

Variations pédagogiques autour de l'enseignement du modèle relationnel et des SGBDR en 1^{ère} STG

« L'enseignement peut être regardé comme la mise à disposition de l'étudiant d'occasions où il puisse apprendre. C'est un processus interactif et une activité intentionnelle. Les buts ... peuvent être des gains dans les connaissances, un approfondissement de la compréhension, le développement de compétences en "résolution de problèmes" ou encore des changements dans les perceptions, les attitudes, les valeurs et le comportement. » G. Brown et M. Atkins (1988)

« L'apprentissage peut être vu comme un processus actif et constructif au travers duquel l'apprenant manipule stratégiquement les ressources cognitives disponibles de façon à créer de nouvelles connaissances en extrayant l'information de l'environnement et en l'intégrant dans sa structure informationnelle déjà présente en mémoire. » R.B. Kozman (1991)

1. Pourquoi le modèle relationnel dans l'enseignement en STG ?

L'enseignement du modèle relationnel s'appuie sur les technologies de l'information et de la communication pour développer des connaissances déclaratives et des savoir-faire procéduraux appliqués à la compréhension et à la manipulation des bases de données.

Mais il offre surtout l'occasion de développer des savoir-faire génériques (abstraction, conceptualisation, logique, résolution de problèmes notamment) qui pourront être transférés dans de nombreux domaines connexes. Il s'inscrit dans un véritable enseignement technologique qui donne aux élèves l'accès aux capacités d'abstraction et de conceptualisation qui leur permettront tout au long de leurs vies professionnelle et personnelle de distinguer le réel de ses représentations et d'exploiter les représentations du réel pour le comprendre.

La culture des bases de données relationnelles est donc essentielle pour l'enseignement technologique d'Économie-gestion, en répondant aux objectifs de la rénovation :

« Les enseignements réunis sous la dénomination *Information et gestion* associent les méthodes, les démarches et les techniques de la gestion des entreprises et des organisations, de la comptabilité et de l'informatique de gestion. Ils visent à permettre aux élèves :

- d'acquérir la connaissance et la pratique des systèmes d'information de gestion et en particulier du système d'information comptable ;
- d'être en mesure de ***produire, d'analyser, de contrôler et d'interpréter les informations obtenues*** ;
- de constater la contingence des instruments de gestion et des conditions de leur utilisation.

Ces finalités visent à développer chez les élèves des compétences spécifiques à la série sciences et techniques de gestion :

- ***appréhender, analyser et modéliser des situations caractéristiques du fonctionnement des organisations*** ;
- ***synthétiser des informations de gestion permettant d'assurer la préparation des décisions et le contrôle des opérations*** ;
- ***communiquer et mettre en œuvre les solutions retenues.*** »

« ***L'étude du modèle relationnel permet d'aborder les notions fondamentales en matière d'organisation et d'interrogation des données. Les bases de données relationnelles constituent la référence technologique en matière de support et de gestion des systèmes d'information automatisés.*** »

« L'objectif général de la série vise ***l'acquisition des connaissances, des techniques et des méthodes de travail nécessaires à la poursuite des études.***

L'enseignement d'*Information et gestion* suppose le recours à des environnements de travail en réseau adaptés à ses objectifs de formation et à son projet pédagogique. ***Les fonctionnalités disponibles doivent permettre d'utiliser, à des fins de simulation, des bases de données échantillons, tirées de la réalité, permettant une approche réaliste des systèmes d'information de gestion.***

L'enseignement d'*Information et gestion* associe la connaissance des fondements théoriques et conceptuels à la mise en œuvre et la pratique des techniques de référence.

L'approche pédagogique ne peut donc se réduire ni à la pratique répétitive et mécanique d'exercices ou d'enregistrements comptables ni à l'apprentissage systématique de logiciels.

Les acquis conceptuels et méthodologiques de la classe de première revêtent une grande importance pour la poursuite d'études. Il convient d'insister en particulier sur :

- *l'acquisition et la compréhension des notions fondamentales ;*
- *l'apprentissage des méthodes de modélisation et de simulation ;*
- *le jugement sur l'utilité des informations pour la prise de décision ;*
- *la communication des réflexions et des commentaires sur les solutions envisagées.*

2. Pourquoi cette production ?

L'objet de ce dossier n'est ni de faire la présentation savante du modèle relationnel à destination des futurs professeurs, ni de proposer un support de cours pour les futurs élèves, ni même d'élaborer une progression pédagogique prête à l'emploi... Nous faisons seulement l'hypothèse que la réflexion active et les interrogations qui nourrissent ce document vous permettront de conforter en vous les compétences nécessaires pour construire votre propre enseignement.

Nous vous proposons donc une immersion dans le monde du relationnel comme support d'une activité de réflexion pédagogique, un voyage plus ou moins initiatique au cours duquel nous nous interrogerons tour à tour sur les savoirs savants liés à ce domaine, sur le rapport entre ces savoirs et ceux à enseigner, sur les obstacles que le professeur et les élèves auront à surmonter et sur les moyens pour les y aider.

3. Quel contenu ?

Le contenu du programme faisant référence à l'apprentissage du relationnel est rappelé ci-dessous (deuxième et quatrième colonne).

« Le modèle relationnel apporte les notions essentielles en matière de structuration des données. Il permet d'étudier la conception d'une base de données (**comment elle est construite**) indépendamment de son implantation technique. Sa connaissance est indispensable à la maîtrise des langages d'interrogation des bases de données dans les différents domaines d'application de l'informatique de gestion. L'étude du modèle relationnel est abordée dans le but :

- **d'introduire les principes fondamentaux qui régissent l'organisation et l'exploitation de l'information en gestion ;**
- **d'aborder la logique de construction d'une représentation des informations sous la forme d'un ensemble structuré de données non redondantes ;**
- **de mettre en œuvre, sur des exemples simples, les règles et la démarche d'interrogation d'une base de données.**

La notion de **dépendance fonctionnelle** est utilisée pour repérer les **attributs** qui caractérisent un objet identifiable du système d'information. La nécessité de la **normalisation** sera illustrée par la mise en évidence des problèmes liés à la **redondance** et à la **mise à jour** des données (insertion, suppression, modification). L'étude des **opérateurs relationnels** prépare l'expression de **requêtes** dans le langage **SQL**»

« La **base de données** est un ensemble d'informations structurées en **tables** dont l'implantation, la mise à jour et l'exploitation sont réalisées à l'aide d'un système de gestion de bases de données. Elle est définie par son **schéma** (la structure) et son **contenu** (les valeurs). L'étude de l'organisation des données dans les tables (**nom, type, clés**) met en évidence les mécanismes utilisés par les SGBD pour assurer le contrôle de la cohérence des données lors d'une mise à jour.

Le recours au langage SQL fonde l'usage d'un raisonnement partant de l'expression du résultat à obtenir. Son apprentissage, indépendant de l'interface propre à chaque outil, contribue à l'acquisition progressive **de la capacité de passer de l'expression d'un besoin en langage naturel à sa définition dans un langage formel.**

Le langage de requête (SQL) est enseigné dans le but de permettre à l'élève d'exprimer de façon précise et normalisée les traitements à réaliser sur les données.

L'étude est volontairement limitée à l'ordre SELECT, en utilisant les clauses d'identification des sources de données sur une ou plusieurs tables (FROM), de restriction et de jointure (WHERE) et de tri (ORDER BY). Les fonctions usuelles d'agrégat et de traitement des chaînes de caractères, des nombres et des dates sont abordées en fonction des besoins. L'utilisation d'un "requêteur" graphique, pour autant qu'elle permette de mettre en œuvre une logique de construction structurée, peut faciliter la visualisation et la compréhension des requêtes exprimées en langage SQL»

L'objet de ce document n'est pas de détailler l'ensemble des concepts énoncés (qui d'ailleurs ne représentent qu'une partie des concepts du relationnel) mais d'y faire référence pour illustrer quelques partis pris pédagogiques. Les théories, les modèles pédagogiques sur lesquels ils reposent seront exposés sommairement : l'[annexe 1](#) vous propose une bibliographie qui vous permettra, si besoin, d'en parfaire la connaissance. En posant cet objectif particulier, nous avons bien conscience de ne pas répondre directement à vos besoins variés de formation théorique sur le modèle relationnel : l'[annexe 2](#) vous fournit une liste d'ouvrages savants ou utiles à une initiation progressive.

Enfin vous trouverez en [annexe 3](#) les définitions qui serviront notre propos. Vous pourrez également tout au long du dossier retrouver ces définitions en activant les liens hypertextuels associés à la première apparition de chaque concept.

Nous demandons à cette occasion l'indulgence des spécialistes pour les quelques libertés prises dans l'expression des concepts liés au modèle relationnel. Nous qualifierions la matière de notre discours de premier niveau de transposition didactique à l'usage de nombreux de nos collègues qui découvrent le domaine.

4. Quelles pratiques pédagogiques en STG ?

L'enseignement en classe de première s'appuie plus particulièrement, mais pas uniquement, sur des méthodes actives. Il repose sur un constat qu'il est bon de rappeler : *apprendre nécessite effort et travail pour surmonter les difficultés.*

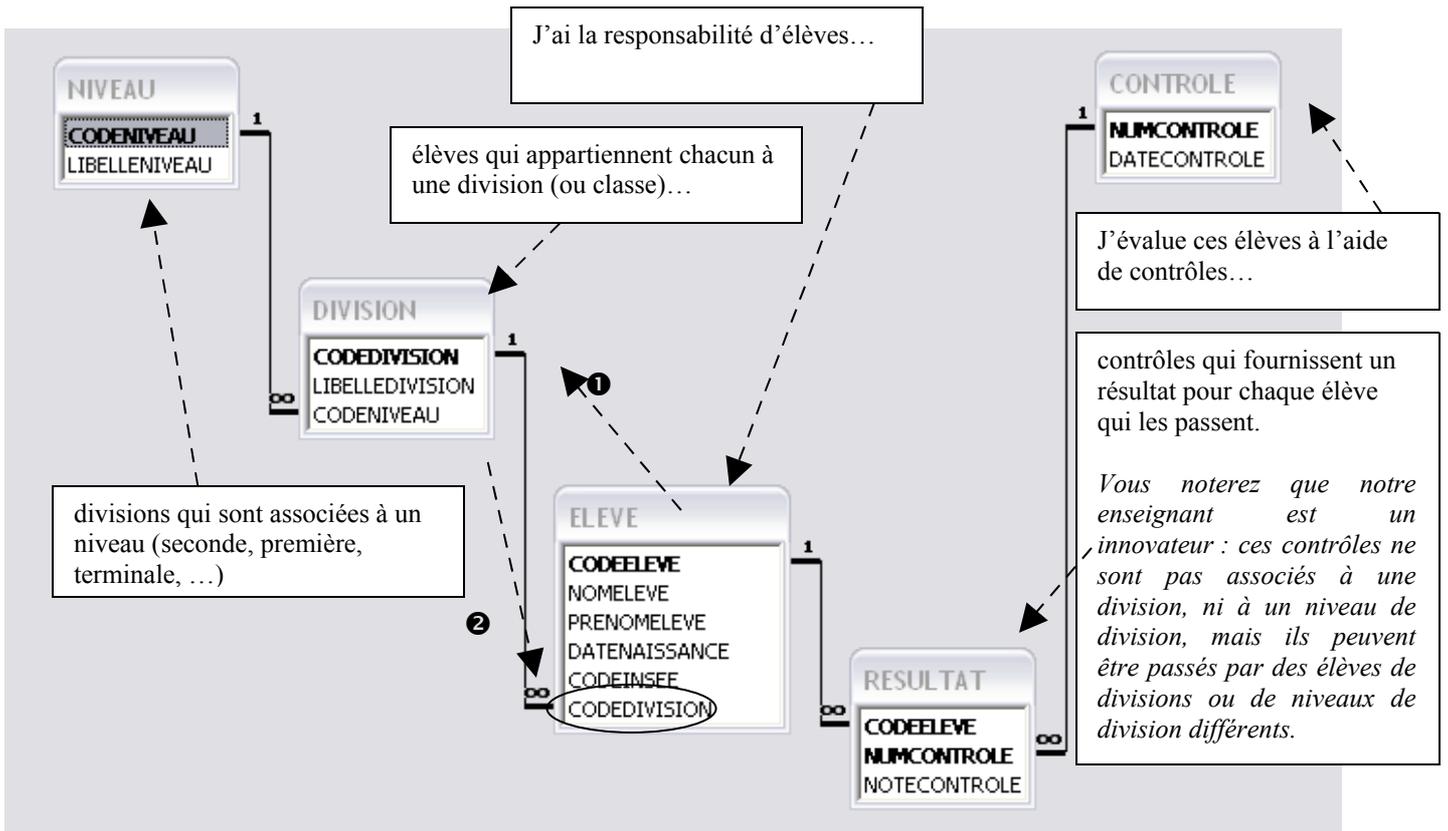
Il s'agit donc de mettre en œuvre des situations d'apprentissage qui permettent aux élèves de construire de nouvelles connaissances à travers l'analyse, la compréhension et la résolution de problèmes et de les transférer dans d'autres situations. Par connaissances, nous entendons ici les savoirs, les savoir-faire opérationnels (habiletés), les savoir-faire intellectuels (capacités, compétences) et les savoir-être (attitudes, comportements) mais aussi le « vouloir-faire » et le « savoir-travailler ».

4.1 Pour commencer

Nous vous proposons ci-après une première situation qui servira à illustrer quelques démarches pédagogiques possibles toutes axées sur l'apprentissage des concepts du modèle relationnel.

L'exemple est volontairement pris dans le domaine métier de l'enseignant afin de réduire la taille de « l'obstacle cognitif » tout à la fois pour l'enseignant et pour l'élève. Il s'agit ici de partir d'une base de données utilisée par un enseignant pour gérer l'ensemble des contrôles qu'il fait passer à ses élèves. Cette base est bien entendu très simplifiée mais elle permet d'aborder tous les concepts du programme de première. L'enseignant est l'acteur principal et l'utilisateur unique du processus de gestion des notes.

- Le professeur a construit une base de données qui répond actuellement à ses besoins et vous explique les informations qu'il a ainsi représentées :



Notre professeur a choisi comme modèle de représentation le modèle graphique associé au SGBDR ACCESS qui fait apparaître les noms des tables (ELEVE, DIVISION, ...) et, pour chaque table, le nom de ses champs (CODEELEVE, NOMELEVE, ...). Le nom du champ (comme CODEELEVE dans la table ELEVE) ou les noms des champs (comme CODEELEVE et NUMCONTROLE dans la table RESULTAT) qui jouent le rôle de clé primaire sont écrits en caractères **gras**.

Chaque table correspond à un ensemble d'objets ou concepts du système d'information : les élèves, les divisions, les niveaux, les contrôles, les résultats des élèves aux contrôles.

Un lien entre deux tables exprime une information supplémentaire.

Exemples :

- ❶ « Un élève appartient à une seule division ».
- ❷ « Une division comporte un ou plusieurs élèves ».

Ces exemples montrent que le trait reliant deux tables possède une sémantique, traduite généralement par un verbe (non précisé sur le schéma). La « quantification » (« un » ou « un ou plusieurs ») se lit devant l'autre table :

❶ « Un élève (table ELEVE) appartient à (signification donnée au trait dans ce sens de lecture) une (1) division (table DIVISION) ». Un tel lien est dit « fonctionnel ».

❷ « Une division (table DIVISION) comporte (signification donnée au trait dans ce sens de lecture) un ou plusieurs (∞) élèves (table ELEVE) ». Un tel lien est dit « hiérarchique ».

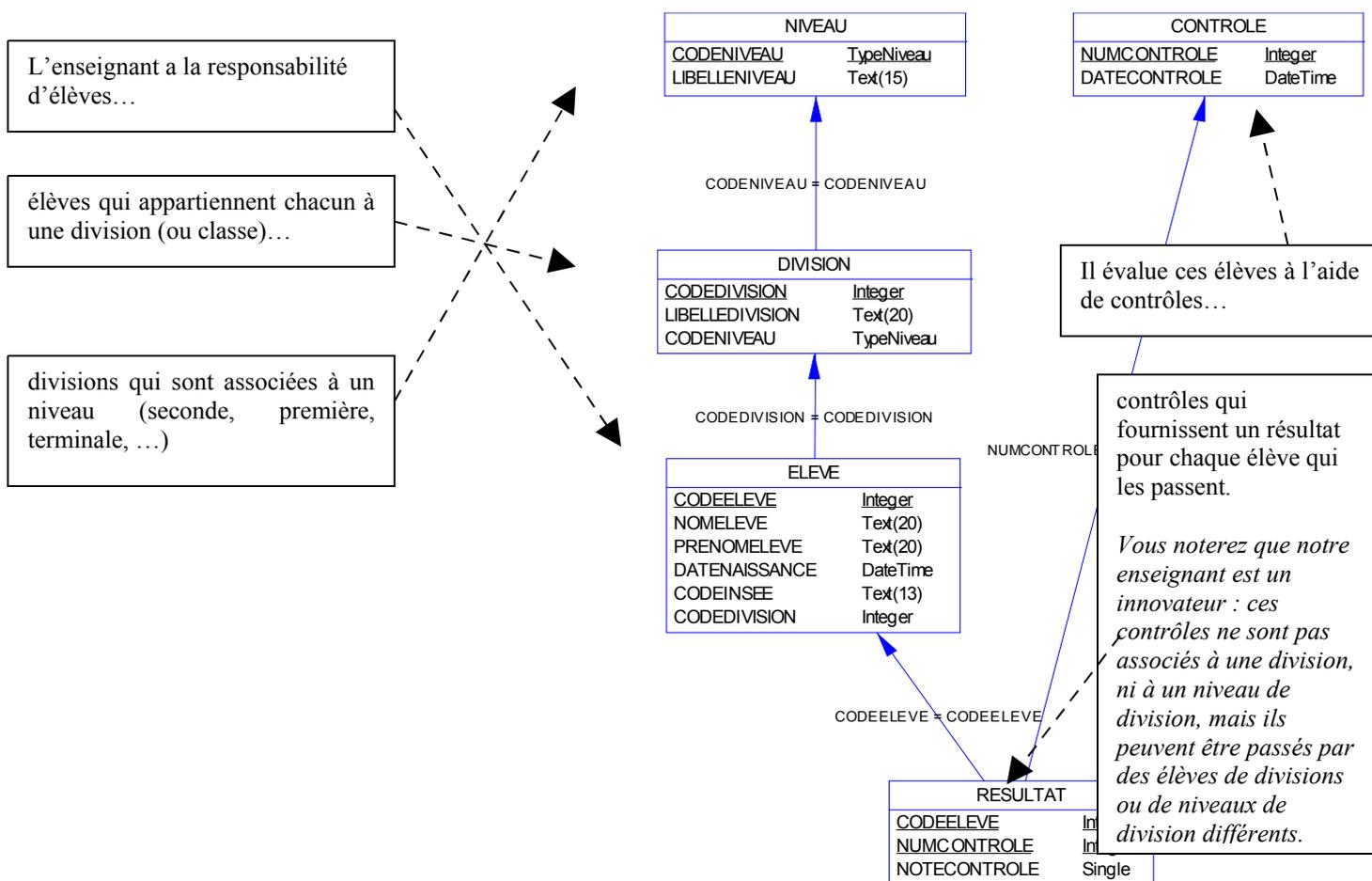
Dans chaque cas, cette double information est représentée par la présence d'une clé étrangère à la source du lien fonctionnel (CODEDIVISION, attribut clé étrangère dans la table ELEVE).

La description d'une table correspond à sa définition en compréhension (en intention). Dans la théorie mathématique des ensembles finis, les définitions en intention et en extension sont équivalentes. Par exemple, l'ensemble des nombres premiers compris entre 1 et 10 n'a qu'une extension connue, toujours la même : {1, 2, 3, 5, 7}. Or, l'extension d'une table relationnelle n'est pas forcément la même d'un instant à l'autre. Dans le domaine qui nous intéresse, la définition en intention exprime **le type de fait retenu** par la base de données et cette définition est stable ; l'extension de la table présente **les faits retenus à un instant donné**. De ce dernier point de vue, chaque table contient des **lignes** (appelés aussi enregistrements). Une ligne comporte une valeur par champ (ou colonne), prise dans un ensemble de valeurs possibles appelé **type de données**. Par extension de langage, on dit qu'un champ possède un type. Voici, à titre d'exemple, des lignes extraites de la table NIVEAU.

NIVEAU	
CODENIVEAU	LIBELLENIVEAU
1	SECONDE
2	PREMIERE
3	TERMINALE
4	BTS1
5	BTS2

- Une autre représentation graphique s'appuie plus directement sur le modèle relationnel. Elle permet de ne plus faire référence à un SGBDR particulier et utilise un vocabulaire propre :

Une **relation** est décrite par un schéma de relation qui liste les **attributs** qui la composent. Un ou plusieurs jouent le rôle de **clé primaire**. Les valeurs d'un attribut sont prises dans un **domaine**.



Comme pour les tables évoquées précédemment, nous pouvons passer du « modèle » aux réalisations : à un instant donné, une relation est composée de n-uplets¹. Les réalisations respectent bien sûr les contraintes énoncées dans le schéma :

- Ainsi, dans tout n-uplet, la valeur prise par un attribut sera compatible avec son domaine de définition.
- Semblablement, deux n-uplets quelconques de la relation ne possèdent jamais la même valeur de clé primaire (comme CODENIVEAU dans la table NIVEAU) sans quoi la clé primaire ne jouerait pas son rôle d'identifiant.
- Enfin, il en va de même pour les liens orientés entre les relations qui expriment la présence d'une contrainte d'intégrité référentielle : cela signifie qu'à une valeur de clé étrangère dans la table à l'origine du lien correspond obligatoirement la présence de cette même valeur de clé primaire dans la table de destination. Sous une apparence complexe, il s'agit d'une règle de bon sens lorsqu'elle est contextualisée : dans notre schéma, nous exprimons par exemple le fait « qu'un élève doit obligatoirement appartenir à une division, ou qu'un résultat concerne obligatoirement un élève et un contrôle : pas de résultat sans élève, pas de résultat sans contrôle »...

Un bon SGBDR devra contrôler toutes ces contraintes, en s'assurant par exemple qu'il est impossible de créer un nouvel élève s'il n'est pas associé à une division, mais aussi en interdisant de supprimer une division si des élèves y sont encore associés.

Un débutant se pose souvent la question de l'intérêt de mentionner sur chaque lien présent dans le schéma l'égalité entre clé étrangère et clé primaire : c'est tout simplement qu'**il est possible² que le nom de la clé étrangère soit différent de celui de la clé primaire**. Cela permet souvent de rendre le schéma plus lisible et d'éviter des erreurs d'interprétation.

4.2 Enseigner : entre rigueur et improvisation

Enseigner, c'est conduire, assister, diriger les élèves dans leurs tâches pour leur permettre de franchir les obstacles qu'ils rencontrent, dans une approche positive qui accorde à l'erreur une place autre que celle de faute. Il faut donc éveiller progressivement chez les élèves leur sens de l'autonomie et de la responsabilité en les amenant à travailler en équipe, à échanger, à coopérer et à collaborer. Il faut aussi développer leur goût de la recherche et de l'effort en les aidant au besoin dans leur démarche, sans pour autant leur apporter de solutions toutes faites.

Maîtriser l'incertitude, c'est donc aussi maîtriser la conduite de la classe, grâce à une bonne connaissance de la psychologie de l'élève, de ses représentations mentales, des mécanismes généraux de développement et d'apprentissage et de la discipline enseignée, afin de pouvoir envisager toutes les erreurs et les réactions possibles (et les autres...). Finalement, « enseigner, c'est prévoir »...

Le rôle du professeur est de faire découvrir les concepts et les connaissances associées à travers des obstacles cognitifs (mises en évidence d'absence de connaissance ou de connaissances insuffisantes, d'absences de méthodes ou de méthodes inadéquates) de telle manière que les élèves puissent s'approprier le savoir en surmontant leurs représentations mentales actuelles. Il permet ainsi aux élèves de conforter leurs compétences en augmentant graduellement la complexité des situations d'apprentissage (des tâches de plus en plus difficiles) et la variété des tâches (des tâches de plus en plus nombreuses).

Préparer une séquence d'enseignement est donc un savant cocktail qui passe par un recensement des objectifs à atteindre, des problèmes qui peuvent se poser et par le choix de stratégies pour atteindre les objectifs, de critères d'évaluation pertinents et des modes d'évaluation associés. Enseigner est alors un acte de communication (de communion ?) qui s'appuie sur plusieurs démarches pédagogiques choisies en fonction des concepts à faire acquérir, du profil des élèves et de leur progression mais aussi quelquefois de la situation.

Nous vous proposons d'illustrer quelques approches pédagogiques possibles à travers des phases d'apprentissage de concepts liés au modèle relationnel.

¹ Une relation comportant deux attributs est composée de couples ; une relation comportant trois attributs est composée de triplets ; une relation comportant n attributs est composée de n-uplets.

² Et même souvent souhaitable, comme nous le verrons plus loin !

4.3 Transposition didactique : adapter la réalité aux besoins des apprentissages

Il est difficile de passer des savoir savants aux savoir enseignés et au contexte de leur enseignement : c'est le rôle de la transposition didactique qui consiste à transformer (sans les déformer) les connaissances pour les adapter au public cible et à les ordonner pour en faciliter l'acquisition.

Dans ce domaine, l'informatique souffre d'une grande jeunesse, comme en témoigne son histoire (voir à ce propos : Histoire et épistémologie de l'informatique de G. Verroust, Université Paris VIII - 1997 : http://pauillac.inria.fr/~weis/info/histoire_de_l_info.html) et s'appuie essentiellement sur des références mathématiques, (J. Arzac, J. Rogalski, C. Pair, ...).

Heureusement pour nos élèves et nous-mêmes, l'informatique est le support de nombreuses activités professionnelles. Elle intervient dans un ensemble de **pratiques sociales de référence**, liant notamment la gestion des entreprises et l'exploitation des bases de données.

Il s'agit donc de recenser et d'exploiter ces pratiques sociales de référence pour construire des situations d'apprentissage en classe de première. Il s'agit aussi de déterminer le degré optimal de simplification (tant sur le plan du contexte professionnel que du contenu des savoirs) pour éviter :

- d'un côté l'apprentissage décontextualisé d'outils informatiques (séries d'exercices à la mode taylorienne, travaux dirigés ou activités occupationnelles vidées de leur sens),
- de l'autre côté la réalisation de projets surdimensionnés entraînant une surcharge cognitive ou une inadaptation à la structure mentale des élèves (objectifs non perceptibles, échéance trop éloignée, compétences à mettre en œuvre trop nombreuses ou inaccessibles, etc.).

A cela s'ajoute la nécessité de conserver l'aspect « authentique » de la pratique de référence, de faire réaliser aux élèves une tâche motivante et d'éviter leur conditionnement en diversifiant les situations et les questionnements.

Ordonner et regrouper les concepts

La première étape consiste à ordonner les concepts, c'est-à-dire à passer du réseau de concepts statique (ou carte conceptuelle) à une séquence dynamique de groupes de concepts. Il faut pour cela d'abord regrouper les concepts voisins (dont les apprentissages sont fortement liés) qui pourront être abordés dans le cadre d'une même séquence puis ordonner ces groupes en fonction des choix didactiques et pédagogiques.

L'enseignement de première nécessite de s'appuyer sur la technologie pour mener les apprentissages. À la lecture du programme, il semble qu'un passage préalable par les concepts des bases de données et ceux associés à la conception des requêtes s'impose avant d'aborder les concepts du modèle relationnel. Un choix possible est représenté dans le tableau suivant :

Ordre	Concepts associés	Remarques
1	BD relationnelle, SGBDR, structure, table, ligne, enregistrement, colonne, champ, nom, valeur, type, clé primaire, clé étrangère, contraintes d'intégrité, manipuler, interroger, mettre à jour, insérer, supprimer, modifier	Par la manipulation d'une base de données en mode graphique et l'observation de sa structure
2	Schéma relationnel, structure, données, relation, attribut, nom, valeur, domaine, n-uplet, clé, clé primaire, clé étrangère, contrainte d'intégrité	Par l'interprétation et la modélisation de la structure d'une base de données
3	Anomalie, manipuler, interroger, mettre à jour, insérer, supprimer, modifier, forme normale, dépendance fonctionnelle, normalisation	Par la manipulation d'une base de données comportant des anomalies de structure et éventuellement la modification de sa structure
4	BD relationnelle, SGBDR, LMD, manipuler, interroger, mettre à jour, insérer, supprimer, modifier, requêtes, SQL, SELECT, FROM WHERE, ORDER BY	Par la manipulation d'une base de données en mode SQL
5	Opérateurs relationnels : restriction, projection, jointure	Par l'interprétation de la structure d'une base de données et des résultats obtenus par des requêtes SQL

Recenser les capacités

La deuxième étape permet de construire une liste des capacités attendues des élèves. Le terme de capacité étant très polysémique, précisons le sens donné ici : c'est une habileté intellectuelle ou technique en rapport étroit avec la discipline.

Ainsi « Transformer une question simple exprimée en langage naturel en une requête SQL », « Tester une requête SQL sur une base de données » ou « Comprendre le sens d'un schéma relationnel » sont des capacités attendues d'un élève de première, mais « Savoir argumenter un choix » ne l'est pas. Cela ne veut pas dire que, dans ce domaine, l'enseignement de première n'améliorera pas la capacité argumentative des élèves (bien au contraire), mais cela signifie qu'elle représente une habileté intellectuelle « collatérale » ou « transversale », un savoir-faire générique transdisciplinaire, une finalité éducative à laquelle participe notre enseignement.

L'analyse du programme permet de réaliser la liste suivante :

- Maîtriser le vocabulaire associé au SGBDR utilisé.
- Maîtriser le vocabulaire associé au modèle relationnel.
- Transformer une question simple exprimée en langage naturel en une requête [SQL](#).
- Repérer dans une requête les opérateurs algébriques mis en œuvre.
- Tester une requête SQL sur une base de données et évaluer la validité du résultat obtenu.
- Représenter le [schéma relationnel](#) d'une base de données.
- Comprendre le sens d'un schéma relationnel.
- Détecter une anomalie de structure dans une base de données.

Recenser les pratiques sociales de référence utiles

La troisième étape vise à recenser les pratiques sociales de référence en rapport avec les domaines traités et à y associer un ou plusieurs groupes de concepts et une ou plusieurs capacités. Il s'agit ensuite de retenir les pratiques qui permettent de couvrir l'ensemble des concepts et capacités et qui sont compatibles avec le niveau d'exigence du programme.

Une pratique sociale de référence n'est pas une « fonction-métier » : il ne s'agit donc pas de recenser les activités professionnelles, mais d'isoler les processus intellectuels qui leur sont communs. Ainsi « exploiter une base de données » est une pratique sociale de référence, « Construire un bilan » ou « Gérer la relation client » n'en sont pas.

Le programme nous fournit de nombreux éléments, listés ci-dessous sans exhaustivité ni ordre particulier :

- Produire, analyser, contrôler et interpréter les informations issues d'une base de données.
- Synthétiser des informations de gestion pour assurer la préparation de décisions et le contrôle des opérations.
- Utiliser une base de données à des fins de simulation.
- Modéliser un système d'information simple à l'aide du modèle relationnel.
- Vérifier la validité d'un schéma relationnel simple et y apporter des modifications.
- Alimenter une base de données à partir de documents représentatifs de l'activité de l'organisation (commandes, factures, livraisons, ...).
- ...

Le tableau suivant peut alors être construit :

Ordre	Concepts associés	Pratique sociale de référence	Capacités
1	BD relationnelle, SGBDR, structure, table, ligne, enregistrement, colonne, champ, nom, valeur, type, clé primaire, clé étrangère, contraintes d'intégrité, manipuler, interroger, mettre à jour, insérer, supprimer, modifier.	Appréhender, analyser et modéliser des situations caractéristiques du fonctionnement des organisations. Alimenter une base de données à partir de documents représentatifs de l'activité de l'organisation (commandes, factures, livraisons, ...).	Maîtriser le vocabulaire associé au SGBDR utilisé. Comprendre le sens d'un schéma relationnel.
2	Schéma relationnel, structure, données, relation, attribut, nom, valeur, domaine, n-uplet, clé, clé primaire, clé étrangère, contraintes d'intégrité.	Modéliser un système d'information simple à l'aide du modèle relationnel.	Maîtriser le vocabulaire associé au modèle relationnel. Représenter le schéma relationnel d'une base de données. Comprendre le sens d'un schéma relationnel.
3	Anomalies, manipuler, interroger, mettre à jour, insérer, supprimer, modifier, forme normale, dépendance fonctionnelle, normalisation.	Vérifier la validité d'un schéma relationnel simple et y apporter des modifications.	Maîtriser le vocabulaire associé au SGBDR utilisé. Maîtriser le vocabulaire associé au modèle relationnel. Représenter le schéma relationnel d'une base de données. Comprendre le sens d'un schéma relationnel. Détecter une anomalie de structure dans une base de données.
4	BD relationnelle, SGBDR, LMD, manipuler, interroger, mettre à jour, insérer, supprimer, modifier, requêtes, SQL, SELECT, FROM WHERE, ORDER BY.	Produire, analyser, contrôler et interpréter les informations issues d'une base de données. Utiliser une base de données à des fins de simulation.	Maîtriser le vocabulaire associé au SGBDR utilisé. Maîtriser le vocabulaire associé au modèle relationnel. Transformer une question exprimée en langage naturel en une requête SQL (et inversement). Repérer dans une requête les opérateurs algébriques mis en œuvre. Tester une requête SQL sur une base de données. Représenter le schéma relationnel d'une base de données. Comprendre le sens d'un schéma relationnel. Détecter une anomalie de structure dans une base de données.
5	Opérateurs relationnels : restriction, projection, jointure.	Synthétiser des informations de gestion pour assurer la préparation de décisions et le contrôle des opérations.	Maîtriser le vocabulaire associé au SGBDR utilisé. Maîtriser le vocabulaire associé au modèle relationnel. Transformer une question exprimée en langage naturel en une requête SQL (et inversement). Repérer dans une requête les opérateurs algébriques mis en œuvre. Tester une requête SQL sur une base de données. Représenter le schéma relationnel d'une base de données. Comprendre le sens d'un schéma relationnel.

La dernière étape est la transposition pédagogique de chaque pratique sociale de référence retenue pour aboutir à une situation d'apprentissage qui permet aux élèves de construire de nouvelles connaissances. Il faut garder à l'esprit le caractère formateur des situations d'apprentissage. L'objectif **principal** n'est pas d'obtenir un résultat, mais de comprendre comment on l'obtient, afin de pouvoir mobiliser de nouveau les compétences acquises dans un autre contexte.

Il est possible d'isoler quelques invariants dans les situations-problèmes issues de la « didactisation » d'une pratique de référence pour des élèves de première :

- une problématique d'organisation visant à préparer une prise de décision ou un choix, à partir d'observations, de modélisations et de simulations, pris dans le champ du programme, tant dans le module « Information et communication » que dans le module « Information et gestion »,
- une interaction entre activités pratiques et réflexion, individuelles et collectives,
- la nécessité de l'exploitation ou de la construction d'un modèle statique (schéma relationnel) ou dynamique (requête ou algorithme),
- l'utilisation raisonnée de l'outil informatique.

Nous pouvons distinguer plusieurs situations-problèmes issues de notre exemple de départ (gestion des notes) ou portant sur d'autres domaines d'activité. Il semble dans un premier temps souhaitable d'utiliser le même contexte pour l'apprentissage de plusieurs groupes de concepts, afin d'optimiser le temps disponible en évitant aux élèves d'avoir à s'approprier trop de contextes différents. Le passage à d'autres contextes permet ensuite de mesurer la capacité de mobilisation des compétences acquises. Le tableau ci-dessous présente un exemple de progression possible.

Ordre	Concepts associés	Pratique sociale de référence	Domaine
1	BD relationnelle, SGBDR, structure, table, ligne, enregistrement, colonne, champ, nom, valeur, type, clé primaire, clé étrangère, contraintes d'intégrité, manipuler, interroger, mettre à jour, insérer, supprimer, modifier.	Appréhender, analyser et modéliser des situations caractéristiques du fonctionnement des organisations. Alimenter une base de données à partir de documents représentatifs de l'activité de l'organisation (commandes, factures, livraisons, ...).	Gestion de production Gestion des notes
2	Schéma relationnel, structure, données, relation, attribut, nom, valeur, domaine, n-uplet, clé, clé primaire, clé étrangère, contraintes d'intégrité.	Modéliser un système d'information simple à l'aide du modèle relationnel.	Gestion des notes
3	Anomalies, manipuler, interroger, mettre à jour, insérer, supprimer, modifier, forme normale, dépendance fonctionnelle, normalisation.	Vérifier la validité d'un schéma relationnel simple et y apporter des modifications.	Gestion commerciale
4	BD relationnelle, SGBDR, LMD, manipuler, interroger, mettre à jour, insérer, supprimer, modifier, requêtes, SQL, SELECT, FROM WHERE, ORDER BY.	Produire, analyser, contrôler et interpréter les informations issues d'une base de données. Utiliser une base de données à des fins de simulation.	Gestion de production Gestion des notes
5	Opérateurs relationnels : restriction, projection, jointure.	Synthétiser des informations de gestion pour assurer la préparation de décisions et le contrôle des opérations.	Gestion de production Gestion des notes

Gérer les contraintes

Les contraintes sont nombreuses dans l'organisation d'une progression : temps effectivement disponible, organisation des espaces, effectifs, équipements mis à disposition, répartition entre travail de groupe et travail en classe entière.

Il s'agit dans un premier temps de choisir des associations entre les concepts et les pratiques sociales de référence, puis de les placer dans un contexte. Ce choix du contexte est essentiel car il vise à intégrer les dimensions techniques et organisationnelles et à « surfer » en permanence sur l'interrelation entre ces deux domaines afin d'optimiser le temps disponible : il est possible par exemple de présenter les concepts du modèle relationnel à travers l'exploitation d'une gestion de production dont les concepts sont déjà maîtrisés par les élèves, ou présenter les concepts de la gestion de production à travers l'interprétation d'un schéma relationnel, en s'appuyant sur les concepts du modèle relationnel précédemment abordés.

Produire...

Produire une situation permettant de placer les élèves dans une démarche active est une tâche difficile qu'il est souhaitable de mener à plusieurs. En effet, il ne s'agit pas d'inventer tous les jours de nouvelles situations mais de croiser les regards et les expériences pour s'approprier des situations intéressantes et en mesurer les qualités mais aussi les pièges afin de les exploiter dans les meilleures conditions possibles.

Nous présenterons ci-dessous plusieurs situations s'appuyant sur la gestion des notes et visant à présenter différentes démarches pédagogiques complémentaires dont la seule ambition est de proposer quelques pistes de réflexion.

5. Faciliter l'apprentissage des élèves : entre didactique et pédagogie

5.1 Un apprentissage de l'abstraction basée sur l'observation (B.-M. Barth)

Britt-Mari Barth propose une démarche active basée sur l'observation d'exemples et de contre-exemples pour identifier les caractéristiques d'un concept³. Cela se fait à travers un débat collectif permettant de proposer des hypothèses, puis de les vérifier à partir d'autres exemples⁴. Nous illustrerons cette démarche sur l'exemple suivant (très simplifié) : « Comment faire comprendre le concept de clé primaire ? ».

Première phase d'observation et d'exploration

Cette phase s'appuie sur le principe du droit à l'erreur et encourage à l'argumentation.

Le professeur présente d'abord aux élèves des exemples de relations en extension (au sens d'ensembles de n-uplets) avec pour chaque relation une assertion « tel attribut peut être une clé primaire » ou « tel attribut ne peut pas être une clé primaire ». Il s'agit de répondre à la question : « qu'est-ce qui caractérise une clé primaire ? ». L'analyse des différents exemples amènent les élèves à formuler différentes hypothèses sur les caractéristiques d'une clé, comme par exemple « il n'y a jamais deux fois la même valeur » (unicité), « il y a toujours une valeur » (contrainte d'entité).

Le professeur propose ensuite une deuxième série d'exemples consistant à ajouter ou supprimer une ligne dans les tables exemples et à fournir la nouvelle assertion correspondante. Les élèves reprennent les hypothèses et les confrontent aux nouveaux exemples.

Remarque : La méthode nécessite une préparation soignée et une grande réactivité du professeur. Il faut éviter que les élèves ne proposent trop de caractéristiques hors sujet et au besoin pouvoir adapter les exemples proposés pour conforter une idée ou en rejeter une autre (par exemple, une caractéristique « hors sujet » comme « une clé primaire contient toujours des chiffres »).

³ Britt-Mari Barth parle en fait d'attribut, mais nous connaissons les dangers de la polysémie : nous utiliserons donc dans la suite le terme de caractéristique.

⁴ La première illustration fournie par l'auteur est une figure géométrique appelée artificiellement « wez », d'où le nom donné parfois à cette méthode : la méthode du Wez.

Deuxième phase de représentation mentale (évaluation formative)

Après un certain nombre d'exemples, lorsque l'enseignant a le sentiment que les élèves ont compris comment se définit le concept (découverte de toutes ses caractéristiques), il vérifie son hypothèse à partir de la présentation de nouveaux exemples.

Lorsque les élèves ont proposé comme caractéristiques, *au besoin dans leur langage*, l'unicité de la clé et l'impossibilité pour une clé de comporter une « valeur nulle » (absence de valeur), le professeur fournit d'autres exemples et demande pour chacun de répondre à la question « tel attribut peut-il être une clé primaire ? » en argumentant la réponse.

Troisième phase d'abstraction

Le transfert du concept dans un autre contexte (exemples jamais rencontrés auparavant) amène l'élève à nommer toutes les caractéristiques pour justifier sa réponse (cela prouve sa compréhension du concept) et le concept présenté (l'idée générale abstraite).

Pour cela le professeur présente aux élèves de nouveaux exemples dans lesquels les clés primaires sont maintenant multi-attributs et leur demande d'indiquer, en argumentant, si « tel ensemble d'attributs peut être une clé primaire ». Il fait ensuite construire la définition d'une clé à partir des caractéristiques trouvées par les élèves dans la première puis la deuxième phase.

Cette méthode favorise le cheminement vers l'abstraction car elle sollicite des processus psychologiques spécifiques :

- La perception est le mécanisme qui consiste à sélectionner intuitivement les caractéristiques d'un concept. Si l'enseignant n'est pas conscient de l'écart qu'il y a entre sa perception et celles des élèves, il ne peut pas efficacement diriger leur attention sur ce qu'ils doivent percevoir, ce qu'ils doivent apprendre.
- La comparaison (ici entre les concepts de clé mono-attribut et de clé multi-attribut) est le mécanisme utilisant le produit de la perception. Elle se divise, selon B.-M. BARTH, en cinq niveaux :
 - Niveau 0 : L'élève peut nommer les caractéristiques ou des exemples d'un des concepts; sans réelle comparaison.
 - Niveau 1 : L'élève peut nommer, sans les structurer, les caractéristiques ou des exemples des deux concepts à comparer.
 - Niveau 2 : L'élève peut comparer les concepts par rapport à une catégorie, mais ne peut donner d'exemple.
 - Niveau 3 : L'élève peut comparer les concepts par rapport à une catégorie et en donner des exemples.
 - Niveau 4 : L'élève peut comparer les concepts par rapport à une catégorie, en donner des exemples et indiquer en plus s'il s'agit d'une différence ou d'une ressemblance.
- L'inférence et sa vérification représentent le mécanisme qui permet à l'élève de tirer les conséquences de la comparaison. Elle peut être inductive (découvrir et saisir les caractéristiques principales d'un concept par l'intermédiaire d'exemples et en construire une règle générale) ou déductive (construire un concept par syllogisme, à partir d'un concept connu). L'apprenant inductif se sent à l'aise dans une démarche de découverte de ce qui n'est pas encore établi alors que l'apprenant déductif recherche une démarche d'application de ce qui est déjà acquis ou établi.

Vous pouvez arguer que cette méthode nécessite plus de temps qu'une démarche expositive ou frontale qui présente les concepts puis propose des exercices d'applications : mais est-il préférable de passer une heure avec 100 % de compréhension, ou 45 mn avec seulement 50 %...

Cette démarche sollicite une grande maîtrise de la part du professeur. Il lui faut maîtriser d'abord son sujet pour composer des exemples et des contre-exemples de qualité, être en mesure de déterminer dans les propositions parfois inattendues des élèves celles qu'il convient d'occulter et celles qui relèvent d'hypothèses plausibles dans l'état actuel de la réflexion et seront infirmées par la suite... Mais cette démarche n'est pas un simple outil pédagogique pour former à des concepts plus ou moins abstraits. Elle possède également un objectif méta-cognitif, celui de former... à l'abstraction ! Ainsi, le professeur ne sera pas seulement amené à refuser une proposition mais à faire comprendre (ou à démontrer) à l'élève en quoi la proposition faite résulte d'une faute de raisonnement. Le professeur doit prendre conscience de cela et, au-delà du contenu, il devra maîtriser les méthodes de raisonnement que nous qualifierons de « scientifique ». Nous avons à dessein choisi l'exemple de la clé primaire pour être en mesure d'illustrer cette question.

Il est dit plus haut que le professeur montre aux élèves des exemples et des contre-exemples de clé primaire. Quelle peut être la nature de ces illustrations « positives » et « négatives » ? Des extraits de tables présentées en extension ? Des schémas de relation ? La première piste est tentante car elle paraît offrir aux élèves un cadre de réflexion féconde, la deuxième laissant plutôt l'impression que la réponse sera toujours dans la question. Examinons cette piste à partir d'un extrait de la table ELEVE :

ELEVE					
CodeElève	NomElève	PrénomElève	DateNaissance	CodeINSEE	CodeDivision
18	WILDA	Rachid	14-07-1989	1 89 07 75 021 342	7
12	MEZZ	Janine	20-02-1988	2 88 02 77 103 045	4
47	CHICKEN	Aude	12-04-1988	2 88 04 75 021 342	5
115	GROCK	Alain	20-11-1988	1 88 11 77 103 045	2
14	SEM	Alain	13-05-1989	1 89 05 93 018 112	
39	SÉTALA	Maud	23-07-1990	2 90 07 95 145 112	8

Remarque : Les lignes extraites de la table ELEVE n'ont pas été choisies sans critère, vous vous en doutez. Mais, en dehors de la démonstration qui nous occupe (la notion de clé primaire), il convient de ne pas « habituer » l'élève à des présentations à partir desquelles il pourrait incidemment établir des généralités. Ainsi, dans l'exemple présenté, les lignes ont été délibérément choisies sans critère de tri : aucun ordre apparent n'est perceptible pour éviter à l'élève de déduire qu'une table est « toujours rangée », ce qui l'amène souvent à confondre systématiquement l'ordre d'enregistrement des lignes avec un critère d'affichage des lignes, critère répondant au besoin propre d'un utilisateur particulier.

Sur l'exemple présenté, il est possible d'affirmer ce qui suit pour amener les élèves à déterminer les caractéristiques d'une clé primaire, selon la méthode du Wez :

1. Exemple : CodeElève peut être une clé primaire de la table ELEVE.
2. 1^{er} contre-exemple : PrénomElève n'est pas une clé primaire de la table ELEVE.
3. 2^e contre-exemple : CodeDivision n'est pas une clé primaire de la table ELEVE.

Peut-être aurez-vous remarqué la subtilité de la formulation. Commençons par les contre-exemples :

- Nous sommes certains que, dans la table telle qu'elle est présentée, le prénom n'est pas une clé car il y a deux lignes comportant la même valeur ("Alain").
- Nous sommes certains que, dans la table telle qu'elle est présentée, le code de la division à laquelle appartient l'élève n'est pas une clé car il y a une ligne où cette information n'est pas renseignée.

Dans l'objectif qui est le nôtre, nous mettons bien en exergue, au travers de ces contre-exemples, certaines caractéristiques d'une clé primaire : les contraintes d'unicité et d'existence. Mais qu'en est-il de l'exemple « positif » ? Bien sûr, dans l'exemple présenté, le code de l'élève est unique et toujours renseigné. Mais il en va de même du nom, de la date de naissance et du code INSEE de l'élève. Il est toujours possible de réduire petit à petit le champ des investigations en accumulant d'autres exemples : deux élèves portant le même nom, deux élèves nés à la même date (le code INSEE posant un problème particulier). Mais, surtout, ce qu'il faut comprendre, c'est que, à partir d'un extrait de table, **un seul contre-exemple** (doublon, « valeur nulle ») permettra d'affirmer que tel champ **n'est pas** clé primaire, tandis que l'absence de contre-exemple ne permet pas d'affirmer le contraire. Sur l'exemple présenté, il n'est pas possible de déduire « tel champ **est** clé primaire » ; il est seulement possible de **formuler des hypothèses** : telle que la table nous est présentée, tels champs **peuvent être** des clés primaires !

Pour aller plus loin dans la détermination de **la** clé primaire, à supposer qu'il n'y ait pas d'autres clés candidates, il faut éventuellement présenter d'autres exemples explorant le domaine du possible et, surtout, dépasser l'illustration mécanique pour prendre en compte la **signification** de chaque champ. Une clé primaire est en effet chargée d'identifier toute ligne de la table, **présente et à venir**. Sa définition relève finalement d'une **règle de gestion** qui ne considère pas seulement l'objet comme une entité intrinsèque (ici, l'élève qui, en tant qu'être vivant, possède bien sûr un nom, un prénom, une date de naissance... avant même qu'un établissement scolaire ne « songe à l'enregistrer ») mais comme un objet informationnel, une composante de l'organisation, que l'organisation a besoin d'identifier précisément.

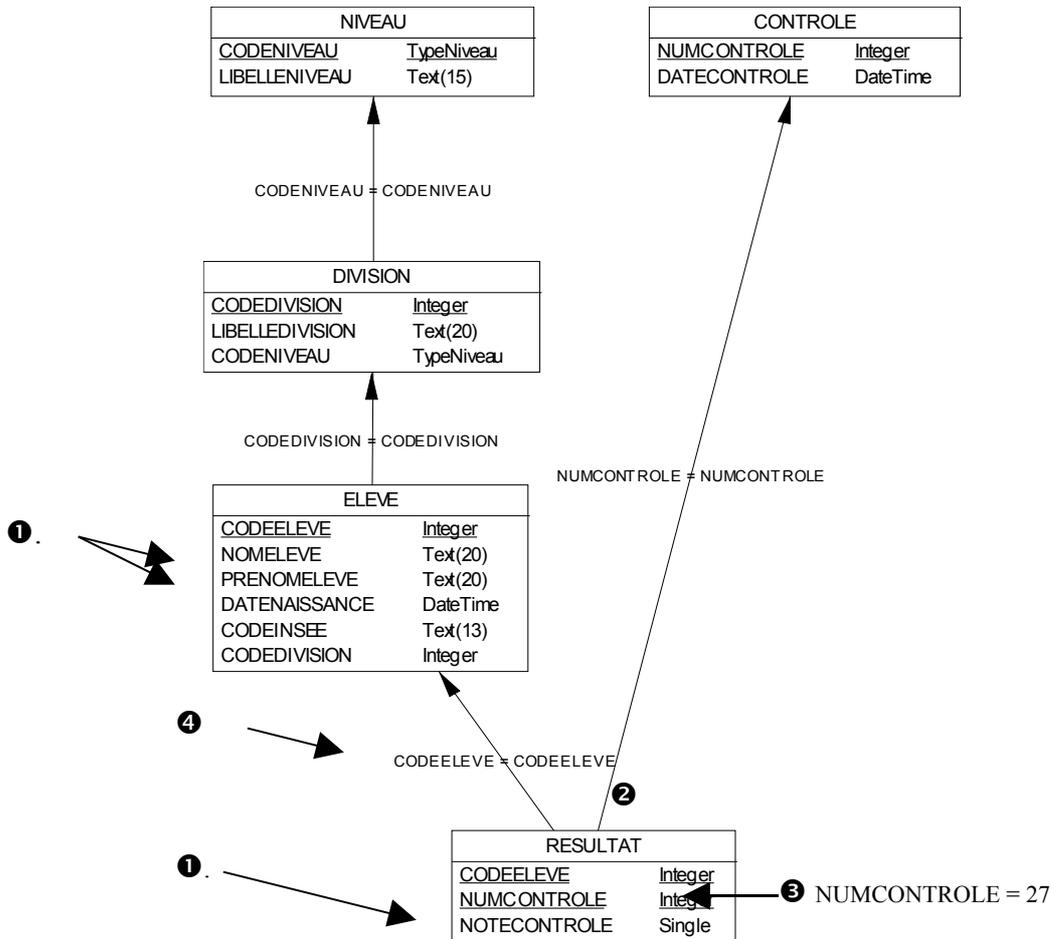
Dans de telles conditions, fréquentes dans ce domaine de connaissance, il nous semble que la démarche s'appuyant sur la résolution de problèmes (voir [5.3](#)) est moins délicate à mettre en œuvre de la part du professeur.

5.2 La technique de modelage

Le modelage consiste pour l'enseignant à réaliser une tâche en position d'expert de telle sorte que les élèves puissent observer sa manière de faire et puissent construire un modèle mental des processus nécessaires à cette réalisation.

Cette démarche est très utile pour des connaissances de type opérationnel. Nous en donnerons un exemple concernant l'apprentissage de la conception d'une [requête](#).

Nous supposons comme prérequis la connaissance de la base de données servant de support, la connaissance des opérateurs relationnels mis en œuvre, étudiés sur des exemples et testés sur la base de données. Nous reprendrons ici le cas « Gestion des notes » :



Le professeur annonce qu'il va résoudre un problème et demande aux élèves de relever les étapes de son raisonnement. Il construit ensuite une requête en raisonnant à voix haute, en plaçant son expertise au niveau de perception de ses élèves et en accompagnant sa réflexion d'une annotation du schéma (mise en évidence des attributs, des tables et des liens mis en jeu) :

« Il s'agit de réaliser la requête qui me permet d'obtenir les résultats des élèves au contrôle 27. Pour chaque résultat, je veux afficher le nom et le prénom de l'élève concerné et sa note.

Je pars du résultat que je veux obtenir donc je recense d'abord les informations à afficher et je cherche les attributs correspondants : *NomElève*, *PrénomElève* et *NoteContrôle* **1**. Les deux premiers se trouvent dans la table ELEVE, le troisième dans la table RESULTAT. Parmi ces deux tables, c'est la table RESULTAT qui représente le mieux le but à atteindre⁵ puisqu'on cherche des résultats (je l'appelle la colonne vertébrale de ma requête) **2**.

Est-ce que je dois afficher tous les résultats ? Non, seulement ceux du contrôle 27. Je recherche donc la condition qui permet d'exprimer cette [restriction](#) : il faut que le numéro de contrôle soit égal à 27. Je dois donc vérifier que *NumContrôle* = 27 **3**.

⁵ Cette notion de colonne vertébrale est un « artefact » dont l'objectif est d'amener les élèves à toujours revenir au sens de la question : il s'agit de déterminer l'objet ou le concept qui est le plus proche du résultat à atteindre et qui servira de point de départ à la conception de la requête.

Il me faut donc aussi l'attribut *NumContrôle*. Est-il dans les tables que j'ai déjà repérées ? Oui, il est dans la table RESULTAT. Je n'ai donc pas besoin d'une autre table.

J'ai donc besoin des tables RESULTAT et ELEVE. Existe-t-il un lien entre elles ? Oui : il est représenté par la clé étrangère *NumElève* qui fait référence à la clé primaire de la table ELEVE pour exprimer le fait qu'un résultat à un contrôle ne concerne qu'un seul élève. Il faut donc que ces deux attributs aient la même valeur : chaque fois que j'ai un résultat, je récupère ainsi le nom et le prénom de l'élève qui est repéré par ce code ④. Nous avons vu qu'il s'agissait d'une [jointure](#).

Je veux donc afficher les attributs *NomElève*, *PrénomElève*, *NoteContrôle* (*nous avons vu qu'il s'agissait d'une projection*) qui se trouvent dans les tables RESULTAT et ELEVE pour lesquelles les attributs *CodeElève* ont la même valeur et l'attribut *NumContrôle* a pour valeur 27.

Je peux maintenant passer à l'écriture de ma requête SQL en plaçant les attributs à afficher derrière la clause SELECT, les tables utilisées derrière la clause FROM et les conditions à utiliser derrière la clause WHERE. Je relie les deux conditions par l'opérateur AND pour traduire le fait que les deux conditions doivent être systématiquement réunies.

```
SELECT      NomElève, PrénomElève, NoteContrôle
FROM        RESULTAT, ELEVE
WHERE       RESULTAT.Elève = ELEVE.CodeElève
AND        NumContrôle = 27
```

Après cette première étape, l'enseignant questionne les élèves pour les amener à repérer les différentes phases de son raisonnement. Il s'appuie pour cela sur un réseau fonctionnel qu'il construit tout au long du questionnement et qui décrit sa démarche de conception de requêtes comme une succession de tâches permettant d'obtenir le résultat attendu (*ce peut être une première approche de la notion d'algorithme*).

Il propose ensuite une deuxième requête et se place en situation d'exécutant : c'est un élève qui dirige le processus de conception, le professeur demandant simplement de préciser les opérations à réaliser en cas d'ambiguïté (une approche de l'exécution d'un programme...) et produisant au fur et à mesure les tâches demandées. Le mode opératoire est éventuellement modifié en conséquence.

Les élèves sont ensuite invités à réaliser une ou plusieurs requêtes en utilisant le mode opératoire qu'ils ont ainsi construit. Une synthèse collective permet d'évaluer la compréhension des élèves et d'envisager quelques requêtes à préparer pour la prochaine séance qui permettront aux élèves d'asseoir leurs nouvelles connaissances.

5.3 Un apprentissage de l'abstraction basée sur la résolution de problèmes

L'apprentissage par résolution de problèmes est une des approches pédagogiques les plus adaptées à notre enseignement. Il fournit aux élèves un but et une méthode de travail communs au sein d'un petit groupe d'apprentissage. Il leur permet ainsi de relever ensemble des défis et d'augmenter leur confiance en leurs capacités en surmontant les obstacles rencontrés.

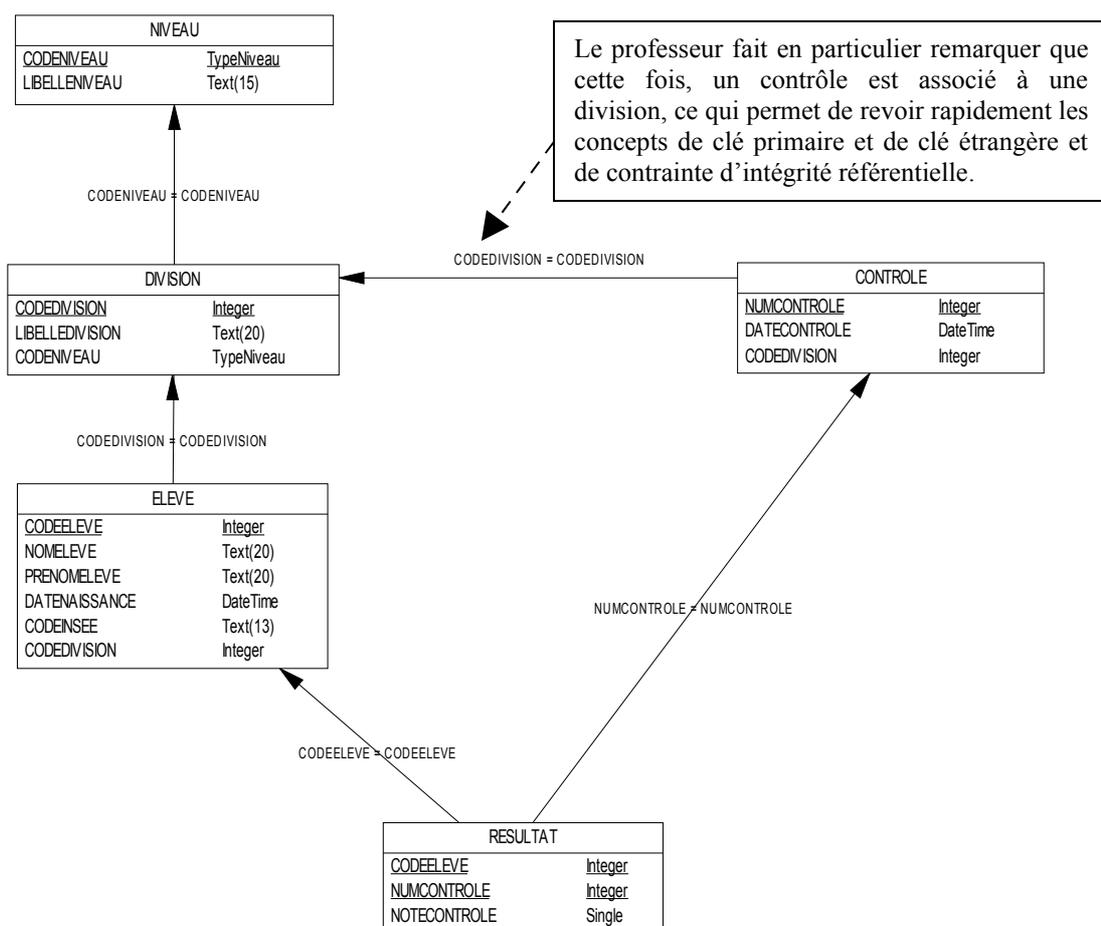
Nous vous proposons d'aborder cette technique à travers l'objectif suivant : « Développer une démarche de construction d'une requête complexe ».

- Nous partons d'une situation concrète « posant problème » qui garde un caractère réel, sans simplification excessive (*phase préparatoire*).

Nous nous appuyons pour cela sur une deuxième version de la gestion des notes *dans laquelle désormais un contrôle est toujours associé à une division*⁶. L'objectif choisi est de mettre en évidence l'impossibilité de répondre à des questions complexes en utilisant la technique employée jusqu'ici. L'exercice doit amener les élèves à mobiliser et compléter leurs compétences pour atteindre cet objectif (décomposition d'un problème, construction de requêtes intermédiaires). Elle s'appuie sur la pratique de référence « Exploiter une base de données », dans sa dimension « observation et interrogation ».

- Les ressources appropriées sont fournies (documents, base de données) et les concepts employés sont au besoin clarifiés (*phase de présentation collective*).

La base de données est fournie (avec un jeu d'essai le plus complet possible), ainsi que les explications concernant les règles de gestion.



Selon la situation dans l'année, ces explications pourront prendre un aspect formel ou être diluées dans une série de documents à exploiter.

⁶ Dans le nouveau schéma, se pose la question de la « transitivité ». Cette difficulté est traitée au point 6.2.

- Le problème est posé (*phase de compréhension*) et sont entamées des démarches d'observation, d'analyse, de recherche, d'évaluation, et de réflexion (*phase des hypothèses*). Le professeur fait alterner des temps de travail en équipe (*phase d'analyse du problème, de remue-méninges initial*) et des temps de travail individuel (*phase de réflexion, d'évaluation*), pour amener les élèves à proposer des démarches pour répondre à la question (*phase de planification*).

« Voyez-vous, j'ai construit une base de données qui me permet de mémoriser tous les résultats de mes élèves. Mais j'ai un problème : j'aimerais savoir quels sont les élèves qui n'ont jamais été absents à mes contrôles. En début d'année c'est facile : il me suffit d'observer le contenu des tables. Mais cela devient impossible en fin d'année. Maintenant que vous savez réaliser des requêtes d'interrogation, je compte sur vous pour m'indiquer comment faire. »

Les élèves sont invités à comprendre le système d'information représenté. Ils peuvent s'appuyer sur la réalisation de requêtes simples, l'analyse du schéma relationnel et des documents. Ainsi est mis en évidence le conflit cognitif qui réside ici dans l'impossibilité de résoudre le problème par une simple requête, avec leurs connaissances actuelles.

Les élèves recherchent ensuite des ébauches de solution, individuellement ou en petit groupe. Chaque ébauche de solution est soumise au groupe et passée au crible des connaissances acquises. Pour chacune, les connaissances qui manquent sont recensées (pour faire...il faudrait être capable de faire...).

Il faut prendre soin dans cette phase d'avoir une approche facilitatrice, sans exclure aucune des propositions. Une phase de synthèse permettra ensuite d'isoler les propositions réalisables et utiles pour l'apprentissage et de planifier le travail (recherche d'informations, apports de nouvelles connaissances, pratique, etc.).

Dans un premier temps, il est possible par exemple d'accepter l'idée de modifier la base de données pour enregistrer explicitement les absences. Mais le professeur pourra la mettre en attente en précisant qu'il convient de s'intéresser d'abord aux solutions qui permettent d'éviter une modification, en rappelant la contrainte de départ (observation et interrogation).

Si une proposition consiste à compter les contrôles passés par chaque élève et à comparer avec le nombre de contrôles proposés à la classe, le professeur pourra apporter la notion de fonctions agrégatives, éventuellement la notion de regroupement (ou partitionnement) selon le niveau atteint par les élèves. Il pourra aussi s'appuyer sur des savoir-faire maîtrisés en exportant des données dans une feuille de calcul et en exploitant des fonctions de cumuls. Il pourra enfin initier les élèves à la logique algorithmique en proposant un programme itératif à analyser.

- Le professeur accompagne le processus de construction des solutions et met en place les évaluations nécessaires pour réguler le processus d'apprentissage et mesurer le chemin accompli (*phase de travail en groupe restreint, avec ou sans assistance, phase de travail individuel et activité pratique*).

Les élèves développent leur proposition de solutions en groupe ou individuellement. Pour chaque solution, le professeur prend soin d'apporter les connaissances complémentaires nécessaires en s'appuyant sur différentes techniques (voir le point [5.4](#)) en fonction du degré d'avancement des élèves dans leur acquisition des connaissances. Les apports de connaissance et l'évaluation peuvent aussi s'appuyer sur des points d'étape : analyse et décomposition du problème, apport d'une solution à chaque partie du problème, association des éléments de solution pour répondre au problème, test du résultat. La démarche peut s'appuyer sur un travail coopératif (les élèves travaillent ensemble sur la résolution du problème) ou sur un travail collaboratif (les élèves travaillent séparément à la résolution d'une partie du problème et se coordonnent pour résoudre le problème).

- Le professeur propose d'examiner collectivement le cheminement parcouru pour aider les élèves à formaliser les stratégies mises en œuvre (*phase de réflexion collective*).

Un débat permet à chaque groupe de présenter sa solution et de la comparer aux objectifs définis en début d'activité. L'enseignant amène ainsi les élèves à découvrir que répondre à une question complexe nécessite de comprendre la question, de comprendre le contenu de la base manipulée et de transformer le problème en un ou plusieurs problèmes dont la résolution permet d'obtenir la réponse attendue. La synthèse collective permet de recenser les nouvelles connaissances acquises.

- Le professeur termine en demandant à chaque élève de rédiger la synthèse des connaissances acquises (*phase de travail individuel*)

Il est demandé aux élèves de construire ou d'enrichir un réseau de concepts (carte conceptuelle) et de rédiger des définitions et des modes opératoires personnels (réseau fonctionnel, algorithme) qui seront ensuite publiés sur

l'intranet du lycée. Cette phase doit donner toute son importance à l'écrit comme outil de structuration et d'acquisition des connaissances.

Ces différentes phases peuvent être accompagnées d'activités en groupe classe (exposés de synthèse, modélisation).

Remarquons qu'il est parfois difficile de distinguer la pédagogie par le projet et l'apprentissage par résolution de problèmes.

De manière un peu caricaturale, nous dirons que, pour la première, le but à atteindre est clairement défini et il s'agit de construire la démarche qui permettra d'atteindre ce but (il s'agit en priorité d'appliquer des connaissances déjà acquises).

Pour la seconde, le but est de trouver la (ou les) solution(s) au problème et il s'agit de dénouer la situation-problème pour y parvenir (il s'agit en priorité d'acquérir de nouvelles connaissances).

Tout l'art de la didactique est certainement de trouver un juste compromis entre ces deux approches qui sache tenir compte du contexte (période de l'année, profil des élèves, temps à consacrer, etc.).

5.4 Quelques techniques utiles

Nous présentons ici quatre techniques pédagogiques à travers une situation d'apprentissage.

La technique de l'étayage

L'étayage consiste à soutenir l'élève dans sa démarche, à l'accompagner en lui apportant des suggestions, des conseils, des supports afin de le mettre progressivement sur la voie. L'objectif est bien sûr d'arriver à réduire progressivement ce soutien pour permettre à l'élève d'accéder à l'autonomie.

La technique du guidage

Lorsque les élèves sont plus autonomes, le professeur peut pratiquer le guidage : une observation attentive des élèves dans leur tâche permet à l'enseignant de proposer des pistes de réflexion, des rappels, des ressources ou de nouvelles tâches pour leur permettre de maîtriser la démarche, mais aussi les amener à en parler.

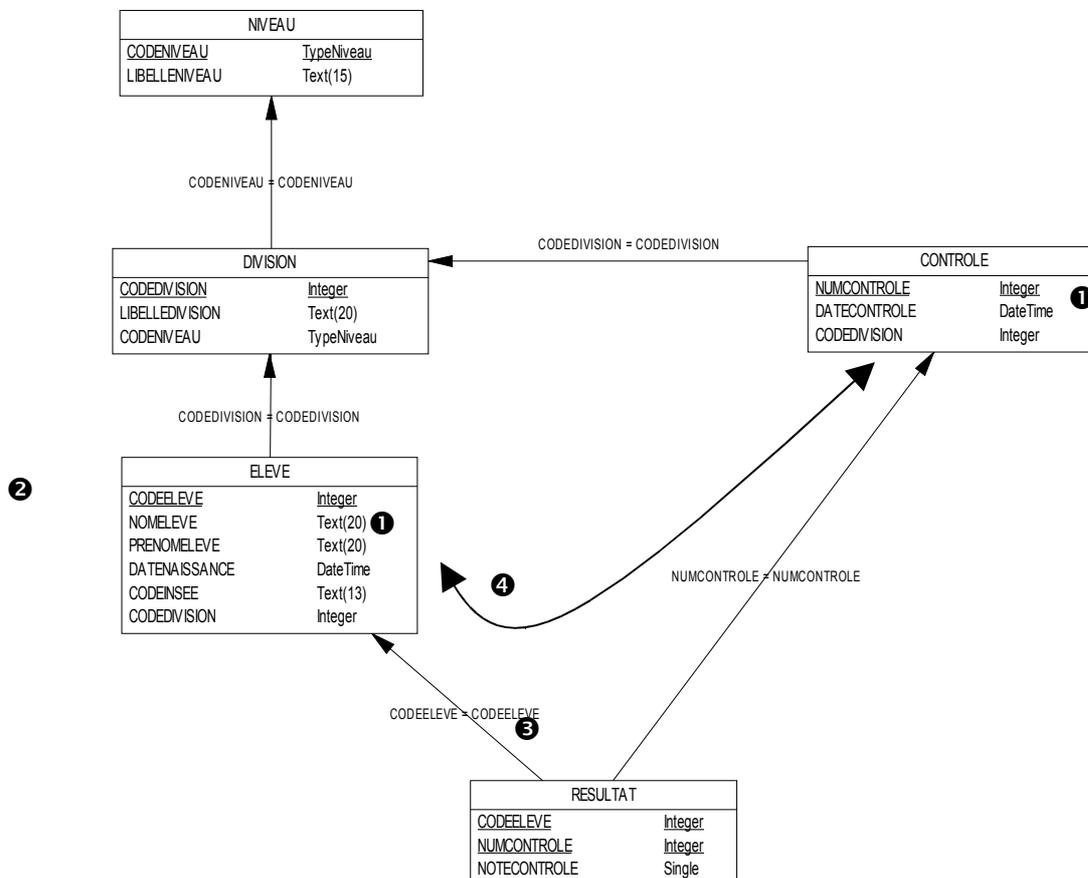
La technique de réflexion collective

La technique de réflexion collective vise à mettre les élèves en situation de comparaison critique de leur démarche respective de résolution d'un problème (et non pas du résultat obtenu). Il vise à renforcer l'acquis par la mise en place d'un débat argumentatif et d'une analyse des pratiques, ainsi qu'à améliorer les techniques de communication.

La technique du détour pédagogique

La technique du détour pédagogique consiste à remédier à l'impossibilité pour les élèves de lever l'obstacle cognitif repéré avec les moyens prévus ou à l'apparition d'un obstacle cognitif inattendu. Il s'agit alors de rechercher une autre méthode d'apprentissage pour permettre aux élèves de surmonter la difficulté en s'appuyant sur leur représentation mentale du moment.

Nous illustrerons ces techniques dans une séance consacrée à la réalisation de requêtes plus complexes permettant de conforter la compétence des élèves dans leur démarche de conception. Cette séance s'appuie toujours sur la gestion des notes (ici dans sa deuxième version).



Prenons ainsi l'exemple suivant : « Je veux afficher les dates de tous les contrôles passés par l'élève de code 235, en indiquant son nom ».

En utilisant la démarche précédente, les élèves vont repérer les attributs à afficher (*DateContrôle et NomElève*) ❶ et en déduire les relations nécessaires (ELEVE et CONTROLE) et la relation « colonne vertébrale » (CONTROLE). Ils doivent trouver la condition (*CodeElève = 235*) et confirmer qu'ils ont besoin de la relation ELEVE ❷.

Mais à la question « Existe-t-il un lien entre les relations CONTROLE et ELEVE ? », certains se trouvent face à un cas apparemment non prévu car le lien n'est pas direct et nécessite d'utiliser la relation RESULTAT (et non pas la table DIVISION) ❸.

Il est alors possible de montrer sur le graphique représentant le schéma relationnel le chemin pertinent ❹ permettant de relier ces deux relations et expliquer ensuite la nécessité de trouver les correspondances (*approche par étayage qui se place au niveau opératoire*).

Il est possible aussi d'utiliser un questionnement adapté pour faire découvrir la notion de lien indirect et le besoin d'utiliser la relation intermédiaire RESULTAT plutôt que la relation DIVISION en invitant à préciser la question en revenant au sens porté par le schéma (*approche par guidage qui agit au niveau du sens*) :

« Comment exprime-t-on dans le schéma le fait qu'un élève passe un contrôle ? On le fait à l'aide de la relation RESULTAT (qui représente l'ensemble des résultats obtenus par les élèves aux contrôles). Comment peut-on donc exprimer la question de façon plus précise ? On veut afficher les dates de tous les contrôles pour lesquels l'élève 235 a obtenu un résultat. Il faut donc bien utiliser la relation RESULTAT ».

Le professeur peut également mettre les élèves en confrontation en les invitant à présenter et à argumenter leurs hypothèses (*technique de réflexion collective*).

S'il détecte un problème de compréhension du concept de clé étrangère et de son intérêt dans la réalisation d'une requête, il peut leur proposer une manipulation concrète (*approche par le détour pédagogique*).

Il fournit aux élèves des relations en extension sous forme de liste papier et leur demande de construire manuellement le résultat attendu en détaillant la démarche utilisée. La démarche sera ensuite mise en œuvre par un élève jouant le rôle d'exécutant et suivant à la lettre les consignes. La mise en évidence d'une analogie de structure avec la base de données manipulée permet de revenir au problème étudié.

Une fois la méthode et les concepts maîtrisés par les élèves, le professeur pourra utiliser une autre base de données présentant une structure similaire afin d'amener les élèves à prendre conscience des analogies de structure à travers la réalisation de requêtes de même type.

6. Du savoir savant au savoir enseigné : anticiper les erreurs

L'enseignement du modèle relationnel peut être basé sur une stratégie organisationnelle (stratégie mentale du sujet visant à traiter l'information) qui permet à l'élève, en comprenant comment l'ordinateur s'y prend pour structurer les données et les rendre exploitables par les utilisateurs, de structurer ses propres connaissances par une réflexion sur sa façon de les construire.

Il faut pour cela transformer un ou plusieurs objectifs d'apprentissage en une situation d'apprentissage, ce qui passe par la recherche des difficultés que peuvent rencontrer les élèves et la compréhension des causes d'erreur possibles pour aider les élèves à les surmonter.

6.1 Erreur : statut et typologie

Apprendre, c'est avant tout prendre conscience de ses erreurs et de son profil d'apprentissage ; c'est passer de la perception d'un concept à une autre, par enrichissement ou par transformation. C'est nécessairement être confronté à des obstacles, à des périodes de doute qui nécessitent de faire preuve de volonté et d'effort pour faire évoluer sa perception du monde. L'erreur est donc un acte nécessaire pour progresser, que l'enseignant doit intégrer à sa stratégie pédagogique (« Identifier son erreur et en prendre conscience, c'est la résoudre » Stella Baruk).

Astolfi [« L'erreur, un outil pour enseigner » -1997] propose une typologie des erreurs, que nous pouvons regrouper en deux catégories : celles qu'il faut éviter et celles qu'il faut exploiter (et donc susciter).

Parmi les erreurs à éviter, il convient de distinguer celles qui sont d'origine sociale au sens large et celles qui sont imputables à l'enseignant.

Les erreurs d'origine sociale résultent le plus souvent d'un mauvais décodage du contrat pédagogique : il va de l'acceptation totale de la « panoplie du bon élève » (l'élève répond au professeur avant de répondre au problème) au refus aussi total du système éducatif en passant par la perte de l'estime de soi (l'élève ne répond pas au problème parce qu'il est persuadé que cela ne sert à rien ou qu'il ne saura pas). L'apprentissage du modèle relationnel peut participer à un changement de comportement dans ce domaine, en confrontant les élèves à des problèmes porteurs de sens et en mettant l'accent sur le fait qu'un problème n'a pas forcément une solution unique.

Les autres erreurs à éviter sont le plus souvent de la responsabilité de l'enseignant, par ses choix de transposition didactique et ses options pédagogiques.

Les erreurs dues à une mauvaise compréhension des consignes.

Elles sont souvent la conséquence d'un questionnement insuffisamment clair ou précis : il importe donc de fixer clairement les objectifs et d'éviter les consignes passe-partout qui commencent par des termes comme analyser, interpréter, décrire, etc. Ces erreurs peuvent également être provoquées par une méconnaissance du vocabulaire spécifique à la discipline ou par une confusion avec un vocabulaire du langage courant. Vous pouvez facilement imaginer en quoi le caractère polysémique des termes *relation* et *clé* peut être source d'ambiguïté, ou le passage régulier et délicat entre deux niveaux d'abstraction.

Les erreurs dues à la complexité du contenu

Elles dénotent souvent (lorsqu'elles surviennent pour une majorité des élèves) d'une inadéquation des concepts présentés ou des opérations intellectuelles demandées au niveau des élèves, donc d'une transposition didactique imparfaite. Ainsi, il semble peu raisonnable de vouloir aborder le modèle relationnel en classe de première à partir des fondements mathématiques (théorie ensembliste ou théorie des prédicats) et d'envisager la construction et la normalisation de schémas complexes.

Les erreurs dues à une surcharge cognitive

Elles résultent le plus souvent d'une accumulation d'informations et de consignes qui entraînent une trop forte mobilisation de la mémoire à court terme. L'élève se concentre alors sur certains aspects en délaissant les autres. Il importe donc de sérier les difficultés et de s'appuyer davantage sur la notion de situation-problème [Astolfi 1997] que de projet. Nous avons déjà abordé ce point.

Toutes les autres erreurs s'adosent aux représentations mentales de l'élève. Il faut les mettre en évidence (susciter) pour faire évoluer ces représentations à travers la proposition d'obstacles cognitifs à surmonter.

Enseigner, c'est donc prévoir ou susciter les erreurs des élèves en recensant d'abord les difficultés auxquelles ils devront faire face.

6.2 Modèle relationnel : Quelques difficultés repérées

Première difficulté : Représenter l'information n'est pas la maîtriser

Une même donnée peut conduire à plusieurs informations selon le sujet ou le contexte associé. Elle ne prend son sens (ne devient une information) qu'une fois mise en relation avec d'autres données et intégrée à un ensemble de liens existants.

L'interprétation d'une base de données consiste ainsi à repérer sa structure et à construire une représentation mentale permettant de lui donner un sens, parmi une multiplicité d'interprétations possibles.

Il importe donc de prendre conscience de la part d'incomplétude d'un schéma relationnel et de lever toutes les ambiguïtés, en mettant à disposition de l'élève tous les éléments permettant de résoudre le problème (explications, documents, interview, etc.).

Ainsi, dans le schéma relationnel suivant :

VEHICULE (CodeVéhicule, Immatriculation, Kilométrage, PuissanceFiscale, CodePersonnel,...)
CodeVéhicule : clé primaire
CodePersonnel : clé étrangère en référence à CodePersonnel de PERSONNEL
PERSONNEL (CodePersonnel, NomPersonnel, CodeVéhicule)
CodePersonnel : clé primaire
CodeVéhicule : clé étrangère en référence à CodeVéhicule de VEHICULE

Bien malin qui pourra dire quels liens, représentés d'une part par la clé étrangère *CodePersonnel* entre les relations *VEHICULE* et *PERSONNEL*, d'autre part par la clé étrangère *CodeVoiture* entre les relations *PERSONNEL* et *VEHICULE*, sont exprimés parmi les suivants :

- un membre du personnel s'est vu attribué un véhicule ;
- un membre du personnel est propriétaire d'un véhicule qu'il est autorisé à utiliser pour son travail ;
- un véhicule a été conduit la dernière fois par un membre du personnel ;
- un membre du personnel est responsable du suivi du véhicule ; ...

Et dans le schéma suivant :

SERVICE (CodeService, NomService, CodePersonnel, ...)
CodeService : clé primaire
CodePersonnel : clé étrangère en référence à CodePersonnel de PERSONNEL
PERSONNEL (CodePersonnel, NomPersonnel, CodeService, ...)
CodePersonnel : clé primaire
CodeService : clé étrangère en référence à CodeService de SERVICE

Est-il si évident au premier abord pour un élève que le premier lien exprime une responsabilité (un service a un seul chef) et que le second exprime une appartenance (un membre du personnel n'est affecté qu'à un seul service) ?

Les schémas suivants ne sont-ils pas plus lisibles lorsque les rôles sont précisément nommés :

VEHICULE (CodeVéhicule, Immatriculation, Kilométrage, PuissanceFiscale, ChargéDuSuivi...)
CodeVéhicule : clé primaire
ChargéDuSuivi : clé étrangère en référence à CodePersonnel de PERSONNEL
PERSONNEL (CodePersonnel, NomPersonnel, VéhiculeAffecté)
CodePersonnel : clé primaire
VéhiculeAffecté : clé étrangère en référence à CodeVéhicule de VEHICULE

SERVICE (**CodeService**, NomService, **ChefDeService**, ...)
 CodeService : clé primaire
 ChefDeService : clé étrangère en référence à CodePersonnel de PERSONNEL

PERSONNEL (**CodePersonnel**, NomPersonnel, **ServiceDAffectation**, ...)
 CodePersonnel : clé primaire
 ServiceDAffectation : clé étrangère en référence à CodeService de SERVICE

Quand est-il utile de distinguer le nom d'une clé étrangère du nom de la clé primaire référencée ? Une idée simple peut être exploitée : ne pas donner le même nom à des ensembles différents. Nous pouvons alors envisager deux cas :

- La clé étrangère admet des « valeurs nulles ».

Ainsi, dans la relation ELEVE(CodeEleve, ... , CodeDivision), où CodeDivision, clé étrangère, fait référence à la clé primaire de la relation DIVISION : admettons que, à tout moment, tous les élèves ne sont pas forcément rattachés à une division (sauf règle de gestion contraire, on n'a pas forcément $ELEVE[CodeDivision]^7 = DIVISION[CodeDivision]$ mais $ELEVE[CodeDivision] \subseteq DIVISION[CodeDivision]$) ; dans ce cas, il est fortement conseillé d'employer un nom de rôle pour la clé étrangère, comme « DivisionDAffectation ».

- Toutes les valeurs de la clé primaire ne sont pas forcément référencées.

Dans le cas évoqué précédemment, nous avons écrit « sauf règle de gestion contraire ». Cette précision a son importance. En effet, le système d'information peut très bien comporter les deux règles de gestion suivantes :

1. *Un élève n'est pas forcément rattaché à une division.*
2. *Une division ne peut pas exister sans qu'au moins un élève y soit rattaché.*

Cette deuxième contrainte implique que, malgré l'éventualité de « valeurs nulles » pour la clé étrangère, l'égalité $ELEVE[CodeDivision] = DIVISION[CodeDivision]$ sera toujours assurée. Logiquement, la clé étrangère peut porter le même nom que la clé primaire qu'elle référence.

Dans l'autre exemple qui concerne des véhicules, admettons les règles de gestion suivantes :

1. *Tous les véhicules sont suivis par un membre du personnel, mais seuls certains membres du personnel assurent le suivi des véhicules.*
2. *Certains membres du personnel se voient affecter un véhicule, mais il n'existe pas d'autres véhicules que ceux qui sont affectés aux membres du personnel.*

Ces règles conduisent à conseiller le schéma suivant :

VEHICULE (**CodeVéhicule**, Immatriculation, Kilométrage, PuissanceFiscale, **ChargéDuSuivi**...)
 CodeVéhicule : clé primaire
 ChargéDuSuivi : clé étrangère en référence à CodePersonnel de PERSONNEL
 /* D'après la règle de gestion n° 1 : $VEHICULE[ChargéDuSuivi] \subseteq PERSONNEL[CodePersonnel]$ */

PERSONNEL (**CodePersonnel**, NomPersonnel, **CodeVéhicule**)
 CodePersonnel : clé primaire
 CodeVéhicule : clé étrangère en référence à CodeVéhicule de VEHICULE
 /* D'après la règle de gestion n° 2 : $PERSONNEL[CodeVéhicule] = VEHICULE[CodeVéhicule]$ */

Ce deuxième type de raisonnement est sans doute un peu plus difficile à expliquer : c'est pourquoi nous conseillons de ne pas tenir compte de ce cas particulier et d'appliquer les règles énoncées dans l'ordre :

1. *Je donne un nom de rôle à toutes les clés étrangères qui admettent des « valeurs nulles ».*
2. *Pour les autres clés étrangères, je donne un nom de rôle si toutes les valeurs de la clé primaire référencée ne sont pas forcément prises par la clé étrangère.*

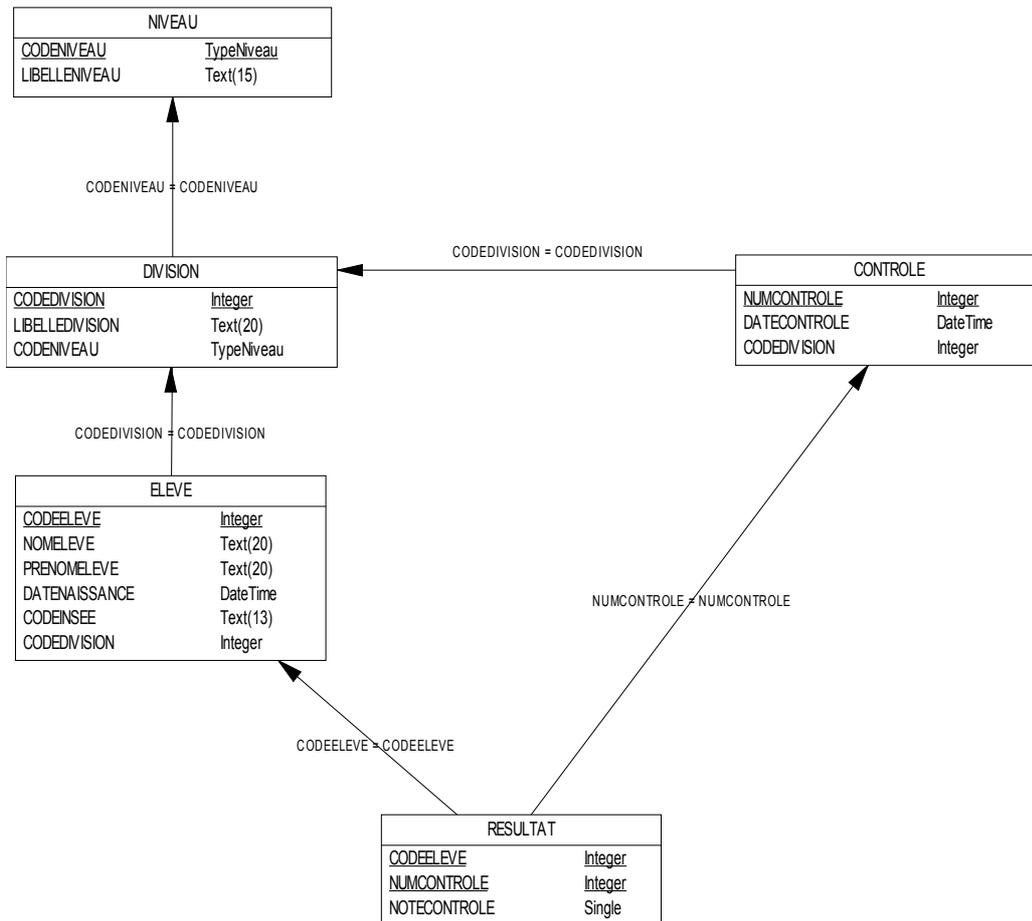
Ces règles nous semblent s'imposer *a minima*. Car, au-delà du risque d'interpréter faussement un schéma relationnel, désigner systématiquement une clé étrangère avec le même nom que la clé primaire référencée conduit à des représentations mentales erronées du modèle relationnel lui-même. Il existe des exemples de ces erreurs dans la littérature pédagogique : il est notamment écrit dans certains ouvrages qu'il y a « clé étrangère » quand « deux relations partagent le même attribut » ou encore « quand deux relations ont un attribut commun ». Il y a fort à parier que, le professeur usant de cette facilité en apparence anodine, l'élève accède à ce genre de représentation mentale erronée. Il ne serait donc pas inconvenant de systématiser la prise en compte du rôle dans la désignation des clés étrangères !

⁷ $ELEVE[CodeDivision]$ se lit : projection de la relation ELEVE sur l'attribut CodeDivision. Une autre écriture possible est proposée en annexe 3 : PROJ(ELEVE, (CodeDivision))

D'une façon plus générale, il est indispensable de privilégier une approche porteuse de sens plutôt que de tenir un discours techniciste. Et pour cela il faut que le schéma s'appuie sur une réalité exprimée par des règles de gestion clairement identifiées qui peuvent être fournies à travers des textes ou des interviews décrivant le fonctionnement de l'organisation décrite.

Deuxième difficulté : vraie et fausse transitivité

Reprenons l'exemple dans lequel les contrôles sont destinés à une division :



À partir de ce schéma, nous allons évoquer de façon pragmatique le « piège » de la transitivité, auquel conduit une interprétation étroite de la théorie de la normalisation des relations.

Sur le schéma, nous observons que les flèches indiquant les références des clés étrangères composent deux chemins qui ont la même origine (RESULTAT) et le même but (DIVISION). Dès lors, nous pouvons nous poser la question de la redondance : y a-t-il un chemin superflu ? En l'occurrence, la réponse est négative :

- Pouvons-nous supprimer le lien entre RESULTAT et ELEVÉ ou entre RESULTAT et CONTRÔLE ? La réponse est non : la clé primaire de RESULTAT est composée de clés étrangères faisant précisément référence aux deux autres relations.
- Pouvons-nous supprimer le lien entre ELEVÉ et DIVISION ? La réponse est non : si ce lien est supprimé, comment déterminer les contrôles que doit passer un élève ? Comment connaître la division d'un élève n'ayant encore passé aucun contrôle ?
- Pouvons-nous supprimer le lien entre CONTRÔLE et DIVISION ? La réponse est évidemment non, pour des raisons semblables.

La notion de transitivité est donc à mobiliser avec beaucoup de circonspection. Comme cela est illustré ci-dessus, il faut considérer l'ordonnancement des faits. Il convient également de repérer les cas où les multiples dépendances fonctionnelles ayant la même cible expriment en fait différents rôles joués par le même concept dans un schéma relationnel. Nous allons illustrer ce cas sur un exemple simple.

Nous disposons d'une relation COMMUNE :

COMMUNE(NumCommune, NomCommune, NumDépartement)

NumCommune : clé primaire (par hypothèse)

NumDépartement : clé étrangère en référence à NumDépartement de DEPARTEMENT (relation non présentée ici)

Nous allons exprimer qu'un délinquant est né dans une commune et qu'il est suivi par un éducateur :

DELINQUANT(NumDélinquant, NomPrénomDélinquant, ... , NumComNaissance, NumEduc)

NumDélinquant : clé primaire

NumComNaissance : clé étrangère en référence à NumCommune de COMMUNE

NumEduc : clé étrangère en référence à NumEduc de EDUCATEUR

Nous exprimons maintenant qu'un éducateur réside dans une certaine commune :

EDUCATEUR(NumEduc, NomPrénomEduc, NumComRésidence)

NumEduc : clé primaire

NumComRésidence : clé étrangère en référence à NumCommune de COMMUNE

Si nous considérons, sur le même thème, les seules dépendances fonctionnelles, un problème se pose :

NumDélinquant \rightarrow NumEduc

NumDélinquant \rightarrow NumCommune

NumEduc \rightarrow NumCommune

Une interprétation étroite de la notion de transitivité nous amènerait à considérer que la deuxième DF (en gras) n'est pas directe, qu'elle est reconstituable par transitivité. Mais, dans les faits, comment pourrait-on déduire la commune de naissance d'un délinquant à partir de la commune de résidence de son éducateur ?

En fait, COMMUNE est un concept qui joue bien deux rôles distincts dans notre système : commune de naissance des délinquants d'une part, commune de résidence des éducateurs de l'autre. C'est d'ailleurs la raison pour laquelle nous avons préféré ne pas nommer les clés étrangères du même nom que la clé primaire et tenir compte des rôles dans leur désignation. Il conviendrait alors d'écrire :

NumDélinquant \rightarrow NumComNaissance

NumEduc \rightarrow NumComRésidence

Il faudra alors mentionner que NumComNaissance et NumComRésidence sont deux rôles joués par le même domaine NomCommune. Par exemple⁸ :

NumComNaissance \Rightarrow NomCommune

NumComRésidence \Rightarrow NomCommune

Notez bien que, dans cet exemple, nous avons choisi pour la commodité de l'exposé des rôles se distinguant très nettement : naissance, résidence. Mais le même problème se poserait si nous ne traitons que des communes de naissance des délinquants d'une part, des éducateurs de l'autre : comment déduire la commune de naissance d'un délinquant à partir de celle de son éducateur ? Il y a là aussi deux rôles à distinguer, même s'ils sont plus subtils. Nous pouvons les présenter sous la forme condensée suivante :

NumDélinquant \rightarrow NumComNaisDélinquant \Rightarrow NomCommune

NumEduc \rightarrow NumComNaisEduc \Rightarrow NomCommune

⁸ Proposition faite par Flory, Rolland et consorts en... 1986 !

Troisième difficulté : un enseignement technologique basé sur des abstractions

Pour construire leurs connaissances, les élèves, même si nous partons de situations concrètes, ne manipulent ni ne produisent que des objets abstraits. Ils sont dès le départ face à des abstractions, tant dans la représentation des données que dans la façon dont elles sont traitées par un ordinateur.

Il n'est pas vain par exemple de s'interroger sur la façon dont les élèves se représentent les opérateurs relationnels sans maîtriser les concepts mathématiques associés puisqu'en la matière le SGBDR agit comme une « boîte noire » qui retourne un résultat à une question posée.

Il manque en la matière des outils de simulation permettant de suivre le déroulement d'une requête, comme par exemple les outils de mise au point associés aux environnements de développement.

Il peut être alors intéressant de revenir à une représentation plus concrète, plus palpable, qui amène les élèves à construire ou à reconstruire une représentation mentale des objets qu'ils manipulent. Ainsi la réalisation de recherches à partir d'une représentation sur papier du contenu des différentes tables d'une base de données permet de faire apparaître la notion de lien entre tables en amenant l'élève à « parcourir ces liens »⁹ pour passer d'une information à une autre (rechercher le numéro de téléphone d'un client à partir d'une de ses commandes, rechercher les notes d'un élève connaissant son nom, etc.).

Quatrième difficulté : jongler avec deux niveaux d'abstraction oblige à gérer les polysémies

Il est d'autant plus difficile de combattre les représentations déjà construites par les élèves dans le domaine informatique que s'y bousculent des termes fortement polysémiques : des termes propres à la spécialité, des termes ordinaires du langage et des termes construits par les apprentissages précédents. L'enseignant peut par exemple s'interroger sur la signification que peut avoir pour un élève le concept de table, de ligne, de colonne, de relation ou de clé avant d'aborder le modèle relationnel.

A cela s'ajoute la nécessité de travailler sur deux niveaux de représentation : le niveau physique qui est celui du SGBDR et de la base de données et le niveau logique qui est celui du modèle relationnel.

Il importe donc d'être vigilant et précis dans les définitions fournies et de s'appuyer sur des techniques de mémorisation éprouvées, du type de celles qu'on emploie dans l'apprentissage d'une nouvelle langue étrangère (élaboration et exploitation d'un lexique, travail par comparaison : technique du « est » ou « n'est pas », repérage des « faux amis »...).

Cinquième difficulté : s'appuyer sur des non-dits ou des implicites génère le risque d'« erreurs par omission »

Pour l'enseignant, réfléchir en terme d'objectifs amène souvent à fonctionner « à rebours » en cherchant le meilleur cheminement à travers les concepts et les démarches permettant de les atteindre. Le danger consiste alors à aboutir à un cheminement logique qui oublie le droit des élèves à l'erreur ou tout simplement celui à ébaucher des hypothèses : c'est un peu comme si on tirait une flèche et qu'on peignait ensuite la cible autour... Deux situations délicates peuvent alors survenir :

- Les élèves suivent le chemin tracé sans se poser de questions : ils ne perçoivent pas les difficultés, se construisent une image déformée, souvent simpliste et techniciste, des connaissances présentées et ne sauront pas mobiliser ces compétences dans un autre contexte.
- Les élèves refusent de suivre le chemin tracé en se posant les questions que nous avons évacuées et nous pouvons nous trouver en difficulté face à une situation non prévue.

Quelques exemples vont nous permettre d'éclairer notre propos.

Le danger de la tentation de la conception

La conception d'une base de données n'est pas au programme de classe de première. Les notions de dépendance fonctionnelle et de normalisation sont utilisées pour repérer les attributs qui caractérisent un objet identifiable du système d'information et montrer les risques que fait courir l'incohérence d'une table sur l'intégrité d'une base de données (anomalies de mise à jour). Il s'agit de montrer qu'à chaque table ne peut correspondre qu'un concept (comme le résultat d'un élève à un contrôle) ou un « objet » (comme un contrôle ou un élève). L'argumentation s'élabore de façon assez simple : « choisir, par exemple, de placer le nom de l'élève dans la table RESULTAT

⁹ Certains experts parlent de « chemins de jointure »...

plutôt que dans la table ELEVE entraîne des anomalies de mises à jour. Pourquoi ? Parce que la table RESULTAT ainsi conçue représente à la fois un concept (le résultat d'un élève à un contrôle) et un « objet » (l'élève). A ce niveau, la normalisation n'est que la traduction de règles de bon sens...

La tentation d'utiliser normalisation et dépendances fonctionnelles pour concevoir une base de données peut être grande mais ne peut trouver sa place en classe de première pour au moins deux raisons :

- Une raison de nature pédagogique : le programme l'exclut explicitement...
- Une raison de nature didactique : tous les auteurs s'accordent sur le fait que ces outils ne sont ni efficaces ni complets pour permettre la conception d'une base de données. Le recours à la fameuse « relation universelle »¹⁰ qu'il faut décomposer progressivement pour aboutir à un schéma normalisé (et ses avatars que sont les matrices de dépendances fonctionnelles et autres SAT¹¹) est le plus souvent inadapté et ne permet pas de représenter la réalité d'une organisation. Pour ne citer qu'un exemple simpliste : comment représenter à partir de DF le fait que, dans votre lycée, chaque élève peut s'inscrire à différentes options ? Cela nécessite la construction d'une relation « toute clé »¹², sans aucune dépendance fonctionnelle :

INSCRIPTION (CodeElève, CodeOption)

CodeElève, CodeOption : clé primaire

CodeElève : clé étrangère en référence à Code de ELEVE

CodeOption : clé étrangère en référence à Code de OPTION

Quelqu'un pourra rétorquer qu'il suffit de considérer qu'il y a un ordre dans le choix des options de chaque élève et obtenir alors : CodeElève, CodeOption → NumOrdre. Il peut alors en déduire la relation suivante :

INSCRIPTION (CodeElève, CodeOption, NumOrdre)

CodeElève, CodeOption : clé primaire

CodeElève : clé étrangère en référence à Code de ELEVE

CodeOption : clé étrangère en référence à Code de OPTION

Les professeurs auteurs d'exercice sont souvent tentés par ce genre d'adaptation pour éviter les relations « toute clé ». Attention aux « effets de bord » ! Ainsi, dans le cas précédent, l'ajout de l'attribut NumOrdre n'a pas fait naître qu'une DF mais deux : CodeElève, NumOrdre → CodeOption (connaissant un élève et un numéro d'ordre, on connaît l'option choisie).

Certains suggèrent de se poser ce genre de question au cours de l'élaboration de la base de données, en quelque sorte de s'inventer des données non présentes dans le système pour mettre à jour les relations « toute clé ». Mais, pour « inventer » un déterminé, il faut disposer d'un déterminant et si un déterminant pertinent a été trouvé... le problème est résolu !

L'intérêt de ne pas utiliser des noms de champs « sur-signifiants »

La meilleure façon de montrer aux élèves qu'ils se situent au niveau physique (SGBDR) et non au niveau conceptuel consiste à exploiter au mieux la notion de qualification. Dans un SGBDR, le nom d'un champ doit en effet être unique dans une table. Sa signification est donc déductible de son nom et du nom de la table à laquelle il appartient. Ainsi le champ Nom dans la table ELEVE représente naturellement le nom de l'élève, sans qu'il soit nécessaire d'appeler ce champ NomElève. La qualification consiste à faire précéder le nom d'un champ du nom de la table (ELEVE.Nom), ce qui lève toute ambiguïté et apporte une lecture assez naturelle (de droite à gauche) : le nom de l'élève (et encore plus naturelle en anglais : STUDENT's name...).

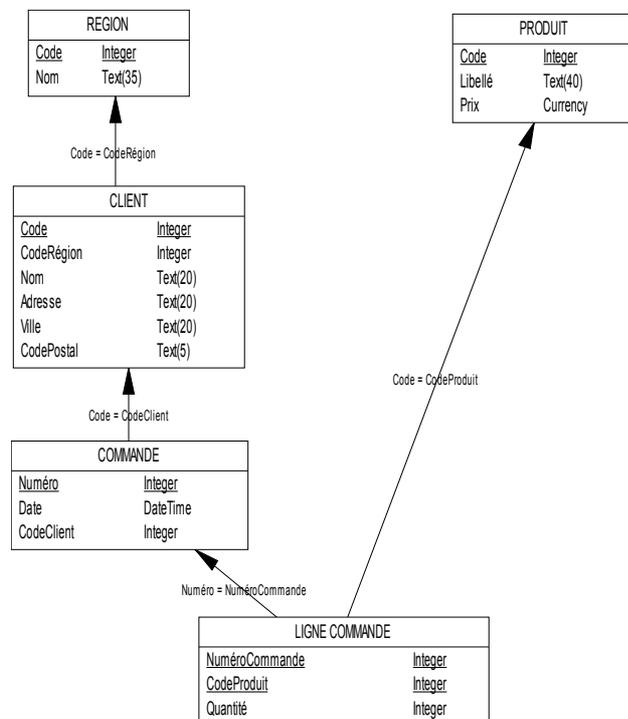
Le respect de cette écriture, outre son aspect logique (pourquoi donner des noms « sur-signifiants » qui aboutissent à une lecture redondante : ELEVE.NomElève se lit ainsi le nom d'élève de l'élève...), permet d'aider les élèves lors de la conception d'une requête en les amenant naturellement à poser la question « de quels champs avons-nous besoin et dans quelles tables se trouvent-ils ? »

Ainsi, dans le schéma relationnel suivant, sont distingués le code d'une région (REGION.Code), celui d'un client (CLIENT.Code) et celui d'un produit (PRODUIT.Code). Le référencement des clés étrangères est adapté en conséquence. Par exemple, entre CLIENT et REGION la référence s'écrit Code = CodeRégion, façon simplifiée d'écrire REGION.Code = CLIENT.CodeRégion.

¹⁰ Relation regroupant tous les attributs représentatifs des données d'un système d'information.

¹¹ Structure d'accès théorique.

¹² Relation ne possédant pas d'autre attribut que sa clé primaire.



Remarquons toutefois que la pratique professionnelle la plus courante, **dans le cadre d'un processus de conception**, est d'intégrer dans la désignation de la donnée un morphème significatif du type d'objet propriétaire de cette donnée (NomClient pour le nom du client...). La méthode Merise l'impose, et la notion de « repository » (dictionnaire) comprend généralement la règle « un nom de donnée est unique dans le dictionnaire ». *En respectant ce principe au départ et en donnant systématiquement des noms de rôles aux clés étrangères au niveau logique, on se trouve également avec une base de données dont les noms d'attributs sont uniques.*

Le danger des activités trop contraintes

Le professeur est souvent tenté de proposer des tâches parcellisées et encadrées afin de mieux diriger les élèves vers le but qu'il s'est fixé pour eux, comme un train sur ses rails. Cette tendance le place dans une posture de chef de projet et les élèves en posture d'exécutants avec le risque de passer du « faire pour apprendre » au « faire pour faire ».

Les sciences humaines sont riches de diversité et de complexité et il serait dommage qu'à être trop contraints les élèves puissent penser que tout est toujours soit blanc soit noir, soit vrai soit faux. Mettons donc un peu de couleurs dans leur vie d'élèves, faisons-leur prendre conscience, par des mises en situation sollicitant réflexion pour prendre une décision, que tout n'est pas si simple.

Un exemple concret concerne la conception de requêtes. Le problème à résoudre réside en effet plus souvent dans la compréhension de la question posée par l'utilisateur et sur la nécessité d'un questionnement complémentaire pour éliminer les ambiguïtés. Ainsi poser la question « Donner le numéro, la date et la note de chaque contrôle passé par l'élève dont le nom est LEFEVRE et le prénom François » (en précisant qu'il n'y a pas d'homonyme...) est certainement adapté pour insister sur la phase de construction de la requête et sur la syntaxe du langage SQL. Mais poser la question « Donnez-moi la moyenne de François LEFEVRE » sera beaucoup plus proche de la réalité et davantage porteur de sens, et mettra l'élève dans la nécessité de comprendre la question avant d'y répondre.

Le danger des cas particuliers ou l'arbre qui cache la forêt

L'apprentissage en classe de première réside sur l'exploitation de bases de données simples (mais pas forcément simplistes). Cela ne veut pas dire que les requêtes à construire doivent limiter les interventions des élèves à une simple application d'un mode opératoire. Il faut au contraire qu'elles soient conçues pour provoquer la réflexion et interroger la démarche employée.

Ainsi, la réalisation systématique de requêtes dans lesquelles les noms des champs associés aux conditions apparaissent toujours dans la clause SELECT ne permet pas aux élèves de distinguer le double questionnement nécessaire pour élaborer une requête (« Quels sont les champs à afficher ? Quels sont les champs nécessaires à l'évaluation d'une condition de restriction ? »).

La question « Quelles notes a obtenues François LEFEVRE durant le mois de mars ? » illustre ce cas : l'évaluation de la condition nécessite de connaître la date des contrôles dans la table CONTRÔLE alors que les notes se trouvent dans la table RESULTAT.

```
SELECT NoteContrôle
FROM RESULTAT, CONTROLE, ELEVE
WHERE ELEVE.CodeElève = RESULTAT.CodeElève
AND RESULTAT.CodeContrôle = CONTROLE.CodeContrôle
AND NomElève = "LEFEVRE"
AND PrénomElève = "François"
AND MONTH(DateContrôle) = 3
```

Nous avons déjà parlé de la nécessité de donner des noms significatifs aux clés étrangères qui ne soient donc pas identiques à celui du champ clé primaire référencé. Cela permet de toujours revenir au sens porté par le schéma en évitant une approche mécanique (« quel champ a le même nom ? »), et cela évite les confusions et les amalgames (comme la question « quels sont les champs communs aux deux tables ? »...). L'emploi d'un schéma dans lequel il existe deux « chemins » possibles entre deux tables permet de montrer le risque d'une telle approche. Imaginons ainsi une organisation dans laquelle des produits sont toujours livrés par les mêmes fournisseurs (ils sont référencés) et que parmi ceux-ci l'un soit considéré comme le fournisseur principal.

Une écriture ne respectant pas les deux conseils précédemment cités (pas de noms de champs « sur-signifiants » et noms des clés étrangères rappelant leur rôle) donnera le résultat suivant :

```
PRODUIT(CodeProduit, LibelléProduit, CodeFournisseur, ...)
    CodeProduit : clé primaire
    CodeFournisseur : clé étrangère en référence à CodeFournisseur de FOURNISSEUR
FOURNISSEUR(CodeFournisseur, NomFournisseur, ...)
    CodeFournisseur : clé primaire
REFERENCEMENT(CodeProduit, CodeFournisseur, ...)
    CodeProduit, CodeFournisseur : clé primaire
    CodeProduit : clé étrangère en référence à CodeProduit de PRODUIT
    CodeFournisseur : clé étrangère en référence à CodeFournisseur de FOURNISSEUR
```

Si nous posons les questions « Qui est le fournisseur principal du produit 12 ? » et « Qui fournit le produit 12 ? », il y a fort à parier que ces questions mettront en difficulté les élèves adeptes d'une démarche mécanique en les plaçant devant un choix difficile : quel champ CodeFournisseur faut-il utiliser ?

Le fournisseur principal du produit 12 :

```
SELECT NomFournisseur
FROM PRODUIT, FOURNISSEUR
WHERE PRODUIT.CodeFournisseur = FOURNISSEUR.CodeFournisseur
AND CodeProduit = 12
```

Les fournisseurs du produit 12 :

```
SELECT NomFournisseur
FROM REFERENCEMENT, FOURNISSEUR
WHERE REFERENCEMENT.CodeFournisseur = FOURNISSEUR.CodeFournisseur
AND CodeProduit = 12
```

L'application de nos conseils donne l'écriture suivante :

```
PRODUIT(Code, Libellé, FournisseurPrincipal, ...)
  Code : clé primaire
  FournisseurPrincipal : clé étrangère en référence à Code de FOURNISSEUR
FOURNISSEUR(Code, Nom, ...)
  Code : clé primaire
REFERENCEMENT(CodeProduit, CodeFournisseur, ...)
  CodeProduit, CodeFournisseur : clé primaire
  CodeProduit : clé étrangère en référence à Code de PRODUIT
  CodeFournisseur : clé étrangère en référence à Code de FOURNISSEUR
```

La démarche de conception des requêtes en est facilitée et conduit à obtenir les réponses suivantes :

Le fournisseur principal du produit 12 :

```
SELECT Nom
FROM PRODUIT, FOURNISSEUR
WHERE FournisseurPrincipal = FOURNISSEUR.Code
AND PRODUIT.Code = 12 // notez la qualification obligatoire pour les attributs nommés Code
```

Les fournisseurs du produit 12 :

```
SELECT Nom
FROM REFERENCEMENT, FOURNISSEUR
WHERE CodeFournisseur = Code
AND CodeProduit = 12 // ici la qualification n'est pas nécessaire
```

Mais employer la qualification uniquement lorsqu'elle est nécessaire n'apparaît pas comme un objectif à atteindre prioritairement en classe de première. La majorité des élèves (à ce niveau comme à des niveaux supérieurs) juge souvent utile de rappeler l'origine des champs utilisés dans la requête : qui les en blâmerait ?

Remarquons que le choix est fait de limiter l'apprentissage en classe de première à l'équijointure basée sur les liens entre les tables (représentés par les clés étrangères), même si cela entraîne une sorte de « mensonge par omission ».

Il est important de se souvenir qu'il est possible de réaliser des équijointures entre d'autres champs, à condition qu'ils appartiennent au même domaine. Ainsi :

La question « Trouvez-moi les élèves qui ont obtenu la même note que François LEFEVRE au contrôle 15 » nécessite une autojointure¹³ sur le champ NoteElève.

```
SELECT NomElève, PrénomElève
FROM ELEVE ELV1, RESULTAT RES1, RESULTAT RES2, ELEVE ELV2
WHERE ELV2.NomElève = "LEFEVRE"
AND ELV2.PrénomElève = "François"
AND RES2.CodeElève = ELV2.CodeElève
AND RES2.CodeContrôle = 15
AND RES1.NoteElève = RES2.NoteElève
AND ELV1.CodeElève = RES1.CodeElève
AND ELV1.NomElève <> "LEFEVRE"
AND ELV1.PrénomElève <> "François"
```

ELV1, ELV2, RES1 et RES2 sont des alias qui permettent de faire référence à une même table plusieurs fois dans une requête.

¹³ Autojointure : jointure entre une table et elle-même.

Il est également possible de réaliser des inéquijointures, mettant en œuvre des opérateurs d'inégalité :

« Quels sont les contrôles qu'ont subis les élèves de la 1STG3 avant le contrôle 75 ? » nécessite une inéquijointure sur le champ DateContrôle.

```
SELECT CodeContrôle, DateContrôle
FROM CONTROLE CTR1, CONTROLE CTR2, ELEVE
WHERE CodeDivision = "1STG3"
AND ELEVE.CodeElève = CTR1.CodeElève
AND CTR1.DateContrôle < CTR2.DateContrôle
AND CTR2.CodeContrôle. = 75
```

Ces notions sont clairement à exclure des apprentissages conduits en 1^e STG.

Sixième difficulté : un enseignement technologique n'est ni un enseignement pratique, ni un enseignement professionnel

Le souci d'enseigner par la technologie peut très vite se transformer en la facilité d'enseigner la technologie, en mettant le SGBDR au centre de la formation. Il est donc important de bien considérer que le SGBDR est avant tout un outil de notre enseignement et non son objet. Au risque d'être caricatural, nous dirons que l'objectif est d'aborder les fonctionnalités de l'outil strictement nécessaires à l'apprentissage des concepts et d'évaluer la compréhension de ces concepts et non la maîtrise des fonctionnalités...

Si notre compétence professionnelle d'enseignant doit intégrer une maîtrise suffisante des technologies de l'information et de la communication, cette même maîtrise ne doit pas être exigée des élèves. Il est même essentiel que notre discours insiste sur le nécessaire recul que l'être humain doit rechercher quant à l'emploi de la technique et alimente une réflexion sur les dangers d'en faire un usage non réfléchi.

7. En guise de conclusion

En matière de pédagogie, il n'y a pas de stratégie miracle : une bonne séance de formation n'est pas une séance programmée et gérée à la minute près. C'est une séance dont la préparation rigoureuse permet de donner toute sa place à l'improvisation, au sens de l'adaptation à la réaction des élèves, à travers le choix et la modulation des situations et des techniques d'apprentissage.

Savoir improviser, c'est pratiquer une pédagogie différenciée, non pas dans le sens d'un enseignement différent pour chacun, mais dans celui d'une prise en compte et d'une adaptation à la diversité des profils d'apprentissage dans la préparation de son enseignement.

Savoir s'adapter, c'est pouvoir puiser dans la diversité des méthodes et des pratiques pédagogiques afin d'appliquer les plus efficaces aux caractéristiques dominantes du groupe le plus souvent, à un élève en difficulté quand c'est nécessaire.

Enseigner, c'est finalement piloter un groupe vers de nouvelles connaissances en « utilisant la bonne méthode au bon moment ».

Mais nous ne pouvons terminer sans évoquer les limites des méthodes actives, tiraillées entre « forcing » et laisser-faire.

Le professeur doit en permanence jongler entre les objectifs d'apprentissage (mettre les élèves aux prises avec des problèmes concrets qu'ils peuvent résoudre à leur rythme, avec la possibilité de refaire si nécessaire, de prendre leur temps) et les objectifs institutionnels (respect du programme, contrainte de temps, présence d'une évaluation sommative et nécessité d'y préparer les élèves).

La tentation est alors grande de privilégier un gain de production au détriment d'un gain en apprentissage en étant plus directif, en se calant sur les élèves les plus compétents ou en faisant à leur place.

Une réflexion collective semble donc plus que nécessaire sur notre culture de l'évaluation et sur nos pratiques didactiques et pédagogiques afin de passer d'une culture de la pédagogie sous contrainte de l'évaluation à celle d'une évaluation pilotée par les objectifs d'apprentissage.

Et pourquoi alors ne pas se prendre à rêver à la mise en place dans les disciplines de l'économie et de la gestion de ponts plus praticables entre enseignants et chercheurs pour travailler sur ces sujets, à l'image de nos collègues mathématiciens avec les IREM ?

ANNEXE 1

Pour en savoir plus sur apprentissage, didactique et pédagogie

- ARSAC J. : La didactique de l'informatique : un problème ouvert, dans *Colloque francophone sur la didactique de l'informatique*, E.P.I. Paris (1989).
- ASTOLFI J.-P. : L'erreur, un outil pour enseigner, Paris, ESF (1997)
- BARTH B.-M. : L'apprentissage de l'abstraction, Retz (1987)
- BRUNEL H., MAILLET P : Le cerveau et l'acquisition des connaissances (2003)
- CHEVALLARD Y : La transposition didactique : du savoir savant au savoir enseigné; La Pensée Sauvage, Grenoble (1985)
- DREVILLON J. : Pratiques éducatives et développement de la pensée opératoire (1980)
- GIORDAN A. : Les origines du savoir, des conceptions des apprenants aux concepts scientifiques, Delachaux & Niestlé (1987)
- GIORDAN A., Un environnement pédagogique pour apprendre : le modèle allostérique (1992)
- GIORDAN A., DE VECCHI G., : Les origines du savoir, Delachaux (1997)
- LAMONTAGNE C. : Vers une pratique du profil d'apprentissage, collection Le profil d'apprentissage, Saint-Hubert (Québec) : Institut de Recherche sur le Profil d'apprentissage (1985)
- LAMONTAGNE C. : Le profil d'apprentissage, vue d'ensemble, collection Outils, techniques et méthodes, Saint-Hubert (Québec) : Institut de recherche sur le profil d'apprentissage, (juin 1985)
- LAMY D. : Développement d'un scénario d'utilisation en contexte d'enseignement d'un logiciel idéateur, Université Québec (1999)
- LEBRUN M. : Des technologies pour enseigner et apprendre : Perspectives en éducation et formation, De Boeck (2002)
- LIEURY A. : Mémoire et réussite scolaire, Dunod
- MERRIEU P. : La pédagogie entre le dire et le faire, ESF Éditions (1995)
- MERRIEU P. : Faire l'école, faire la classe, ESF Éditions (2004)
- PAIR C. : Je ne sais (toujours) pas enseigner la programmation ; *informatiques 2* (1988)
- PERRENOUD P. : Construire des compétences dès l'école Paris, ESF, 3^e éd. (2000)
- PERRENOUD P. : Enseigner : agir dans l'urgence, décider dans l'incertitude. Savoirs et compétences dans un métier complexe, Paris, ESF, 2^e éd. (1999)
- PERRENOUD P. : Dix nouvelles compétences pour enseigner : Invitation au voyage, Paris, ESF (1999)
- PERRENOUD P. : Pédagogie différenciée : des intentions à l'action, Paris, ESF, 2^e éd. (2000)
- REUCHLI et BACHER : Les différences individuelles dans le développement cognitif de l'enfant, PUF (1989)
- VAYER P. : Le principe d'autonomie et l'éducation, Paris : ESF (1993)
- VEZIN J.-F. : Apport informationnel des schémas dans l'apprentissage. Le travail humain, tome 47, n° 1 p. 61-74 (1984)

ANNEXE 2

Pour en savoir plus sur le modèle relationnel et les SGBDR

- Bases de données - 5e tirage 2003 (Georges Gardarin) - Eyrolles
- Bases de données : Architectures, modèles relationnels et objets, SQL 3 (Serge Miranda) - Dunod
- Introduction aux bases de données relationnelles (Ramon A. Mata-Toledo, Pauline K. Cushman) - Ediscience
- Introduction pratique aux bases de données relationnelles (Andreas Meier) - Springer
- Les Bases de Données Relationnelles (André Flory, Frédérique Laforest) - Economica
- Programmation SQL (Ramon A. Mata-Toledo, Pauline K. Cushman) - Dunod
- SQL pour les nuls (Allen G. Taylor) - First Interactive
- SQL L'intro (Judith S. Bowman, Sandra L. Emerson, Marcy Darnovsky) - Campus Press
- SQL et Algèbre relationnelle : Notions de base (Jérôme Gabillaud) –ENI Collection Technote
- Le Modèle relationnel : Algèbre, langages, applications (Bouzeghoub)- Hermes Sciences Publication
- SQL (Ronald R. Plew, Ryan K. Stephens) – Campus Press
- Introduction aux requêtes SQL (Michael J. Hernandez, John L. Viescas) - Eyrolles

ANNEXE 3

Connaître le domaine de savoir : quelques éléments de contenu

Remarque : Ce qui suit ne représente en aucun cas une progression applicable en classe de première STG. Il s'agit d'un condensé de la théorie relative au modèle relationnel. Le lecteur qui éprouverait des difficultés à appréhender ce discours quelque peu formel est invité à se reporter aux ouvrages cités en annexe 2.

Le modèle relationnel a été proposé par E.F. Codd en 1970. Sa simplicité repose sur une vision tabulaire des données et sa robustesse résulte de ses bases mathématiques (la théorie des ensembles et la logique des prédicats notamment). Ses principaux objectifs opérationnels sont de permettre un haut degré d'indépendance entre les applications (programmes, interfaces) et la représentation interne des données (fichiers, index, chemins d'accès) ainsi que de faciliter le traitement des problèmes de cohérence et de redondance des données. Le modèle est fondé sur la notion de **relation**.

L'extension d'une **relation** peut être vue comme un ensemble de lignes appelées **n-uplets**. Chaque n-uplet est une liste ordonnée de n valeurs prises dans des ensembles de valeurs appelés **domaines**. Chacun de ces domaines joue donc au moins un rôle dans la relation et chacun de ces rôles est représenté par un **attribut**.

La définition (en intention) d'une relation recense les correspondances entre attributs et domaines et se présente sous la forme d'un **schéma de relation** R, noté $R(A_1:D_1, A_2:D_2, \dots, A_n:D_p)$ où A_i est le nom d'un attribut et D_j le nom d'un domaine. Un même domaine peut jouer plusieurs rôles dans une même relation. Une **relation** est alors définie formellement comme un sous-ensemble du produit cartésien des domaines cités. Pour représenter les données d'un système d'information, les relations, les domaines et les attributs acquièrent une dimension sémantique. Un ensemble de schémas de relations, liées entre elles pour représenter une réalité donnée, est souvent appelé un **schéma relationnel**¹⁴.

Exemple : Le schéma d'une relation dénommée RESULTAT.

RESULTAT (CodeElève:Entier, NomElève:Chaîne[20], PrénomElève:Chaîne[20], DateNaissance:Date, CodeINSEE:Chaîne[13], CodeDivision:Entier, NomDivision:Chaîne[20], CodeNiveau:TypeNiveau, LibelléNiveau:Chaîne[20], NumContrôle:Entier, DateContrôle:Date, NoteContrôle:Réel)

Cette relation exprime une réalité (simplifiée) du système d'information pédagogique d'un établissement scolaire (gestion des notes) : c'est l'ensemble des résultats à des contrôles passés par des élèves qui appartiennent à des divisions, chaque division étant associée à un niveau. Chaque n-uplet représente donc un fait mémorisé dans le système d'information (ici un résultat à un contrôle). Un contrôle est identifié par un numéro, se déroule à une date donnée :

(234, 'DUMONT', 'Pierre', '12/03/1986', '18603350175', 13, '1STG3', 'B', 'PREMIERE', 256, '07/01/2005', 12.5) exprime le fait que l'élève 234, se nommant DUMONT, se prénommant Pierre, né le 12 mars 1986, ayant pour numéro d'INSEE le n° 18603350175, appartenant à la classe 13 dénommée 1STG3 et associée au niveau PREMIERE, a subi le contrôle 256 le 7 janvier 2004 et obtenu la note de 12,5.

Voici d'autres faits enregistrés :

(234, 'DUMONT', 'Pierre', '12/03/1986', '18603350175', 13, '1STG3', 'B', 'PREMIERE', 254, '25/12/2004', 15.5)

(239, 'DUVAL', 'Pascal', '17/05/1988', '188805150234', 13, '1STG3', 'B', 'PREMIERE', 254, '25/12/2004', 17.5)

(278, 'DELPIERRE', 'Jean', '26/12/1987', '18712145354', 13, '1STG3', 'B', 'PREMIERE', 256, '07/01/2005', 9)

Remarque : L'utilisation des « apostrophes » (ou des guillemets anglais, c'est-à-dire le caractère ") pour encadrer les caractères, les chaînes de caractères et les dates permet de les distinguer des valeurs numériques. Il existe bien d'autres conventions...

Chaque **attribut** A_i est le nom d'un rôle joué par son domaine D_j dans le schéma de relation R.

Exemples : L'attribut DateContrôle fait jouer au domaine Date un rôle (la date d'un contrôle passé par un élève) et l'attribut DateNaissance fait jouer au domaine Date un autre rôle (la date de naissance d'un élève).

Un **domaine** D est un ensemble de valeurs caractérisé par un nom. Un domaine peut être défini en extension en donnant la liste des valeurs qui le composent ou en compréhension (en intention) en définissant une propriété caractéristique. Dans

¹⁴ Il y a là une petite difficulté de vocabulaire à prendre en compte lorsqu'il faut s'adresser aux élèves : un schéma de relation est le schéma d'une relation, un schéma relationnel est un ensemble de schémas de relation. Pour éviter les confusions, beaucoup de professeurs n'utilisent pas la première expression et parlent de « relation » tout court. Exemple : Voici le schéma relationnel de la gestion des notes composé des relations suivantes : ... », ce qui suit étant des « schémas de relation ».

la théorie de Codd, le domaine possède une dimension sémantique : ainsi il ne serait pas convenable de relier au même domaine le nombre entier mesurant un stock d'articles et le nombre entier mesurant l'ancienneté en années d'un employé. Mais, du point de vue de la réalisation informatique, le domaine est souvent confondu avec la notion de **type** de données (dans l'exemple précédent, le type entier). Dans la suite de ce document, nous nous plierons à cette facilité.

Exemples :

Le domaine Date exprime toutes les dates valides : '13/03/1987' appartient à ce domaine, '17/15/2003' n'appartient pas à ce domaine (date invalide).

Le domaine appelé ici TypeNiveau est composé des caractères 'A', 'B', 'C', 'D' et 'E'.

Remarque : Par souci de simplification, le schéma d'une relation est présenté sous une forme réduite à la liste des attributs qui la constituent, sans faire référence aux domaines associés¹⁵. C'est cette forme dite « simplifiée » que nous utilisons par la suite.

Une **clé** de relation est un ensemble minimal d'attributs permettant d'identifier chacun des n-uplets de la relation, aujourd'hui et demain. « Minimal » signifie que, si l'on y supprime un attribut de la clé, le résultat n'est plus une clé.

- *L'ensemble des attributs (CodeElève, NumContrôle) est une clé de la relation RESULTAT que nous venons de décrire : si le code d'un élève et le numéro d'un contrôle qu'il a passé sont connus, sont aussi connues toutes les caractéristiques de cet élève (et notamment celles concernant sa division d'appartenance) et toutes les caractéristiques du contrôle qu'il a passé, ainsi que le résultat qu'il a obtenu (la note du contrôle). Mais un seul de ces attributs ne suffit pas : pour un élève dont le code est connu, il peut y avoir plusieurs notes (il a pu passer plusieurs contrôles), pour un contrôle également (il est probable que plusieurs élèves ont passé ce contrôle). L'ensemble (CodeINSEE, NumContrôle) est une autre clé.*
- *L'ensemble des attributs (CodeElève, CodeNiveau) n'est pas une clé : la connaissance d'une valeur pour chacun de ces attributs ne permet pas d'obtenir une valeur unique de chacun des autres attributs. Un même élève peut avoir plusieurs notes.*

La **clé primaire** d'une relation est une clé qui a été choisie parmi les clés possibles dites « **clés candidates** ». A son tour, le domaine sur lequel est définie une clé est dit lui aussi primaire.

(CodeElève, NumContrôle) a été choisie comme clé primaire, (CodeINSEE, NumContrôle) est une autre clé candidate.

*RESULTAT (CodeElève, NomElève, PrénomElève, DateNaissance, CodeINSEE, CodeDivision, NomDivision, CodeNiveau, LibelléNiveau, NumContrôle, DateContrôle, NoteContrôle)
CodeElève, NumContrôle : clé primaire*

Il existe de nombreuses façons conventionnelles de montrer la clé primaire d'une relation. La façon la plus claire de la montrer est de la désigner de façon explicite ; la plus économique consiste à la placer en tête de liste et à la souligner.

Désignation explicite :

*RESULTAT (CodeElève, NomElève, PrénomElève, DateNaissance, CodeINSEE, CodeDivision, NomDivision, CodeNiveau, LibelléNiveau, NumContrôle, DateContrôle, NoteContrôle),
Clé primaire : CodeElève, NumContrôle*

Convention d'écriture :

RESULTAT (CodeElève, NumContrôle, NomElève, PrénomElève, DateNaissance, CodeINSEE, CodeDivision, NomDivision, CodeNiveau, LibelléNiveau, DateContrôle, NoteContrôle),

¹⁵ Il faudra bien sûr que ces références soient fournies par ailleurs.

Une **clé étrangère** dans une relation est un attribut qui fait référence à la clé primaire d'une relation¹⁶. Une clé étrangère est donc définie sur le même domaine primaire que la clé primaire qu'elle référence. Par convention (et à la suite d'une erreur de traduction...), la clé étrangère est souvent désignée en faisant suivre (ou précéder) son nom du symbole #. Il semble souhaitable d'éviter cette représentation qui crée des confusions dans l'esprit de nombreux élèves et de préciser ce rôle à l'aide d'une assertion nommant la relation cible.

Exemple : Nous distinguons cette fois deux relations liées entre elles, RESULTAT et DIVISION.

*RESULTAT (CodeElève, NumContrôle, NomElève, PrénomElève, DateNaissance, CodeINSEE, CodeDivision, CodeNiveau, LibelléNiveau, DateContrôle, NoteContrôle)
CodeDivision, clé étrangère faisant référence à CodeDivision de la relation DIVISION*

DIVISION (CodeDivision, NomDivision)

Il n'existe pas de valeur prise par CodeDivision dans RESULTAT qui ne se trouve pas dans la relation DIVISION.

Un **schéma relationnel**¹⁷ est un ensemble de relations liées sémantiquement, notamment par la présence de clés étrangères dans ces relations.

L'exemple précédent montre un schéma relationnel composé de deux relations : RESULTAT et DIVISION. La clé étrangère CodeDivision, attribut de la relation RESULTAT, opère un lien entre les deux relations pour exprimer qu'un élève appartient à une division.

Une **contrainte d'intégrité** est une propriété que tous les n-uplets composants les relations doivent respecter. Certaines de ces contraintes sont liées aux règles de gestion du système d'information représenté : par exemple, le stock actuel d'un article ne doit pas être inférieur à son stock de sécurité. D'autres contraintes sont liées au modèle relationnel lui-même. Il s'agit de :

- **La contrainte de relation.** Il n'y a pas de doublon dans une relation, donc pas de doublon dans la clé primaire (unicité de clé) ; la clé primaire est toujours renseignée (contrainte d'entité) ; la clé primaire ne peut pas changer de valeur (contrainte de stabilité).
- **La contrainte de domaine.** Dans un n-uplet donné, un attribut a une valeur qui correspond au domaine de définition de l'attribut (le cas de la « valeur nulle » est abordé ci-après).
- **La contrainte référentielle** (ou de clé étrangère). Dans un n-uplet donné, la valeur d'une clé étrangère doit apparaître comme valeur de clé primaire dans la relation cible.

Si, dans un n-uplet donné, l'un des attributs n'est pas renseigné, il est usuel de dire qu'il possède une « **valeur nulle** ». Cette expression vient du terme anglo-saxon correspondant (*null*). La traduction utilisée n'est pas sans défaut. En effet, il ne faut pas confondre la « valeur nulle » qui signifie absence de valeur avec la valeur 0 (zéro) d'un attribut de type numérique¹⁸.

Ainsi, il est possible d'admettre que la relation RESULTAT enregistre tous les contrôles passés, même ceux dont les notes ne sont pas connues. Dans ce cas, il n'est pas improbable que, pour certains n-uplets de cette relation, la note ne soit pas renseignée, qu'elle possède une « valeur nulle ». Il ne faut pas confondre cette situation avec celle d'un élève ayant passé un contrôle et dont la note, connue, est égale à 0 !

Rappelons que la clé primaire d'une relation n'admet pas de valeur nulle : elle doit toujours être renseignée. Il arrive qu'une clé étrangère admette des « valeurs nulles ».

Reprenons le schéma relationnel composé des deux relations RESULTAT et DIVISION. Si la règle de gestion est qu'un élève est forcément rattaché à une division, la clé étrangère CodeDivision, attribut de RESULTAT, n'admettra pas de « valeur nulle » : cette caractéristique fait partie de la définition de l'attribut.

¹⁶ Rien n'interdit qu'une clé étrangère fasse référence à la clé primaire de la relation à laquelle elle-même appartient.

¹⁷ Il y a là une petite difficulté de vocabulaire à prendre en compte lorsqu'il faut s'adresser aux élèves : un schéma de relation est le schéma d'une relation, un schéma relationnel est un ensemble de schémas de relation.

¹⁸ La valeur « chaîne vide » pour un attribut de type chaîne de caractères n'est pas toujours assimilable non plus à la notion de « valeur nulle ».

Il est fréquent d'admettre les « valeurs nulles » pour les autres attributs. Dans le cas où cela est admis, qu'est-ce que cela peut signifier ? Dans la grande majorité des cas, la « valeur nulle » signifie :

- soit que l'attribut n'est pas significatif pour le n-uplet concerné¹⁹ (exemple : le nom d'épouse pour un homme),
- soit que l'attribut est significatif mais que sa valeur pour le n-uplet concerné n'est pas encore connue²⁰ (exemple : la note non encore attribuée à un contrôle passé).

En théorie, les domaines sur lesquels sont définis les attributs intègrent dans leur description les « valeurs nulles » qui leur sont spécifiques, une valeur particulière pour chacune des significations possibles. En pratique, seule l'absence de valeur est considérée.

Une **dépendance fonctionnelle** (DF) entre les ensembles d'attributs X et Y exprime le fait que X détermine Y ou que Y dépend fonctionnellement de X. Elle s'écrit $X \rightarrow Y$. Cela implique qu'à une valeur de X ne peut correspondre qu'au plus une valeur de Y, autrement dit qu'il ne peut pas y avoir deux couples (x_1, y_1) et (x_1, y_2) avec $y_1 \neq y_2$ ²¹.

Exemple : Dans la relation RESULTAT (CodeElève, NumContrôle, NomElève, PrénomElève, DateNaissance, CodeINSEE, CodeDivision, NomDivision, CodeNiveau, LibelléNiveau, DateContrôle, NoteContrôle), le rôle des attributs permet de recenser les DF suivantes :

CodeElève \rightarrow NomElève, PrénomElève, DateNaissance, CodeINSEE, CodeDivision, NomDivision, CodeNiveau, LibelléNiveau

Qu'on peut lire : la connaissance d'un code d'élève donne la connaissance de son nom (unique), de son prénom (unique), ...

CodeINSEE \rightarrow NomElève, PrénomElève, DateNaissance, CodeElève, CodeDivision, NomDivision, CodeNiveau, LibelléNiveau

CodeDivision \rightarrow NomDivision, CodeNiveau, LibelléNiveau

CodeNiveau \rightarrow LibelléNiveau

NumContrôle \rightarrow DateContrôle

CodeElève, NumContrôle \rightarrow NoteContrôle

CodeINSEE, NumContrôle \rightarrow NoteContrôle

Lorsque X ne comporte qu'un attribut, la DF est dite simple (exemple : CodeElève \rightarrow NomElève), lorsque X en comporte plusieurs, la DF est dite composée (exemple : CodeElève, NumContrôle \rightarrow NoteContrôle).

Certains auteurs font la distinction entre DF forte (à une valeur de la source correspond une et une seule valeur de la cible) et dépendance fonctionnelle faible (à une valeur de la source correspond au plus une valeur de la cible). Ce dernier cas correspond au fait que la cible admet des « valeurs nulles ». Dans la démonstration qui nous occupe, il est *impératif de se limiter à la première définition qui est la seule permettant d'appliquer sans souci les règles ci-dessous.*

Les dépendances fonctionnelles obéissent à un certain nombre de règles parmi lesquelles :

Soient X, Y, Z et W des ensembles d'attributs :

Réflexivité : Si $Y \subset X$, alors $X \rightarrow Y$

Exemple : CodeElève, DateContrôle \rightarrow DateContrôle (il s'agit d'une trivialité)

Augmentation : Si $X \rightarrow Y$, alors $ZX \rightarrow ZY$ pour un attribut Z quelconque

Exemple : Si NumContrôle \rightarrow DateContrôle alors NumContrôle, NomElève \rightarrow DateContrôle, NomElève (il s'agit d'une autre trivialité).

¹⁹ Dans les tableaux usuellement conçus, le symbole « - » est souvent utilisé dans ce sens.

²⁰ Dans les tableaux usuellement conçus, le symbole « ? » est souvent utilisé dans ce sens.

²¹ $x_i \in X$ et $y_i \in Y$

Union : $\text{Si } X \rightarrow Y \text{ et } X \rightarrow Z, \text{ alors } X \rightarrow YZ$

Exemple : Si CodeElève \rightarrow NomElève et CodeElève \rightarrow PrénomElève alors CodeElève \rightarrow NomElève, PrénomElève (une nouvelle trivialité).

Décomposition : $\text{Si } X \rightarrow YZ, \text{ alors } X \rightarrow Y \text{ et } X \rightarrow Z$

Exemple : Si CodeElève \rightarrow NomElève, PrénomElève alors CodeElève \rightarrow NomElève et CodeElève \rightarrow PrénomElève (on enfonce des portes ouvertes...).

Transitivité : $\text{Si } X \rightarrow Y \text{ et } Y \rightarrow Z, \text{ alors } X \rightarrow Z$

Exemple : Si CodeDivision \rightarrow CodeNiveau et CodeNiveau \rightarrow LibelléNiveau alors CodeDivision \rightarrow LibelléNiveau (la connaissance du code d'une division permet d'en déduire son code de niveau – unique pour une division – et la connaissance d'un code de niveau permet d'en déduire son libellé. Donc connaître un code de division permet de connaître le libellé de niveau – unique - qui lui correspond).

Pseudo-transitivité : $\text{Si } X \rightarrow Y \text{ et } WY \rightarrow Z, \text{ alors } WX \rightarrow Z$

Exemple : Si CodeElève \rightarrow CodeInsee et NumContrôle, CodeInsee \rightarrow NoteContrôle alors NumContrôle, CodeElève \rightarrow NoteContrôle (un peu moins trivial : le code de l'élève peut être substitué à son code INSEE dans toute DF qui l'utilise en partie source).

Ces règles permettent de définir la **dépendance fonctionnelle élémentaire (DFE)** : c'est une dépendance fonctionnelle $X \rightarrow Y$ telle que Y ne comporte qu'un seul attribut et que pour tout $X' \subset X, X' \rightarrow Y$ n'est pas vérifiée. Il s'agit donc d'une **dépendance minimale, non triviale**.

Exemple : Dans la relation RESULTAT (CodeElève, NumContrôle, NomElève, PrénomElève, DateNaissance, CodeINSEE, CodeDivision, NomDivision, CodeNiveau, LibelléNiveau, DateContrôle, NoteContrôle), il est possible de recenser les DFE suivantes :

<i>CodeElève \rightarrow NomElève</i>	<i>(CodeINSEE \rightarrow NomElève)</i>
<i>CodeElève \rightarrow PrénomElève</i>	<i>(CodeINSEE \rightarrow PrénomElève)</i>
<i>CodeElève \rightarrow DateNaissance</i>	<i>(CodeINSEE \rightarrow DateNaissance)</i>
<i>CodeElève \rightarrow CodeINSEE</i>	<i>(CodeINSEE \rightarrow CodeElève)</i>
<i>CodeElève \rightarrow CodeDivision</i>	<i>(CodeINSEE \rightarrow CodeDivision)</i>
<i>CodeElève \rightarrow NomDivision</i>	<i>(CodeINSEE \rightarrow NomDivision)</i>
<i>CodeElève \rightarrow CodeNiveau</i>	<i>(CodeINSEE \rightarrow CodeNiveau)</i>
<i>CodeElève \rightarrow LibelléNiveau</i>	<i>(CodeINSEE \rightarrow LibelléNiveau)</i>

CodeDivision \rightarrow NomDivision
CodeDivision \rightarrow CodeNiveau
CodeDivision \rightarrow LibelléNiveau

CodeNiveau \rightarrow LibelléNiveau

NumContrôle \rightarrow DateContrôle

CodeElève, NumContrôle \rightarrow NoteContrôle
CodeINSEE, NumContrôle \rightarrow NoteContrôle

Remarques :

- Cette énumération n'est ni le fruit du hasard, ni celui de la nécessité... Il est le résultat du sens porté par les données et des règles de gestion qui ont été choisies. En l'occurrence, certains trouveront certainement inadmissible de ne pas avoir la DF NumContrôle \rightarrow CodeDivision en argumentant qu'un contrôle est toujours préparé pour une division particulière. Dans notre premier exemple, nous avons fait le choix d'une autre « réalité » : en exprimant que ce lien n'existe pas, nous exprimons que nous avons affaire à un enseignant innovateur qui prépare des contrôles « à la carte », auxquels pourront participer des élèves de divisions, voire de niveaux de division différents.
- La relation RESULTAT possède en fait deux clés candidates puisque, dans toutes les DF dont CodeElève fait partie de la source, cet attribut peut être remplacé par CodeINSEE. Par souci de simplification, nous supposons que, dans ce cas, l'une des clés candidates est choisie comme clé primaire et les DF qui ont pour source les autres clés candidates sont éliminées. C'est ce que nous appliquons par la suite où CodeINSEE n'apparaît plus comme source de DF.

A l'inverse, $X \rightarrow Y$ est une dépendance **fonctionnelle partielle** lorsque Y ne comporte qu'un seul attribut mais qu'il existe un sous-ensemble X' de X suffisant pour déterminer Y (soit $X' \rightarrow Y$).

Exemple : CodeElève, NumContrôle, CodeINSEE \rightarrow NoteContrôle est une dépendance fonctionnelle partielle.

Une **dépendance fonctionnelle directe (DFD)** $X \rightarrow Y$ est une DF qui ne peut pas être déduite par transitivité : il n'existe pas d'attribut Z tel que $X \rightarrow Z$ et $Z \rightarrow Y$.

Exemple : CodeElève \rightarrow LibelléNiveau n'est pas une DFD car il existe CodeElève \rightarrow CodeNiveau et CodeNiveau \rightarrow LibelléNiveau.

Quel est l'intérêt d'étudier les dépendances fonctionnelles entre les données ? Pour éviter les anomalies de mise à jour ! Une **anomalie de mise à jour** représente une difficulté dans la gestion d'une base de données. Elle provient généralement d'une relation qui possède deux attributs X et Y fonctionnellement dépendants, **sans que X soit une clé de la relation**. Les anomalies remettent en cause la cohérence de la base de données.

- **Anomalie d'insertion.** On ne peut pas mémoriser une nouvelle dépendance entre les valeurs de X et Y sans créer un nouveau n-uplet de la relation.
- **Anomalie de suppression.** On ne peut pas supprimer un dernier n-uplet avec certaines valeurs pour X sans perdre l'information de dépendance qui est stockée dans ce n-uplet.
- **Anomalie de modification.** On ne peut pas changer une dépendance sans changer les n-uplets de la relation.

Exemple : Supposons que la relation RESULTAT est réduite aux quatre n-uplets présentés précédemment. En supprimant le quatrième n-uplet, on perd la connaissance du contrôle 254 (anomalie de suppression). Pour enregistrer un nouveau contrôle, il faut l'associer obligatoirement à une note (anomalie de création). Si la division ISTG3 est renommée IEG3, il faut modifier les quatre n-uplets (anomalie de modification). La relation RESULTAT n'est pas bien conçue et l'étude des dépendances fonctionnelles va nous aider à l'améliorer.

L'objectif est donc d'éliminer toute anomalie en évitant la présence de dépendances fonctionnelles redondantes. Le processus de recherche de la qualité s'appelle la normalisation, il permet d'atteindre successivement différents niveaux de qualité appelés **formes normales**.

La **normalisation** est l'opération qui permet de transformer un schéma relationnel en un autre schéma relationnel permettant d'éviter toute redondance et anomalie de mise à jour. Elle s'applique à chaque relation composant le schéma relationnel.

Relation en première forme normale (1FN). C'est une relation dont aucun attribut n'est lui-même une relation. La 1FN sert à rendre opérationnel le raisonnement en termes de DF. De ce fait, le modèle relationnel ne permet pas d'utiliser des attributs dont les valeurs sont des ensembles ou des n-uplets. Une relation qui ne respecterait pas cette règle peut être normalisée en 1FN soit en décomposant l'attribut composé, soit en dupliquant l'attribut de type ensemble.

La relation RESULTAT (CodeElève, NumContrôle, NomElève, PrénomElève, DateNaissance, CodeINSEE, CodeDivision, NomDivision, CodeNiveau, LibelléNiveau, DateContrôle, NoteContrôle) est en 1FN : tous les attributs sont atomiques. L'habitude s'est ancrée de concevoir les données d'un système d'information de façon atomique, même s'il faut pour cela faire une approximation du type de la donnée. Ainsi, un élève peut avoir plusieurs prénoms ; s'il y a besoin de les retenir comme pour l'état civil, il est d'usage d'intégrer tous les prénoms dans une chaîne de caractères qui ne représente finalement qu'une valeur. Un autre usage consiste à dupliquer l'attribut en postulant que certains attributs admettront des valeurs nulles. Ainsi, une entreprise peut disposer de plusieurs numéros de téléphone. Si les attributs NoTél1, NoTél2, NoTél3 sont choisis, nous ne pourrions pas retenir un quatrième numéro. Dans ces conditions, les cas de relations qui ne sont pas en 1FN sont très rares.

Relation en deuxième forme normale (2FN) : c'est une relation en 1FN dont tous les attributs non clés dépendent pleinement de la clé. Toute relation en 1NF peut être décomposée en plusieurs relations en 2FN.

La relation RESULTAT (CodeElève, NumContrôle, NomElève, PrénomElève, DateNaissance, CodeINSEE, CodeDivision, NomDivision, CodeNiveau, LibelléNiveau, DateContrôle, NoteContrôle) n'est pas en 2FN car il existe des attributs qui ne dépendent pas pleinement de la clé :

CodeElève \rightarrow NomElève

CodeElève \rightarrow PrénomElève

CodeElève \rightarrow DateNaissance

CodeElève \rightarrow CodeINSEE

NumContrôle → *DateContrôle*

Cela amène à constituer plusieurs relations en regroupant tous les attributs en dépendance pleine de chaque partie de la clé puis à éliminer ces attributs de la relation initiale. Chaque nouvelle relation prend pour clé la partie de la clé origine de ces dépendances directes. Dans la relation initiale, chaque partie de la clé devient clé étrangère sur la nouvelle relation créée.

ELEVE (***CodeElève***, *NomElève*, *PrénomElève*, *DateNaissance*, *CodeINSEE*, *CodeDivision*, *NomDivision*,
CodeNiveau, *LibelléNiveau*)

CONTROLE (***NumContrôle***, *DateContrôle*)

RESULTAT (***CodeElève***, ***NumContrôle***, *NoteContrôle*)

CodeElève clé étrangère sur *ELEVE*

NumContrôle clé étrangère sur *CONTROLE*

Relation en troisième forme normale (3FN). C'est une relation en 2FN dont toute DF est directe. **Il ne peut y avoir de dépendance entre deux attributs non clés** (source, nous l'avons vu, des anomalies de mise à jour). Toute relation en 2FN peut être décomposée en plusieurs relations en 3FN.

Les relations *CONTROLE* (*NumContrôle*, *DateContrôle*) et *RESULTAT* (*CodeElève*, *NumContrôle*, *NoteContrôle*) sont en 3FN : elles sont en 2FN et il n'y a pas de dépendance entre deux attributs non clés (c'est évident pour les deux relations puisqu'elles ne possèdent chacune qu'un attribut non clé).

La relation *ELEVE* (*CodeElève*, *NomElève*, *PrénomElève*, *DateNaissance*, *CodeINSEE*, *CodeDivision*, *NomDivision*, *CodeNiveau*, *LibelléNiveau*) n'est pas en 3FN : elle est en 2FN (sa clé étant mono-attribut, les DF sont forcément élémentaires) mais il existe des DF qui ne sont pas directes, autrement dit il existe des **DF entre attributs non clés** :

CodeElève → *NomDivision*

CodeElève → *CodeNiveau*

CodeElève → *LibelléNiveau*

CodeElève → *LibelléNiveau*

CodeElève → *CodeDivision* et ***CodeDivision*** → ***NomDivision***

CodeElève → *CodeDivision* et ***CodeDivision*** → ***CodeNiveau***

CodeElève → *CodeDivision* et ***CodeDivision*** → ***LibelléNiveau***

CodeElève → *CodeNiveau* et ***CodeNiveau*** → ***LibelléNiveau***

Pour passer la relation *ELEVE* en 3FN, il faut la décomposer en autant de relations qu'il y a de sources de DFD. Dans chaque nouvelle relation, la source des DFD devient la clé de la relation. Les liens entre les relations sont établis par les clés étrangères.

ELEVE (***CodeElève***, *NomElève*, *PrénomElève*, *DateNaissance*, *CodeINSEE*, *CodeDivision*)

CodeDivision clé étrangère en référence à *CodeDivision* de *DIVISION*

DIVISION (***CodeDivision***, *NomDivision*, *CodeNiveau*)

CodeNiveau clé étrangère en référence à *CodeNiveau* de *NIVEAU*

NIVEAU (***CodeNiveau***, *LibelléNiveau*)

CONTROLE (***NumContrôle***, *DateContrôle*)

RESULTAT (***CodeElève***, ***NumContrôle***, *NoteContrôle*)

CodeElève clé étrangère en référence à *CodeElève* de *ELEVE*

NumContrôle clé étrangère en référence à *NumContrôle* de *CONTROLE*

Toutes les relations obtenues sont maintenant en 3FN.

La **normalisation** est donc l'opération qui permet de transformer un schéma relationnel composé de relations non normalisées en un autre schéma relationnel composé de relations en 3FN afin d'éviter toute redondance et anomalie de mise à jour.

Maintenant que le décor est planté, il faut dynamiser tout cela, en se donnant les moyens de manipuler l'information ainsi représentée : c'est là qu'intervient l'algèbre relationnelle.

L'**algèbre relationnelle** propose une collection d'opérateurs permettant d'interroger (et de mettre à jour) les informations décrites dans un schéma relationnel à travers des **requêtes**. Le résultat de toute opération de l'algèbre est une nouvelle relation (cela signifie qu'il n'y a jamais de doublon : nous y reviendrons). L'algèbre relationnelle utilise les **opérateurs ensemblistes** (union, intersection, différence et produit cartésien) et introduit des **opérateurs relationnels** (restriction – appelée aussi sélection -, projection, jointure, division). Nous ne nous intéressons ici qu'aux principaux opérateurs relationnels.

- Opérateur **Restriction (ou Sélection²²)** (REST). La restriction $REST(R, \langle \text{condition} \rangle)^{23}$ prend en entrée une relation R de schéma S et produit en sortie une nouvelle relation de même schéma ayant comme n-uplets les n-uplets de R qui satisfont la condition de restriction. La condition de restriction utilise les opérateurs de comparaison (< ; ≥ ; ≤ ; > ; = ; ≠), les connecteurs logiques (ET, OU, NON) et les parenthèses.

Exemple : $REST(CONTROLE, DateContrôle > '05/01/2005')$ renvoie la liste des contrôles réalisés après le 5 janvier 2005.

- Opérateur **Projection** (PROJ). La **projection** $PROJ(R, (A_1, A_2, \dots, A_n))$ prend en entrée une relation R de schéma S et produit en sortie une nouvelle relation de schéma $A_1; A_2; \dots; A_n$ (schéma inclus dans S) ayant comme n-uplets ceux de R restreints au sous-schéma $A_1; A_2; \dots; A_n$ (sans doublon).

Exemple : $PROJ(ELEVE, (NomElève, DateNaissance))$ renvoie une relation contenant tous les couples (nom et date de naissance) des élèves. Attention : un seul n-uplet apparaît pour des homonymes nés le même jour.

- Opérateur **Jointure** (JOIN). Soient R1 et R2 deux relations de schémas respectifs S1 et S2. La relation résultat $JOIN(R1, R2, A1 \text{ op } A2)$ a pour schéma (S1, S2) et comme n-uplets la concaténation des n-uplets de R1 avec ceux de R2 s'ils répondent à la condition $A1 \text{ op } A2$, où A1 est un attribut de R1, A2 un attribut de R2 et op un opérateur de comparaison. La jointure est donc équivalente à un produit cartésien suivi d'une restriction. A1 et A2 doivent appartenir au même domaine. Lorsque les attributs comparés sont définis sur un domaine primaire, c'est-à-dire lorsqu'il s'agit de comparer soit une clé étrangère et une clé primaire, soit deux clés étrangères ayant une référence commune, il s'agit d'une « jointure d'enrichissement ». Il nous semble que seul ce type de jointure est à étudier en 1^{er} STG.

Lorsque op est l'égalité, la jointure est une **équijointure**. Si la relation résultante est présentée comme une table, elle comportera deux colonnes ou deux groupes de colonnes identiques. La **jointure naturelle** correspond à une équijointure suivie de la suppression de l'une des colonnes ou de l'un des groupes de colonnes identiques. Le schéma de la relation résultante n'est plus obtenu par juxtaposition des attributs des deux relations opérantes mais par l'union de ces attributs. Lorsque la jointure naturelle est une jointure d'enrichissement et que le critère de jointure est sans ambiguïté possible (lien entre clé étrangère et clé primaire par exemple), le critère de jointure peut rester implicite : $JOIN(R1, R2)$.

Exemple : $JOIN(ELEVE, DIVISION)$, équivalent à $JOIN(ELEVE, DIVISION, ELEVE.CodeDivision = DIVISION.CodeDivision)$ permet d'obtenir une nouvelle relation dans laquelle le nom de la division de chaque élève est désormais explicite. Pour ce faire, un lien a été exploité, pour chaque élève on connaît maintenant le nom de la division : on a exploité le lien entre les deux relations, exprimé par la clé étrangère pour récupérer le nom de division correspondant au code, cela pour chaque élève : c'est finalement le moyen de reconstruire ce que la normalisation a déconstruit...

Remarque : La notation R.A est appelée qualification (attribut A de la relation R) : elle permet de lever l'ambiguïté en cas de nom d'attributs identiques dans des relations différentes. Dans l'exemple précédent, on lit « la valeur de l'attribut CodeDivision de la relation ELEVE est égale à celle de l'attribut de même nom dans la relation DIVISION ».

²² Nous suivons ici l'usage de nombreux enseignants qui désigne cette opération avec le terme « restriction ». Il s'agit d'éviter la confusion du vocable « sélection » avec le mot clé SELECT du langage SQL, suivi dans ce langage par la liste des attributs... projetés ! Notez toutefois que, dans la littérature, des acceptions très différentes sont données au terme « restriction ». Ainsi, dans des ouvrages de référence, il est souvent question de « restriction » pour désigner deux opérateurs : la sélection interprétée comme une restriction « horizontale » et la projection interprétée comme une restriction « verticale ».

²³ Il existe de multiples façons de représenter ces opérations, de façon littérale ou même graphique. Nous avons choisi une expression littérale plutôt bavarde pour ne pas dérouter le lecteur non averti.

- Opérateur **Affectation**²⁴ (\leftarrow). L'affectation permet d'identifier le résultat d'une expression afin de pouvoir la réutiliser dans une autre expression : $R (A_1; A_2; \dots; A_n) \leftarrow JOIN(R1, R2)$

Exemple : Pour obtenir une liste des résultats des élèves au contrôle 15, le calcul peut se décomposer en :

```
R1 ← REST(CONTROLE, NumContrôle = 15)
      // permet de ne retenir que les résultats du contrôle 15
R2 ← JOIN(R1, ELEVE) // jointure naturelle sur l'attribut CodeElève
      // permet de récupérer les informations concernant les élèves ayant composé au contrôle 15
R3(Code, Nom, Prénom, Note) ← PROJ(R2, (CodeElève, NomElève, PrénomElève, NoteContrôle))
      // permet de ne conserver que les attributs indiqués, en les renommant
```

Remarque : Si on ne précise pas de liste d'attributs pour les relations résultats, les noms des champs récupérés sont utilisés implicitement.

Une **base de données relationnelle** est le résultat de l'**implantation** d'un schéma relationnel sur un support informatique dont la gestion est assurée par un **SGBDR** (système de gestion de bases de données relationnel).

Une **table** (relationnelle) représente l'implantation d'une relation. Elle est constituée de **lignes** (ou enregistrements) et de **colonnes** (ou **champs**, correspondant aux attributs de la relation). Elle est la représentation **en extension** d'une relation : chaque ligne y représente un fait avéré (une assertion).

Pour assurer la cohérence d'une base de données, un SGBDR prend en charge un certain nombre de contraintes (contraintes d'entité, liées aux clés primaires, contraintes d'intégrité référentielle, liées aux clés étrangères, contraintes de domaine, liées aux types des attributs, etc.). Le plus souvent, ces contraintes ne sont pas implicites : ne pas déclarer une clé primaire ne permet pas de garantir l'unicité des lignes d'une table, ne pas déclarer une clé étrangère ne permet pas de respecter la contrainte d'intégrité référentielle²⁵.

La création d'une base de données repose sur un langage normalisé : **SQL** (*Structured Query Language*), implémenté sur pratiquement tous les SGBDR. Ce langage a fait l'objet de plusieurs normes ISO dont la plus répandue aujourd'hui est la norme SQL2, définie en 1992. Le succès du langage SQL repose sur sa simplicité : il propose un langage de requêtes ensembliste et assertionnel : écrire une requête, c'est donc avant tout décrire ce que l'on veut obtenir, et pas comment.

Le langage SQL comporte trois parties essentielles :

- le langage de définition de données (LDD) qui permet de définir la structure de la base de données (et notamment les tables et les contraintes d'intégrité),
- le langage de manipulation des données (LMD) qui permet d'interroger et de mettre à jour une base de données sous forme déclarative sans se préoccuper de l'organisation physique des données, mais seulement de sa structure (les tables et les liens représentés par les clés étrangères),
- le langage de contrôle de données (LCD) qui permet de contrôler la sécurité et les accès aux données en définissant notamment les utilisateurs et leurs droits (ce qu'ils peuvent faire sur les tables).

Nous nous intéressons ici uniquement au LMD (langage de manipulation des données), et plus particulièrement à l'interrogation, représentée par l'ordre **SELECT** :

- l'ordre **SELECT** exprime le résultat attendu sous la forme d'une liste d'attributs auxquels il est possible d'appliquer différents opérateurs et fonctions ;
- la clause **FROM** liste les relations utilisées pour évaluer les requêtes ;
- la clause **WHERE** qui est facultative énonce une condition que doivent respecter les enregistrements pour être sélectionnés.

La syntaxe élémentaire d'un ordre **SELECT** est la suivante :

²⁴ L'emploi de l'opérateur « = » n'est cependant pas incongru. En algorithmique, l'affectation se place au sein d'une action procédurale et évoque le transfert d'une valeur, au besoin calculée, dans la variable cible. Ici, la même interprétation peut être faite (\leftarrow) mais il est également possible de penser à l'établissement pérenne d'une formule de calcul entre les relations opérands et la relation résultat (=) : la formule demeure, le résultat dépend du moment où la formule est calculée réellement.

²⁵ Dans ce dernier cas, il arrive même que la déclaration ne soit pas suffisante et qu'il faille explicitement demander à ce que la contrainte référentielle soit respectée.

```
SELECT [DISTINCT] * | {[<Nom_de_table> | <Alias> .]<Nom_de_colonne> [AS <Nouveau_nom>]}
FROM {<Nom_de_table> [<Alias>]}
[WHERE <condition>]
```

Les signes « inférieur » et « supérieur » (<...>) encadrent le nom d'un élément propre à chaque requête. Exemple : <Nom_de_relation> indique que, dans une requête donnée, il faut remplacer cette expression par le nom d'une table.

Les crochets ([...]) encadrent un élément facultatif. Exemple : DISTINCT, AS et WHERE sont des clauses facultatives.

Les accolades ({...}) contiennent la description de l'élément type d'une liste non vide. Exemple : Derrière le mot clé FROM, une liste de tables est citée (au moins une).

La barre (...|...) sépare des options alternatives. Exemple : Derrière SELECT, il est possible d'écrire soit le symbole « * », soit une liste d'attributs.

Exemple : La requête SQL suivante permet d'obtenir la liste des résultats des élèves au contrôle 15.

```
SELECT      CodeElève, NomElève, PrénomElève, NoteContrôle
FROM        CONTROLE, ELEVE
WHERE       CONTROLE.CodeElève = ELEVE.CodeElève
AND         NumContrôle = 15
```

Il est possible d'utiliser des alias (inscrits en gras dans l'exemple suivant) :

```
SELECT      CodeElève, NomElève, PrénomElève, NoteContrôle
FROM        CONTRÔLE C, ELEVE E
WHERE       C.CodeElève = E.CodeElève
AND         NumContrôle = 15
```

Il est également possible de renommer des colonnes obtenues à l'aide de clauses AS (inscrites en gras dans l'exemple suivant) :

```
SELECT      CodeElève AS Code, NomElève AS Nom, PrénomElève AS Prénom, NoteContrôle AS Note
FROM        CONTROLE, ELEVE
WHERE       CONTROLE.CodeElève = ELEVE.CodeElève
AND         NumContrôle = 15
```

Le lien entre algèbre relationnelle et langage SQL est somme toute assez simple, tout au moins pour la partie qui nous occupe. Nous pouvons le représenter à travers quelques situations extraites de notre schéma.

Projection : $PROJ(R, (A,B,C)) \equiv SELECT [DISTINCT] A, B, C FROM R^{26}$

Exemple : $PROJ(R2, (CodeElève, NomElève, PrénomElève, NoteContrôle))$ se traduit par
 $SELECT CodeElève, NomElève, PrénomElève, NoteContrôle FROM R2^{27}$

Restriction : $REST(R, <condition>) \equiv SELECT * FROM R WHERE <condition>$

Exemple : $REST(CONTROLE, NumContrôle = 15)$ se traduit par
 $SELECT * FROM CONTROLE WHERE NumContrôle = 15$

Le caractère * permet d'indiquer qu'on veut afficher tous les attributs des tables citées après la clause FROM.

²⁶ Pour assurer une véritable équivalence, l'emploi de la clause DISTINCT est en principe requis si la clé primaire de R ne fait pas partie des attributs projetés. SQL tolère les doublons ce qui n'est pas le cas du langage algébrique qui respecte la notion d'ensemble.

²⁷ Nous admettrons que la clé de R2 est contenue dans la liste des attributs projetés.

Jointure naturelle : $JOIN(R1, R2) \equiv SELECT * FROM R1, R2 WHERE R1.A = R2.B$ ²⁸

Exemple : *JOIN(R1, ELEVE) se traduit par*
*SELECT * FROM R1, ELEVE WHERE R1.CodeElève = ELEVE.CodeElève*

Remarque : Une autre représentation de la jointure (norme SQL 2) se rencontre notamment dans ACCESS : elle possède le mérite de bien dissocier la jointure des critères de restriction.

*SELECT * FROM R1 JOIN ELEVE ON R1.CodeElève = ELEVE.CodeElève*
// sélectionner toutes les colonnes de R1 lié à ELEVE sur l'égalité de l'attribut CodeElève
Ou

*SELECT * FROM R1 NATURAL JOIN ELEVE*
// sélectionner toutes les colonnes de la jointure naturelle entre R1 et ELEVE

Rappel : Le point entre le nom d'une table et le nom d'un de ses attributs représente la qualification : on y recourt en cas d'ambiguïté, ce qui est le cas ici, puisque le terme CodeElève désigne à la fois un attribut de la table ELEVE et un attribut de la table R1.

Vous pouvez alors vous poser la question : « Comment passer de la suite d'opérations relationnelles à la requête SQL correspondante ? ». En fait nous ne pouvons pas vraiment parler d'un passage, car nous avons affaire à deux approches différentes :

- La construction d'une suite ordonnée d'opérations de l'algèbre relationnelle pour obtenir un résultat peut s'assimiler à la rédaction d'un algorithme : il s'agit de construire ce qu'on appelle un arbre de requêtes qui représente les opérations qu'effectuera le moteur d'interrogation du SGBDR pour fournir le résultat. Nous sommes davantage dans le domaine du COMMENT, utile pour comprendre le principe d'exécution d'une requête (approche algorithmique) et notamment les principes d'optimisation mis en œuvre par le moteur SQL²⁹.

Nous récupérons les lignes du contrôle 15
R1 ← REST(CONTROLE, NumContrôle = 15)
Pour chaque ligne, nous récupérons toutes les informations concernant l'élève
R2 ← JOIN (R1, ELEVE)
Nous ne conservons que les attributs qui nous intéressent
R3 ← PROJ(R2, (CodeElève, NomElève, PrénomElève, NoteContrôle))

- La construction d'une requête SQL pour obtenir le même résultat peut s'assimiler à la rédaction d'une assertion (phrase réputée vraie) portant sur le schéma relationnel. Nous sommes dans le domaine du QUOI (et éventuellement du OÙ) qui s'appuie sur la compréhension du schéma relationnel et du système d'information qu'il représente.

// Donner les attributs qui nous intéressent
SELECT CodeElève, NomElève, PrénomElève, NoteContrôle
// qui se trouvent dans les tables CONTROLE et ELEVE
FROM CONTROLE, ELEVE
// reliées par le code de l'élève
WHERE CONTROLE.CodeElève = ELEVE.CodeElève
// et qui correspondent au contrôle 15
AND NumContrôle = 15

²⁸ Où A est clé primaire et B clé étrangère référençant A, ou l'inverse.

²⁹ Même si l'utilisation d'autres règles d'écriture, associées au principe du parenthésage évite une telle décomposition, en écrivant, par exemple :

$R = ((CONTROLE \otimes ELEVE)[NumContrôle = 15])[CodeElève, NomElève, PrénomElève, NoteContrôle]$ ou

$R = ((CONTROLE [NumContrôle = 15]) \otimes ELEVE)[CodeElève, NomElève, PrénomElève, NoteContrôle]$

Où \otimes représente la jointure naturelle et les crochets représentent la restriction et la projection, selon leur contenu.

Dans les deux cas, il est nécessaire de comprendre la signification du schéma relationnel pour pouvoir transformer une question exprimée en langage naturel (« *donner la liste des résultats des élèves au contrôle 15* ») en une requête. Compte tenu du temps disponible et des objectifs à atteindre, il semble raisonnable pour le professeur de limiter l'approche des élèves au QUOI tout en maîtrisant la démarche du COMMENT pour être capable de présenter les opérateurs relationnels à travers la description de requêtes SQL et de requêtes en mode graphique.

Nous finirons par un parallèle avec les mathématiques qui semble ici judicieux. Il peut y avoir (au moins) trois types d'élèves après un cours sur les systèmes de deux équations à deux inconnues :

- ceux qui ne savent toujours pas les résoudre...
- ceux qui savent les résoudre mais qui ne savent pas à quoi cela sert...
- ceux qui sont capables de transformer un problème en un système d'équations avant de le résoudre.

Notre objectif est bien de former des élèves qui soient capables :

- d'abord d'écrire une requête (minimum syntaxique...);
- ensuite de construire une requête pour répondre à un problème clairement exprimé (utilisation d'une démarche de conception);
- enfin de transformer une question posée en un problème clairement exprimé et construire la requête correspondante (utilisation d'une démarche d'analyse et compréhension du sens de la base de données manipulée).

Nous avons là deux des objectifs essentiels de l'apprentissage du relationnel en classe de première :

- *maîtriser une méthode permettant de construire une requête,*
- *comprendre la signification d'un schéma relationnel.*