Transcription d'une vidéo de Jacques Riguet : Base de données relationnelles et catégories

Dans ce premier écran, j'ai rappelé très rapidement l'état de ce qu'on pouvait appeler une base de données et 1970 est l'année de la naissance des bases de données relationnelles. Avant il y avait un grand désordre dans ce domaine et Codd[1] est le premier à clarifier tout cela par sa notion de relation au sens des bases de données.

Alors donc j'ai donné quelques exemples, ici, un chef d'établissement, mais nous avons tous dans nos poches ou dans nos dossiers des tables de ce genre et à chaque fois que nous avons un rendez-vous, nous consultons un certain carnet dans lequel se trouve des tableaux de ce genre. Et bien souvent aussi, nous avons à nous remémorer un certain tableau, et puis un autre, de façon à rassembler les informations que chacun de ces tableaux nous donne séparément.

Voici un autre exemple ici, un tableau qui donne le nombre d'étudiants, il s'agit d'un chef d'établissement, n'est-ce pas, qui a dans son dossier un feuillet où se trouve un tableau qui donne pour chaque étudiant son nom, son âge, la profession de ses parents.

Alors si j'essaye de donner très rapidement ce que ça signifie du point de vue mathématique, il s'agit d'un ensemble avec des couples. Alors habituellement, les couples, on les écrit avec une parenthèse, un élément, une virgule, un autre élément, et on ferme la parenthèse. On les donne aussi sous forme de flèches : le couple (a, b) est remplacé par un petit dessin ainsi $a \mapsto b$. Par exemple, la flèche $a \overset{\text{père}}{\mapsto} b$.

Ou encore, on peut utiliser cette notation que j'emploie dans la suite, qui est utilisée dans le domaine des bases de données. C'est une notation qui ressemble à une fraction, n'est-ce pas, car chacune de ces données se réfère à un des attributs : ici l'attribut, eh bien, c'est le nom de l'étudiant, dont le numéro, ou plus exactement ici le nom de la personne, ici son âge, et ici la profession du père. Alors donc une base de données, ça va être un certain ensemble d'ensembles de ce type-là, mais avec des lignes successives. Puis il y a ce qu'on appelle des keys (clés), mais je n'ai pas le temps d'insister beaucoup là-dessus.

Voici encore ici la suite de cette énumération et avec ces tableaux, il faut avoir le moyen de synthétiser l'information que deux ou plusieurs de ces tableaux comportent. L'opération fondamentale du point de vue informatique qui est, si vous voulez, ce qu'on appelle la fusion des tables.

Alors je donne un exemple : ici voici un premier tableau qui donne comme information des numéros de factures auxquels sont associés des noms de clients et des montants de factures. Ici, je vais fusionner les tableaux en un seul tableau qui va me donner ce qu'on appelle le recollement, si vous voulez, de ces deux informations. Voici ce recollement et on dira en général que cette information, en quelque sorte, est décomposable en les deux précédentes.

Alors une petite remarque en passant : une relation peut être décomposable, mais peut-être de

Référence : https://www.youtube.com/watch?v=INmhWTn0jgEt=3s
Denise Vella-Chemla, octobre 2026.

1. Edgar Frank Codd (dit Ted Codd) est considéré comme l'inventeur de la notion de base de données relationnelle.

plusieurs manières ; il y a toute une élaboration de ces concepts. Cette notion a eu des successeurs jusque dans les années 1980-1985, et puis on a commencé à faire un peu plus d'algèbre sur ces bases-là.

Alors, un tableau n'est acceptable que s'il satisfait à certaines contraintes. Par exemple, si on se réfère aux premiers exemples ci-dessus, un tableau d'étudiants n'est acceptable que si dans la colonne âge, chaque âge doit nécessairement être un entier positif inférieur à 150 (on est plein d'espoir puisque des personnes qui ont dépassé l'âge de cent cinquante ans, il n'y en a pas eu beaucoup) ; si un symbole d'une autre nature se trouve être présent dans la colonne âge, cela signifierait que ce tableau est le fruit d'une détérioration ou provient d'une erreur et le test de l'âge est donc un moyen, aussi du point de vue informatique, de contrôler que tout se passe bien.

Très simplement, une base doit satisfaire aussi des contraintes d'un autre type. Par exemple, elle doit satisfaire une contrainte de dépendance fonctionnelle : alors c'est une notion très importante, cette notion de dépendance fonctionnelle ; elle a été développée et baptisée au cours des années 1970-1980 et a fait l'objet de plusieurs normes et actions de standardisation informatiques.

Eh bien, reprenons ce que nous voudrions dire, on est friand de notations rigoureuses, c'est-à-dire algébriques.

Passons alors à un autre exemple, lui aussi pris dans un autre domaine qui celui des stations électriques. Ça, c'est le schéma d'un dispositif classique d'interrupteurs va-et-vient qui permettent d'allumer ou d'éteindre en deux endroits différents une certaine lampe et ceci peut se représenter par le tableau qui est à droite ; il y a trois sortes d'attributs si vous voulez, et sur le tableau correspondant, vous voyez d'une façon assez évidente comment le va-et-vient se transpose en tableau.

Alors, j'en reviens à une approche un peu plus algébrique. Eh bien, il faut rappeler qu'une famille d'ensembles S, c'est ce qu'on appelle un schéma, est un élément d'une catégorie. On a habituellement la puissance qui est la catégorie des parties de S et vous voyez ici la catégorie dont les objets fondateurs appartiennent à la catégorie des ensembles discrets dans la catégorie des ensembles et dans les flèches, ce sont les transformations naturelles, les foncteurs.

Alors habituellement aussi, c'est-à-dire que la notion de foncteur de produits se définit dans une catégorie quelconque et dans la catégorie des ensembles, eh bien, il y a un foncteur et tous les foncteurs produits possibles qui sont touchés d'abord. Il y en a un qu'on choisit en particulier, celui qui est défini ainsi : le foncteur qui à chaque famille d'objets d'un ensemble fait correspondre l'ensemble des applications de I dans l'union de tous les ensembles, et la transformation naturelle de ce foncteur, et je note les projections de S, la famille des projections, dans le cas de la notion duale, que la notion de foncteurs produits ou coproduits indique qu'il y a de façon habituelle un foncteur dans l'autre sens.

Voici maintenant la définition précise de ce que c'est : c'est simple : c'est constitué par un ensemble I, qu'on appelle l'ensemble des attributs de la base, c'est noté ici en gris clair, et une famille d'ensembles est ce qu'elle est : un sous-ensemble du produit des E_s pour $s \in I$, c'est-à-dire un sous-ensemble constitué par un ensemble de données qui sont en quelque sorte encadrées par S.

Alors ceci va s'éclairer par le tableau que je vais vous montrer maintenant, qui est une espèce de géométrisation de ce que je viens de dire.

Ça, c'est un peu tôt. Voilà la schématisation d'une base de données. Voici le schéma d'une famille d'ensembles. C'est une famille de données, il y en a trois ou si vous voulez qui sont sur un tirage, tout ce qu'on peut lire en diagonale ou en rond, en ligne ou autre. Souvent ça va être écrit en tables détaillées. Il faut tout de suite dire que la différence entre la notion habituelle et la notion d'algèbre qu'on utilise dans les bases de données, c'est à cette différence près que les attributs dans l'algèbre ordinaire sont ordonnés, définis sur un ensemble fini de données, alors que pour les bases de données, comme on a des sortes d'attributs, eh bien, c'est beaucoup plus souple, mais cela alourdit beaucoup l'écriture.

Voici donc cette base de données. Voici strictement comment on peut la représenter, il s'agit d'un cube n'est-ce pas qui montre, en particulier, qu'à partir de la donnée d'un devis qui est ici représenté sur un cube à trois dimensions, car les semaines que vous voyez cerclées de rouge ne disent pas le ba si ce sont des ab et bab si c'est la deuxième ligne, mais cela ne suffira pas, il faut avoir les trois éléments de la base de données. Eh bien ces éléments peuvent se lire totalement, et vous voyez qu'il y a donc une notion qui s'impose géométriquement, en quelque sorte, cette notion de projection des membres verticalement va donner lieu à la base sur les enregistrements qui ont cet attribut, et de l'autre côté, pour l'autre attribut, on va avoir une projection sur la verticale. Et là, je peux définir, je reviens en arrière, je crois, voilà, les projections sont définies avant, je pense. Oui, voici donc la définition formelle et catégorique même des projections. Alors vous voyez que j'utilise ce symbole pour dire que lorsqu'on a une application, il y a une équivalence associée à la projection sur le sous-ensemble d'attributs J et à ceci une relation d'équivalence, c'est ce qu'on appelle souvent l'équivalence cylindrique, n'est-ce pas : c'est la projection sur un même plan. Donc il y a un cylindre horizontal, cylindre vertical par exemple dans le cas que j'ai établi, alors il y a un jeu de...

Presque naturellement s'est imposée la notion de ce que j'ai appelé des équivalences conjuguées ou surconjuguées, il y a bien longtemps, je ne sais pas quelle est la terminologie actuelle : deux équivalences sont dites conjuguées si leur intersection est la diagonale et si l'équivalence qui est engendrée par elles est la relation grossière. Ceci vous donne une vague idée de ce que j'avais à dire.

Alors ici il y a un fait qui s'impose, c'est que si cela est très général, indépendamment des bases de données, on peut aussi utiliser que l'équivalence sur un ensemble, c'est une relation relationnelle très banale mais importante, c'est que si on a une équivalence sur un ensemble, elle définit sur l'ensemble ordonné des parties une fermeture qui à toute partie quelconque X fait correspondre l'union des classes d'équivalence qui ont une intersection non vide avec X ; eh bien, cette fermeture est topologique, c'est-à-dire qu'elle satisfait, si j'appelle ϕ cette fermeture, à la fermeture de l'union égale à l'union des fermetures. Eh bien, c'est le cas des champs de cubes cylindriques, et dans les bases de données, nous avons donc là, presque immédiatement à partir de faits algébriques qui sont très simples mais importants, le fait que dans une base de données, il y a des tas de topologies qui prennent naissance parce qu'automatiquement à partir de là...

Voilà, alors ça, ça fait allusion à la catégorie des carquois (en anglais quivers) c'est ce qu'on appelle

un carquois, je l'appelle ainsi parce qu'en général on l'appelle la catégorie des graphes, mais je préfère faire une distinction entre les carquois et les graphes, c'est-à-dire qu'on ne tient pas dans un graphe en général du sens des catégoriciens, on impose une section, c'est-à-dire on impose que chaque objet définisse une flèche d'identité, tandis que dans un quiver, dans un carquois, c'est juste une sorte de flèches, ou si vous voulez du produit algébrique, c'est juste des ensembles et des applications d'un ensemble de flèches dans un ensemble d'objets. Bon, à cette catégorie de quivers, de carquois, cette propriété fondamentale est qu'elle admet des pullbacks. Alors j'ai essayé de donner une idée. Alors bon, ça c'était quelque fois très important pour la suite, que je ne devrais bien sûr pas hésiter à développer, de la catégorisation des bases de données et de dire que cette catégorie des quivers, ou si vous préférez des pullbacks, permet de définir les grands schémas.

Alors voilà un exemple : ici vous avez un graphe, enfin un quiver, carquois 2, et puis un carquois 0, et le pullback, le pushout, excusez-moi, le pushout ici, a un diagramme dual, mais celui-là je voulais le dessiner, vous voyez les écoprojections ici, à 2. Alors quand on dessine en tant que pushout, on ne voit pas bien comment ceci correspond à des sortes de recollements ; en fait dans les quivers, la catégorie des quivers, c'est bien d'aller aussi au grand col blanc qui apparaît sur ce schéma-là.

Alors j'espère que vous avez le temps de voir qu'il y a presque ici la quasi-identité entre ce qui est sur cette partie supérieure et ce qui est ici, enfin le fait de recoller le quiver de base A ici, le quiver de base B, il a suffi de les aplatir en quelque sorte, restant bien entendu sous la dépendance des parents et des ados, dont je n'ai pas le temps d'insister.

Oui, enfin j'y passerai rapidement parce que c'est un petit essai que j'avais fait pour essayer d'expliquer la notion, enfin ma manière de voir, si vous voulez, les théories et algèbres, ceci avait pour but d'expliquer pourquoi je trouve nécessaire au point de vue algébrique, même quand elles sont déclarées de sorte qu'elles sont en fait toujours multi-sortes, parce qu'il y a à mon sens nécessité d'introduire une sorte qui est une sorte de logique. Alors ça c'est un petit schéma qui essayait d'expliquer, bon, je n'ai pas le temps d'insister davantage, parce que c'est la géo-morphogenèse dans ce sens-là.

Bon, alors ceci pour dire que déjà à ce niveau-là, on a une, on voit déjà non seulement le lien mais la prise en charge de ce que j'ai dit précédemment sur les bases de données dans le cadre de la logique, à cette différence près que dans la théorie des algèbres avec spécifications, etc., on fait appel à des opérations, il y a des sortes de flèches, vous voulez, des flèches qui sont des opérations et des flèches qui sont des flèches de relations, de prédicats, qui vont donner naissance à des prédicats, alors qu'au niveau de l'algèbre des bases de données relationnelles, il y a seulement des prédicats.

Bon alors là il y aurait une discussion à avoir sur la possibilité de supprimer finalement les opérations coordonnées, de ne considérer que des prédicats.

Voilà, alors ça je le dis rapidement, je laissais du blanc là pour avoir la possibilité de vous expliquer en gros ce que je venais de faire, et pour les prédicats, en théorie des bases de données, les variables ont des types non ordonnés ici là, ce à quoi j'ai déjà fait allusion, il n'y a pas seulement des opérations, il y a seulement des prédicats. Mais alors quand on a des bases de données, on les interroge, c'est-à-dire qu'on fait des requêtes. Alors les requêtes en fait, dans les bases de données, sont des

modèles de certaines formules dans cette algèbre que j'appelle U, c'est-à-dire que les requêtes sont aussi des formules.

Alors bon, je renvoie à un livre qui est très détaillé sur ces sujets, qui est le livre de Abiteboul, Hull et Vianu, dont je suis le préfacier, ce livre est en anglais, je ne sais pas s'il y a une édition française, en tout cas il a paru il y a une dizaine d'années maintenant.

Vous avez des questions ? Oui, il y a encore... je croyais qu'il y avait une dernière diapositive, c'est ce bouton-là, non, il n'y en a plus qu'un. Bon, eh bien juste deux minutes encore pour dire que c'est un sujet qui est en plein développement et que si vous désirez en savoir plus, j'ai vraiment dit très peu de choses sur l'aspect catégorique donc des bases de données, alors si vous désirez en savoir plus et beaucoup plus, eh bien vous savez, sur Internet, vous cherchez "bases de données relationnelles et catégories" et vous verrez alors que dès des résultats de 2012 et 2013, c'est-à-dire de Spivak, il y a déjà une masse de documents qui reprennent nombre de ces travaux, qui représentent 4500 pages et qui font appel aux tout derniers développements de la théorie des catégories. Voilà, je n'en dirai pas plus.