

# BASES DE DONNÉES RELATIONNELLES

---

Flight reservation systems decide whether or not you exist. If your information isn't in their database, then you simply don't get to go anywhere.

Arthur Miller

## À la fin de ce chapitre, je sais :

- ✎ faire la différence entre une base de données et un SGBD
- ✎ modéliser des entités relatives à une réalité simple
- ✎ modéliser une association simple entre deux entités en précisant les cardinalités
- ✎ traduire une association dans un modèle relationnel
- ✎ interpréter un modèle relationnel de base de données

## A Pourquoi?

L'ubiquité de l'information n'est plus une vue de l'esprit aujourd'hui. La vie contemporaine n'est parfois qu'une suite ininterrompue de sollicitations de systèmes d'information qui ne cessent de générer des opérations sur des bases de données via des systèmes de gestion de bases de données. Par exemple :

- prendre un train, un bus, un avion, un bateau, une autoroute,
- louer un vélo, une voiture, une place de stationnement,
- réserver un hôtel, un article d'un magasin, une activité sportive ou culturelle, recevoir un colis,
- consulter un médecin, acheter des médicaments en pharmacie, faire des analyses médicales,
- payer ses impôts, demander une aide à la CAF, chercher un texte légal en vigueur dans le journal officiel,
- prendre une photo ou une vidéo sur son smartphone, envoyer un message à un proche, partager des informations via une application,

- jouer à un jeu en ligne, démarcher des clients ou chercher des collaborateurs via un réseau social...

Ces systèmes interagissent<sup>1</sup> entre eux à la suite de nos actions, souvent à l'insu même de l'instigateur. La plupart des requêtes sont tellement rapides qu'elles sont imperceptibles à l'être humain, à moins d'analyser finement ce que l'on est en train de faire et l'information que l'on manipule. Nous produisons collectivement et à travers eux des quantités inimaginables d'information.

■ **Exemple 1 — Quantité d'information sur les serveurs à l'échelle mondiale.** À l'échelle mondiale, on estime actuellement la quantité d'information créée et sauvegardée annuellement<sup>a</sup> à une centaine de Zio (zebioctets), soit  $100 \times 2^{70}$  octets. Pour se donner une idée, si on estime qu'une machine personnelle possède en standard un disque dur de 128 Gio, cela représente 858993459200 disques durs, soit environ 858 milliards. Nous sommes pour l'instant moins de 8 milliards d'habitants sur Terre... Gageons que ces données créées, sauvegardées, diffusées et recopiées sont toujours utiles!?

a. Ces données ne tiennent pas compte des machines personnelles (ordinateurs ou smartphones) ni des objets connectés.

 **Vocabulary 1 — bit**  $\longleftrightarrow$  bit Unité binaire de l'information. Une information est pour l'instant représentée en machine par un ensemble de symboles binaires généralement notées 0 et 1.

 **Vocabulary 2 — Byte**  $\longleftrightarrow$  octet. Un octet est constitué de huit bits.

Les bases de données étaient déjà au cœur des systèmes d'information, même avant l'ère informatique. Elles sont présentes aujourd'hui dans les systèmes d'exploitation, les applications des smartphones et même sur les objets connectés.

Ce chapitre traite de la modélisation de l'information pour les bases de données. Car, avant de pouvoir manipuler concrètement l'information de manière cohérente, il faut dégager un modèle qui décrit la réalité qu'on est en train de manipuler pour lui donner un sens. C'est l'enjeu de la modélisation conceptuelle et logique et du modèle relationnel associé.

## B Données et gestion des données

■ **Définition 1 — Base de données.** Une base de données est un ensemble de données enregistré représentant une partie du monde ou une activité humaine.

Il est de l'intérêt du concepteur et de l'utilisateur de limiter le périmètre d'une base de données. Par exemple, on s'intéressera aux documents d'une médiathèque, aux pneumatiques distribués en France, aux légumes cultivables en agriculture biologique ou aux dossiers scolaires des étudiants.

1. Les canadiens parlent d'infonuagique pour l'informatique dans le cloud.

Une base de données est une idée que l'on peut concrétiser en écrivant sur des cahiers ou en programmant des ordinateurs selon ce que l'on veut en faire.

■ **Définition 2 — Système de Gestion de Base de Données (SGBD).** Un système de gestion de base de données est un ensemble logiciel dont le but est de gérer une base de données stockée sur support informatique. Il s'agit de garantir l'accès, la lecture, l'écriture, l'interrogation ou la modification de la base de données d'une manière cohérente et efficace.

■ **Exemple 2 — SGBD.** Parmi les SGBD les plus utilisés, on peut citer Oracle, MySQL, PostgreSQL, SQLite ou Microsoft SQL Server.

L'interaction avec les SGBD contemporains a été standardisée et on peut aujourd'hui utiliser le langage SQL pour interagir avec la plupart des SGBD. Au-delà de la manipulation des données, un SGBD doit pouvoir veiller à la cohérence et à l'intégrité des informations tout en garantissant des transactions rapides.

■ **Définition 3 — Structured Query Language.** SQL est un langage informatique normalisé servant à interagir avec des bases de données relationnelles. Il permet de manipuler (rechercher, ajouter ou supprimer, modifier) des informations, mais aussi de créer et de contrôler l'intégrité des données et des transactions sur une base de données relationnelle.

## C De la conception à l'implémentation physique

La réalisation d'une base de données n'est pas une agrégation brute de données : c'est la réalisation d'une structure cohérente avec un contexte. Pour que les recherches s'effectuent simplement et efficacement, il est nécessaire de décrire, au-delà des données elles-mêmes, les liens sémantiques qui existent et qui relient les données entre elles, mais également la manière dont on stocke logiquement et physiquement les données ainsi que les transactions possibles.

La conception d'une base de données s'effectue la plupart du temps en trois étapes :

1. on établit d'abord un modèle **conceptuel** de la réalité à modéliser,
2. puis on choisit une représentation **logique** de ce modèle (généralement le modèle relationnel)
3. enfin, on implémente **physiquement** la base de données à l'aide d'un SGBD.

**R** Un modèle de base de données n'est pas unique. Selon le degré des exigences spécifiées par les utilisateurs de la base, différents modèles peuvent convenir. Les modèles proposés par la suite sont donc toujours perfectibles. C'est normal. En général en ingénierie, on peut parvenir à des solutions différentes pour résoudre un même problème. On parle alors d'équifinalité.

## a Modélisation conceptuelle

L'étape de modélisation conceptuelle s'affranchit de toute considération d'implémentation de la base de données pour se concentrer sur la **description sémantique de la réalité à modéliser**, c'est à dire le sens de cette réalité. Une réalité consiste le plus souvent en une activité humaine décrite simplement par un texte comme le montre l'exemple 6. L'analyse de ce texte et des mots le composant permet de dégager les éléments à modéliser. Cette modélisation est donc avant tout une activité du *langage* et une activité d'abstraction<sup>2</sup>. Elle s'appuie généralement sur le modèle entité-association ou le langage Unified Modeling Language (UML).

■ **Définition 4 — Entité ou classe.** Une entité est un concept qui représente une abstraction d'un ensemble de données que l'on peut regrouper et caractériser par des attributs communs

■ **Exemple 3 — La voiture, une entité simple.** Une voiture est une entité : elle représente les véhicules à quatre roues motorisés. Elle est caractérisée d'être d'une marque particulière et par le fait de posséder quatre roues, des portes, un moteur, un poste de pilotage.

■ **Exemple 4 — Un conducteur de voiture, une entité simple.** Un conducteur est une entité : elle représente la personne qui conduit un véhicule. Elle est caractérisée un nom et un numéro de permis.

■ **Définition 5 — Association.** Une association entre entités définit un ensemble de liens entre ces entités. Généralement, une association est représentée par un verbe. Elle est pondérée par des cardinalités qui précisent le nombre d'entités mises en jeu dans cette association. Une association peut réunir une, deux ou trois entités, on dit alors qu'elle est unaire, binaire ou ternaire.

■ **Exemple 5 — Conduire, une association simple.** Conduire est une association binaire : elle relie une entité conducteur à une entité véhicule. Selon la réalité à modéliser, on pourra pondérer cette association différemment. Par exemple, une personne peut ne conduire qu'une seule voiture. Mais certains conducteurs conduisent plusieurs voitures. De la même manière, une voiture peut être conduite par un ou plusieurs conducteurs selon la situation.

■ **Définition 6 — Modèle entité-association.** Le modèle entité-association cherche à représenter sur un même schéma des entités et des associations identifiables dans une réalité à modéliser.

■ **Exemple 6 — Une réalité à modéliser.** Afin d'illustrer le modèle entité-association, on a capturé une réalité à modéliser. Cette capture se fait la plupart du temps grâce à des entretiens avec les personnes impliquées dans le contexte. Voici le texte décrivant la réalité à modéliser :

2. Il n'y a pas que les mathématiques dans la vie, il y a le langage aussi!

*Une société de production recense les concerts donnés par des orchestres sur une saison. Chaque orchestre est décrit par son nom. Elle qualifie les orchestres en catégories selon leur style (baroque, symphonique, chambre, jazz, expérimental), sachant qu'un même orchestre ne peut faire partie que d'une seule catégorie. Un orchestre est dirigé par un chef d'orchestre dont on connaît le nom et le prénom. Au cours d'une même saison, un même orchestre peut accueillir plusieurs chefs différents. De la même manière, un chef peut diriger plusieurs orchestres. La résidence d'un orchestre définit son lieu de répétition. Une résidence est occupée par un seul orchestre mais un orchestre peut ne pas avoir de résidence. Elle est caractérisée par une ville et un pays.*

*On souhaite disposer d'une base de données pour accéder facilement à :*

- la liste des chefs d'un orchestre,
- la liste des pays de résidence des orchestres,
- la liste des orchestres de même style.

La réalité décrite dans l'exemple 6 peut être modélisée et représentée graphiquement comme sur la figure 1. L'intérêt de cette modélisation n'est pas uniquement son intelligibilité pour l'être humain : des outils<sup>3</sup> capables de transformer ce modèle graphique en modèle logique voire physique<sup>4</sup> existent et sont couramment utilisés.

Sur la figure 1, les cardinalités viennent quantifier le nombre d'entités qui participent à une association. Le premier nombre quantifie la cardinalité minimale de l'association, le second la cardinalité maximale.

■ **Définition 7 — Types d'association.** On définit plusieurs types d'association en fonction des cardinalités **maximales** qu'elles mettent en jeu :

1. de un à un,
2. de un à plusieurs,
3. de plusieurs à plusieurs.

■ **Exemple 7 — Interprétations des cardinalités des associations de l'exemple 6.** La notation 0 – 1 à droite de l'association *occupe* sur la figure 1 signifie qu'un orchestre occupe zéro ou une seule résidence, conformément à l'énoncé de l'exemple 6.

Le 1 – 1 au-dessous de l'association *occupe* signifie qu'une résidence est occupée au plus par un orchestre et au moins par un orchestre, c'est à dire exactement par un orchestre, conformément à l'énoncé.

Cette association *occupe* est de type **un à un**, car de chaque côté la cardinalité maximale vaut un.

La notation 0 – *n* au dessus de l'association *qualifie* sur la figure 1 signifie qu'un style qualifie zéro ou plusieurs orchestres.

Le 1 – 1 au-dessous de l'association *qualifie* signifie qu'un orchestre est qualifié exactement par un style.

3. des compilateurs

4. c'est à dire en base de données concrète et utilisable



FIGURE 1 – Un modèle conceptuel de type entité-association associé à la réalité de l'exemple 6. Les entités sont inscrites dans des rectangles, les attributs des entités dans des ovales et les associations entre les entités dans des losanges. Des cardinalités sont précisées sur les arcs qui relient les entités aux associations

Cette association **qualifie est de type un à plusieurs**, car la cardinalité maximale de part et d'autre de l'association vaut un et  $n$ .

La notation  $0 - n$  au dessus de l'association **dirige** sur la figure 1 signifie qu'un chef d'orchestre peut diriger zéro ou plusieurs orchestres. Le zéro laisse la possibilité à un chef d'être libre de tout engagement.

Le  $0 - n$  au dessous de l'association **dirige** signifie qu'un orchestre peut être dirigé par zéro ou plusieurs chefs. Le zéro laisse la possibilité à un orchestre de ne pas avoir de chef.

L'association **dirige est de type plusieurs à plusieurs**, car la cardinalité maximale de chaque côté de l'association vaut  $n$ .

**R** Pour comprendre les cardinalités sur la figure 1, une bonne maîtrise de la voix active et de la voix passive en français est nécessaire. On lit une association dans un sens à la voix active et dans l'autre sens à la voix passive. L'interprétation des cardinalités découle de cette lecture à double voix.

## b Modélisation logique

■ **Définition 8 — Modèle logique.** Un modèle logique s'élabore à partir d'un modèle conceptuel. Il explique comment sont structurées les données dans le logiciel de base de données. Il ne dit pas comment implémenter les structures mais spécifie :

- le type de structure qui enregistre les entités et les associations,
- le type de données des attributs et les valeurs possibles,
- les clefs qui identifient les entités,
- les contraintes référentielles liées à l'intégrité entre les éléments du modèle.

**R** Par essence, une base de données est partagée entre plusieurs utilisateurs<sup>a</sup>. Certains utilisateurs ont plus de droits que d'autres. Par exemple, ils peuvent avoir le droit d'insérer des nouveaux enregistrements. Il n'est pas rare de trouver dans les bases de données mal conçues :

- des doublons,
- des valeurs aberrantes,
- des valeurs incohérentes.

Pour remédier à ceci, on impose des contraintes sur la modèle logique de la base de données.

<sup>a</sup>. voire des millions...

■ **Définition 9 — Contrainte d'unicité.** Une contrainte d'unicité précise qu'un attribut ou un ensemble d'attributs d'une entité est unique dans la base de données. On ne peut donc plus insérer deux informations différentes dont ces attributs seraient identiques. Cela permet d'éviter l'insertion d'informations redondantes. Elle est réalisée grâce à une clef primaire.

■ **Définition 10 — Clef primaire.** Une clef primaire est un attribut ou ensemble d'attributs d'une entité qui permet de distinguer un représentant de cette entité d'un autre dans une base de données. C'est un identifiant.

■ **Exemple 8 — Identifier une voiture et un conducteur.** L'identifiant choisi pour une entité voiture est généralement sa plaque d'immatriculation qui est unique au moins à l'échelle d'un pays. Ainsi, on ne pourra pas confondre deux voitures. De même, pour un conducteur, le plus sûr est de choisir son numéro de permis de conduire. On pourrait imaginer utiliser la combinaison de son nom et de sa date de naissance. Cela peut malheureusement ne pas suffire dans certaines situations <sup>a</sup>.

<sup>a</sup>. Un conducteur nommé Jean Martin, étant donné qu'il porte le prénom et le nom les plus courants en France, peut avoir un homonyme né le même jour. Ce n'est pas exclu.

Si la contrainte d'unicité protège des doublons, elle ne protège pas des valeurs aberrantes ou incohérentes.

■ **Définition 11 — Contrainte référentielle.** Une contrainte référentielle spécifie qu'un attribut d'une entité faisant référence à une autre doit nécessairement référencer l'autre entité en question pour ne pas dupliquer les informations <sup>a</sup>. Ce référencement se fait via une clef étrangère.

<sup>a</sup>. et les rendre incohérente par la même occasion...

■ **Définition 12 — Clef étrangère.** Une clef étrangère est un attribut ou ensemble d'attributs d'une entité qui permet de faire référence à un représentant d'une autre entité dans une base de données. C'est, le plus souvent, la clef primaire d'une autre entité dans la base.

Construire un modèle logique d'une base de données, c'est donc se poser de nombreuses questions sur les données et ce qu'on peut ou doit en faire. Toutes les réponses formulées ont un impact sur la structure logique de la base de données.

Le modèle logique généralement utilisé dans le domaine des bases de données est le modèle relationnel qui est décrit à la section D.

## c Modélisation physique

■ **Définition 13 — Modèle physique.** Un modèle physique est l'implémentation d'un modèle logique dans un SGBD.

Pour implémenter ce modèle physique en mémoire, le SGBD crée alors :

- la base de données (sur un serveur ou dans un fichier),
- toutes les structures de données nécessaires pour enregistrer les données,
- les contraintes logicielles d'unicité et de référencement,
- des vues pour simplifier la visualisation des données,
- des index pour accélérer l'exécution des requêtes.



La réalisation d'une base de données physiquement est hors programme. Dans toutes les épreuves, on considère que la base de données a déjà été implémentée, elle est donc fournie.

## D Le modèle relationnel

Cette section décrit plus précisément le modèle relationnel qui est au programme de l'épreuve d'informatique commune. C'est à partir de ce modèle que l'on peut effectuer des requêtes sur des bases de données.

Le modèle relationnel repose sur l'algèbre relationnelle, de solides fondements mathématiques qui ne sont pas exposés ici car hors programme, mais qui justifient pleinement son adoption massive comme modèle logique de référence.

■ **Définition 14 — Modèle relationnel.** Le modèle relationnel est un modèle logique de base de données pour lequel :

- l'information est structurée en tableaux à deux dimensions : les relations.
- les entités décrites dans ces tableaux doivent l'être indépendamment des associations qui les relient.

Un exemple de modèle relationnel est donné sur la figure 2 : on y retrouve les définitions données ci-dessous.

■ **Définition 15 — Relation.** Une relation est tableau à deux dimensions constitué :

1. d'un entête qui contient la liste des noms des attributs associés à leur domaine,
2. d'un corps qui est un ensemble de lignes dont les éléments sont des valeurs qui appartiennent aux domaines des attributs correspondant dans l'entête.

Les attributs d'une relation relèvent tous d'une entité commune.

■ **Définition 16 — Ligne ou enregistrement.** Un enregistrement est une ligne dans le corps d'une relation. Un enregistrement représente une instance d'une entité. Généralement on parle de ligne au niveau logique et d'enregistrement au niveau physique.

Ⓡ Au niveau physique, on désigne une relation par le terme **table**, ce qui est légitime car son implémentation est un tableau. Les **lignes** d'une relation sont les enregistrements de la table. Les colonnes d'une table sont les attributs de la relation.

■ **Définition 17 — Domaine d'un attribut.** Le domaine d'un attribut est l'ensemble des valeurs qu'il peut prendre.

■ **Exemple 9 — Domaine d'un attribut *paiement*.** Si l'attribut d'une entité *paiement* est une monnaie, alors le domaine associé à cet attribut est l'ensemble de toutes monnaies possibles : euro, dollar, rouble, yen, livre sterling...



FIGURE 2 – Modèle relationnel construit à partir du modèle conceptuel 1. Seules les entêtes de colonne des relations sont représentées. Les associations du modèle conceptuel ont été intégrées aux tables en ajoutant des colonnes qui contiennent les clefs primaires et les clefs étrangères nécessaires ou en créant une nouvelle table (direction). Les flèches mettent en évidence ces associations.

**R** Certains SGBD n'utilisent le modèle relationnel, on parle de systèmes NoSQL. Ils sont souvent utilisés lorsque les modèles de données à gérer varient rapidement.

La représentation des entités du modèle conceptuel est naturel dans le modèle relationnel : une entité est représentée par une table. Par contre, représenter une association demande un travail supplémentaire.

## E Traduction des associations dans le modèle relationnel

Cette section s'attache à décrire comment on peut traduire les associations d'un modèle conceptuel dans un modèle relationnel.

**M** **Méthode 1 — Transformation d'une association de un à un** Une association de un à un se traduit par l'apparition d'une clef étrangère dans une des deux tables du modèle relationnel.

Par exemple, sur le modèle conceptuel de la figure 1, la relation *occupe* est une association de un à un. Concrètement, au niveau des tables du modèle relationnel, on peut ajouter une colonne à la table *résidence*, colonne de même type que la clef primaire de la table *orchestre*. Cette colonne de la table *résidence* devient une clef étrangère qui référence la table *orchestre*, signifiant ainsi qu'une résidence est occupée par un orchestre comme le montre la figure 3.

**R** Les colonnes ajoutées aux tables et représentant des clefs étrangères portent souvent le même nom que la clef primaire à laquelle elles se réfèrent. C'est une bonne pratique. Cependant, il est possible de leur donner des noms différents.

**R** Les cardinalités minimales sont généralement traduites dans le modèle relationnel par des contraintes qui n'apparaissent pas sur les schémas mais dans la manière de compléter les colonnes des clefs étrangères.

Pour une association dont la cardinalité est minimale est 1, on impose à la clef étrangère d'être définie : par exemple, pour le modèle de la figure 1, le fait qu'une résidence soit nécessairement occupée par un orchestre implique que le SGBD doit vérifier qu'à chaque nouvelle insertion d'une résidence, cette clef soit présente et référence un orchestre en particulier.

Pour une cardinalité minimale 0, lors de l'insertion d'un nouvel enregistrement, une clef étrangère peut ne pas être définie.



FIGURE 3 – Traduction d’une association de un à un dans un modèle relationnel. Modèle relationnel spécifiant les tables résidence et orchestre du modèle relationnel de la figure 2.

**M Méthode 2 — Transformation d'une association de un à plusieurs** Une association de un à plusieurs se traduit par l'apparition d'une clef étrangère dans la table dont la cardinalité maximale vaut un.

Par exemple, sur le modèle conceptuel de la figure 1, la relation *qualifie* est une association de un à plusieurs. Concrètement, au niveau des tables du modèle relationnel, on peut ajouter une colonne à la table *orchestre*, colonne de même type que la clef primaire de la table *style*. Cette colonne de la table *orchestre* devient une clef étrangère qui référence la table *style*, signifiant ainsi qu'un orchestre est qualifié par un style, comme le montre comme le montre la figure 4.



FIGURE 4 – Traduction d'une association de un à plusieurs dans un modèle relationnel. Modèle relationnel spécifiant les tables *orchestre* et *style* du modèle relationnel de la figure 2.

**M** **Méthode 3 — Transformation d'une association de plusieurs à plusieurs** Une association de plusieurs à plusieurs se traduit par la création d'une nouvelle table dans la base de données. On choisit généralement de nommer cette table d'après le nom de l'association. Cette nouvelle table contient deux clefs étrangères, chacune pointant vers une extrémité différente de l'association.

Par exemple, sur le modèle conceptuel de la figure 1, la relation *dirige* est une association de plusieurs à plusieurs. Concrètement, au niveau des tables du modèle relationnel, on ajoute une table nommée *direction*. Cette table comporte au moins deux colonnes, deux clefs étrangères : l'une pointe vers la clef primaire de la table *orchestre* et l'autre vers la clef primaire de la table *chef*. Une *direction* est le fait qu'au cours de la saison, un chef dirige un orchestre. La figure 5 montre comment transformer une relation de plusieurs à plusieurs dans le cas du modèle conceptuel de la figure 1.



FIGURE 5 – Traduction d’une association de plusieurs à plusieurs dans un modèle relationnel. Modèle relationnel spécifiant les tables orchestre, direction et chef qui implémente l’association *dirige* du modèle conceptuel de la figure 1.



FIGURE 6 – Modèle physique de la base de données du modèle relationnel 2.

## F Du modèle physique à la création des tables --> HORS PROGRAMME

Il existe donc trois types de modèle pour décrire les base de données selon que l'on se place au niveau :

1. conceptuel,
2. relationnel (logique),
3. ou physique.

Sur la figure 6 est schématisé le modèle physique associé au modèle relationnel de la figure 2 et au modèle conceptuel de la figure 1.

Sur ce modèle physique sont notamment précisés les types des attributs en essayant de minimiser la taille en mémoire de chaque attribut.

Le langage SQL permet la création de ce modèle physique. Le code 1 montre comment créer les tables concrètes associées au modèle physique de la figure 6 dans la base de données. En



plus des types des attributs y figurent également les contraintes de clefs primaires et étrangères.

#### Code 1 – Création du modèle physique d'une base de données

```
1  CREATE TABLE "chef"
2  (
3      "id_chef" INTEGER,
4      "nom"      VARCHAR(80),
5      "prenom"   VARCHAR(80),
6      PRIMARY KEY ("id_chef")
7  );
8
9  CREATE TABLE "style"
10 (
11     "nom" VARCHAR(80),
12     PRIMARY KEY ("nom")
13 );
14
15 CREATE TABLE "orchestre"
16 (
17     "id_orch" INTEGER,
18     "nom"      VARCHAR(80),
19     "id_style" VARCHAR(80),
20     PRIMARY KEY ("id_orch"),
21     FOREIGN KEY ("id_style") REFERENCES "style"
22 );
23
24 CREATE TABLE "residence"
25 (
26     "id_res" INTEGER,
27     "ville"   VARCHAR(80),
28     "pays"    VARCHAR(80),
29     "id_orch" INTEGER,
30     PRIMARY KEY (id_res),
31     FOREIGN KEY ("id_orch") REFERENCES orchestre (id_orch)
32 );
33
34
35 CREATE TABLE "direction"
36 (
37     "id_dir" INTEGER,
38     "id_chef" INTEGER,
39     "id_orch" INTEGER,
40     PRIMARY KEY ("id_dir"),
41     FOREIGN KEY ("id_chef") REFERENCES chef (id_chef),
42     FOREIGN KEY ("id_orch") REFERENCES orchestre (id_orch)
43 );
```

---