspondant à la réponse.
- **Session** : Définie quand un élève répond à un questionnaire. Contient un id unique et une date de session.
- **Elève, Programme, Module, Activité** : Ces entités représentent comme leur nom l'indique l'élève, les programmes de l'EISTI et enfin les modules et les activités des différents programmes.

2.2 Les associations

- **rep_associer** : Une question proposant des réponses, cette association fait le lien entre la réponse et son unique (cardinalité 1,1) question correspondante. Si une question est ouverte, les réponses qui lui sont associées vont être les réponses entrées par les utilisateurs lors des sessions.
- **qu_associer** : Un questionnaire proposant des questions, cette association fait le lien entre la question et le questionnaire où elle figure.
- **qcm_être, eval_être, ctg_être** : Définition du type de questionnaire
- **qcm_poser, eval_poser** : Définition du contexte dans lequel le questionnaire est posé (dans quelle activité pour les QCM, dans quel module pour les évaluations)
- **act_associer, mod_correspondre** : Association hiérarchique de données : une activité est contenue dans un module, un module est contenu dans un programme
- **elv_répondre** : Quand un élève répond à un questionnaire, il génère une session. elv_répondre fait le lien entre la session et l'élève.
- **qaire_choisir** : Une session choisit forcément un et un seul questionnaire
- **rep_valider** : Une session enregistrera des réponses choisies

2.3 Précisions

Par rapport au MCD du dernier rapport, nous avons enlevé l'association *elv_intégrer* car nous nous sommes rendus compte que le programme qu'intègre l'élève n'est pas pertinent à la finalité de notre projet.

La manière dont nous enregistrons les réponses des élèves est la suivante : une session est créée lorsqu'un élève veut répondre à un questionnaire. Ainsi, nous aurons un lien entre l'élève et le questionnaire auquel il veut répondre. De ce questionnaire, on fait découler les questions lui appartenant à l'élève et on lui propose de choisir une réponse (question fermée) ou d'en entrer une (question ouverte). On enregistre finalement les réponses choisies ou entrées par l'élève avec *rep_valider*.

L'intérêt de cette structure est la possibilité de pouvoir répondre plusieurs fois au même questionnaire. De ce fait, les questionnaires sont indéfiniment réutilisables. La définition de la date de session nous permet en plus de trier chronologiquement les opérations.

Chapitre 3

Modèle logique de données

Le modèle conceptuel de données décrit ci-avant se traduit ainsi en modèle logique de données :

- Questionnaire(id, titre, qa_date);
- Question(id, libelle, qu_type, #qa_id);
- Reponse(id, libelle, #qu_id);
- Eleve(id, nom, prenom, datenaiss);
- Programme(id, libelle);
- Module(id, libelle, #prg_id);
- Activite(id, libelle, #mod_id);
- QCM(id, #qa_id, #act_id);
- Evaluation(id, #qa_id, #mod_id);
- Categorisation(id, #qa_id);
- Session(id, ses_date, #elv_id, #qa_id);
- rep_valider(#ses_id, #rep_id);

Pour les références des clés étrangères, on se réfère au préfixe de l'attribut en question, avec :

- qa : Questionnaire
- qu : Question
- prg : Programme
- mod : Module
- act : Activité
- elv : Elève
- ses : Session
- rep : Réponse

Chapitre 4

Opérations SQL

4.1 Utilisation

Nous fournissons dans le dossier *src* de ce livrable trois fichiers .sql :

- create.sql : Ce fichier permet de créer les tables de notre base de données
- insert.sql : Ce fichier permet d’insérer les données de notre jeu d’essai dans la base de données
- clean.sql : Ce fichier permet de supprimer les tables (et leurs données) de notre base de données

Nous avons effectué notre travail à l’aide d’Oracle 10g : il est donc recommandé d’utiliser cette version pour utiliser nos fichiers. Dedans, on crée et on remplit la base de données avec ces deux commandes successives :

```
start <chemin du fichier>/create.sql  
start <chemin du fichier>/insert.sql
```

On supprime les tables quand on en a plus besoin avec la commande :

```
start <chemin du fichier>/clean.sql
```

4.2 Création et suppression de tables

La création de tables suit rigoureusement l'ordre que nous avons donné dans le MLD. En effet, la définition de certaines tables implique la définition de clés étrangères, et il est impossible de définir une clé étrangère faisant référence à une table non existante.

La suppression de tables suit l'ordre inverse pour des raisons analogues : si on supprime une table et qu'un autre attribut y fait référence, il est impossible de supprimer cette table.

Les *NOT NULL* insérés dénotent des attributs obligatoires lors d'insertion de données. Autrement, des valeurs par défaut (notamment pour les dates : Questionnaire.qa_date et Session.ses_date ; par défaut : current_date) ont été définies pour le confort de l'utilisateur.

Des *ON DELETE SET NULL/CASCADE* ont été également mis en place pour l'utilisateur. Ainsi, il peut aisément supprimer un questionnaire et les questions/réponses seront automatiquement supprimées (*ON DELETE CASCADE*). Les attributs QCM.act_id et Evaluation.mod_id sont *ON DELETE SET NULL* pour une ultérieure et éventuelle mise à jour de référence.

4.3 Jeu d'essai

Nous fournissons dans ce jeu d'essai une liste d'élèves, une liste de programmes (ainsi que des modules et activités sous-jacentes), des questionnaires prédéfinis (reprise partielle du jeu de test effectué sur Excel) ainsi que des exemples de sessions.

Nous faisons apparaître les résultats des requêtes suivantes pour un aperçu non exhaustif et rapide du jeu d'essai :

```
SELECT * FROM Eleve;
```

ID	NOM	PRENOM	DATENAISS
1	Dos	Hugo	05-NOV-93
2	Tyndal	Pierre-Alexandre	27-JUN-93
3	Ingouff	Christian	28-APR-93
4	Simpson	Bart	07-APR-93
5	Simpson	Maggie	23-SEP-96
6	Monkey D.	Luffy	28-APR-92
7	Uzumaki	Naruto	25-DEC-93
8	Uchiha	Sasuke	03-FEB-93
9	Uchiha	Itachi	20-APR-91
10	San	Gohan	17-JUN-90
11	Elric	Edward	17-OCT-94
ID	NOM	PRENOM	DATENAISS
12	Elric	Alphonse	22-NOV-95
13	Van Damme	Jean-Claude	02-FEB-02
14	Kurosaki	Ichigo	11-NOV-92
15	San	Goten	14-JUL-95

FIGURE 4.1 – Liste d'élèves

```
SELECT P.libelle as programme, M.libelle as module, A.libelle as activite
FROM Activite A, Module M, Programme P
WHERE A.mod_id = M.id
AND M.prg_id = P.id
ORDER BY P.id ASC;
```

PROGRAMME	MODULE	ACTIVITE
CPI 1	Mathématiques	TD
CPI 1	Mathématiques	DM
CPI 1	Mathématiques	DS
CPI 2	Mathématiques	DS
CPI 2	Mathématiques	DM
CPI 2	Mathématiques	TD
ING 1	Mathématiques	TD
ING 1	Mathématiques	DM
ING 1	Mathématiques	DS
ING 1	Sport	Competition
ING 1	Transverse	Ninjutsu
PROGRAMME	MODULE	ACTIVITE
ING 1	Langues	Présentation orale
ING 1	Langues	DS
ING 2	Informatique	Projet
ING 2	Informatique	TD
ING 2	Informatique	DS
ING 3	Informatique	TD
ING 3	Informatique	Projet
ING 3	Informatique	DS

FIGURE 4.2 – Listes des activités, modules et programmes correspondants

```

SELECT qu.libelle as question, r.libelle as reponse
FROM Question qu, Reponse r
WHERE r.qu_id = qu.id
AND qu.id = 14;

```

```

QUESTION
-----
REPONSE
-----
Quel âge a l'EISTI ?
15 ans

Quel âge a l'EISTI ?
30 ans

Quel âge a l'EISTI ?
45 ans

QUESTION
-----
REPONSE
-----
Quel âge a l'EISTI ?
50 ans

Quel âge a l'EISTI ?
100 ans

```

FIGURE 4.3 – Enoncé de la question 14

```

SELECT e.nom as eleve, q.libelle as question, r.libelle as reponse
FROM Eleve e, Question q, Reponse r, Session_ s, rep_valider v
WHERE e.id = s.elv_id
AND v.ses_id = s.id
AND v.rep_id = r.id
AND r.qu_id = q.id
AND s.id = 2;

```

```

ELEVE
-----
QUESTION
-----
REPONSE
-----
Tyndal
Il est samedi soir. Que faites vous ?
Je sors pour faire des activités

Tyndal
Quel genres de jeux vidéo preferez-vous ?
Strategie(RTS, MOBA, FPS ...)

Tyndal
Quel genre de musique preferez-vous ?
Classique/Jazz

Tyndal
A l'EISTI, vous preferez...
Les mathématiques

Tyndal
A quelle fréquence faites-vous du sport ?
Une fois de temps en temps

```

FIGURE 4.4 – Réponses enregistrées dans la session 2

Chapitre 5

Conclusion

Nous avons désormais au terme de ce livrable une base de données concrète et complète qui nous permettra d'effectuer diverses opérations dans le prochain livrable, à savoir les requêtes d'interrogation.

Il est intéressant de noter la progression depuis la maquette Excel à la base de données Oracle, qui n'est pas forcément comme on aurait pu se l'imaginer, mais dont les résultats, au final, se conforment à nos attentes.

Finalement, c'est notre deuxième occasion (après celle rencontrée en CPI) de manipuler le langage SQL, ce qui ravive nos doigts et notre esprit technique.