11

Bases de données relationnelles

Un des grands défis du XXI^{e} siècle, rendu possible par l'émergence d'internet à la fin du XX^{e} siècle, est de regrouper, d'ordonner, et de mettre à disposition un grand nombre de données, en vue d'une utilisation par des experts (études statistiques...) ou des particuliers (renseignements...). Cette immense mise en commun de données éparpillées nécessite des structures spéciales : outre la mémoire nécessaire pour le stockage des données, il faut aussi pouvoir structurer ces données de façon à pouvoir en extraire facilement ce qui nous intéresse : extraire par exemple d'une base de donnée musicale les oeuvres de compositeurs russes, ou les sonates pour flûte écrites entre 1700 et 1750. Il faut donc une structure permettant de dégager rapidement les principales caractéristiques d'une oeuvre (compositeur, date, pays, instruments, type d'oeuvre, etc).

La structure de base de donnée (BDD) relationnelle répond à cette question. Nous nous proposons dans ce chapitre d'étudier le façon intuitive cette structure de BDD relationnelle, avant d'en proposerons une formalisation algébrique. Nous verrons dans le chapitre suivant un langage adapté à la manipulation de ces bases de données (SQL), ainsi que les différentes opérations algébriques possibles en vue de faire des requêtes (extraire les informations précises qui nous intéressent).

Nous illustrons ce chapitre et le suivant par la création d'une base de donnée musicale.

I Environnement client / serveur

Le principe-même d'une base de données impose que de nombreuses personnes peuvent y avoir accès. Cependant, il est peu souhaitable que tout le monde ait un accès direct à cette base, afin d'éviter les accidents de manipulation qui pourraient faire perdre des données de la base. Pour cette raison, on opère en général une séparation stricte entre la machine (ou machine virtuelle) sur laquelle est stockée la base, et les machines des utilisateurs. Les utilisateurs n'ont en fait accès à la base (et la machine qui la contient) que *via* des « requêtes » dont la syntaxe et l'inoffensivité sont bien contrôlées. Par ailleurs, les utilisateurs ont des droits bien établis : la plupart d'entre eux n'ont qu'un droit de consultation ; les utilisateurs ayant un droit d'ajout ou de modification sont en petit nombre et souvent supervisés par un superutilisateur unique (root).
Cette séparation nette entre la base elle-même et les utilisateurs conduit à la notion d'environnement client/serveur, ou à l'architecture 3-tiers qui en est une variante améliorée. Dans une telle structure, les logiciels sont scindés en 2 : une partie légère du coté du client, et une partie lourde (le traitement) du côté du serveur.

I.1 Le serveur informatique

Définition 11.1.1 (Serveur informatique)
Le serveur informatique est un dispositif informatique offrant des services à des clients.

Ces services peuvent être de différents types suivant la nature du serveur :

- partage de fichiers ;
- partages de données (BDD) ;
- hébergeurs web : accès au Web, stockage de pages webs ;
- serveurs de courrier électronique : stockage et envoi de messages ;
- réseaux locaux : mise en commun au sein d'une entreprise des ressources informatiques, que ce soit matérielles (imprimantes...) ou logicielles (cela évite d'avoir à intaller les logiciels lourds sur toutes les machines) ;
- serveur local à un ordinateur : gère les différents utilisateurs de l'ordinateur ;
- cloud computing : offre aux clients une grande puissance de calcul. Cela permet de mutualiser la puissance : plutôt que d'augmenter individuellement leur capacité de calcul par l'achat de matériel neuf, les clients payent le droit de se servir des ressources calculatoires mises à disposition par un fournisseur.

Un serveur est en communication avec plusieurs clients, généralement un grand nombre (figure 11.1). La communication entre le serveur et les clients se fait suivant un certain protocole de communication strict et sécurisé, dans la mesure où la sécurité et l'intégrité du serveur sont bien assurées par ce protocole.

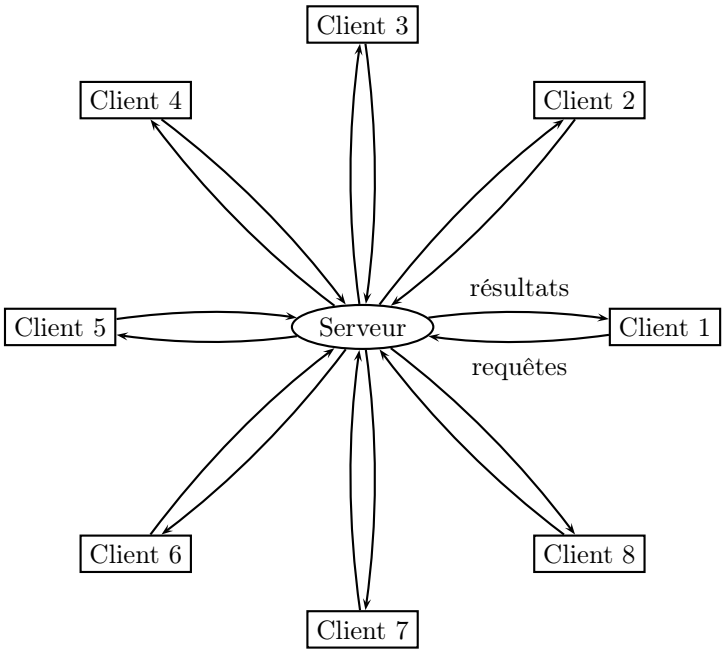


FIGURE 11.1 – Architecture client-serveur

Les caractéristiques demandées à un serveur sont :

- la rapidité de communication (débit)
- la rapidité de traitement
- la capacité d'adaptation à la demande (augmentation de la taille en mémoire, du nombre de requêtes simultanées...)
- la disponibilité 24h sur 24.

Les serveurs sont donc très souvent des machines beaucoup plus puissantes que la moyenne, et très fiables (afin d'éviter les pannes matérielles).

I.2 Le client

Définition 11.1.2 (Client)
Le client est un matériel logiciel ou informatique, permettant l'envoi de requêtes à un serveur donné, et la réception des réponses.

Il s'agit donc d'une interface légère gérant de façon conviviale la communication avec un serveur, traduisant les requêtes de l'utilisateur de façon adéquate.

Les logiciels associés à une architecture client-serveur sont organisés en trois couches :

- la gestion du stockage de l'information ;
- le traitement des requêtes, donc la partie calculatoire ;
- le client : l'interface avec l'utilisateur (entrée des requêtes, affichage des résultats).

La première est essentiellement destinée au maintien et à la mise à jour du serveur. La seconde est souvent la partie lourde du logiciel ; quant à la troisième, elle n'est qu'une interface légère de communication. Les deux premières couches sont installées sur le serveur lui-même, alors que la troisième couche (le client) est installée chez tous les utilisateurs. Cela présente une économie logicielle (seule une interface légère est installée en plusieurs exemplaires), ainsi qu'une sécurité, puisque l'utilisateur n'a pas un accès physique direct au serveur, mais ne peut communiquer avec lui que par l'intermédiaire de l'interface client, qui ne transmet que des requêtes inoffensives.

I.3 Architecture 3-tiers

Définition 11.1.3 (Architecture 3-tiers)
Il s'agit d'une évolution de l'architecture client-serveur, dans laquelle les trois couches logicielles exposées ci-dessus sont physiquement séparées. Ainsi, le serveur lui-même est scindé en deux serveurs :

- serveur 1 : stockage de l'information
- serveur 2 : serveur métier, i.e. serveur effectuant la gestion logicielle.

La terminologie provient de l'anglais « tier » signifiant « couche ».

Le client communique avec le serveur métier qui lui-même communique avec le serveur de stockage (figure 11.2) : le client est davantage séparé du serveur de stockage, ce qui minimise les risques de pertes de données suite à des erreurs (involontaires ou malveillantes) de manipulation.

Les bases de données actuelles sont contruites sur ce schéma (figure 11.3). Le serveur métier s'appelle le système de gestion de base de donnée (SGBD).

On peut voir plusieurs intérêts à une telle dissociation :

- dissociation BDD-SGBD : il n'y a pas de communication directe entre le client et la base : la passage obligatoire par le serveur de gestion, contrôlé rigoureusement par un superutilisateur, est une garantie supplémentaire de sécurité pour la base. Ce superutilisateur (administrateur de la base de donnée) est en théorie l'unique personne pouvant modifier la base. C'est elle qui attribue des droits aux clients (droits de consultation, de modification, d'ajout...)
- dissociation client-BDD : outre la sécurité, cette dissociation permet un enrichissement de la base indépendante des utilisateurs.
- dissociation client-SGBD : de même, les logiciels de gestion peuvent être actualisés indépendamment du client : ainsi, une amélioration logicielle peut se faire uniquement au niveau du serveur, et ne nécessite pas une actualisation au niveau de chaque client. Par ailleurs, seule une interface légère est nécessaire du côté client, ce qui ne nécessite pas une machine particulièrement performante.

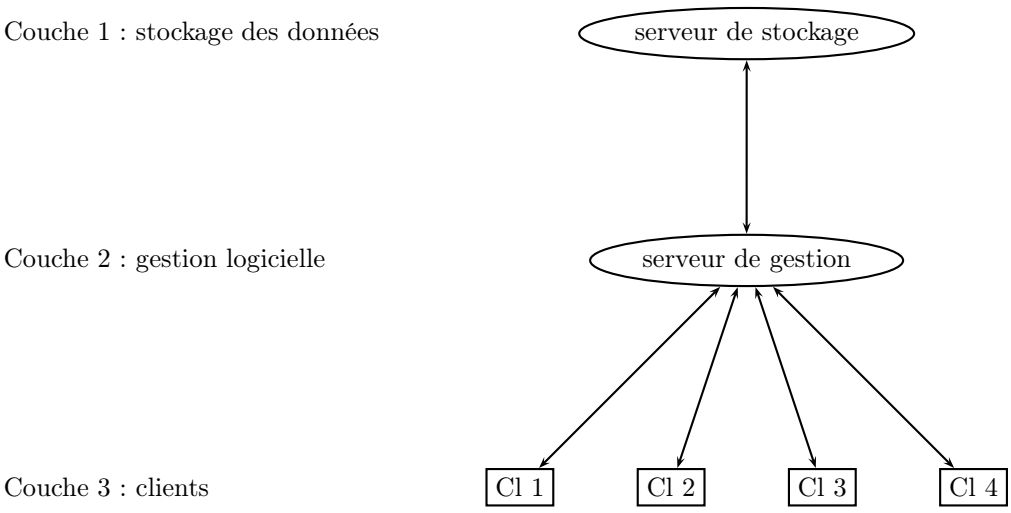


FIGURE 11.2 – Architecture 3-tiers

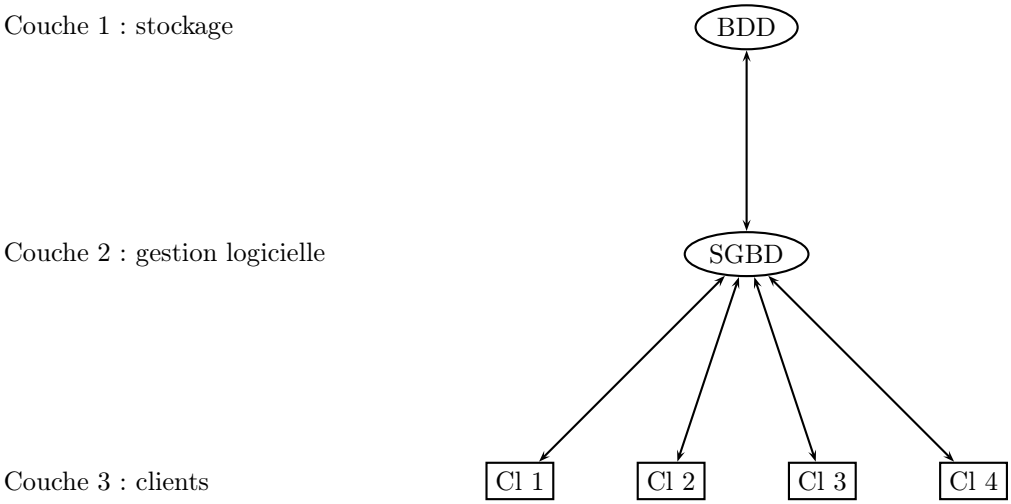


FIGURE 11.3 – Architecture 3-tiers pour une BDD

II Bases de données

II.1 Présentation intuitive

Il s'agit de stocker des données de façon adaptée, de sorte à :

- pouvoir facilement faire des recherches
- prendre le moins de place possible en mémoire.

Imaginons par exemple que nous souhaitions faire une base de donnée musicale avec les types de données suivantes :

- Compositeur
- Dates de naissance et de mort
- Pays
- Titre de l'oeuvre
- Date de composition
- Type d'oeuvre (symphonie, concerto...)
- instrumentation (orchestre, choeur, soliste, ensemble de chambre...)

- Instruments solistes
- Famille des instruments solistes (vents, cordes...)

La première idée pouvant venir à l'esprit est une présentation linéaire sous forme d'un tableau (figure 11.4).
On se rend compte assez rapidement que cette présentation assez simpliste présente un certain nombre d'inconvénients, comme par exemple les redondances (on indique les dates et pays des compositeurs autant de fois qu'il y a d'oeuvre d'eux dans la base !), ainsi que les entrées multiples (dans l'instrumentation ou les solistes par exemple), qui compliquent les recherches.

On peut alors penser à séparer les données en plusieurs tableaux. Par exemple un tableau des compositeurs, un tableau des oeuvres et un tableau des instruments (figure 11.5)
Les redondances ne sont pas complètement supprimées : par exemple indiquer dans l'instrumentation qu'il y a des solistes est redondant avec la colonne soliste, qui n'est remplie que dans ce cas. On peut supprimer cette information. Par ailleurs, il reste le problème des entrées multiples. Ce problème-là n'est pas très dur à régler : il suffit de dupliquer la ligne autant de fois qu'il y a d'entrée de l'attribut multiple. Mais cela au risque de créer de nouvelles redondances qu'il faudra supprimer (tableau 11.6, on n'y indique que le tableau OEUVRES, les autres étant inchangées)
Pour supprimer les redondances ainsi apparues, on scinde à nouveau le tableau :

- un premier tableau identifiera l'oeuvre (on peut y mettre aussi la date, déterminée entièrement et de façon unique par l'oeuvre, ainsi que le type). On créera un identifiant de l'oeuvre, construit par exemple sur les premières lettres du compositeur puis une numérotation, de façon à ce que la donnée de cet identifiant caractérise de façon unique l'oeuvre.
- Un deuxième tableau donnera les ensembles instrumentaux utilisés (orchestre, choeur, quatuor...) pour chacune des oeuvres, identifiées par l'identifiant du premier tableau.
- Un troisième donne les instruments solistes.

Cela nous donne les tableaux de la figure 11.7.
Évidemment, les lignes des deux derniers tableaux comportant une entrée NIL sont inutiles. La non-présence de ces éléments dans le tableau suffit à retrouver cette information. On peut donc supprimer toutes ces lignes. Par ailleurs, si l'on veut par la suite pouvoir faire des références précises à ces tableaux, il faut pouvoir identifier chacune des lignes de ces tableaux de façon unique et non équivoque. Pour cette raison, on a souvent recours à un **identifiant** (une numérotation, qui peut être purement utilitaire et ne pas avoir de sémantique particulière).

Enfin, comme il est plus facile de créer la structure de la base de donnée initialement que de faire des modifications par la suite (surtout si de nombreuses données sont rentrées initialement), on réfléchit attentivement à toutes les contraintes et possibilités liées aux attributs, de façon à limiter le nombre de dépendances pouvant apparaître par la suite, ou la rupture d'un identifiant. Nous illustrons ces propos par deux exemples :

- Supposons que nous voulions ajouter Haendel à notre liste de compositeurs. Quel pays mettre ? L'Allemagne ou l'Angleterre ? Il serait préférable de pouvoir mettre les deux. Ainsi, le compositeur ne détermine pas le pays, et on a une entrée multiple. Comme dans le cas des instrumentations, cela peut inciter à scinder le tableau des compositeurs, en séparant les nationalités. Ce cas n'est pas unique, on peut aussi songer à Scarlatti (Italie ou Espagne ?), Rossini (Italie ou France ?), Stravinski (Russie ou France ?), Rachmaninov (Russie ou USA ?) ou encore Johann-Christian Bach (Allemagne, Italie ou Angleterre ?). Cela dépend aussi de savoir si on veut indiquer dans cette colonne la nationalité administrative, ou le pays d'accueil. Si c'est cette dernière interprétation qu'on choisit, il convient également de reconsidérer le cas de Beethoven.
- Le cas Bach soulève un autre problème : le patronyme ne caractérise pas le compositeur, et ne peut pas servir d'identifiant. Ainsi, nombreuses sont les familles musiciennes dans l'histoire de la musique : la prolifique famille Bach, bien entendu, mais également les Couperin, les Scarlatti, les Mozart, ou plus récemment les Strauss ou les Alain ; sans compter les homonymes non familiaux (comme Johann et Richard Strauss). Le prénom même ne suffit pas à régler le problème

Nom	naissance	mort	pays	titre	date	type	instrumentation	solistes	famille
Couperin	1668	1733	France	Messe des couvents	1690	messe	soliste	orgue	claviers
Bach	1685	1750	Allemagne	Variations Goldberg	1740	variations	soliste	clavecin	claviers
Bach	1685	1750	Allemagne	L'art de la fugue	1750	recueil	soliste	clavecin	claviers
Bach	1685	1750	Allemagne	Le clavier bien tempéré I	1722	recueil	soliste	clavecin	claviers
Bach	1685	1750	Allemagne	Messe en si mineur	1748	messe	orchestre, choeur, chanteurs		
Mozart	1756	1791	Autriche	Don Giovanni	1787	opéra	orchestre, choeur, chanteurs		
Mozart	1756	1791	Autriche	Die Zauberflöte	1791	opéra	orchestre, choeur, chanteurs		
Mozart	1756	1791	Autriche	Concerto pour flûte et harpe	1778	concerto	orchestre, solistes	flûte, harpe	bois, cordes
Mozart	1756	1791	Autriche	Messe du couronnement	1779	messe	orchestre, choeur, chanteurs		
Mozart	1756	1791	Autriche	Requiem	1791	messe	orchestre, choeur, solistes		voix
Beethoven	1770	1827	Allemagne	Symphonie n° 5	1808	symphonie	orchestre		
Beethoven	1770	1827	Allemagne	Symphonie n° 9	1824	symphonie	orchestre, choeur, chanteurs		
Beethoven	1770	1827	Allemagne	Quatuor n° 13	1825	quatuor	quatuor à cordes		
Beethoven	1770	1827	Allemagne	Grande Fugue	1825	quatuor	quatuor à cordes		
Beethoven	1770	1827	Allemagne	Sonate n° 29	1818	sonate	soliste	piano	claviers
Beethoven	1770	1827	Allemagne	Sonate pour violon et piano n° 5	1801	sonate	solistes	violon, piano	cordes, claviers
etc									

FIGURE 11.4 – Tableau des données brutes

COMPOSITEURS			
Nom	naissance	mort	pays
Couperin	1668	1733	France
Bach	1685	1750	Allemagne
Mozart	1756	1791	Autriche
Beethoven	1770	1827	Allemagne

INSTRUMENTS	
instrument	famille
orgue	claviers
clavecin	claviers
flûte	bois
harpe	cordes
piano	claviers
violon	cordes

OEUVRES					
Nom	titre	date	type	instrumentation	solistes
Couperin	Messe des couvents	1690	messe		orgue
Bach	Variations Goldberg	1740	variations	soliste	clavecin
Bach	L'art de la fugue	1750	recueil	soliste	clavecin
Bach	Le clavier bien tempéré I	1722	recueil	soliste	clavecin
Bach	Messe en si mineur	1748	messe	orchestre, choeur, chanteurs	
Mozart	Don Giovanni	1787	opéra	orchestre, choeur, chanteurs	
Mozart	Die Zauberflöte	1791	opéra	orchestre, choeur, chanteurs	
Mozart	Concerto pour flûte et harpe	1778	concerto	orchestre, solistes	flûte, harpe
Mozart	Messe du couronnement	1779	messe	orchestre, choeur, chanteurs	
Mozart	Requiem	1791	messe	orchestre, choeur, chanteurs	
Beethoven	Symphonie n° 5	1808	symphonie	orchestre	
Beethoven	Symphonie n° 9	1824	symphonie	orchestre, choeur, chanteurs	
Beethoven	Quatuor n° 13	1825	quatuor	quatuor à cordes	
Beethoven	Grande Fugue	1825	quatuor	quatuor à cordes	
Beethoven	Sonate n° 29	1818	sonate	soliste	piano
Beethoven	Sonate pour violon et piano n° 5	1801	sonate	solistes	violon, piano

Figure 11.5 – Première repartition des données

(comme l'indique le problème des Johann Strauss père et fils). Ainsi, là aussi, il faut créer un identifiant. Prendre comme identifiant les 3 premières lettres du nom n'est pas suffisant (problème d'homonymie, ou plus généralement, de noms commençant de la même façon, comme Chopin et Chostakovitch, ou Schubert et Schumann). On peut par exemple prendre les 3 premières lettres, suivies d'un numéro pour distinguer les homonymes.

On arrive au final au schéma de la figure 11.8, comportant 6 tableaux.

OEUVRES					
Nom	titre	date	type	instrumentation	solistes
Couperin	Messe des couvents	1690	messe	soliste	orgue
Bach	Variations Goldberg	1740	variations		clavecin
Bach	L'art de la fugue	1750	recueil		clavecin
Bach	Le clavier bien tempéré I	1722	recueil		clavecin
Bach	Messe en si mineur	1748	messe	orchestre	
Bach	Messe en si mineur	1748	messe	choeur	
Bach	Messe en si mineur	1748	messe	chanteurs	
Mozart	Don Giovanni	1787	opéra	orchestre	
Mozart	Don Giovanni	1787	opéra	choeur	
Mozart	Don Giovanni	1787	opéra	chanteurs	
Mozart	Die Zauberflöte	1791	opéra	orchestre	
Mozart	Die Zauberflöte	1791	opéra	choeur	
Mozart	Die Zauberflöte	1791	opéra	chanteurs	
Mozart	Concerto pour flûte et harpe	1778	concerto	orchestre	flûte
Mozart	Concerto pour flûte et harpe	1778	concerto	orchestre	harpe
Mozart	Messe du couronnement	1779	messe	orchestre	
Mozart	Messe du couronnement	1779	messe	choeur	
Mozart	Messe du couronnement	1779	messe	chanteurs	
Mozart	Requiem	1791	messe	orchestre	
Mozart	Requiem	1791	messe	choeur	
Mozart	Requiem	1791	messe	chanteurs	
Beethoven	Symphonie n° 5	1808	symphonie	orchestre	
Beethoven	Symphonie n° 9	1824	symphonie	choeur	
Beethoven	Symphonie n° 9	1824	symphonie	chanteurs	
Beethoven	Symphonie n° 9	1824	symphonie	orchestre	
Beethoven	Quatuor n° 13	1825	quatuor	quatuor à cordes	
Beethoven	Grande Fugue	1825	quatuor	quatuor à cordes	
Beethoven	Sonate n° 29	1818	sonate		piano
Beethoven	Sonate pour violon et piano n° 5	1801	sonate		violon
Beethoven	Sonate pour violon et piano n° 5	1801	sonate		piano

FIGURE 11.6 – Tableau sans entrée multiple

OEUVRES				
IdOeuvre	Compositeur	Titre	Date	Type
COU 1-1	Couperin	Messe des couvents	1690	messe
BAC 1-1	Bach	Variations Goldberg	1740	variations
BAC 1-2	Bach	L'art de la fugue	1750	recueil
BAC 1-3	Bach	Le clavier bien tempéré I	1722	recueil
BAC 1-4	Bach	Messe en si mineur	1748	messe
MOZ 1-1	Mozart	Don Giovanni	1787	opéra
MOZ 1-2	Mozart	Die Zauberflöte	1791	opéra
MOZ 1-3	Mozart	Concerto pour flûte et harpe	1778	concerto
MOZ 1-4	Mozart	Messe du couronnement	1779	messe
MOZ 1-5	Mozart	Requiem	1791	messe
BEE 1-1	Beethoven	Symphonie n° 5	1808	symphonie
BEE 1-2	Beethoven	Symphonie n° 9	1824	symphonie
BEE 1-3	Beethoven	Quatuor n° 13	1825	quatuor
BEE 1-4	Beethoven	Grande Fugue	1825	quatuor
BEE 1-5	Beethoven	Sonate n° 29	1818	sonate
BEE 1-6	Beethoven	Sonate pour violon et piano n° 5	1801	sonate

INSTRUMENTATION	
IdOeuvre	formation intrumentale
COU 1-1	NIL
BAC 1-1	NIL
BAC 1-2	NIL
BAC 1-3	NIL
BAC 1-4	orchestre
BAC 1-4	choeur
BAC 1-4	chanteurs
MOZ 1-1	orchestre
MOZ 1-1	choeur
MOZ 1-1	chanteurs
MOZ 1-2	orchestre
MOZ 1-2	choeur
MOZ 1-2	chanteurs
MOZ 1-3	orchestre
MOZ 1-4	orchestre
MOZ 1-4	choeur
MOZ 1-4	chanteurs
MOZ 1-5	orchestre
MOZ 1-5	choeur
MOZ 1-5	chanteurs
BEE 1-1	orchestre
BEE 1-2	orchestre
BEE 1-2	choeur
BEE 1-2	chanteurs
BEE 1-3	quatuor à cordes
BEE 1-4	quatuor à cordes
BEE 1-5	NIL
BEE 1-6	NIL

SOLISTES	
IdOeuvre	instrument soliste
COU 1-1	clavecin
BAC 1-1	clavecin
BAC 1-2	clavecin
BAC 1-3	NIL
MOZ 1-1	NIL
MOZ 1-2	NIL
MOZ 1-3	flûte
MOZ 1-3	harpe
MOZ 1-4	NIL
MOZ 1-5	NIL
BEE 1-1	NIL
BEE 1-2	NIL
BEE 1-3	NIL
BEE 1-4	NIL
BEE 1-5	piano
BEE 1-6	violon
BEE 1-6	piano

Figure 11.7 – Tableau sans les redondances pour les oeuvres

OEUVRES				
IdOeuvre	Compositeur	Titre	Date	Type
COU 1-1	COU 1	Messe des couvents	1690	messe
BAC 1-1	BAC 1	Variations Goldberg	1740	variations
BAC 1-2	BAC 1	L'art de la fugue	1750	recueil
BAC 1-3	BAC 1	Le clavier bien tempéré I	1722	recueil
BAC 1-4	BAC 1	Messe en si mineur	1748	messe
MOZ 1-1	M0Z 1	Don Giovanni	1787	opéra
MOZ 1-2	MOZ 1	Die Zauberflöte	1791	opéra
MOZ 1-3	MOZ 1	Concerto pour flûte et harpe	1778	concerto
MOZ 1-4	MOZ 1	Messe du couronnement	1779	messe
MOZ 1-5	MOZ 1	Requiem	1791	messe
BEE 1-1	BEE 1	Symphonie n° 5	1808	symphonie
BEE 1-2	BEE 1	Symphonie n° 9	1824	symphonie
BEE 1-3	BEE 1	Quatuor n° 13	1825	quatuor
BEE 1-4	BEE 1	Grande Fugue	1825	quatuor
BEE 1-5	BEE 1	Sonate n° 29	1818	sonate
BEE 1-6	BEE 1	Sonate pour violon et piano n° 5	1801	sonate

NATIONALITÉ		
IdNat	IdComp	pays
1	COU 1	France
2	BAC 1	Allemagne
3	MOZ 1	Autriche
4	BEE 1	Allemagne
5	BEE 1	Autriche
6	HAE 1	Allemagne
7	HAE 1	Angleterre
8	STR 1	Russie
9	STR 1	France
10	BAC 2	Allemagne
11	BAC 2	Italie
12	BAC 2	Angleterre

COMPOSITEURS				
IdComp	Nom	Prénom	naissance	mort
COU 1	Couperin	François	1668	1733
BAC 1	Bach	Johann Sebastian	1685	1750
MOZ 1	Mozart	Wolfgang Amadeus	1756	1791
BEE 1	Beethoven	Ludwig (van)	1770	1827
HAE 1	Haendel	Georg Friedrich	1685	1759
STR 1	Stravinski	Igor	1882	1971
BAC 2	Bach	Johann Christian	1735	1782

SOLISTES		
IdSol	IdOeuvre	instrument soliste
1	COU 1-1	clavecin
2	BAC 1-1	clavecin
3	BAC 1-2	clavecin
4	MOZ 1-3	flûte
5	MOZ 1-3	harpe
6	BEE 1-5	piano
7	BEE 1-6	violon
8	BEE 1-6	piano

INSTRUMENTS	
instrument	famille
orgue	claviers
clavecin	claviers
flûte	bois
harpe	cordes
piano	claviers
violon	cordes

INSTRUMENTATION		
IdInstr	IdOeuvre	formation intrumentale
BAC 1-4-1	BAC 1-4	orchestre
BAC 1-4-2	BAC 1-4	choeur
BAC 1-4-3	BAC 1-4	chanteurs
MOZ 1-1-1	MOZ 1-1	orchestre
MOZ 1-1-2	MOZ 1-1	choeur
MOZ 1-1-3	MOZ 1-1	chanteurs
MOZ 1-2-1	MOZ 1-2	orchestre
MOZ 1-2-2	MOZ 1-2	choeur
MOZ 1-2-3	MOZ 1-2	chanteurs
MOZ 1-3-1	MOZ 1-3	orchestre
MOZ 1-4-1	MOZ 1-4	orchestre
MOZ 1-4-2	MOZ 1-4	choeur
MOZ 1-4-3	MOZ 1-4	chanteurs
MOZ 1-5-1	MOZ 1-5	orchestre
MOZ 1-5-2	MOZ 1-5	choeur
MOZ 1-5-3	MOZ 1-5	chanteurs
BEE 1-1-1	BEE 1-1	orchestre
BEE 1-2-1	BEE 1-2	orchestre
BEE 1-2-2	BEE 1-2	choeur
BEE 1-2-3	BEE 1-2	chanteurs
BEE 1-3-1	BEE 1-3	quatuor à cordes
BEE 1-4-1	BEE 1-4	quatuor à cordes

FIGURE 11.8 – Structure finale de la base de donnée

II.2 Dépendances et redondances

Pour supprimer les redondances, on peut commencer par étudier les dépendances entre les différentes entrées. Par exemple, la date de naissance d'un compositeur ne dépend que du compositeur, et non de l'oeuvre. De même, la famille instrumentale ne dépend que de l'instrument, et non de l'oeuvre.

Définition 11.2.1 (Attribut)

Un attribut est l'une des caractéristiques des données à saisir. Il s'agit donc du titre des colonnes. Ainsi, dans notre exemple, on dispose des attributs COMPOSITEUR, NAISSANCE, MORT, PAYS, TITRE...

Définition 11.2.2 (Dépendance fonctionnelle)

On dit qu'il existe une dépendance fonctionnelle de l'attribut A vers l'attribut B si la valeur de l'attribut B ne dépend de rien d'autre que de la valeur de l'attribut A. Autrement dit, toutes les entrées possèdant la même valeur de l'attribut A possèdent également une même valeur de l'attribut A. Une telle dépendance sera notée $A \longrightarrow B$.
On peut étendre la définition aux sous-ensembles d'attribut : il existe une dépendance fonctionnelle d'un sous-ensemble $\mathcal{A}$ d'attributs vers un sous-ensemble d'attributs $\mathcal{B}$ si les valeurs données aux différents attributs de $\mathcal{A}$ déterminent entièrement les valeurs des attributs de $\mathcal{B}$.

Cette définition dépend des entrées de la base : plus la base est fournie, plus la diversité des cas particuliers est susceptible de briser une dépendance fonctionnelle. Pour l'étude des dépendances fonctionnelles, on supposera la base remplie de toutes les entrées imaginables en rapport avec le thème de la base, afin de prendre en considération l'ensemble des cas possibles. Ainsi, dans notre exemple, nous avons les dépendances fonctionnelles suivantes :

- IDOEUVRE $\longrightarrow$ NOM, PRÉNOM, MORT, NAISSANCE, PAYS, DATE, TYPE
- IDCOMP $\longrightarrow$ MORT, NAISSANCE, NOM, PRÉNOM
- INSTRUMENT $\longrightarrow$ FAMILLE
- l'oeuvre détermine un sous-ensemble de formations instrumentales, mais pas une formation instrumentale. Identifier séparément tous les couples (oeuvre/formation instrumentale) permet de traduire cela par la dépendance :
  IDINSTR $\longrightarrow$ OEUVRE, FORMATION
- IDSOL $\longrightarrow$ OEUVRE, INSTRUMENT de la même manière.
- IDNAT $\longrightarrow$ PAYS, COMPOSITEUR
- En revanche, on n'a pas de dépendance fonctionnelle de NOM vers PRÉNOM, comme le montre l'exemple de la famille Bach, ni entre NOM, PRÉNOM et PAYS, comme le montre l'exemple de Haendel.
- On peut aussi s'interroger sur la dépendance NOM, PRÉNOM $\longrightarrow$ IDCOMP, qui n'est valable que s'il n'y a pas d'homonyme. L'exemple de Johann Strauss montre que ce point est discutable.
- Quant au TITRE d'une oeuvre, il ne détermine pas grand chose : nombreux sont les compositeurs ayant composé une symphonie n° 5. On ne peut même pas en déduire la formation instrumentale (pensez aux symphonies pour orgue de Widor par exemple).

Remarquez que si une famille $\mathcal{A}$ détermine un attribut B, mais aucun sous-ensemble strict de $\mathcal{A}$, on peut toujours se ramener au cas où B est déterminé par un unique attribut, en créant un nouvel attribut A, identifiant de l'ensemble des données fournies par $\mathcal{A}$. C'est ce qu'on a fait en introduisant l'indentifiant IDCOMP, permettant d'exprimer de façon plus élémentaire les dépendances induites par le groupe NOM, PRÉNOM.

On a de façon évidente :

Proposition 11.2.3 (Dépendances composées)
Si $\mathcal{A} \to \mathcal{B}$, $\mathcal{B}' \subset \mathcal{B}$ et $\mathcal{B}' \to \mathcal{C}$, alors $\mathcal{A} \to \mathcal{C}$.

Ainsi, la dépendance fonctionnelle IDOEUVRE $\longrightarrow$ NOM peut se retrouver en composant les dépendances IDOEUVRE $\longrightarrow$ IDCOMP et IDCOMP $\longrightarrow$ NOM.

Définition 11.2.4 (Dépendances élémentaires)
Une dépendance $A \longrightarrow B$ est élémentaire, si elle ne s'écrit pas comme composition non triviale de dépendances.

Ainsi, dans l'exemple ci-dessus, IDOEUVRE $\longrightarrow$ IDCOMP est une dépendance élémentaire, mais pas IDOEUVRE $\longrightarrow$ NOM.

Définition 11.2.5 (Graphe ADF)
Le graphe ADF (graphe des attributs et des dépendances fonctionnelles) est le graphe dont les sommets sont les attributs et les arêtes orientées indiquent les dépendances élémentaires.

Ainsi, dans notre exemple, on obtient le graphe de la figure 11.9.

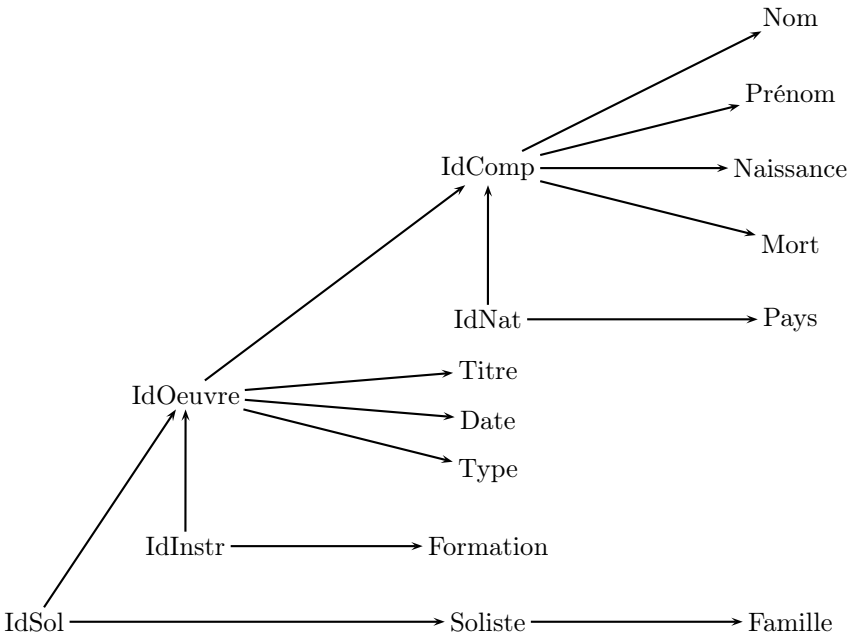


FIGURE 11.9 – Graphe ADF

Remarque 11.2.6

- Un graphe ADF linéaire $A_1 \longrightarrow A_2 \longrightarrow \cdots \longrightarrow A_n$ est telle que la donnée de A_1 détermine successivement les données de $A_2, \cdots, A_n$. Cela ne signifie pas que les entrées ne sont pas redondantes. En effet, suivant les valeurs de A_1, A_2 peut prendre plusieurs fois la même valeur, et les attributs suivants sont alors tous répétés.
- Un graphe ADF arborescent orienté de hauteur 1 (donc constitué uniquement de la racine A et de fils directs $B_1, \ldots, B_n$, ces derniers n'ayant pas de fils) est tel que A_1 détermine $B_1, \ldots, B_n$ indépendamment. Le tableau dont les entrées sont les attributs $B_1, \ldots, B_n$ est donc sans redondance.

Méthode 11.2.7 (Trouver la structure de la BDD à partir du graphe ADF)

Nous supposons ici que le graphe (non orienté) ADF est arborescent (connexe, sans cycle), non réduit à un sommet. Un sommet duquel part au moins une flèche est appelé noeud, sinon il est appelé feuille. On appelle fils d'un noeud A tout sommet accessible de A en une étape, en suivant une arête orientée convenablement. Toute feuille est fils d'au moins un noeud.

- La remarque précédente nous permet de retrouver assez rapidement la structure finale de notre base de donnée : à chaque noeud A va correspondre une table T_A, dont les attributs seront ce noeud et ses fils (mais pas les descendants suivants). Ainsi chaque sous-graphe ADF de ces tables sera un graphe de hauteur 1, donc sera non redondant.
- Chacune de ces tables T_A aura un identifiant, c'est à dire une colonne dont l'entrée déterminera toutes les autres. Cet identifiant est la colonne correspondant à la racine du sous-graphe ADF (donc au noeud considéré lors de sa construction)

Remarque 11.2.8

- Il y a des contraintes imposées sur la compatibilité des tableaux. Ainsi, si le fils B de A est lui-même un noeud, l'attribut B apparaît à la fois dans T_A et dans T_B, et sert d'identifiant à la table T_B, mais pas à la table T_A : une entrée possible pour B peut apparaître plusieurs fois ou pas du tout dans T_A ; mais si elle apparaît au moins une fois, elle doit être présente aussi dans T_B (la table T_B est à voir comme une définition et précision de toutes les valeurs de l'attribut B qu'on peut trouver dans les autres tables). Pour un attribut C d'une table T, notons $T(C)$ l'ensemble des valeurs prises par l'attribut C dans la table T. La contrainte ci-dessus s'exprime de la façon suivante : $T_A(B) \subset T_B(B)$ (contrainte d'inclusion). De plus, on a une contrainte d'unicité pour la colonne B de T_B, formalisant la notion d'identifiant : chaque valeur possible de B dans ce tableau n'apparaît qu'une fois.
- Ce sont ces contraintes qui définissent la notion de clé : l'attribut B de T_B constitue une clé primaire (un identifiant auquel on peut se référer dans une autre table, avec contrainte d'unicité). L'attribut B de T_A constitue une clé étrangère (il est accompagné d'une référence à une clé primaire, donc à un attribut d'une autre table, avec contrainte d'inclusion).

Remarque 11.2.9

La structure ADF n'est pas nécessairement arborescente. Pouvez-vous imaginer un exemple de structure contenant une boucle ?

III Algèbre relationnelle

III.1 Schéma relationnel et relation

Nous formalisons maintenant le concept de base de donnée.

Définition 11.3.1 (Attribut)

Un *attribut* d'une base de donnée est une des caractéristiques dont on souhaite enregistrer la valeur, lors d'une entrée. Ainsi, il s'agit du nom des colonnes des différentes tables ci-dessus. On note $\mathcal{A}$ l'ensemble des attributs.

Exemple 11.3.2

Dans l'exemple ci-dessus, les attributs sont NOM, PRÉNOM, NAISSANCE, MORT, TITRE etc.

Définition 11.3.3 (Domaine d'un attribut)

Le domaine d'un attribut est l'ensemble des valeurs possibles pour cet attribut. Le domaine de l'attribut A sera noté $\mathrm{dom}(A)$.

Exemple 11.3.4

- N'ayant aucune restriction autre que l'utilisation de l'alphabet pour désigner les noms des compositeurs, en notant L l'alphabet latin (avec les lettres accentuées, l'apostrophe et l'espace) donc l'ensemble de toutes les lettres, et L^* le monoïde libre engendré par L (donc tous les mots qu'on peut former avec cet alphabet), on a $\mathrm{dom}(\mathrm{NOM}) = L^*$
- On a $\mathrm{dom}(\mathrm{NAISSANCE}) = \mathbb{Z}$ (même s'il est assez peu probable que notre base contienne des compositeurs nés avant l'an 1).
- On peut restreindre certains domaines : par exemple dom(TYPE) peut être restreint à un certain nombre de types, par exemple :
  dom(TYPE) = { opéra, symphonie, sonate, trio, quatuor, messe, etc. },
  en catégorisant les types d'oeuvres tels qu'on le souhaite.
- De même, dom(INSTRUMENT) ou dom(FAMILLE) peuvent être définis comme des ensembles finis déterminés (à condition de lister tous les instruments, et toutes les familles d'instrument).
- Dans la pratique, on définit souvent un domaine en imposant un format particulier pour les entrées (une chaîne de caractères de longueur inférieure à 10 par exemple, ou une chaîne de longueur fixée, constituée de chiffres ou lettres en positions spécifiées...)

Définition 11.3.5 (Schéma relationnel)

Un schéma relationnel (ou schéma de relation) est une application $f : \mathcal{A} \longrightarrow \mathcal{D}$, où $\mathcal{A}$ est un ensemble d'attributs, $\mathcal{D}$ est un ensemble de domaines (donc un ensemble d'ensembles). La fonction f est la fonction associant à A son domaine. Ainsi, $f(A) = \mathrm{dom}(A)$.

Par exemple, en notant M l'ensemble de tous les caractères, on peut prendre pour $\mathcal{D}$ l'ensemble $\mathcal{P}(M^*)$, mais cela ne donne aucune contrainte sur le format des entrées. On peut imposer que les entrées soient des chaînes de caractères alphabétiques quelconques, ou des chaînes de caractères de longueur bornée, ou imposée, ou des mots d'une certaine langue, ou que ce soient des entiers, ou des dates sous un certain format (JJ/MM/AAAA), ou des éléments d'un ensemble fini etc.

Si $\mathcal{A} = \{A_1, \ldots, A_n\}$, on parlera simplement du schéma relationnel $\mathcal{S} = (A_1, \ldots, A_n)$, la donnée des domaines $\mathrm{dom}(A_i)$ étant implicite. Avec cette convention, les attributs du schéma relationnel sont ordonnés.

Définition 11.3.6 (Relation, valeur, enregistrement)

1. Une relation $\mathcal{R}$ (ou $\mathcal{R}(\mathcal{S})$) associée à un schéma relationnel $\mathcal{S} = (A_1, \ldots, A_n)$ est un ensemble fini de n-uplets de $\mathrm{dom}(A_1) \times \cdots \times \mathrm{dom}(A_n)$.
2. Les éléments de $\mathcal{R}$ sont appelés valeurs (ou enregistrements) de la relation.
3. $|\mathcal{R}|$ est appelé cardinal de la relation, et est généralement noté $\sharp R$.

On représente souvent une relation sous forme d'une table, comme nous l'avons fait précédemment. Ainsi, le schéma relationnel définit les noms des colonnes d'une table, et le type des entrées de chaque colonne, alors que la relation est l'ensemble des données entrées dans la table. Un enregistrement (ou une valeur) correspond à une ligne de la table. Le cardinal de la relation est le nombre de lignes dans la table.

III.2 Clés

Comme nous l'avons constaté sur des exemples, en général, les données stockées dans une base de donnée vont être réparties dans plusieurs tables. Ainsi, une base de donnée sera définie par plusieurs schémas relationnels et plusieurs tables. Certaines colonnes d'une table renvoient à des colonnes d'autres tables. Ce sont ces références qui permettront ensuite de faire des recherches croisées sur les différentes tables simultanément. La formalisation de ces références est faite par la notion de clé, intimement liée à celle d'identifiant.

Définition 11.3.7 (Identifiant)

Un identifiant d'une table est un ensemble d'attributs (pouvant être réduit à un unique attribut) tel que les valeurs prises par ces attributs déterminent toutes les autres valeurs de l'enregistrement (donc déterminent toute la ligne).

Ainsi, la donnée des valeurs prises sur les attributs constituant l'identifiant détermine sans ambiguïté (donc *identifient*) la ligne à laquelle on fait référence : 2 lignes différentes ont des valeurs différentes de ces attributs. Il s'agit donc d'une contrainte d'unicité : toute valeur de l'identifiant ne peut apparaître qu'une fois dans la table (si elle apparaissait deux fois, ces deux occurrences détermineraient la même ligne, ce qui n'est pas possible, les entrées d'une table (la relation) étant définies par une structure d'ensemble, donc sans répétition).

Remarque 11.3.8

1. Un identifiant peut être constitué de plusieurs colonnes. Par exemple, dans la table OEUVRE, l'ensemble { IDOEUVRE, COMPOSITEUR, TITRE } constitue un identifiant, mais cet identifiant n'est pas minimal.
2. Par définition, { IDOEUVRE } est encore une identifiant. Si le singleton $\{A\}$ est un identifiant, on dira simplement que A est un identifiant de la table.
3. Par définition, une relation est un ensemble d'enregistrements : il ne peut donc pas y avoir d'enregistrements multiples (plusieurs fois la même ligne). Ainsi, toute table possède au moins un identifiant qui est l'ensemble de tous ses attributs.
4. Tout ensemble d'attribut $\mathcal{I}$ tel qu'il existe $\mathcal{I}' \subset \mathcal{I}$ tel que $\mathcal{I}'$ soit un identifiant est lui aussi un identifiant. Une table possède donc en général plusieurs identifiants.

Définition 11.3.9 (Identifiant minimal)

Un identifiant est minimal dès lors qu'ôter un attribut de cet identifiant supprime le caractère identifiant.

Exemple 11.3.10

- { IDOEUVRE, COMPOSITEUR, OEUVRE } n'est pas un identifiant minimal
- IDOEUVRE est un identifiant minimal
- { COMPOSITEUR, OEUVRE } est aussi un identifiant minimal. En effet COMPOSITEUR n'est pas un identifiant (la plupart des compositeurs ont écrit plus d'une oeuvre, même ceux dont la notoriété est basée sur une unique oeuvre). OEUVRE n'est pas non plus un identifiant, puisque plusieurs compositeurs peuvent avoir composé une symphonie n° 5.
- Ainsi, il peut y avoir plusieurs identifiants minimaux. Certains identifiants minimaux peuvent être contitués de plusieurs attributs. Il n'existe pas toujours d'identifiant minimal constitué d'un unique attribut, mais on peut toujours s'arranger pour que ce soit le cas, quitte à rajouter un attribut (c'est ce que nous avons fait en ajoutant l'attribut IDOEUVRE).

Remarque 11.3.11

La notion d'identifiant prend tout son sens lorsque la table est suffisamment remplie et contient tous les cas de figure possibles. En effet, si la table OEUVRE n'est pas plus remplie que dans l'exemple donnée, l'attribut OEUVRE constitue également un identifiant, mais évidemment, cela n'a pas beaucoup de pertinence puisqu'on voit facilement que le caractère identifiant a de fortes chances d'être brisé en grossissant la base de donnée (ce qui est la vocation d'une BDD !) Ainsi, la recherche d'un identifiant acceptable ne se fait pas simplement de façon automatique en regardant les données déjà entrées, mais en considérant tous les cas de figures envisageables, même s'ils ne se sont pas encore présentés dans la base.

Remarque 11.3.12

On peut toujours se ramener à la situation où la table possède au moins un identifiant constitué d'un unique attribut. Il suffit pour cela de partir d'un identifiant (constitué de plusieurs attributs), d'ajouter un nouvel attribut ID, et de numéroter de façon artificielle les différentes valeurs possibles prises par les attributs identifiants. C'est ce que nous avons fait en introduisant IDOEUVRE ou IDINSTR ou IDSOL. Pour les deux premiers, nous avons essayé de garder une numérotation significative, pour le dernier, c'est une numérotation qui ne possède en soi pas tellement de sens, et dont le seul intérêt est le caractère identifiant.

Par commodité, on supposera (sans perte de généralité) que toutes les tables que nous considérerons ont au moins un identifiant réduit à un attribut.

Définition 11.3.13 (Clé primaire – Primary Key)

Une clé primaire d'une relation $\mathcal{R}$ est le choix d'un attribut A_{PK} (primary key) identifiant la relation.

On pourrait définir une clé primaire comme un ensemble d'attributs identifiant, mais la remarque précédente nous permet de faire cette simplification sans perte de généralité.

Exemple 11.3.14

- De façon naturelle, on peut choisir les clés primaires suivantes :
  - IDOEUVRE pour la table OEUVRES
  - IDNAT pour la table NATIONALITÉ
  - IDCOMP pour la table COMPOSITEURS
  - IDSOL pour la table SOLISTES
  - IDINSTR pour la table INSTRUMENTATION
  - INSTRUMENT pour la table INSTRUMENT
- Certaines colonnes de ces tables renvoient alors à des identifiants : ainsi, IDCOMP de la table NATIONALITÉ renvoie à l'identifiant IDCOMP de la table compositeur.
- Cette référence peut se faire avec des noms de colonnes distincts, ce qui impose de définir explicitement la référence : par exemple la colonne COMPOSITEUR de la table OEUVRE renvoie à la colonne IDCOMP de la table COMPOSITEUR. Ceci nous amène à la notion de clé étrangère.

Définition 11.3.15 (Clé étrangère – Foreign Key)

Soit $\mathcal{R}(\mathcal{S})$ une relation. Une clé étrangère est la donnée d'un coupe $(A_{FK}, \mathcal{R}'(\mathcal{S}'))$ constitué d'un attribut A_{FK} de $\mathcal{S}$ et d'un schéma relationnel $\mathcal{S}'$ muni d'une clé primaire A'_{PK}, et telle que $\mathcal{R}(A_{FK}) \subset \mathcal{R}'(A'_{PK})$ (contrainte d'inclusion, ou contrainte référentielle), où $\mathcal{R}(A_{FK})$ désigne l'ensemble des valeurs prises par A_{FK} dans la table.

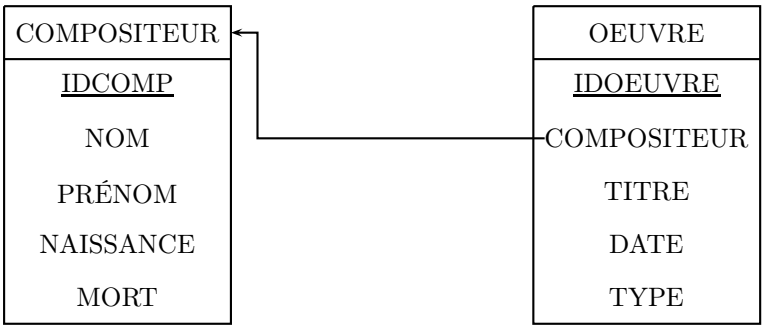
Ainsi, les entrées de la table $\mathcal{R}$ correspondant à l'attribut A_{FK} renvoient aux entrées de la table $\mathcal{R}'$ correspondant à l'attribut A'_{PK}, et toute valeur prise par A_{FK} doit aussi être prise par A'_{PK} (donc avoir été définies dans le tableau référent).

Ces notions peuvent être définies uniquement sur le schéma relationnel, en imposant alors des contraintes sur les entrées à venir de la base (contrainte d'unicité pour la clé primaire, contrainte référentielle pour la clé étrangère) : le SGBD peut refuser une entrée dans la BDD si elle ne respecte par ces contraintes, afin de préserver l'intégrité des clés.

On pourra donc se contenter de représenter ces notions sur les schémas relationnels. Un schéma relationnel sera représenté par un tableau listant ses attributs, par exemple :

COMPOSITEUR
IDCOMP
NOM
PRÉNOM
NAISSANCE
MORT

Dans ce tableau, la clé primaire sera indiquée en la soulignant, comme on l'a fait ci-dessus. Une clé étangère entre un attribut d'une table et une autre table sera représentée en reliant par une flèche l'attribut correspondant et la table référente :



Dans cette représentation, il n'y a pas d'ambiguïté pour savoir à quelle colonne du premier tableau réfère la flèche (la référence se fait vers l'identifiant déclaré, donc la clé primaire).

III.3 Schéma de base de donnée

Définition 11.3.16 (Schéma de BDD)

Un schéma de BDD est un ensemble

$$E = \{\mathcal{T} = (\mathcal{S}, A_{PK}, \{(A_{FK,i}; \mathcal{S}_i), i \in I\})\}.$$

dont les éléments sont des triplets formés de :

- un schéma relationnel $\mathcal{S}$
- une clé primaire A_{PK} associée au schéma relationnel $\mathcal{S}$
- un ensemble de clés étrangères $(A_{FK,i}; \mathcal{S}_i)$ d'attributs de $\mathcal{S}$ vers des schémas relationnels $\mathcal{S}_i$, tels qu'il existe $\mathcal{T}_i$ dans E dont la première coordonnée est $\mathcal{S}_i$.

Il s'agit donc de la donnée de la structure même d'une base de donnée. Chaque triplet élément de E est la donnée d'une table de la base de donnée, de son identifiant (donc on impose que chaque table ait un identifiant), et des références éventuelles vers une ou plusieurs autres tables de la base (par la donnée des clés étrangères).

On peut facilement étendre cette définition au cas où les clés sont données par des ensembles d'attributs plutôt qu'un attribut unique.

Définition 11.3.17 (Base de donnée)

Soit E un schéma de BDD. Une BDD de schéma E est un ensemble $\{\mathcal{R}_\mathcal{T}, \mathcal{T} \in E\}$ de relations (tables), chaque $\mathcal{R}_T$ étant une relation associée au schéma relationnel $\mathcal{T}$, ces relations vérifiant les contraintes d'unicité et les contraintes référentielles liées aux clés primaires et étrangères définies par E.

On verra dans le prochain chapitre comment effectuer des opérations sur une telle structure. Toutes les requêtes (recherches) peuvent se faire à partir de combinaisons de contructions algébriques simples (algèbre relationnelle). Nous étudierons ces requêtes au travers du langage SQL (Structured Query Language), actuellement prédominant dans ce domaine. La force de cette approche des bases de donnée, et de ce langage, vient du fait que les opérations de l'algèbre relationnelle peuvent alors toutes s'écrire avec une seule instruction : SELECT.

IV Exercices

Exercice 15

Décrire sous forme d'un diagramme avec des flèches la structure complète du schéma de BDD défini par l'exemple développé tout au long de ce chapitre.

Exercice 16

Imaginer comment enrichir la structure de cette base de donnée en y incluant la possibilité de répertorier les enregistrements que l'on possède (il faut donc inclure la diversité des interprétations, et des supports musicaux, et la possibilité d'avoir plusieurs oeuvres sur un même support)

Exercice 17

Définir un schéma de BDD dont le but est de stocker les informations suivantes relatives aux classes préparatoires scientifiques au lycée : élèves et renseignements divers sur les élèves, professeurs et renseignements divers, matières, types de classe (PCSI, MPSI...), classes.

Exercice 18

Enrichir l'exemple de l'exercice précédent en y incluant l'emploi du temps et les salles (et leur capacité, ou d'autres caractéristiques)...

Exercice 19

Comment gérer un hôtel composé de 3 bâtiments, et de chambres de capacités diverses et prix divers ?