

2012

EISTI



ANTOINE Sabri,
BELLUOT Vincent,
BENYIAT Mohamed,
GOURRAUD Nicolas

[GENIE LOGICIEL]

Il s'agit de refaire la gestion de la bibliothèque de notre école pour que les professeurs et les élèves puissent emprunter des documents de l'EISTI

Table des matières

Résumé des précédents livrables.....	3
Requêtes.....	5
Conclusion.....	10

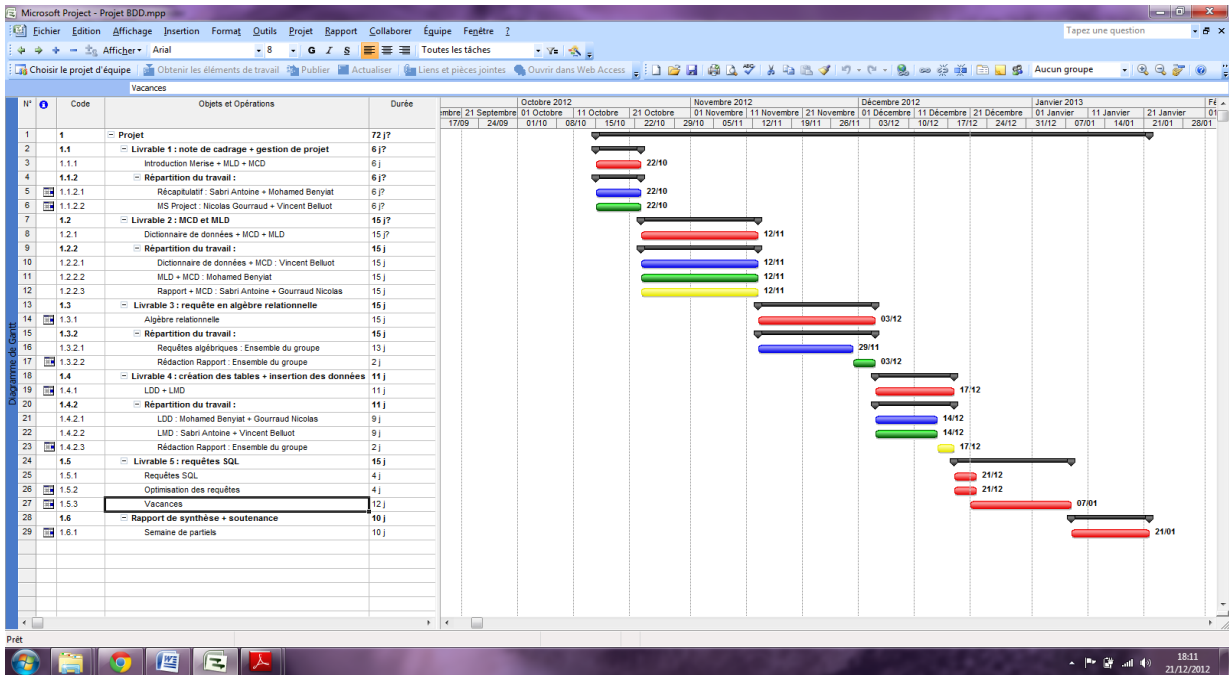
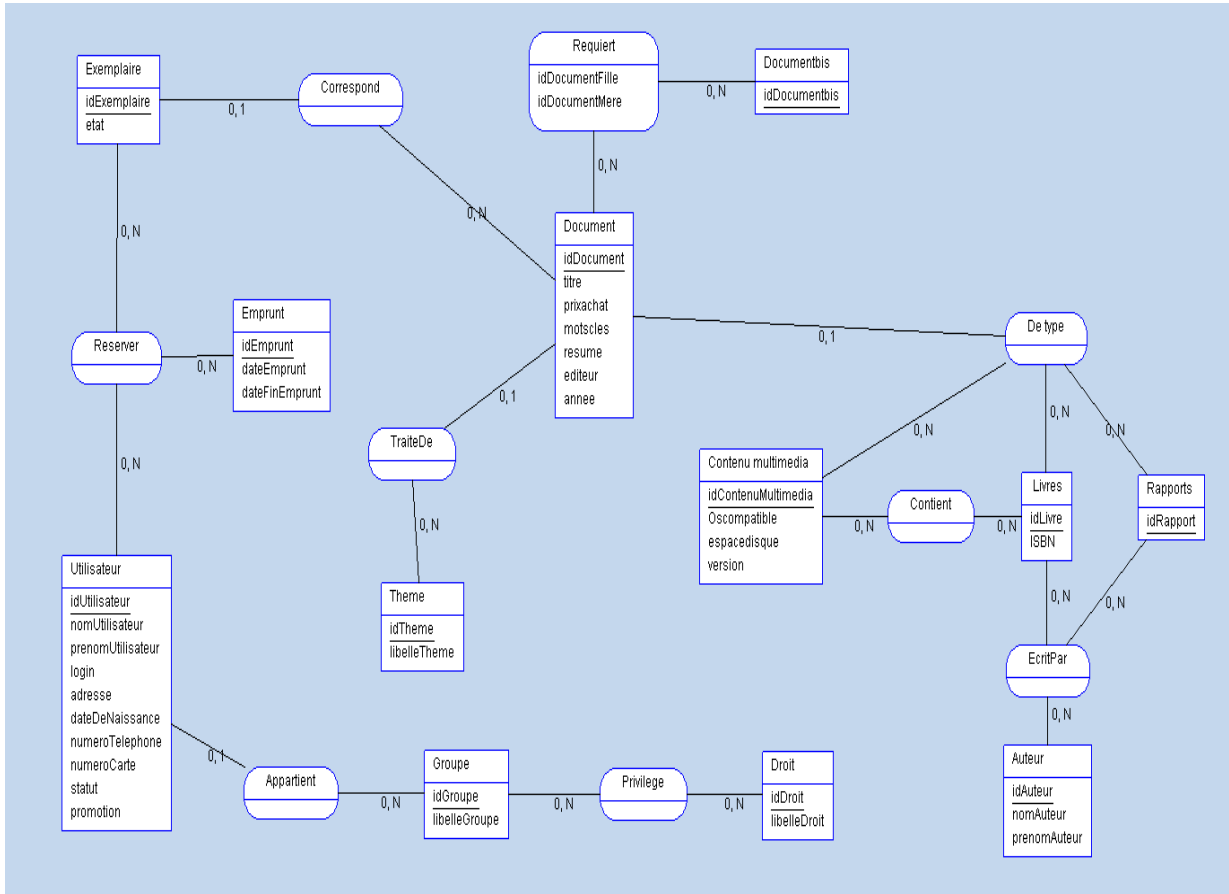
1) Résumé des précédents livrables

L'objectif de ce projet est la création d'une base de données permettant une gestion efficace de la bibliothèque de l'EISTI. Dans ce projet nous appliquerons nos connaissances acquises pendant le semestre. La portée du projet part du début de sa conception jusqu'à la réalisation de la base de données ainsi que des requêtes SQL permettant son traitement ainsi que son analyse.

Pour le premier livrable nous avons expliqué les fonctionnalités du programme ainsi que le planning en utilisant la méthode des SIXO. Dans le deuxième livrable, nous devons faire le Modèle Conceptuel des Données (MCD), puis le Modèle de Logique des Données (MLD) ainsi qu'une bibliothèque de données. Tout ceci afin de représenter la structure du système d'information, du point de vue des données et aussi pour définir les dépendances ou relations entre ces différentes données. Enfin dans un troisième livrable, nous avons travaillé avec l'algèbre relationnelle afin de représenter des requêtes sur la base de données dont le résultat s'exprime sous la forme d'une relation.

Pour ce livrable nous devons travailler sur le langage de définition des données (LDD) et le langage de manipulation des données (LMD). Le LDD s'utilise au moment de la constitution de la base de données et sert à décrire la nature des données telles qu'elles vont exister à l'aide des opérateurs CREATE, DROP et ALTER. Le LMD, quant à lui, permet d'effectuer des opérations sur les données, il concerne le cycle de vie des données dans une application et implémente le modèle « CRUDE ». On y utilise les opérateurs INSERT, UPDATE, DELETE et SELECT.

Voici notre MCD et notre MS Project (mis à jour) :



2) Nos requêtes

Nous sommes, parfois, amenés à supprimer une table d'une base de données pour une raison quelconque. La commande **DROP TABLE** de SQL est donc idéale pour la suppression de tables d'une base de données. On les met au début pour supprimer si les tables existent déjà.

```
DROP TABLE Refere ;
DROP TABLE Emprunter_un;
DROP TABLE associer ;
DROP TABLE EcritPar ;
DROP TABLE Contient ;
DROP TABLE Privilege ;
DROP TABLE Reserver ;
DROP TABLE De_type;
DROP TABLE Requier ;
DROP TABLE Theme;
DROP TABLE Emprunt ;
DROP TABLE Droit ;
DROP TABLE Utilisateur ;
DROP TABLE Contenu_multimedia ;
DROP TABLE Exemple ;
DROP TABLE Rapports ;
DROP TABLE Groupe ;
DROP TABLE Livres ;
DROP TABLE MotCle ;
DROP TABLE Mots ;
DROP TABLE Documentbis ;
DROP TABLE Document ;
DROP TABLE Auteur ;
```

Les tables représentent la structure de base pour stocker des données dans une base de données. Il n'est pas possible pour le fournisseur de bases de données de connaître d'avance vos besoins en matière de stockage de données, il est donc très probable que vous ayez à créer vous-même des tables dans la base de données.

```
CREATE TABLE Theme (
  idThemee INT CONSTRAINT pk_theme PRIMARY KEY,
  libelleTheme VARCHAR2(30));
```

```
CREATE TABLE Utilisateur(  
  idUtilisateur INT CONSTRAINT pk_utilisateur PRIMARY KEY,  
  nomUtilisateur VARCHAR2(20),  
  prenomUtilisateur VARCHAR2(20),  
  login VARCHAR2(20),  
  adresse VARCHAR2(30),  
  dateDeNaissance DATE DEFAULT SYSDATE,  
  numeroTelephone INT,  
  numeroCarte INT,  
  statut VARCHAR2(10),  
  promotion VARCHAR2(10),  
  num_groupe INT,  
  idGroupe_ce1 INT);
```

```
CREATE TABLE Groupe (  
  idGroupe INT CONSTRAINT pk_groupe PRIMARY KEY,  
  libelleGroupe VARCHAR2(20));
```

```
CREATE TABLE Livres (  
  idLivre INT CONSTRAINT pk_livre PRIMARY KEY,  
  ISBN VARCHAR(20));
```

```
CREATE TABLE Rapports (  
  idRapport INT CONSTRAINT pk_rapport PRIMARY KEY);
```

```
CREATE TABLE Contenu_multimedia (  
  idContenuMultimedia INT CONSTRAINT pk_contenuumultimedia PRIMARY KEY,  
  Ocompatible VARCHAR(20),  
  spacedisque INT,  
  version VARCHAR(10));
```

```
CREATE TABLE Exempleire (  
  idExempleire INT CONSTRAINT pk_exempleire PRIMARY KEY,  
  etat VARCHAR(10),  
  idDocument_c5 INT);
```

```
CREATE TABLE Emprunt (  
  idEmprunt INT CONSTRAINT pk_emprunt PRIMARY KEY,  
  dateEmprunt DATE DEFAULT SYSDATE,  
  dateFinEmprunt DATE DEFAULT SYSDATE);
```

```
CREATE TABLE Droit (  
  idDroit INT CONSTRAINT pk_droit PRIMARY KEY,  
  libelleDroit VARCHAR(20));
```

```
CREATE TABLE Document (  
  idDocument INT CONSTRAINT pk_document PRIMARY KEY,  
  titre VARCHAR(20),  
  prixachat FLOAT,  
  resume VARCHAR(100),  
  editeur VARCHAR(20),  
  annee INT,  
  idTheme_c5 INT);
```

```
CREATE TABLE Auteur (  
  idAuteur INT CONSTRAINT pk_auteur PRIMARY KEY,  
  nomAuteur VARCHAR(20),  
  prenomAuteur VARCHAR(20));
```

```
CREATE TABLE Documentbis (  
  idDocumentbis INT CONSTRAINT pk_documentbis PRIMARY KEY);
```

```
CREATE TABLE MotCle (  
  idMotCle INT CONSTRAINT pk_motcle PRIMARY KEY,  
  mot VARCHAR(50));
```

```
CREATE TABLE Mots (  
  idMot INT CONSTRAINT pk_mot PRIMARY KEY,  
  Mot VARCHAR(20));
```

```
CREATE TABLE Requierit (  
  idDocument_c1 INT NOT NULL,  
  idDocumentbis_c1 INT NOT NULL,  
  PRIMARY KEY (idDocument_c1,idDocumentbis_c1)) ;
```

```
CREATE TABLE Reserver (  
  idEmprunt_c1 INT NOT NULL,  
  idUtilisateur_c1 INT NOT NULL,  
  PRIMARY KEY (idEmprunt_c1,idUtilisateur_c1) ) ;
```

```
CREATE TABLE De_type (  
  idContenuMultimedia_c1 INT NOT NULL,  
  idLivre_c1 INT NOT NULL,  
  idRapport_c1 INT NOT NULL,  
  idDocument_c1 INT NOT NULL,  
  PRIMARY KEY (idContenuMultimedia_c1,idLivre_c1,idRapport_c1,idDocument_c1)) ;
```

```
CREATE TABLE Privilege (  
  idGroupe_c1 INT NOT NULL,  
  idDroit_c1 INT NOT NULL,  
  PRIMARY KEY (idGroupe_c1,idDroit_c1)) ;
```

```
CREATE TABLE Contient (  
  idLivre_c2 INT NOT NULL,  
  idContenuMultimedia_c2 INT NOT NULL,  
  PRIMARY KEY (idLivre_c2,idContenuMultimedia_c2)) ;
```

```
CREATE TABLE EcritPar (  
  idLivre_c2 INT NOT NULL,  
  idRapport_c2 INT NOT NULL,  
  idAuteur_c1 INT NOT NULL,  
  PRIMARY KEY (idLivre_c2,idRapport_c2,idAuteur_c1)) ;
```

```
CREATE TABLE Refere (  
  idDocument_c3 INT NOT NULL,  
  idmot_c2 INT NOT NULL,  
  PRIMARY KEY (idDocument_c3,idmot_c2)) ;
```

```
CREATE TABLE Emprunter_un (  
  idExemplaire_c1 INT NOT NULL,  
  idEmprunt_c2 INT NOT NULL,  
  PRIMARY KEY (idExemplaire_c1,idEmprunt_c2)) ;
```

```
CREATE TABLE associer (  
  idMot_c1 INT NOT NULL,  
  idDocument_c2 INT NOT NULL,  
  PRIMARY KEY (idMot_c1,idDocument_c2)) ;
```


Vous pouvez être amené à modifier à n'importe quel moment la structure d'une table créée dans la base de données. Les raisons classiques de modification d'une table sont les suivantes :

- Ajouter une colonne
- Supprimer une colonne
- Changer un nom de colonne
- Changer les types de données d'une colonne

La commande ALTER TABLE permet aussi de modifier la structure d'une table (changer les spécifications de la clé primaire ou ajouter une contrainte unique à une colonne).

```
ALTER TABLE Utilisateur ADD CONSTRAINT FK_Utilisateur_idGroupe FOREIGN KEY (idGroupe_ce1) REFERENCES Groupe (idGroupe);
ALTER TABLE Exemple ADD CONSTRAINT FK_Exemple_idDocument FOREIGN KEY (idDocument_c5) REFERENCES Document (idDocument);
ALTER TABLE Document ADD CONSTRAINT FK_Document_idTheme FOREIGN KEY (idTheme_c5) REFERENCES Theme (idTheme);
ALTER TABLE Requier ADD CONSTRAINT FK_Requier_idDocument FOREIGN KEY (idDocument_c1) REFERENCES Document (idDocument);
ALTER TABLE Requier ADD CONSTRAINT FK_Requier_idDocumentbis FOREIGN KEY (idDocumentbis_c1) REFERENCES Documentbis (idDocumentbis);
ALTER TABLE Reserver ADD CONSTRAINT FK_Reserver_idEmprunt FOREIGN KEY (idEmprunt_c1) REFERENCES Emprunt (idEmprunt);
ALTER TABLE Reserver ADD CONSTRAINT FK_Reserver_idUtilisateur FOREIGN KEY (idUtilisateur_c1) REFERENCES Utilisateur (idUtilisateur);
ALTER TABLE De_type ADD CONSTRAINT FK_De_type_idCM FOREIGN KEY (idContenuMultimedia_c1) REFERENCES Contenu_multimedia(idContenuMultimedia);
ALTER TABLE De_type ADD CONSTRAINT FK_De_type_idLivre FOREIGN KEY (idLivre_c1) REFERENCES Livres (idLivre);
ALTER TABLE De_type ADD CONSTRAINT FK_De_type_idRapport FOREIGN KEY (idRapport_c1) REFERENCES Rapports (idRapport);
ALTER TABLE De_type ADD CONSTRAINT FK_De_type_idDocument FOREIGN KEY (idDocument_c1) REFERENCES Document (idDocument);
ALTER TABLE Privilege ADD CONSTRAINT FK_Privilege_idGroupe FOREIGN KEY (idGroupe_c1) REFERENCES Groupe (idGroupe);
ALTER TABLE Privilege ADD CONSTRAINT FK_Privilege_idDroit FOREIGN KEY (idDroit_c1) REFERENCES Droit (idDroit);
ALTER TABLE Contient ADD CONSTRAINT FK_Contient_idLivre FOREIGN KEY (idLivre_c2) REFERENCES Livres (idLivre);
ALTER TABLE Contient ADD CONSTRAINT FK_Contient_idCM FOREIGN KEY (idContenuMultimedia_c2) REFERENCES Contenu_multimedia(idContenuMultimedia);
ALTER TABLE EcritPar ADD CONSTRAINT FK_EcritPar_idLivre FOREIGN KEY (idLivre_c2) REFERENCES Livres (idLivre);
ALTER TABLE EcritPar ADD CONSTRAINT FK_EcritPar_idRapport FOREIGN KEY (idRapport_c2) REFERENCES Rapports (idRapport);
```

```
ALTER TABLE EcritPar ADD CONSTRAINT FK_EcritPar_idAuteur FOREIGN KEY (idAuteur_c1) REFERENCES Auteur (idAuteur);
ALTER TABLE Refere ADD CONSTRAINT FK_Refere_idDocument FOREIGN KEY (idDocument_c3) REFERENCES Document (idDocument);
ALTER TABLE Refere ADD CONSTRAINT FK_Refere_idmot FOREIGN KEY (idmot_c2) REFERENCES MotCle (idMotCle);
ALTER TABLE Emprunter_un ADD CONSTRAINT FK_Emprunter_un_idExemplaire FOREIGN KEY (idExemplaire_c1) REFERENCES Exemplaire(idExemplaire);
ALTER TABLE Emprunter_un ADD CONSTRAINT FK_Emprunter_un_idEmprunt FOREIGN KEY (idEmprunt_c2) REFERENCES Emprunt (idEmprunt);
ALTER TABLE associer ADD CONSTRAINT FK_associer_idMot FOREIGN KEY (idMot_c1) REFERENCES Mots (idMot);
ALTER TABLE associer ADD CONSTRAINT FK_associer_idDocument FOREIGN KEY (idDocument_c2) REFERENCES Document (idDocument);
```

3) Conclusion

Avec les cours que nous avons eues en cours en base de donnée et de ce que nous avons fait dans nos précédents livrable, nous avons pu commencer à coder notre projet. Nous avons ainsi pu faire toutes nos requêtes et voir si elles allaient bien dans notre projet. Le fait de commencer à coder nous permet de voir les choses plus concrètement et ainsi, de bien avancer dans notre projet et pour avoir une bonne base de donnée.