

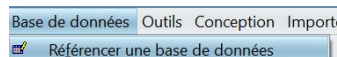
Études de cas (ESSAIS, ESSAIS+) sur IBEasy+ voire sur Looping

Découverte de IBEasy+

IBEasy+ (M Grange, janvier 2022) est un logiciel client gratuit d'un serveur Firebird qui peut fonctionner en mode partagé accessible via le port par défaut 3050, ce qui permet à plusieurs logiciels clients (LibreOffice Base, Apache OpenOffice Base, Flamerobin, IBEasy+) de s'y connecter ou en mode embarqué (exclusif, IBEasy+ étant alors le seul client). Firebird est un fork de Borland Interbase qui a été un temps open-source. Firebird est donc un SGBDR qui accepte des instructions SQL de la part d'un client graphique à la MS Access 2003 qui a la grande originalité d'offrir en son sein un module pour la bonne conception d'une base de données s'appuyant sur un dictionnaire de données qui se focalise sur la description de la donnée pour la rattacher à une catégorie (future relation). Ce rattachement peut être aidé par le logiciel s'il est configuré pour le français ou l'anglais. Les dépendances fonctionnelles du domaine pilotent les relations entre catégories. Une fonction d'export vers Microsoft Access fait de IBEasy+ avec Firebird embarqué un Atelier de Génie Logiciel *lite* pour Microsoft Access même s'il est aussi une alternative gratuite offrant des éditeurs pour les tables, les vues, les domaines, les attributs auto-incrémentés, les déclencheurs, les fonctions externes et les procédures stockées. Le requêtage peut se baser sur le SQL ou le QBE.

Pour bénéficier du module conception de base de données, il faut créer une nouvelle base de données en tant que : **Nom d'utilisateur** : IBEASY **Mot de passe** : ibeasyuser (le logiciel met ce mdp pour vous).

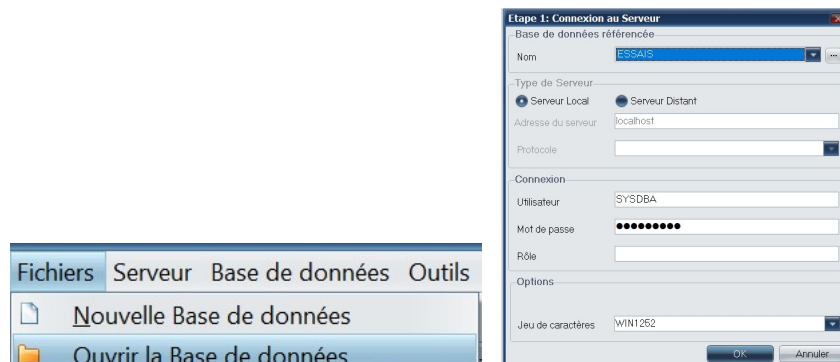
Pour ouvrir une base de données Firebird existante dans IBEasy+, il faut d'abord la référencer avec un alias :



Le bouton **Parcourir** permet de sélectionner le fichier *.FDB atteignable par le serveur Firebird (embarqué ou non). Il faut corriger l'**alias** puis appuyer sur le bouton **OK** pour que l'alias figure dans la liste des alias connus (à gauche). Quand les alias pour les bases de données Firebird ont été définis, vous pouvez appuyer sur **Quitter**. Les boutons **Nouvel alias**, **Tester la connexion** et **Supprimer l'alias** sont au dessus des alias connus.



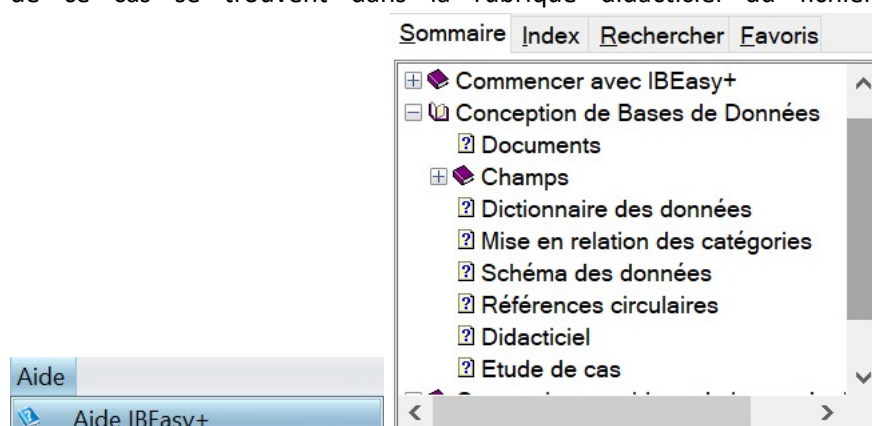
Pour créer ou ouvrir une base de données Firebird, il faut aller dans le menu **Fichier**.



Pour l'ouverture, on peut choisir un alias connu. Ce n'est pas le cas lors de la création mais un sélecteur de fichier vous permettra de préciser où créer le fichier de base de données Firebird vide.

Didacticiel et étude de cas Essais

Étudiez l'étude de cas présente dans le fichier d'aide du logiciel IBEasy+. Les grandes étapes de la résolution de ce cas se trouvent dans la rubrique didacticiel du fichier d'aide que vous devez faire.



À noter que l'étude de cas est aussi disponible [ici](#). Dans cette page, le dictionnaire de données présenté est un dictionnaire sur papier et ne correspond pas exactement au dictionnaire à entrer dans IBEasy+.

Rappel du sujet

La gestion d'étudiants qui suivent les différents enseignements d'un diplôme.

Les documents de travail utilisés sont les suivants :

Liste des étudiants (doc 1)			
N° étudiant	Nom de l'étudiant	Date de naissance	Sexe
1	Dupont, Charles	18-03-1981	H (Homme)
2	Dubois, Jules	02-11-1982	H (Homme)
3	Favier, Isabelle	02-02-1979	F (Femme)
...

Relevé de notes (doc 2)

N° étudiant : 1 Dupont, Charles			
N° matière	Nom	Coefficient	Note / 20
1	Mathématiques	3	10
2	Informatique	2	9
3	Sociologie	2	12,5
4	Histoire	1	13
5	Géographie	1	7
Moyenne Générale	10,3		

Liste des enseignants (doc 3)				
N° enseignant	Nom	Grade	Ancienneté	Matière enseignée
10	Bertrand, Pierre	ASS	2	Sociologie
11	Dupont, Auguste	MCF	3	Mathématiques
...
15	Simon, Etienne	ASS	5	Histoire Géographie

Règles de gestion :

- Une matière n'est enseignée que par un seul enseignant.
- Un étudiant n'a qu'une note par matière.

Dictionnaire de données

Partant des documents, la phase d'analyse consiste à en extraire les informations élémentaires (non décomposables) qui vont constituer la future base de données.

La réunion de l'ensemble des données élémentaires, que l'on appelle des attributs ou des champs, constitue le dictionnaire des données. Chaque attribut (champ) du dictionnaire de données peut être caractérisé par les propriétés suivantes :

Propriété	Signification
Mnémonique	Abréviation du nom de l'attribut.
Libellé	Libellé contenant la signification précise et le rôle de l'attribut.
Type de donnée	Type de l'attribut : entier, réel, chaîne de caractères, date...
Contraintes d'intégrité	Liste des contraintes sur les valeurs possibles de l'attribut
Règle de calcul	Règle de calcul (d'obtention) de l'attribut correspondant.

Le dictionnaire des données relatifs aux documents précédents est le suivant :

Mnémonique	Libellé	Type	Contraintes	Règle de calcul
ANCIEN	Ancienneté dans le grade	Entier	≥ 0	
CDSEXE	Code sexe	Chaîne(1)	H ou F	
COEFF	Coefficient de la matière	Entier	> 0 et < 6	
DTNAISS	Date de naissance	Date		
GRADE	Grade de l'enseignant	Chaîne(4)	VAC, PRCE, LRU, PRAG, ASS ou MCF ou PR	
LBSEXE	Libellé du sexe	Chaîne(7)	Homme ou Femme	
MOYENNE	Moyenne au diplôme	Réel	≥ 0 et ≤ 20	$\text{SUM}(\text{NOTE} * \text{COEFF}) / \text{SUM}(\text{COEFF})$
NOMAT	Nom de la matière	Chaîne(15)		
NOMENS	Nom de l'enseignant	Chaîne(20)		

NOMETU	Nom de l'étudiant	Chaîne(20)		
NOTE	Note obtenue à la matière	Réel	≥ 0 et ≤ 20	
NUMAT	N° de la matière	Entier	> 0	
NUMENS	N° de l'enseignant	Entier	> 0	
NUMETU	N° de l'étudiant	Entier	> 0	

Consignes sur la prise en main du logiciel

Jusqu'à la version 2.4 incluse d'IBEasy+, avant la validation de la conception, il n'était pas possible d'ajouter les contraintes aux attributs. Il est souhaitable maintenant de les ajouter dès la conception.

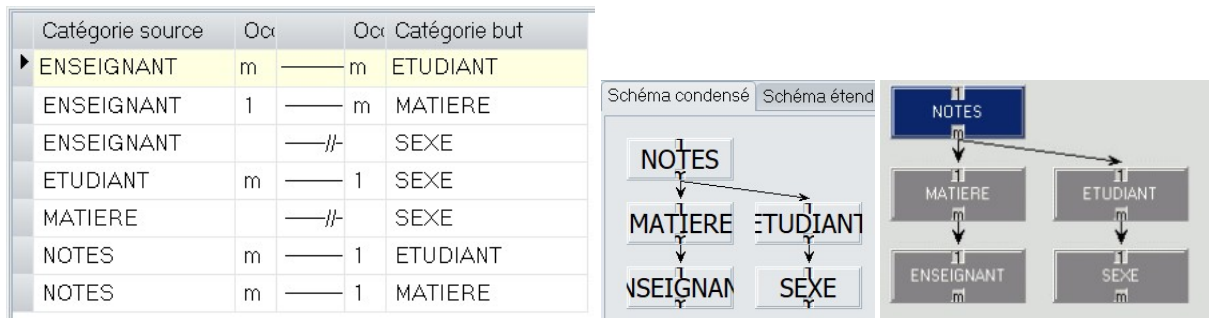
Remarques concernant la longueur des mnémoniques et des catégories du dictionnaire des données : MS Access accepte des noms de tables et des noms de colonnes ayant jusqu'à 64 caractères tandis qu'une table ne peut avoir plus de 255 attributs. Firebird 3.0 embarqué dans la distribution IBEasy+ accepte des noms de tables et des noms de colonnes ayant jusqu'à 31 caractères tandis qu'une table n'a qu'une taille maximale. A noter que Firebird 4.0 passe de 31 caractères à 63 caractères. Cependant la limite des 31 caractères doit prendre en compte les caractères accentués gérés par plusieurs octets avec Firebird (utf8 codé sur 4 octets). La gestion des accents nécessite le choix d'un jeu de caractères. Les plus utilisés sont Win1252 (utilisé en interne par les applications clientes développées en Delphi) et UTF8 (mieux supporté sous OS Linux donc aussi sous OS MacOSX donc jeu de caractères utilisé en interne par l'exécutable des applications clientes développées par Lazarus, aussi pour Flamerobin). Il faut toujours se rappeler que le serveur Firebird ne fait le transcodage que si les jeux de caractères de connexion et de codage dans le champ de la base sont différents, émettant une erreur en refusant les caractères appartenant à un seul des 2 jeux. Par contre si les 2 jeux sont identiques le serveur ne fait aucun contrôle, autorisant l'écriture de caractères inexistant dans le jeu utilisé dans le champ. La base peut alors devenir inutilisable depuis un logiciel utilisant un jeu de caractères de connexion différent.

Avant d'appuyer sur le bouton **Valider** en bas de l'onglet **Base de données** du module conception de base de données, vous sauvegarderez une copie de la base de données Firebird sous le nom ESSAIS+.FDB qui au mieux sera la base de départ de la conception de l'affinement de l'étude de cas.

Après description des champs du document, les différentes catégories du document doivent être mise en relation.

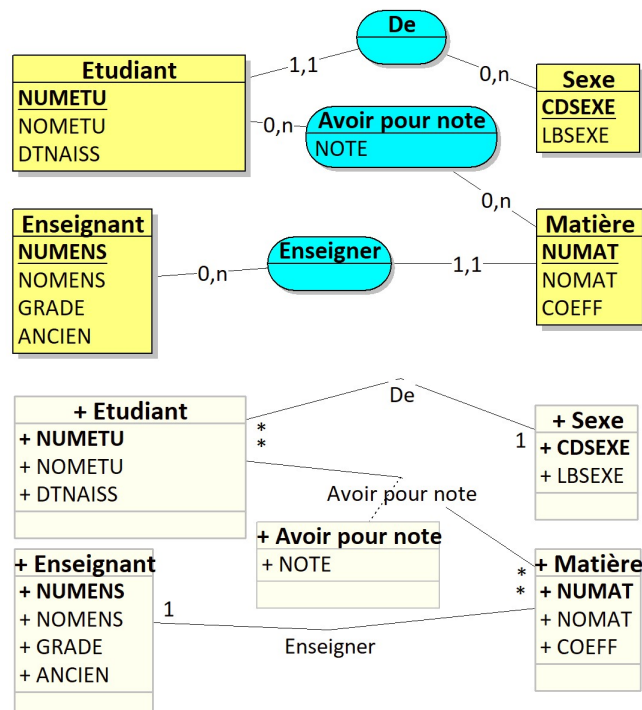
Il y a 4 types de relations possibles :

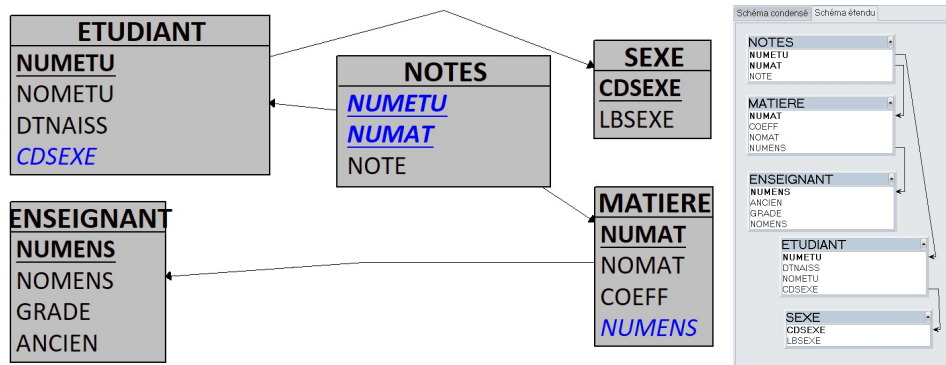
- 1-1 : À une occurrence de la catégorie source on ne peut associer qu'une seule occurrence de la catégorie but et vice-versa.
- 1-m : À une occurrence de la catégorie source on peut associer plusieurs (m pour *many*) occurrences de la catégorie but, mais à une occurrence de la catégorie but on ne peut associer qu'une occurrence de la catégorie source.
- m-m : À une occurrence de la catégorie source on peut associer plusieurs (m pour *many*) occurrences de la catégorie but et vice-versa.
- Pas de relation (----//----) : il n'existe aucune liaison (relation) entre la catégorie source et la catégorie but.



Le didacticiel nous a fait créer la catégorie composée nommée Notes. L'ensemble des descripteurs rattachés à la catégorie composée, constituant le but, dépend fonctionnellement de manière élémentaire et directe de la même source composée des identifiants des catégories définissant cette catégorie composée. Aussi les relations entre la catégorie composée et celles qui la définissent sont gérées automatiquement sans possibilité d'édition.

Les relations entre catégories permettent de déduire dans IBEasy+, le graphe des dépendances fonctionnelles interrelations (ordre de créations des tables si pas de cycle) et le modèle relationnel.





La structuration des données du dictionnaire des données s'effectue en 5 étapes :

1. Détermination de la liste des DF simples
2. Prise en compte des attributs non classés dans l'étape 1 et détermination des DF composées
3. Élimination des transitivités du schéma des DF
4. Construction, à partir des DF simples, des entités de la base de données
5. Construction, à partir des DF composées, des associations de la base de données

L'application de cette démarche à notre exemple nous donne les résultats suivants :

1-DF simples de la base de données ETUDIANTS

Numetu -> Nometu, Dtnaiss, Cdsexe, Lbsexe

Cdsexe -> Lbsexe

Numens -> Nomens, Grade, Ancien

Numat -> Nomat, Coeff, Numens, Nomens, Grade, Ancien

2-Attributs non classés et DF composées de la base de données ETUDIANTS

L'attribut "note" n'a pas été classé dans l'étape précédente, il intervient donc dans une DF composée. Cette DF est la suivante :

Numetu, Numat -> Note

3-Élimination des transitivités de l'ensemble des DF

Numetu -> Nometu, Dtnaiss, Cdsexe

Cdsexe -> Lbsexe

Numens -> Nomens, Grade, Ancien

Numat -> Nomat, Coeff, Numens

4-Construction des entités

ETUDIANT (Numetu, Nometu, Dtnaiss, Cdsexe)

SEXE (Cdsexe, Lbsexe)

ENSEIGNANT (Numens, Nomens, Grade, Ancien)

MATIERE (Numat, Nomat, Coeff, Numens)

5-Construction des associations

NOTES (Numetu, Numat, Note)

Il convient aussi de rechercher s'il n'existe pas de table de correspondance entre deux attributs par ailleurs clé primaire de relations qui serait d'intérêt pour le domaine étudié et qui aurait pu figurer dans le modèle Entité-Association comme association n :m non porteuse d'attribut.

À la suite de la validation de la conception, l'ordre des attributs dans une table Firebird nouvellement créée est le suivant :

- les parties de la clé primaire ;
- les attributs hors clé dans l'ordre lexicographique ;
- les clés étrangères.

Il convient de réorganiser les attributs ordonnés lexicographiquement dans un ordre plus parlant : le libellé ou la désignation (pour un objet) le nom (pour une personne) d'abord. Il y a des attributs comme « ancienneté dans le grade » qui complètent d'autre attribut (le grade) et qui logiquement sont après l'attribut qu'ils complètent.

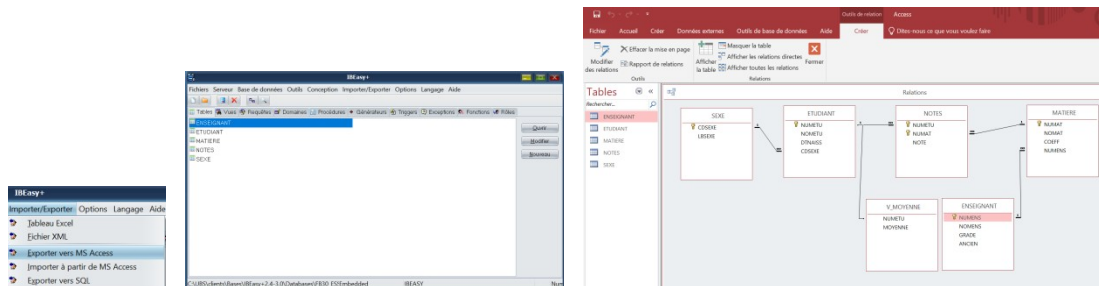
Après avoir validé la conception, on peut ajouter les contraintes dans le dictionnaire de données. La partie correspondant au dictionnaire de données dans un document ne permettait pas de déclarer une contrainte avant la version 2.5 alors que le dictionnaire de données indépendant via **Conception / Dictionnaire de données** le permet. De plus, vous pouvez ajouter des contraintes complexes de tables utilisant plusieurs attributs de la table.

Champs	Utilisé dans...	Description SQL			
Mnémonique	Description	Domaine	Défaut	NUL	Contraintes
▶ ANCIEN	Ancienneté de l'enseignant	SMALLINT		X	value>=0
CDSEXE	Code sexe de l'étudiant	CHAR(1) character set 'WIN1252'		X	value='H' OR value='F'
COEFF	Coefficient de la matière	SMALLINT		X	value>0 AND value<6
DTNAISS	Date naissance de l'étudiant	DATE		X	
GRADE	Grade de l'enseignant	CHAR(4) character set 'WIN1252'		X	value in ('VAC','PRCE','PRAG','LRU','ASS','ATER','MCF','PAST','PR')
LBSEXE	Libellé du code sexe de l'étudiant	CHAR(7) character set 'WIN1252'		X	value in ('Homme','Femme')
MOYENNE	Moyenne de l'étudiant	DECIMAL(18,2)		X	value>=0 AND value<=20
NOMAT	Nom de la matière	CHAR(20) character set 'WIN1252'		X	
NOMENS	Nom de l'enseignant	CHAR(20) character set 'WIN1252'		X	
NOMETU	Nom de l'étudiant	CHAR(20) character set 'WIN1252'		X	
NOTE	Note obtenue par l'étudiant dans la matière	DECIMAL(18,2)		X	value>=0 AND value<=20
NUMAT	Numéro de la matière	SMALLINT		X	value>0
NUMENS	Numéro de l'enseignant	SMALLINT		X	value>0
NUMETU	Numéro de l'étudiant	SMALLINT		X	value>0

Exportation vers MS Access

Le serveur de données Firebird met à disposition un pilote ODBC 32 bits ou 64 bits pour les applications clientes. La distribution Microsoft Access 2019 étant 64 bits, il faut installer le pilote 64 bits ODBC mais pour cela il faut avoir des privilèges d'administration sur le poste de travail. De toute manière, l'importation de données Firebird ou le liage de tables Firebird nous garantit des types les plus proches mais ni les clés primaires et les clés étrangères ne sont reconnus. IBEasy+ 2.4 en utilisant le composant Ace Oledb 12.0, permet de créer un fichier *.mdb qui contiendra les tables bien conçues avec les bons types, les clés primaires et les clés étrangères. IBEasy+ 2.5.2 (février 2023) permet aussi de récupérer le caractère obligatoire ou non, les contraintes valides. Cependant ni les descriptions de tables, ni celles des colonnes

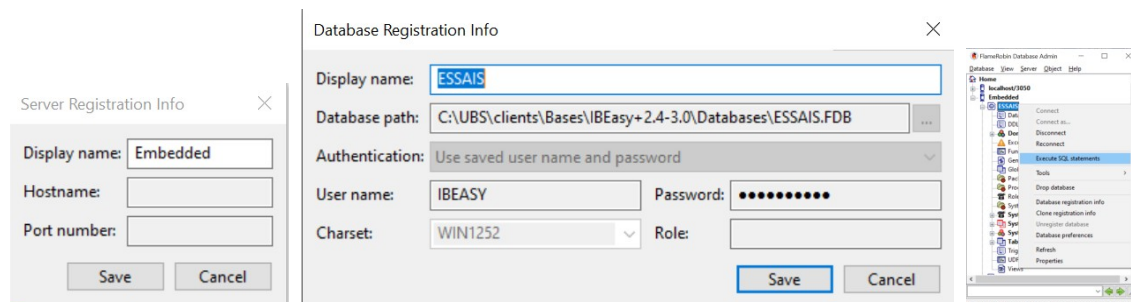
ne sont exportées. Avec Office 2007, Microsoft a remplacé ses fichiers de données d'Office qui étaient dans des formats binaires propriétaires (*.mdb) par des fichiers à la norme OpenXML (accdb). Ils en ont profité pour remplacer le moteur JET par le moteur ACE. Une fois le fichier mdb ouvert, vous pouvez le sauvegarder en accdb.



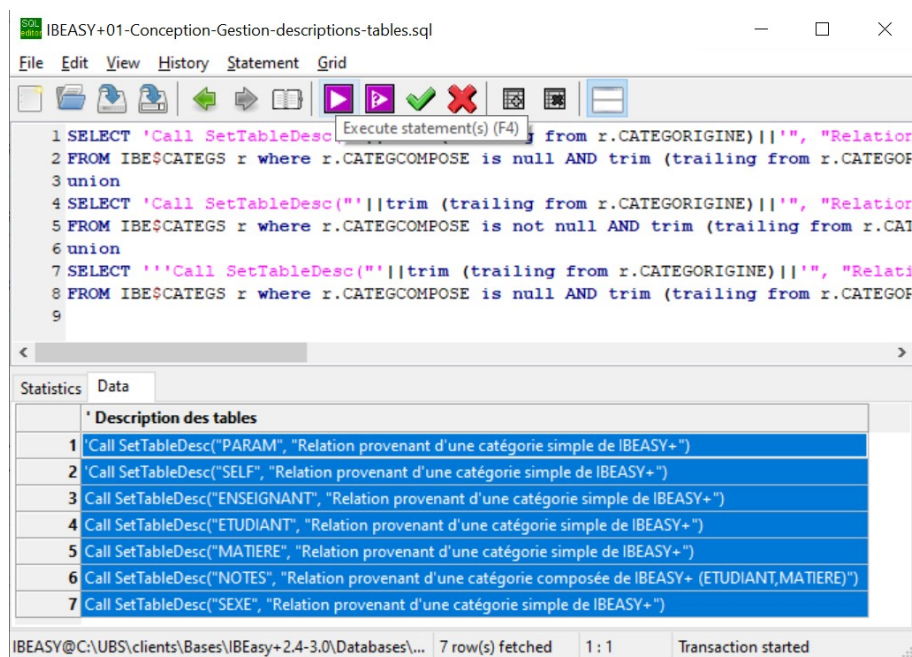
Compléments à l'exportation vers MS Access

Le module VBA contenu dans le fichier `Databases\ExportFromIBEasy+.bas` de la distribution Firebird 3.0 embarquée pour les utilitaires IBEasy+ 2.4 ou 2.5 ou 3.2 et Flamerobin 0.9.3.6 contient les routines VBA (procédures et fonctions) qui permettent de compléter la spécification de la base de données MS Access par rapport à celle de Firebird produite avec IBEasy+. De plus le répertoire `Databases` d'IBEasy+ contient les requêtes pour générer les appels effectifs à ces routines pour perfectionner l'export précédent.

Via l'utilitaire Flamerobin embarqué comme IBEasy+, connectez-vous à la base de données Firebird précédemment créée/modifiée par IBEasy+ (utilitaire qu'il a fallu quitter pour que Flamerobin utilise en mode exclusif/embarqué le serveur Firebird). Pour cela, il faut définir le serveur embarqué (qui n'a pas d'hôte, ni de port) puis enregistrer les informations de connexion à la base `ESSAIS`.



Une fois connecté à la base de données Essais, on la sélectionne dans l'arborescence de Flamerobin, on fait un clic droit et on utilise l'instruction **Execute SQL statements** du menu contextuel. Une fenêtre s'ouvre. Dans son menu File, utilisez l'instruction **Open...** pour charger le premier script `Databases\IBEasy+??-Conception-Gestion-*.sql`, c'est-à-dire `IBEasy+01-Conception-Gestion-descriptions-tables.sql`. Exécutez la commande SQL contenue : elle génère des appels effectifs à la procédure `SetTableDesc` du module VBA `ExportFromIBEasy+` à importer dans l'éditeur VBA après avoir ouvert dans le logiciel Microsoft Access la base de données `ESSAIS.mdb` créée via l'export depuis IBEasy+.



Dans Microsoft Access, la base de données étant ouverte et le contenu étant validé, l'onglet **Outils de bases de données** nous permet d'ouvrir l'éditeur VBE via l'instruction **Visual Basic**. On peut alors ajouter à notre fichier mdb ou accdb le module `ExportFromIBEasy+` via la commande **Importer un fichier...** du menu **Fichier**.

La commande **Affichage / Fenêtre d'exécution** permet d'ajouter en bas de l'écran la fenêtre où s'affichent les messages des instructions VBA `Debug.Print`.

Il convient de compléter la procédure `try()` des instructions générées par l'exécution dans flamerobin des 20 scripts `IBEasy+??-Conception-Gestion-*.sql`.

Une fois que le contenu de `try` vous semble bon, vous pouvez demander l'exécution de la procédure `try()` du module `ExportFromIBEasy+` afin d'obtenir une meilleure synchronisation de la base de données MS Access par rapport à celle de Firebird.

La base de données MS Access par rapport à celle de Firebird qui a été la base de l'exportation :

- possède les type de données les plus proches (Export natif IBEasy+) ;
- voit ses clés primaires spécifiées (Export natif IBEasy+) ;
- voit ses clés étrangères spécifiées (Export natif IBEasy+) ;
- voit les données de ses tables importées (Export natif IBEasy+) ;
- voit les descriptions de tables générées en fonction des catégories d'IBEasy+ (script 01) ;
- voit les descriptions de colonnes importées du dictionnaire des données d'IBEasy+ (script 02) ;
- voit les spécifications de colonnes NOT NULL importées du dictionnaire des données d'IBEasy+ (script 03) ;
- voit les spécifications de colonnes textuelles n'autorisant pas une longueur à zéro importées du dictionnaire des données d'IBEasy+ (script 04) ;
- voit les spécifications de colonnes acceptant une valeur par défaut importées du dictionnaire des données d'IBEasy+ (script 05) ;
- voit les spécifications de colonnes ayant une règle de validation importées du dictionnaire des données d'IBEasy+ (script 06) ;

- voit les spécifications de tables ayant une règle de validation importées de la métabase Firebird (script 07) ;
- voit les listes de choix statiques attachées à certaines colonnes déduites du dictionnaire des données d'IBEasy+ (script 08) ;
- voit les listes de choix dynamiques attachées aux clés étrangères déduites de la métabase Firebird (script 09) ;
- voit les attributs calculés du dictionnaire de données d'IBEasy+ donner lieu à la création de vues (script 10) ;
- voit l'ajout de l'attribut IDPARAM à l'éventuelle table PARAM avec ses contraintes pour garantir la présence d'une unique ligne dans cette table (script 11).
- voit les déclencheurs avant pour insertion ou modification de table donner lieu à la création de contraintes CHECK via ADODB (script 12)
- voit les index uniques ajoutés après la validation conception pour la prise en compte des clés candidates (script 13)
- voit la définition des clés étrangères composées (script 14-0)
- voit les clés étrangères composées que d'attributs significatifs corrigées après la validation donner lieu à des listes de choix dynamiques en cascade (dépendantes entre elles) (script 14-1)
- voit les clés étrangères composées se voir adjoindre un index composé afin de faciliter la vérification de l'intégrité référentielle (scripts 14-2-drop, 14-2)
- voit les clés primaires numériques simple donner lieu à une clé adhoc auto-incrémentée (script 15)
- voit les clés étrangères simples se voir adjoindre un index simple afin de faciliter la vérification de l'intégrité référentielle (script 16)
- voit la colonne significative ou colonne à afficher des tables être des clés secondaires (script 17).

Affinement de l'étude de cas

Par rapport à l'étude de cas proposée dans le fichier d'aide, voici un complément du dictionnaire de données concernant la gestion des contrôles :

NUMETU	Numéro de l'étudiant qui fait le contrôle
NOMAT	Nom de la matière
NUMETUBI	Numéro de l'étudiant qui constitue le binôme de l'étudiant
DTCONTROLE	Date du contrôle, par défaut date du jour
ANNEEUNIV	Année universitaire en cours
HRCONTROLE	Heure de début du contrôle
TYPECONTROLE	Type de contrôle (CC, CT)
DUREECONTROLE	Durée en minute du contrôle
SESSION	Numéro de session : 1 ou 2 (rattrapage)
REFSALLE	Identification de la salle
LBSALLE	Salle où a lieu le contrôle (KB 030, DSEG S216, YC D070, ENSIBS D101, etc.)
PLACE	Place dans la salle affectée pour le contrôle à l'étudiant
CONSIGNE	Consigne sur les documents et matériels (autorisés, non autorisés)
CAPACITE	Nombre maximal d'étudiant dans la salle en configuration pour un examen
DUREEMINDEF	Durée minimale de composition définie de manière uniforme à 30 minutes
DUREEMINTMP	Durée minimale avant départ temporaire (uniformément à 120 minutes)

NOMSURVEILLANT	Nom du responsable de la surveillance de la salle
OBSERVATIONS	Observations faites par le surveillant concernant le déroulé de l'épreuve
PRESENCE	L'étudiant est-il présent lors de l'épreuve (O/N) ?
NB_INSCRITS	Nombre d'étudiants inscrits dans la salle = SELECT COUNT(NUMETU) FROM ETUDIANT
NB_PRESENTS	Nombre d'étudiants présents dans la salle = SELECT COUNT(NUMETU) FROM ... WHERE PRESENCE=TRUE
NB_COPIES_RENDUES	Nombre de copies rendues dans une salle de surveillance

Règle de gestion supplémentaire : Un étudiant a un autre étudiant qui joue le rôle au cours de l'année de son binôme pour tous les TP dans toutes les matières. La base de données doit garder cette information.

La liste d'émargement (doc 4) circule d'une salle de surveillance à l'autre. A posteriori, le secrétariat pédagogique inscrit dans la base de données la présence de chaque étudiant en fonction des signatures. Il y a un Procès-verbal de déroulement de l'épreuve (doc 5) pour chaque salle de surveillance.

Extrait de la charte des examens (doc 6) :

Aucun candidat n'est autorisé à quitter momentanément la salle durant les **deux premières heures**. Au-delà, les candidats qui demandent à quitter provisoirement la salle n'y seront autorisés qu'un par un.

Aucun candidat ne peut quitter définitivement la salle avant la fin de la **première demi-heure**. Le candidat qui quitte définitivement la salle avant la fin de l'épreuve remet obligatoirement sa copie même s'il remet une copie vierge.



Année universitaire 2021-2022



PROCÈS-VERBAL DE DÉROULEMENT DE L'ÉPREUVE
À retourner au secrétariat avec la liste d'émargement

PV CONTRÔLE
2021/2022

Date: 04/02/2022
Matière: ID-reporting / BO
Intervenant: Michel Dubois

ÉPREUVE	ID & reporting : BO
DATE	Vendredi 04/02/2022
Heure / Durée / Lieu de l'épreuve	14h30 / 45 min / B017
Enseignant responsable	Michel Dubois
Nombre d'étudiants inscrits	24
Nombre d'étudiants présents	
Nombre de copies remises	
Observations sur le déroulement de l'épreuve	
Surveillant (nom et signature)	

		BIN	SIGNATURES
1	BAUDET	PERRINE	2
2	BODEC	GWENDAL	3
3	CANTUS	JUSTINE	4
4	CARDEC	ADRIEN	5
5	CASTELLANI	ROMAIN	6
6	CORBEL	MANON	7
7	DANIEL	ALBAN	8
8	DEBUIRE	PAUL	9
9	DELAVET	TIPHAIN	10
10	DENIS	LOANN	11
11	DERVIEUX	PAULINE	12
12	DOS SANTOS	AUDREY	13
13	JEGOU	MAXENCE	14
14	JOUANNEL	KILIAN	15
15	LE BOLLOCH	AMANDINE	16
16	LE GUINER	AXEL	17
17	NEGGAL	AMINE	18
18	NOWAK	ENORA	19
19	OKOMFESSONO	FLORIANE	20
20	ORELLANA	MELANIE	21
21	RUNGETTE	DYLAN	22
22	SORIN	GAETAN	23
23	SOW THIERNO	ABDOULAYE	24
24	STURMA	CLARA	1

Il arrive qu'une donnée telle une date, une heure, un numéro d'ordre, une place ou un rang doive figurer seule dans un schéma de données, où elle forme l'unique propriété d'une entité spatio-temporelle. L'entité spatio-temporelle se justifie pleinement quand elle représente une donnée gérée, tel un calendrier prévisionnel ou une liste de numéros de places que l'on ne peut pas déduire d'un simple calcul. Par contre, certaines entités spatio-temporelles artificielles servent uniquement à introduire la propriété nécessaire pour former la source composée d'une dépendance fonctionnelle. Dans ce cas, les valeurs prises par leurs occurrences sont toutes présentes dans les occurrences de l'association (cardinalité minimale à 1 entre l'entité et l'association). L'entité ne donnera pas de relation dans le modèle relationnel.

Dans le module de conception de IBEasy+, les entités spatio-temporelles sont rattachées à la catégorie SELF. A charge pour vous de créer une catégorie composite qui représente l'association liée à l'entité spatio-temporelle. Rappel : l'ensemble des descripteurs rattachés à la catégorie composée, constituant le but, dépend fonctionnellement de manière élémentaire et directe de la même source composée des identifiants des catégories définissant cette catégorie composée.

Remarques sur les références circulaires: certes une mauvaise conception peut amener à une multiplication des références circulaires mais certaines sont légitimes (car irréductibles qu'au prix d'une perte de sémantique de la base de données) comme l'employé et son supérieur direct. Firebird, contrairement à MS Access en natif (mais simulables en VBA), implante du reste les *Recursive Common Table Expression* pour calculer la fermeture transitive d'un graphe. D'une manière générale, il peut, entre deux tables, exister plusieurs chemins de jointure si ces derniers ne sont pas redondants (une telle transitivité est éliminée par la troisième forme normale), c'est-à-dire si dans la réalité chaque chemin de jointure a un sens différent. C'est au concepteur de requête en mode ébauche (QBE) ou non (SQL) à ne pas inclure tous ces chemins afin d'éviter une intersection de chemin qui réduit le nombre de lignes dans la réponse, si seulement un sens est souhaitable. Il est vrai que la gestion (création, modification et destruction de tables puis l'insertion, modification et suppression de lignes) s'en trouve complexifiées du fait de l'apparition de cycle dans le graphe des dépendances fonctionnelles interrelations. Certains SGBDR implantent des contraintes d'intégrité référentielles soit désactivables temporairement (Oracle, MySQL) soit différables à la validation de la transaction (Oracle, PostgreSQL). Il est vrai que ni MS Access, ni Firebird n'implante un de ces mécanismes, ce qui conduit à détruire une contrainte d'intégrité référentielle avant insertion puis à la recréer à la fin.

Sur un schéma Entité-Association, les références circulaires sont déjà produites par les associations réflexives qui nécessite d'explicitier un rôle sur chaque patte de l'association pour valablement déterminer la cardinalité d'une patte. Dans IBEasy+, au niveau d'une catégorie, on peut définir le rôle.

Malheureusement, si l'auteur de IBEasy+ permet dans le dictionnaire de données d'ajouter l'attribut représentant le lien réflexif (NUMETUBI), en créant une catégorie de type rôle (ETUDIANTBINOME) et en affectant cette catégorie à l'attribut, le module de conception, en ne permettant pas de définir de relation entre la catégorie d'origine (ETUDIANT) et (ETUDIANTBINOME), ne provoque pas ni l'ajout de l'attribut dans la relation, ni la déclaration de la clé étrangère. IL vous faudra le faire en mode conception graphique après validation de la conception.

La catégorie Param dans IBEasy+ permet de rattacher les attributs paramètres qui n'existent qu'en un seul exemplaire dans tout le système et qui ne changent que rarement.

Mnémonique	Description	Catégorie du champ	Catégorie composée de	Type du champ	Domaine
▶ ANNEEUNIV	Année universitaire	PARAM		Descripteur	CHAR(9) character set WIN1252
CAPACITE	Nombre maximal d'é	SALLE		Descripteur	SMALLINT
CONSIGNE	Consigne sur les doc	CONTROLE	DTCONTROLE,HRCONTROLE,MATIERE	Descripteur	CHAR(120) character set WIN1252
DTCONTROLE	Date du contrôle	SELF		Descripteur	DATE
DUREECONTROLE	Durée en minute du	CONTROLE	DTCONTROLE,HRCONTROLE,MATIERE	Descripteur	SMALLINT
DUREEMINDEF	Durée minimale de c	PARAM		Descripteur	SMALLINT
DUREEMINTEMP	Durée minimale avar	PARAM		Descripteur	SMALLINT
HRCONTROLE	Heure de début du c	SELF		Descripteur	TIME
LBSALLE	Libellé de la salle	SALLE		Descripteur	CHAR(20) character set WIN1252
NB_COPIES_RENDUES	Nombre de copies re	SURVEILLANCE	DTCONTROLE,HRCONTROLE,SALLE	Descripteur	SMALLINT
NB_INSCRITS	Nombre d'étudiants	SURVEILLANCE	DTCONTROLE,HRCONTROLE,SALLE	Descripteur	SMALLINT
NB_PRESENTS	Nombre d'étudiants	SURVEILLANCE	DTCONTROLE,HRCONTROLE,SALLE	Descripteur	SMALLINT
NOMAT	Nom de la matière	MATIERE		Descripteur	CHAR(20) character set WIN1252
NOMSURVEILLANT	Nom du responsable	SURVEILLANCE	DTCONTROLE,HRCONTROLE,SALLE	Descripteur	CHAR(30) character set WIN1252
OBSERVATIONS	Observations faites	SURVEILLANCE	DTCONTROLE,HRCONTROLE,SALLE	Descripteur	CHAR(250) character set WIN1252
PLACE	Place dans la salle a	SELF		Descripteur	INTEGER
▶ PRESENCE	L'étudiant est-il prés	CONVOCATION	DTCONTROLE,ETUDIANT,HRCONTROLE,PLACE,SALLE	Descripteur	BOOLEAN
REFSALLE	Identification de la s	SALLE		Identifiant	INTEGER
SESSION	Numéro de session	CONTROLE	DTCONTROLE,HRCONTROLE,MATIERE	Descripteur	SMALLINT

Mnémonique	Description	Catégorie du champ	Catégorie composée de	Type du champ	Domaine
ANNEEUNIV	Année universitaire	PARAM		Descripteur	CHAR(9) character set WIN1252
DTCONTROLE	Date du contrôle	SELF		Descripteur	DATE
NOMENS	Nom de l'enseignant	ENSEIGNANT		Descripteur	CHAR(20) character set WIN1252
NOMETU	Nom de l'étudiant	ETUDIANT		Descripteur	CHAR(20) character set WIN1252
NUMETU	Numéro de l'étudian	ETUDIANT		Identifiant	SMALLINT
NUMETUBI	Numéro de l'étudian	ETUDIANTBINOME		Identifiant	SMALLINT

Les dépendances fonctionnelles élémentaires et directes du cas étendu :

Numetu -> Nometu, Dtnaiss, Cdsexe, Numetubi

Cdsexe -> Lbsexe

Numens -> Nomens, Grade, Ancien

Numat -> Nomat, Coeff, Numens

Refsalle -> Lbsalle, Capacite

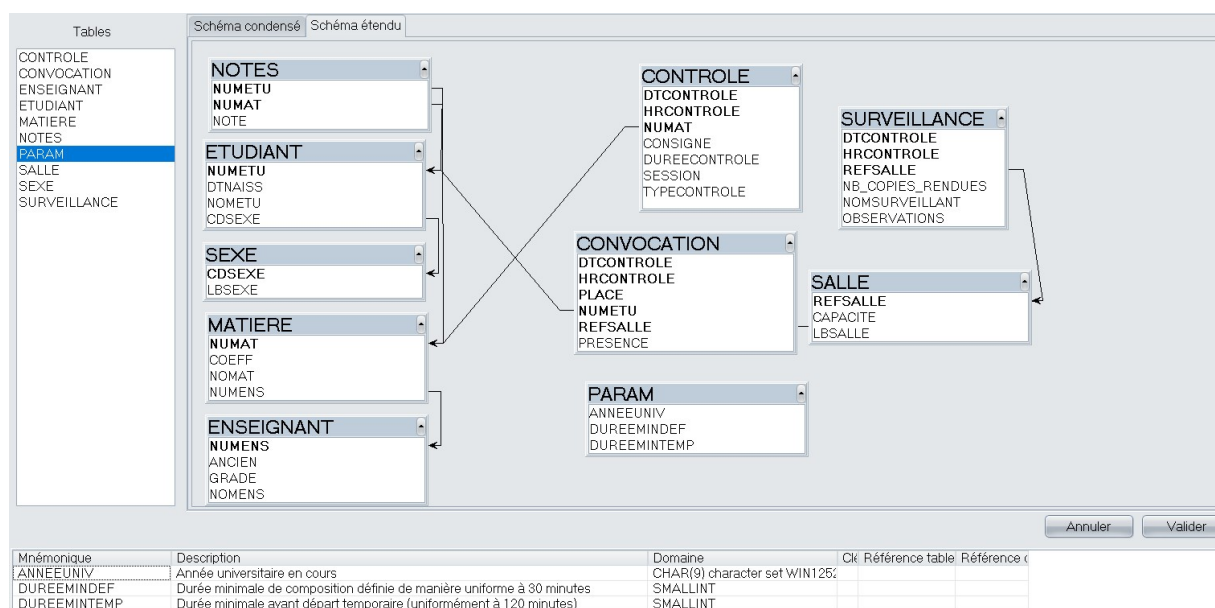
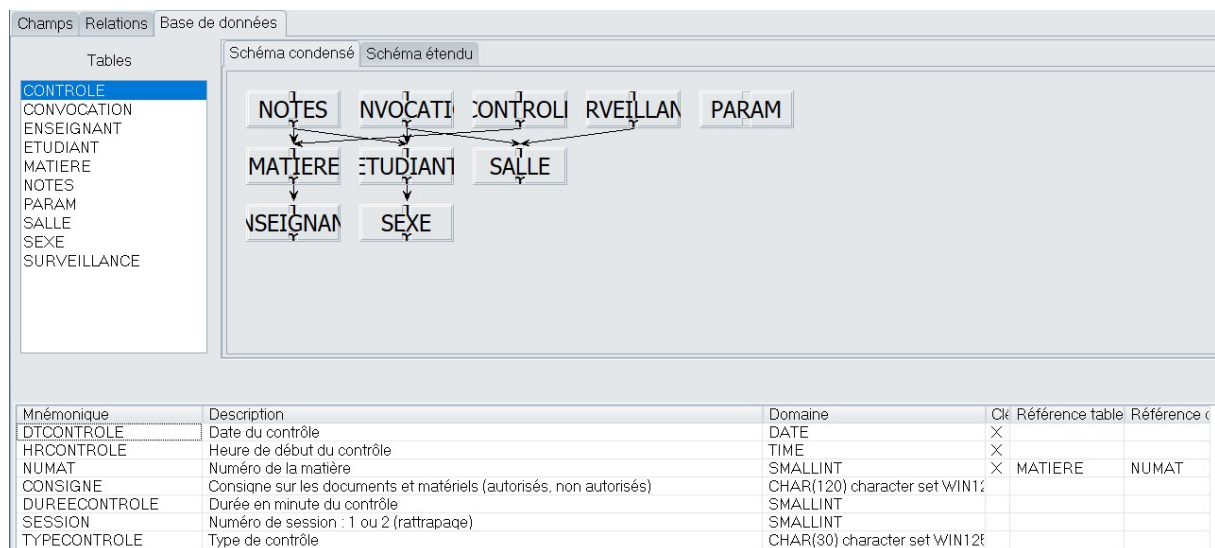
Numetu, Numat -> Note

Dtcontrole, Hrcontrole -> Dureecontrole, Consigne, Typecontrole, Session, Numat

Dtcontrole, Hrcontrole, Refsalle -> Nomsurveillant, Observations, Nb_Copies_Rendues

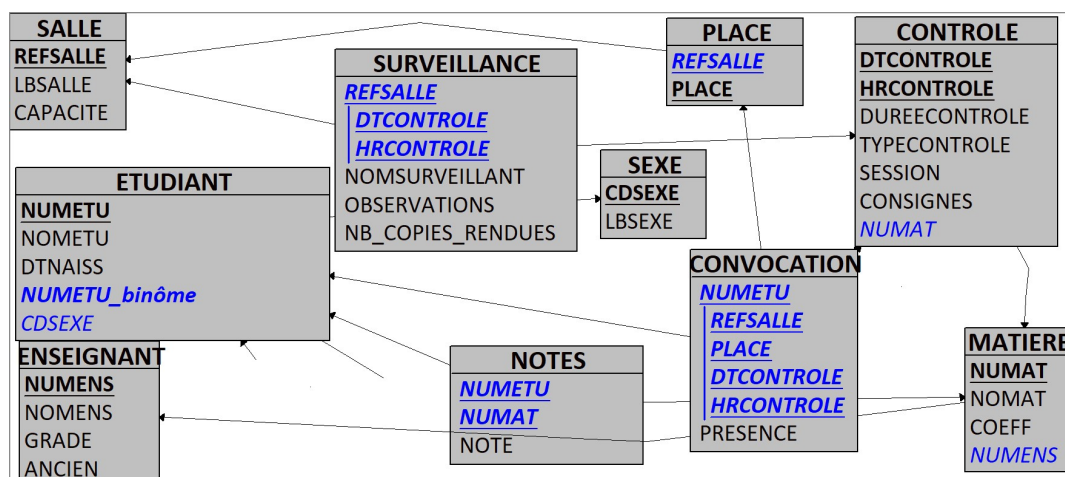
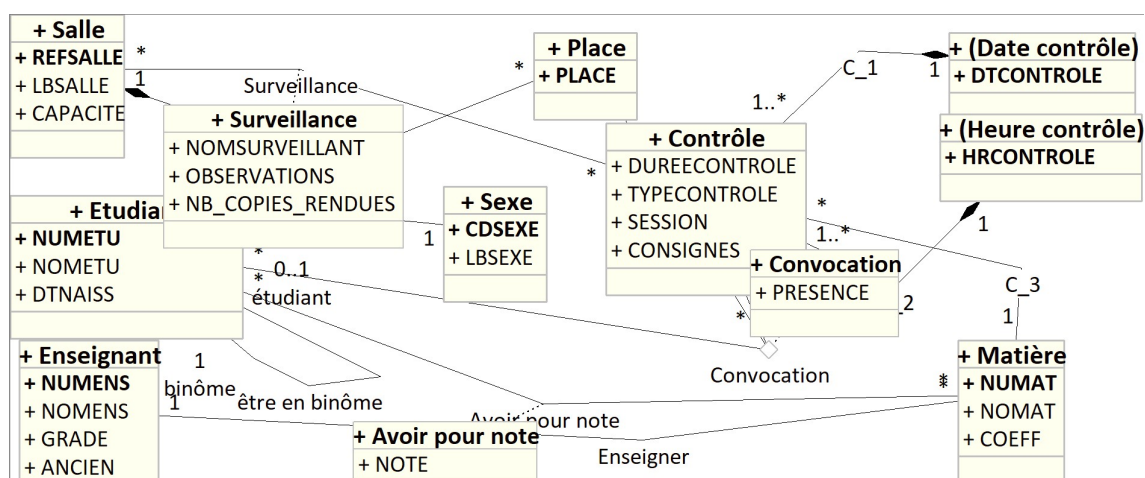
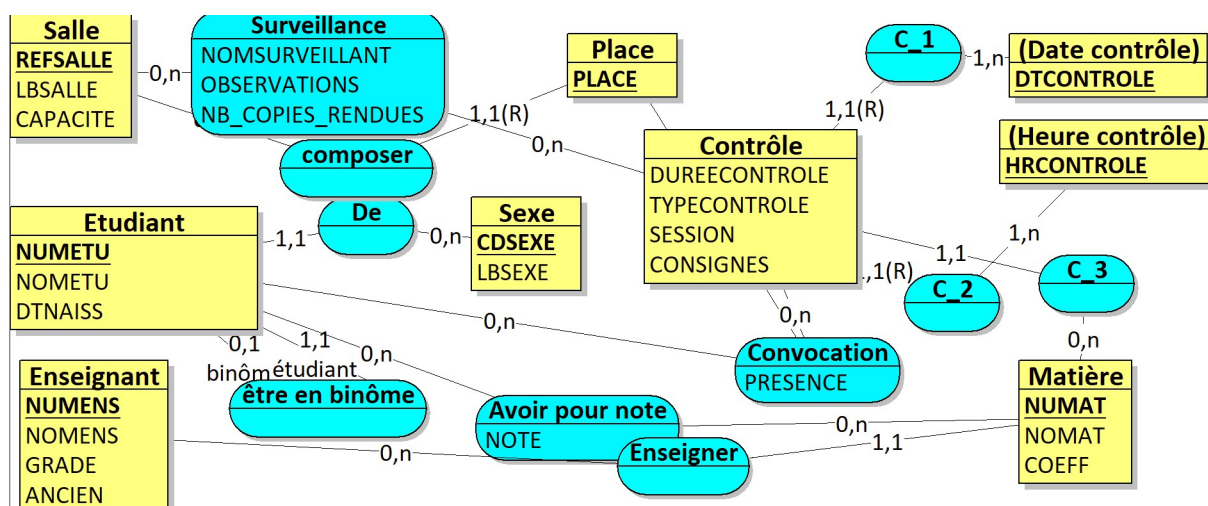
Dtcontrole, Hrcontrole, Refsalle, Place -> Presence, Numetu

Catégorie source	Oc		Oc	Catégorie but
▶ CONTROLE	m	——	1	MATIERE
CONVOCATION	m	——	1	ETUDIANT
CONVOCATION	m	——	1	SALLE
ENSEIGNANT	m	——	m	ETUDIANT
ENSEIGNANT	1	——	m	MATIERE
ENSEIGNANT		——//		SEXE
ETUDIANT	m	——	1	SEXE
MATIERE		——//		SEXE
NOTES	m	——	1	ETUDIANT
NOTES	m	——	1	MATIERE
SURVEILLANCE	m	——	1	SALLE

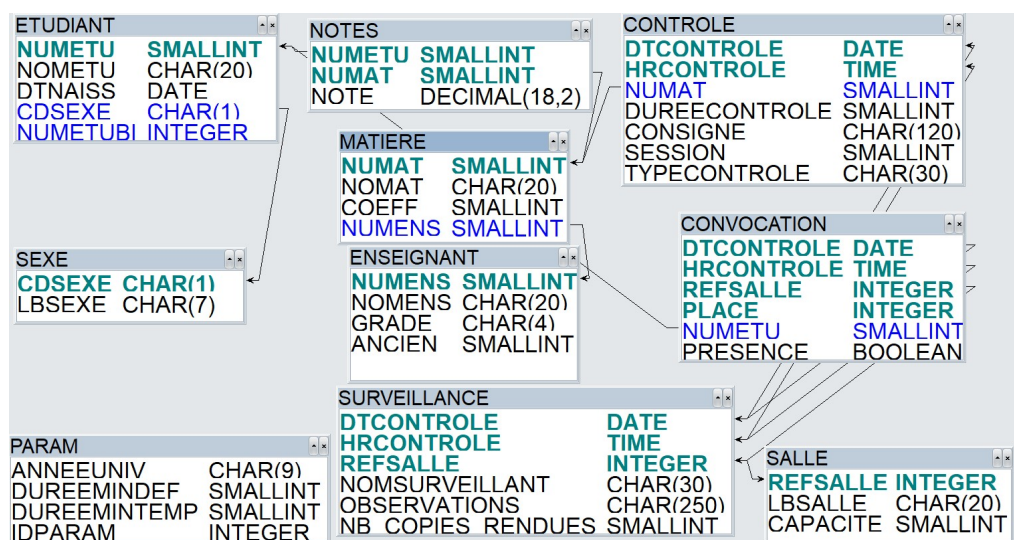


IBEasy+ a du mal à modéliser ce que l'on obtient avec soit l'identification relative, soit avec une CIF sur une n-aire qui concerne toutes les pattes d'une association ou pseudo entité ou agrégation d'association qui dans le modèle relationnel se traduit par moins d'attribut dans la clé primaire composée. De plus, la table PARAM n'a pas d'attribut garantissant qu'elle n'a qu'une seule ligne. Les clés candidates devront se voir ajouter un index unique.

Le cas avec Looping :



Ainsi, voici le résultat de la conception graphique suite aux adaptations dans IBEasy+ :



Les clés primaires de CONTRÔLE (perte de NUMAT) et de CONVOCATION (perte de NUMETU) ont été corrigées, l'attribut NUMETUBI et la clé étrangère réflexive ont été ajoutées ainsi que les contraintes de clés étrangères multiples (DTCONTROLE, HRCONTROLE) dans SURVEILLANCE, (DTCONTROLE, HRCONTROLE, REFSALLE) dans CONVOCATION après avoir enlevé l'option vérifier l'intégrité référentielle. On a créé le champ IDPARAM dans la table PARAM. Des index uniques ont été ajoutés aux champs IDPARAM d'une part, DTCONTROLE, HRCONTROLE, NUMETU d'autre part et NUMETUBI enfin. Vous pouvez rajouter les contraintes et les valeurs par défaut dans le dictionnaire de données :

Mnémonique	Description	Domaine	Défaut	NUL	Contraintes
▶ ANCIEN	Ancienneté de l'enseignant	SMALLINT		×	value<43
ANNEEUNIV	Année universitaire en cours	CHAR(9) character		×	
CAPACITE	Nombre maximal d'étudiant da	SMALLINT		×	
CDSEXE	Code sexe de l'étudiant	CHAR(1) character		×	value in ('H','F')
COEFF	Coefficient de la matière	SMALLINT		×	value in (1,2,3,4,5,6)
CONSIGNE	Consigne sur les documents et	CHAR(120) charact		×	
DTCONTROLE	Date du contrôle	DATE	CURRENT_DATE		
DTNAISS	Date naissance de l'étudiant	DATE		×	
DUREECONTF	Durée en minute du contrôle	SMALLINT		×	
DUREEINDE	Durée minimale de composition	SMALLINT	30	×	
DUREEINTE	Durée minimale avant départ b	SMALLINT	120	×	
GRADE	Grade de l'enseignant	CHAR(4) character		×	value in ('VAC', 'LRU', 'PRCE', 'PRAG', 'ASS', 'MCF', 'PR')
HRCONTROLE	Heure de début du contrôle	TIME		×	
IDPARAM	Garantie de l'unicité de la ligne	INTEGER	1	×	
LBSALLE	Libellé de la salle	CHAR(20) characte		×	
LBSEXE	Libellé du code sexe de l'étudi	CHAR(7) character		×	value in ('Homme','Femme')
MOYENNE	Moyenne de l'étudiant	DECIMAL(18,2)		×	value>=0 and value <=20
NB_COPIES_R	Nombre de copies rendues da	SMALLINT			
NB_INSCRITS	Nombre d'étudiants inscrits da	SMALLINT		×	
NB_PRESENT	Nombre d'étudiants présents c	SMALLINT		×	
NOMAT	Nom de la matière	CHAR(20) characte		×	
NOMENS	Nom de l'enseignant	CHAR(20) characte		×	
NOMETU	Nom de l'étudiant	CHAR(20) characte		×	
NOMSURVEILI	Nom du responsable de la sur	CHAR(30) characte		×	
NOTE	Note obtenue par l'étudiant da	DECIMAL(18,2)		×	value >=0 and value<=20
NUMAT	Numéro de la matière	SMALLINT		×	value>0
NUMENS	Numéro de l'enseignant	SMALLINT		×	value >0
NUMETU	Numéro de l'étudiant	SMALLINT		×	value>0
NUMETUBI	Numéro de l'étudiant qui est sc	SMALLINT			value >0
OBSERVATION	Observations faites par le surv	CHAR(250) charact		×	
PLACE	Place dans la salle affectée pc	INTEGER		×	
PRESENCE	L'étudiant est-il présent lors de	BOOLEAN			
REFSALLE	Identification de la salle	INTEGER		×	value >0
SESSION	Numéro de session : 1 ou 2 (re	SMALLINT		×	value in (1,2)
TYPECONTRO	Type de contrôle	CHAR(30) characte		×	value in ('CC', 'CT')

SEXE = (CDSEXE *VARCHAR(1)*, LBSEXE *VARCHAR(7)*);

ENSEIGNANT = (NUMENS *COUNTER*, NOMENS *VARCHAR(20)*, GRADE *VARCHAR(4)*, ANCIEN *BYTE*);

SALLE = (REFSALLE *COUNTER*, LBSALLE *VARCHAR(20)*, CAPACITE *BYTE*);

PLACE = (#REFSALLE, PLACE *BYTE*);

ETUDIANT = (NUMETU *COUNTER*, NOMETU *VARCHAR(20)*, DTNAISS *DATE*, #NUMETU_binôme, #CDSEXE);

MATIERE = (NUMAT *BYTE*, NOMAT *VARCHAR(20)*, COEFF *BYTE*, #NUMENS);

CONTROLE = (DTCONTROLE *DATE*, HRCONTROLE *TIME*, DUREECONTROLE *SMALLINT*, TYPECONTROLE *VARCHAR(2)*, SESSION *BYTE*, CONSIGNES *VARCHAR(50)*, #NUMAT);

NOTES = (#NUMETU, #NUMAT, NOTE *REAL*);

SURVEILLANCE = (#REFSALLE, #(DTCONTROLE, HRCONTROLE), NOMSURVEILLANT *VARCHAR(20)*, OBSERVATIONS *VARCHAR(250)*, NB_COPIES_RENDUES *BYTE*);

CONVOCATION = (#NUMETU, #(#REFSALLE, PLACE), #(DTCONTROLE, HRCONTROLE), PRESENCE *LOGICAL*);

Des règles concernant les binômes et le chevauchement des examens nécessitent d'accéder à des informations sur plusieurs tables ou à des lignes autres que la ligne courante d'une seule table. Techniquement seuls les déclencheurs de table pour Firebird et les contraintes CHECK créées par ADODB pour MS Access peuvent les implanter et non les contraintes CHECK de colonnes et les contraintes CHECK de tables pour Firebird et les règles ValideSi de colonnes et les règles ValideSi de tables pour MS Access.

Vous pouvez ajouter les déclencheurs BEFORE UPDATE et BEFORE INSERT sur la table ETUDIANT pour vérifier qu'un étudiant est bien le binôme de son binôme (voir Annexe 3).

Vous pouvez ajouter les déclencheurs BEFORE UPDATE et BEFORE INSERT sur la table CONTROLE pour vérifier qu'il n'y a pas de chevauchement entre contrôle d'un même jour (voir Annexe 3).

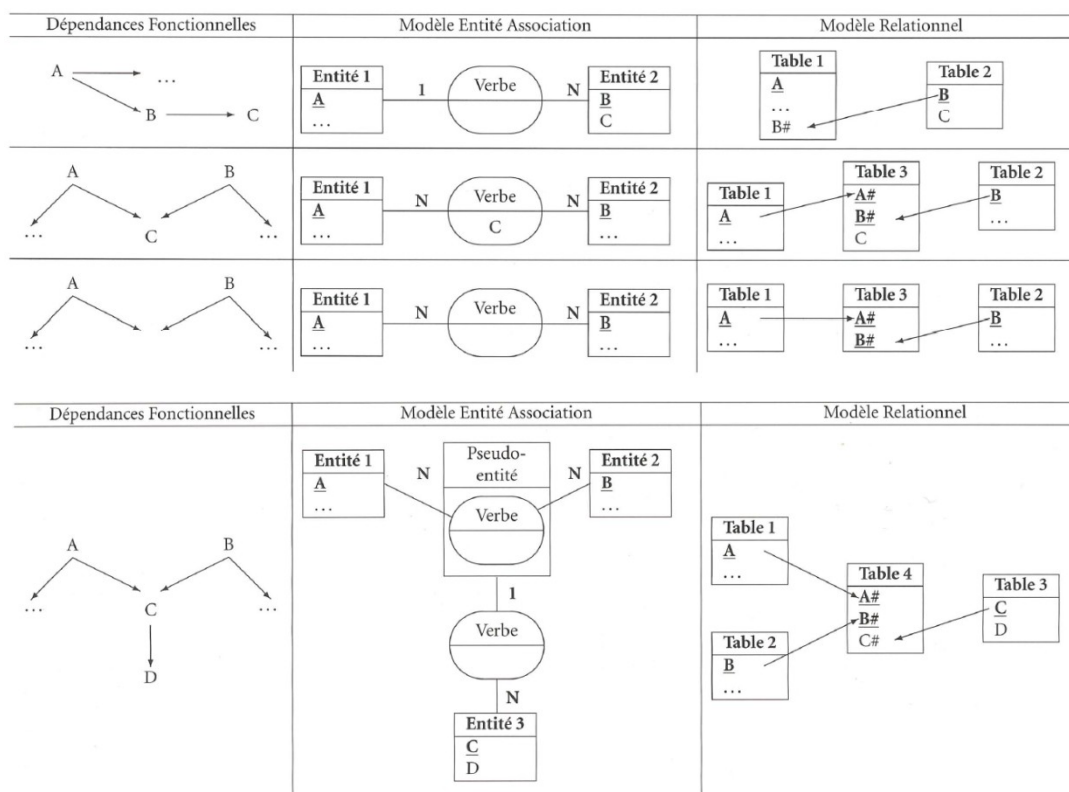
Vous pouvez alors procéder à l'export vers MS Access et à l'utilisation du module VBA pour compléter l'export.

Attention : le type BOOLEAN n'était pas exporté pour les versions 2.4 et 2.5 mais il est exporté par la version 3.2 d'IBeasy+. NOTE est un mot réservé par MS ACCESS, il vaut mieux utiliser dans Looping NOTATION ou dans le SQL ACCESS utiliser [NOTE].

La modification null autorisé à la place de null non autorisé nécessite dans IBEasy+ (limite ALTER DOMAIN de Firebird) de détruire dans la table l'attribut, de supprimer l'attribut dans le dictionnaire de données et de le recréer corrigé puis, dans la table, d'ajouter à nouveau l'attribut avec le bon domaine (corrigé). C'est le module conception de base de données d'IBeasy+ qui force la création de domaine dans Firebird. Juste après la validation, les tables Firebird sont créées en utilisant pour chaque colonne un domaine particulier.

Enfin, il est signalé que la version 3.2 d'IBeasy+ exporte vers Access les clés étrangères composées si ces dernières ont été validées lors de leur ajout dans la conception graphique. Ce n'était pas le cas des versions 2.4 et 2.5 d'IBeasy+.

Annexe 1 : Correspondance entre les modèles



Annexe 2 : Présentation du logiciel gratuit LOOPING

Looping (P. Bergougnoux, mai 2021) est un logiciel gratuit (mais pas open-source), comme IBEasy+ (voir <https://www.looping-mcd.fr/>) qui est un exécutable unique (portable en ce sens qu'il n'écrit pas dans la base de registre Windows) en version soit 32 bits soit 64 bits depuis sa version 3.1 qui se lance avec un double clic de souris permet de faire de la conception SGBD sans pouvoir faire de la rétroconception jusqu'à la version 4.0. Sa principale force est la bijection qu'il implante entre schéma E-A/UML Il va jusqu'à générer des CREATE TABLE adaptés à certains SGBDR. Pour MS Access, il génère le code SQL compatible dans une fenêtre et en mode export textuel, il génère une procédure ADODB pour l'exécution de ce code. Il propose son modèle relationnel sous forme graphique et textuel. Firebird, HSQLDB sont des SGBD cibles possibles en tant que SGBD personnalisé. Les fonctions de modélisation avancées sont :

- Pour les associations réflexives, les pattes peuvent afficher un rôle ;
- Pour les entités spatio-temporelles, on peut ne pas créer la relation ;
- Pour les entités faibles, on peut utiliser un identifiant relatif (=composition UML). Une association peut se transformer en entité faible ;
- Avec la version 3.1, on peut choisir l'ordre des clés étrangères dans la relation issue d'une n : m ou d'un n-aire ; on peut choisir l'ordre des attributs dans l'index unique d'une clé candidate composée ; les commentaires de tables et de colonnes peuvent figurer dans le script.
- Les CIF et les contraintes interrelation ainsi que l'héritage peuvent être modélisés.
- La version 4.0 permet le glisser déposer de propriété vers une autre entité pour la déplacer ou la dupliquer avec la touche Ctrl.

- Ce qui n'est pas encore pris en compte dans les scripts SQL générés : valeur par défaut, contraintes CHECK colonne et table (mais déjà possibilité de mettre soi-même du SQL pour les contraintes inter-relations).
- La version 4.1 propose une rétroconception à partir d'un script SQL de création de tables pour déduire un schéma Entités-Associations.
- Pour MS Access, un script nommé `txt2bas.bat`, fourni par Michel Dubois, permet de convertir les éventuelles instructions COMMENT ON..., contenues dans le fichier VBA produit par l'export de Looping, en appel à son module VBA `ExportFromIBEasy+` afin des créer des descriptions de tables et de colonnes.

Annexe 3 : Déclencheurs de tables pour ESSAIS+

Déclarer sous IBEASY+ l'exception BADSTUDENTREFERENCE :

```
CREATE EXCEPTION BADSTUDENTREFERENCE 'Un étudiant doit être le binôme de son binôme';
```

Déclarer sous IBEASY+ les déclencheurs BEFORE INSERT et BEFORE UPDATE nommés respectivement ETUDIANT_INS_BI_TR et ETUDIANT_UPD_BI_TR pour la table ETUDIANT :

```
AS
DECLARE Result boolean;
begin
Result = false;

SELECT COUNT(*)<>0 FROM ETUDIANT INNER JOIN ETUDIANT AS ETUDIANT_1 ON ETUDIANT.NUMETUBI =
ETUDIANT_1.NUMETU WHERE ETUDIANT_1.NUMETUBI<>ETUDIANT.NUMETU

into :Result;

IF (Result) then
EXCEPTION BADSTUDENTREFERENCE ;
end
```

La contrainte CHECK ADODB traduite pour MS ACCESS :

```
(SELECT COUNT(*)=0 FROM ETUDIANT INNER JOIN ETUDIANT AS ETUDIANT_1 ON ETUDIANT.NUMETUBI =
ETUDIANT_1.NUMETU WHERE ETUDIANT_1.NUMETUBI<>ETUDIANT.NUMETU)=TRUE")
```

Déclarer sous IBEASY+ l'exception EXAMINATIONOVERLAPING :

```
CREATE EXCEPTION EXAMINATIONOVERLAPING 'Il existe un chevauchement de contrôles !';
```

Déclarer sous IBEASY+ les déclencheurs BEFORE INSERT et BEFORE UPDATE nommés respectivement CONTROLE_INS_TR et CONTROLE_UPD_TR pour la table CONTRÔLE :

```
AS
DECLARE Result boolean;
begin
Result = false;

SELECT COUNT(*)<>0 FROM "CONTROLE" C1, "CONTROLE" C2

WHERE C1."DTCONTROLE"=C2."DTCONTROLE" AND C1."HRCONTROLE"<>C2."HRCONTROLE" AND (((
C1."HRCONTROLE" < dateadd (minute, C2."DUREECONTROLE", C2."HRCONTROLE")) AND (C1."HRCONTROLE" >=
C2."HRCONTROLE")) OR ((dateadd (minute, C1."DUREECONTROLE", C1."HRCONTROLE") <= dateadd (minute,
C2."DUREECONTROLE", C2."HRCONTROLE")) AND (dateadd (minute, C1."DUREECONTROLE", C1."HRCONTROLE") >
C2."HRCONTROLE"))))
```

```

into :Result;

IF (Result) then

EXCEPTION EXAMINATIONOVERLAPING;

end

```

La contrainte CHECK ADODB traduite pour MS ACCESS :

```

(SELECT COUNT(*)=0 FROM CONTROLE C1, CONTROLE C2 WHERE C1.DTCONTROLE=C2.DTCONTROLE AND
C1.HRCONTROLE<>C2.HRCONTROLE AND (((C1.HRCONTROLE < dateadd ('n', C2.DUREECONTROLE, C2.HRCONTROLE))
AND (C1.HRCONTROLE >= C2.HRCONTROLE)) OR ((dateadd ('n', C1.DUREECONTROLE, C1.HRCONTROLE) <= dateadd
('n', C2.DUREECONTROLE, C2.HRCONTROLE)) AND (dateadd ('n', C1.DUREECONTROLE, C1.HRCONTROLE) >
C2.HRCONTROLE))))=TRUE")

```

Il convient lors de la création d'une contrainte CHECK ADO de lui affecter un nom long qui résume la règle de gestion car contrairement à une règle ValideSi de table, on ne peut pas définir un message d'erreur. Donc le nom de la contrainte doit être un bon indice pour comprendre pourquoi la contrainte a réagi. Les fonctions personnalisées publiques d'un module VBA ne sont pas autorisées ni pour les règles ValideSi, ni pour les contraintes CHECK ADO.

Depuis MS Access 2010, il est apparu des macros de tables ou de données qui peuvent être utilisées comme alternative aux contraintes CHECK ADODB. La fonctionnalité macro qui permet d'automatiser des traitements dans MS Access sans savoir programmer, a été fortement enrichie dans les versions 2010 et 2013 pour devenir équivalente à la fonctionnalité VBA qui nécessite de savoir programmer au moins de manière procédurale car, lorsque l'on destinait la base de données pour le web, les macros étaient automatiquement traduites dans Microsoft Sharepoint et dans Microsoft SQL Server. Les macros attachées à un formulaire ou à un contrôle sont traduites en Javascript pour les formulaires html de l'application Sharepoint correspondant au formulaire MS Access. Chaque table MS Access a son équivalent dans SQL server. Les macros de données attachées à un événement de table (la table est ouverte en **mode création**, onglet du ruban **Conception de table**, icône **créer des macros de données**, choix de l'évènement) sont traduites en déclencheur Transac-SQL After sur leur table. Les macros de données autonomes ou nommées (la table est ouverte en mode **feuille de données**, onglet du ruban **Table**, icône **Macro nommée**) sont associées à une table spécifique, mais pas à un événement spécifique. Vous pouvez appeler des macros de données nommées à partir de n'importe quelle autre macro de données ou macro standard. Ce type de macro peut prendre des paramètres et est conservé en tant que procédure stockée écrite en Transac-SQL dans SQL Server. Ainsi faut-il considérer que les macros de données ne correspondent pas aux macros utilisées dans les formulaires ou les états. Ce ne sont pas les mêmes actions qui sont disponibles dans ces deux contextes.

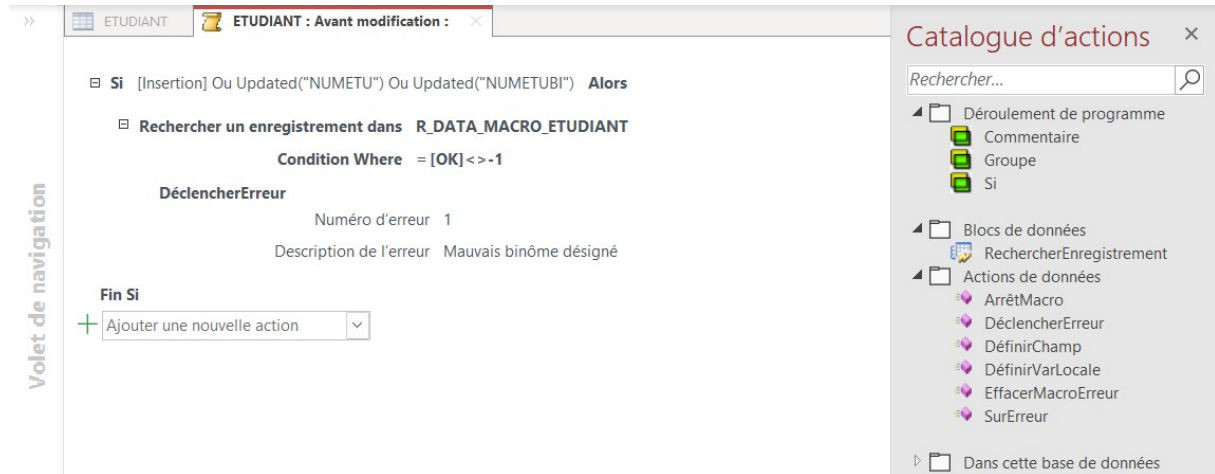
Les macros de données vous permettent de créer des actions lorsque les données de la table répondent à des critères spécifiques. Par exemple, si vous avez un client de Californie, vous pouvez avoir une macro de données qui vous envoie un e-mail indiquant qu'il s'est inscrit. Une macro de données **Avant Modification / Before Change** ou **Avant suppression / Before Delete** peut déclencher une erreur via l'action **DéclencherErreur / RaiseError** et bloquer ainsi la modification non désirée du point de vue d'une règle de gestion. On s'aide aussi de l'action **DéfinirVarLocale / SetLocalVar** pour conserver une valeur qui va être écrasée, de la syntaxe **[Ancien].[Nom de champ] / [Old].[Field Name]** qui met à disposition la valeur précédente dans un champ de données en cours de mise à jour, de l'action **Rechercher un enregistrement** dans sur une table ou une requête sauvegardée restreinte par une condition **Where / Look Up A Record In Where**. Les tests **[Insertion] / [IsInserted]** ou la fonction **Updated("[Nom de champ]")** sont aussi d'intérêt.

Pour la macro de données Avant modification sur la table ETUDIANT, il faut avant créer une requête nommée R_DATA_MACRO_ETUDIANT dont la définition SQL est :


```
SELECT COUNT(*)=0 AS OK
```

```
FROM ETUDIANT INNER JOIN ETUDIANT AS ETUDIANT_1 ON ETUDIANT.NUMETUBI = ETUDIANT_1.NUMETU
WHERE ETUDIANT_1.NUMETUBI<>ETUDIANT.NUMETU;
```

La macro de données Avant modification sur la table ETUDIANT :



Le texte de la macro :

```
Si [Insertion] Ou Updated("NUMETU") Ou Updated("NUMETUBI") Alors
    Rechercher un enregistrement dans R_DATA_MACRO_ETUDIANT
    Condition WHERE : [OK]<>-1
    DéclencherErreur
    Numéro : 1
    Description de l'erreur : Mauvais binôme désigné
Fin Si
```

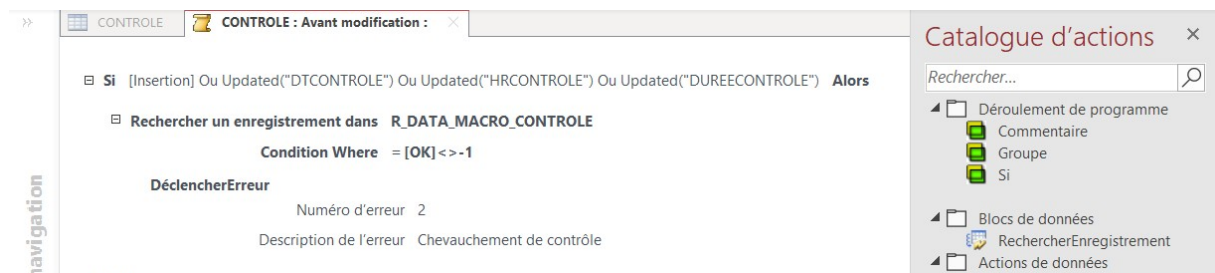
Pour la table CONTROLE, il faut la requête R_DATA_MACRO_CONTROLE :

```
SELECT COUNT(*)=0 AS OK
```

```
FROM CONTROLE AS C1, CONTROLE AS C2
```

```
WHERE C1.DTCONTROLE=C2.DTCONTROLE And C1.HRCONTROLE<>C2.HRCONTROLE And
(((C1.HRCONTROLE<dateadd('n',C2.DUREECONTROLE,C2.HRCONTROLE)) And (C1.HRCONTROLE>=C2.HRCONTROLE)) Or
((dateadd('n',C1.DUREECONTROLE,C1.HRCONTROLE)<=dateadd('n',C2.DUREECONTROLE,C2.HRCONTROLE)) And
(dateadd('n',C1.DUREECONTROLE,C1.HRCONTROLE)>C2.HRCONTROLE))) ;
```

Enfin, le corps de la macro de données Avant modification :



Le texte de la macro :

```
Si [Insertion] Ou Updated("DTCONTROLE") Ou Updated("HRCONTROLE") Ou Updated("DUREECONTROLE") Alors
```

```

Rechercher un enregistrement dans R_DATA_MACRO_CONTROLE

Condition WHERE : [OK]<>-1

DéclencherErreur

Numéro : 2

Description de l'erreur : Chevauchement de controle

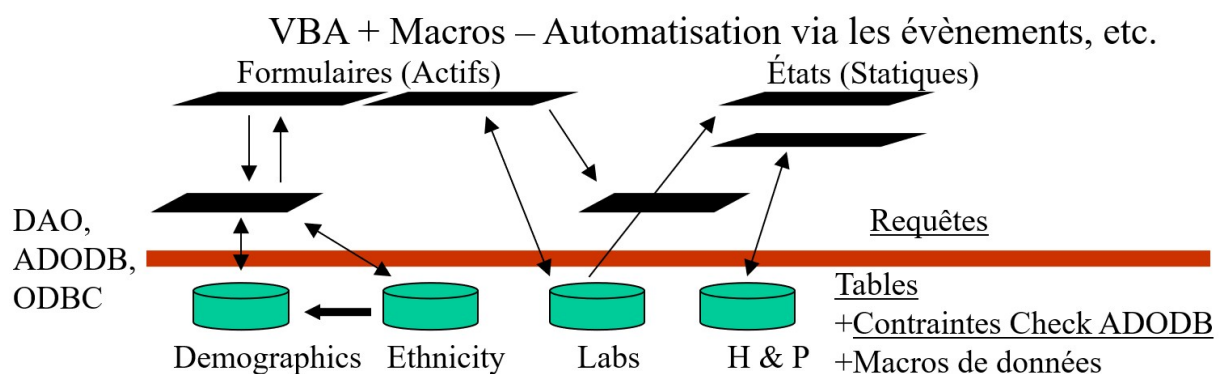
Fin Si

```

Les calculs et les recherches au sein d'une macro de données connaissent aussi des restrictions mais il est aussi possible d'utiliser les fonctions de domaines pour les agrégats. Enfin, l'utilisation d'une fonction publique codée dans un module VBA peut étendre les possibilités mais l'accès à la table depuis une application cliente qui ne dispose pas de VBA rend invalide le calcul et donc rend aléatoire la vérification de la macro de données dans ce cas. En cas de découpage Front-End Access application – Backend Access Storage Engine, le fonction publique VBA se trouvant au niveau du Front-End, on obtient un fonctionnement normal de la macro de données. Le mauvais fonctionnement se rencontre lorsque l'on accède à la table via un module VBA ADO ou DAO depuis une autre application que MS Access dotée du module avec la fonction publique (par exemple MS Excel), via ODBC, OLEDB, via python, via R, via SAS. Il est fortement déconseillé d'utiliser du code VBA dans les événements de tables hormis pendant la phase de développement où le recours à certaines fonctions (msgbox, debug, timer) permet de déboguer et d'optimiser certaines macros. L'appel aux fonctions personnelles publiques VBA sont à proscrire dans les macros de données qui valident les données (avant modification, avant suppression). Les appels à ces fonctions peuvent être tolérés pour les autres macros de données avec des événements après insertion, après modification, après suppression.

Découpage du fichier

Fichier *Front-End* / frontal - Contient toutes les entités (Formulaires, Requêtes, etc.) et maintient les liens vers les tables de données du fichier *Back-End* / dorsal. A noter que vous pouvez avoir plus d'un fichier FE pour gérer plusieurs type d'utilisateurs.



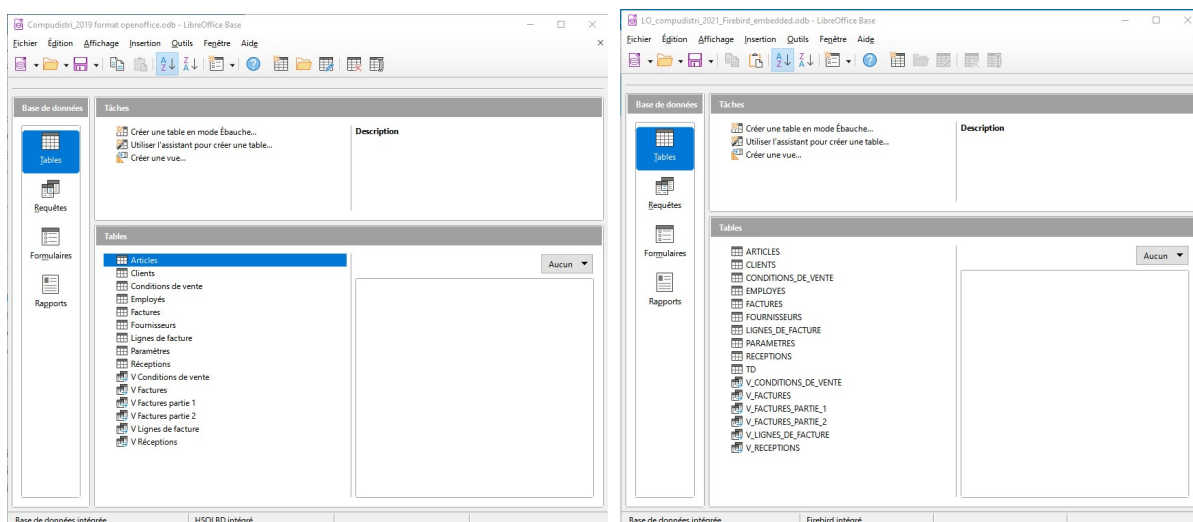
Fichier *Back-End* / dorsal – Contient toutes les tables de données

Annexe 4 Accéder aux bases de données utilisant un moteur HSQLDB ou Firebird intégré dans un fichier ODB via LibreOffice Base

OpenOffice.org Base 3.2 était le pendant gratuit de Microsoft Access qui lui est disponible pour les OS Windows, Linux et MacOSX. Suite au rachat par Oracle en 2009 de Sun Microsystems, un fork nommé LibreOffice Base fait face à Apache OpenOffice Base issu du don par Oracle à la fondation Apache du code de OpenOffice.org.

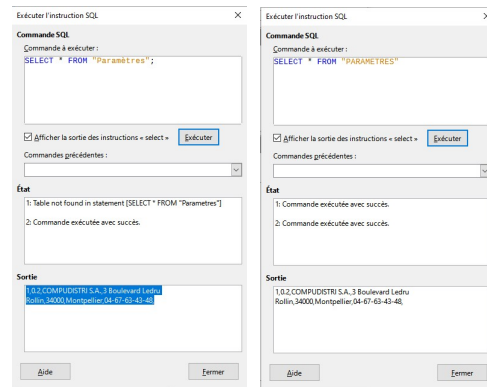
Apache OpenOffice Base peut intégrer dans un fichier ODB un moteur HSQLDB 1.8 et LibreOffice 7.x Base peut intégrer soit un moteur HSQLDB 1.8 (ou ultérieur) intégré (si une extension est installée), soit un moteur Firebird 3.0 intégré. HSQLDB nécessite un JRE Java. Pour utiliser Firebird, il faut activer le mode expérimental.

Actuellement, sur les postes du réseau ENS de l'université Bretagne-Sud, c'est LibreOffice 7.1.3.2 qui est installé. Il est bien configuré pour ouvrir les deux types de fichiers ODB. De plus l'ouverture de chacun des types de documents se fait en double-cliquant sur le fichier.



Pour accéder aux relations entre tables, il faut aller dans le menu **Outils / Relations....**

Pour envoyer un ordre SQL dans le moteur intégré (HSQLDB ou Firebird), il faut aller dans le menu **Outils / SQL....**



Contrairement à MS Access, les métabases de HSQLDB et de FIREBIRD sont riches et précises et peuvent être requêtée.