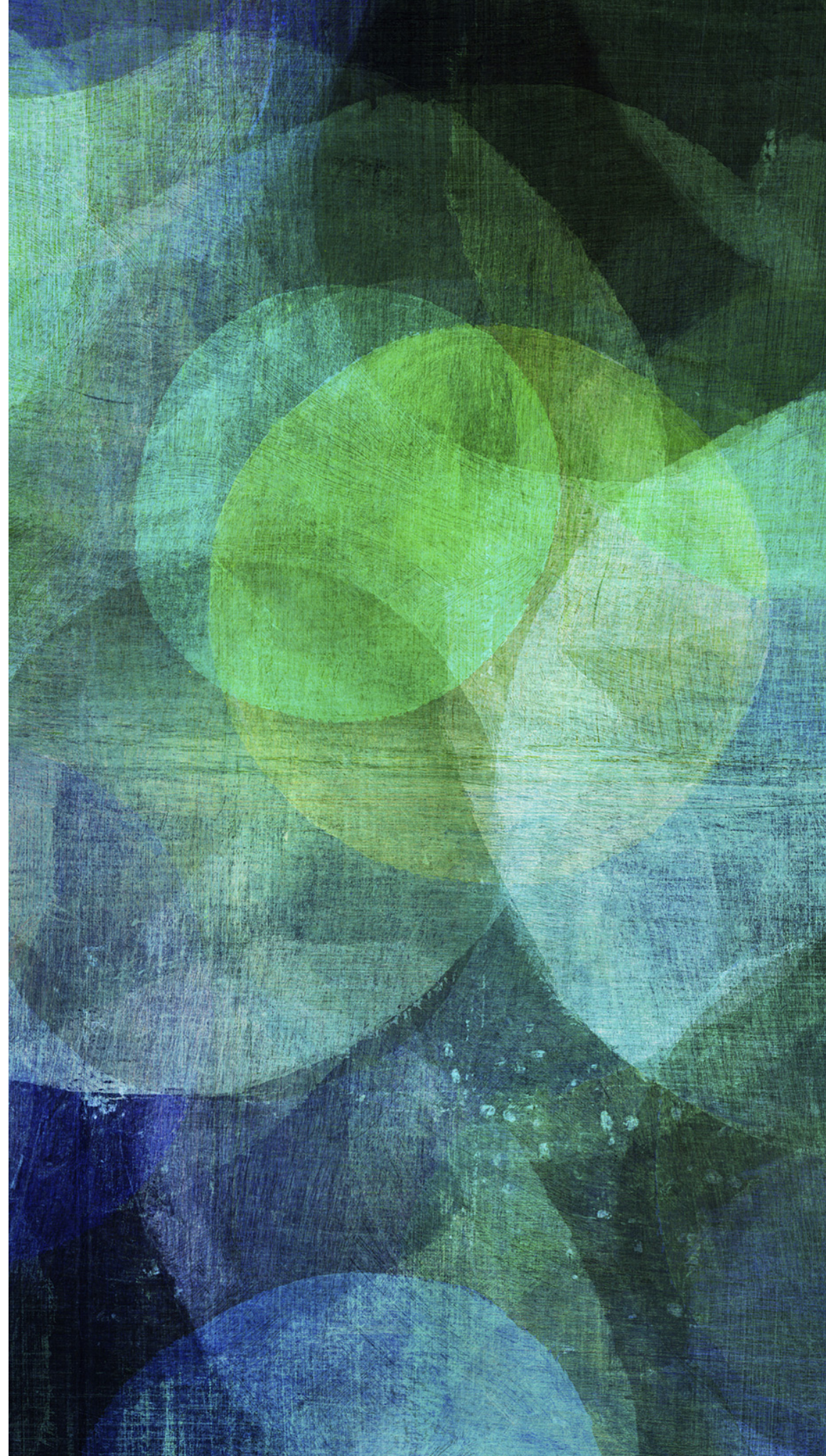


SQL



INTRODUCTION



QU'EST-CE QU'UNE « BASE DE DONNÉES » ?

- Une **base de données** permet de stocker et de retrouver des données structurées, semi-structurées ou des données brutes ou de l'information, souvent en rapport avec un thème ou une activité ; celles-ci peuvent être de natures différentes et plus ou moins reliées entre elles.

TYPES DE BASES DE DONNÉES

- Bases de données relationnelles
- Bases de données orientées « document » (NoSQL)
- Bases de données orientées « graphe »
- Bases de données « clef/valeur »

BASES DE DONNÉES « RELATIONNELLES »

- Une base de données dite relationnelle est un type de base de données fortement structuré, introduit par Edgar Codd en 1960.
- Elles possèdent deux grandes caractéristiques :
 - la notion de **schéma**, représentant le « domaine conceptuel » d'une application (p.e. les concepts nécessaires à la gestion d'une bibliothèque)
 - la notion de **calcul relationnel**, un ensemble d'opérations permettant d'extraire de la base de données les informations demandées

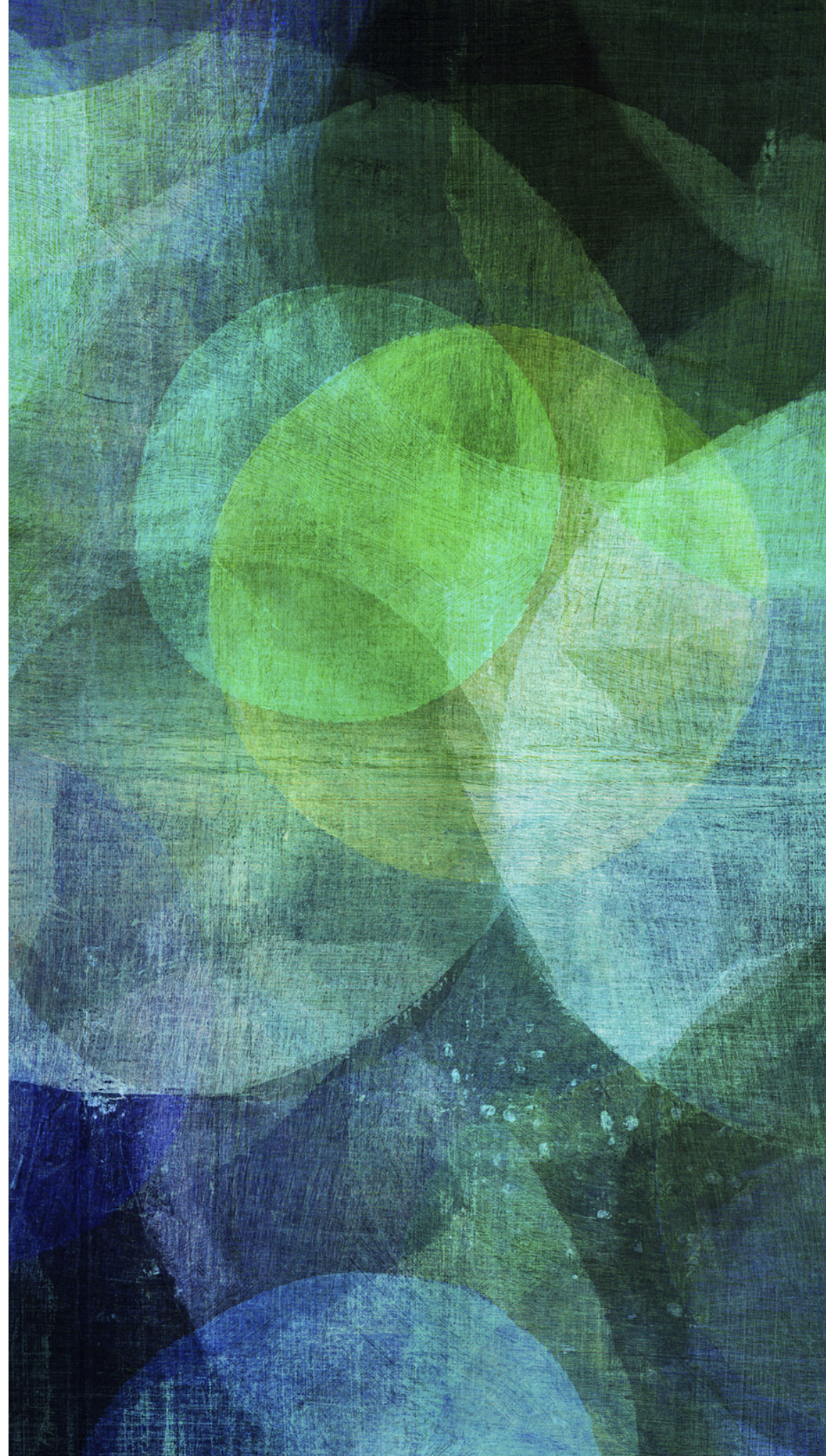
NOTION DE « RELATION »

- On appelle « relation » un ensemble d'attributs qui caractérisent une proposition ou une combinaison de propositions comme
- "un employé a un matricule, il a un nom, il a un employeur".
- Dans cet exemple, les attributs de l'employé sont : son matricule, son nom et son employeur. Chaque combinaison de propositions ainsi formée est appelée **tuple** (collection ordonnée).
- L'ensemble ("1245", "Jean Dupond", "Compagnie des belles lettres") constitue un tuple de la relation définissant « employé ».

VOCABULAIRE

- **Relation** : ensemble des attributs caractérisant un concept donné
- **Tuple** : ensemble des valeurs (pour une certaine relation) caractérisant un objet donné

INSTALLATION



INSTALLATION DE MAMP

URL de téléchargement : <https://www.mamp.info/en/>

LANCEMENT DE PHPMYADMIN

Une fois le serveur MAMP/XAMPP lancé, vous pouvez aller dans le navigateur de votre choix et ouvrir la page :

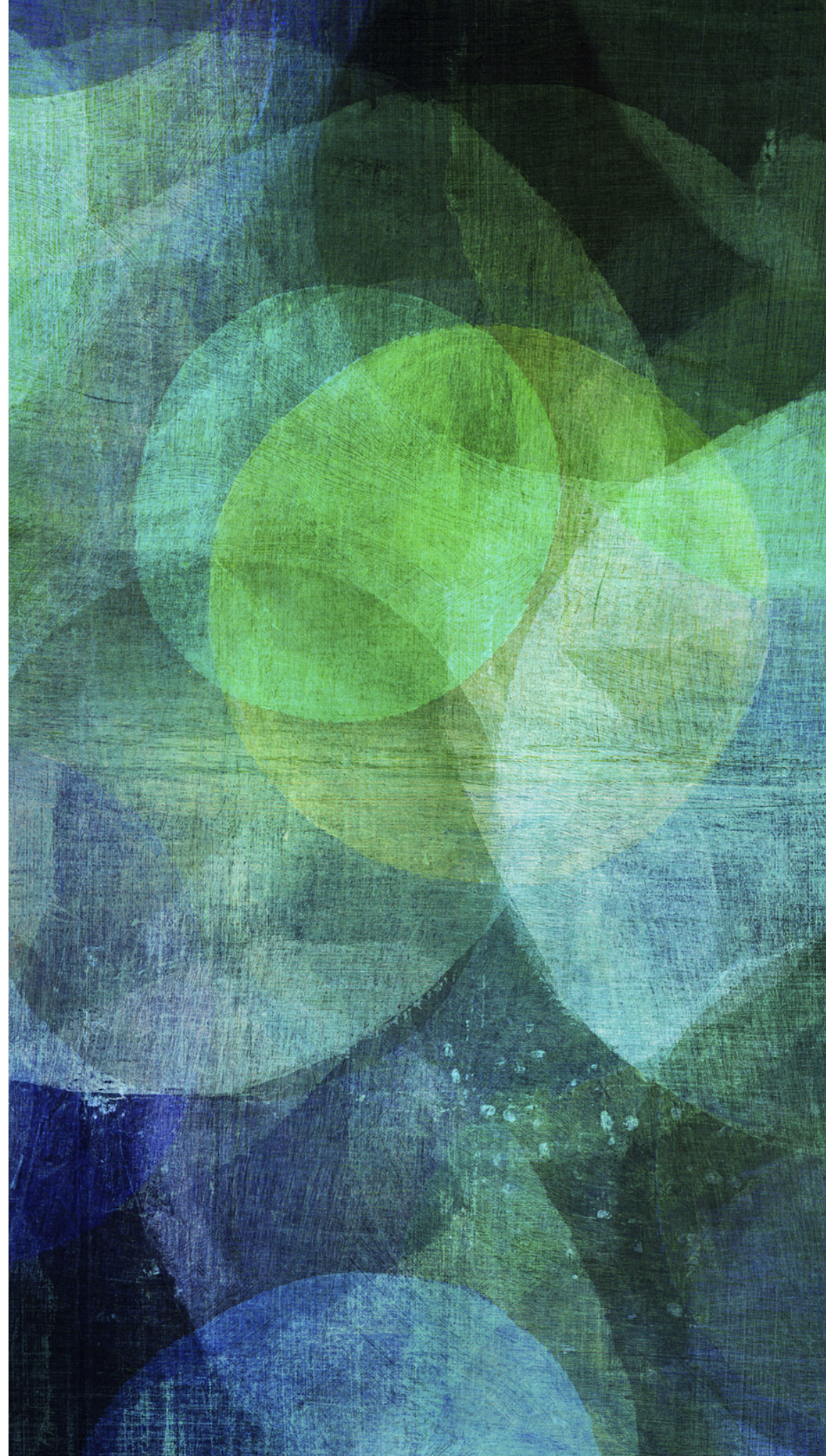
`http://localhost:<port>/phpmyadmin`

Le port dépend de la configuration du serveur. Par défaut, MySQL écoute sur le port 3306

INTERFACE

.....

STRUCTURE DE BASE DE DONNÉES



CRÉER UNE BASE DE DONNÉES

CREATE DATABASE bibliothèque

CRÉER UNE TABLE

```
CREATE TABLE livre (  
— description du concept de livre  
)
```

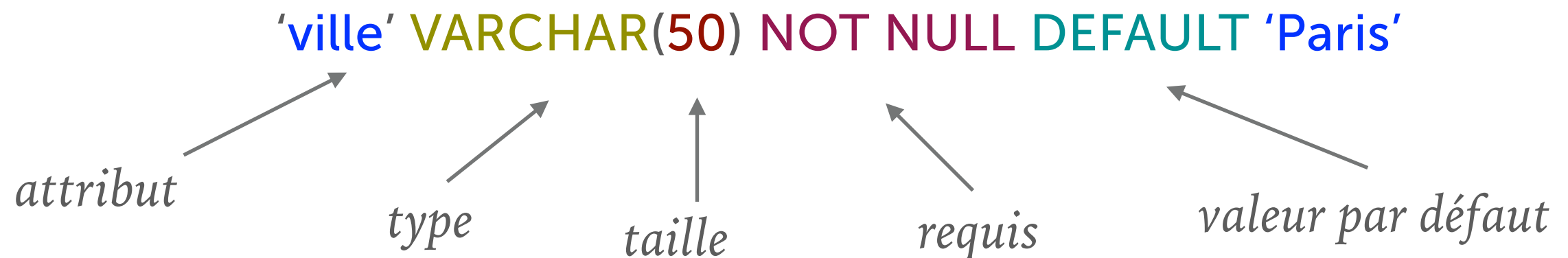

DÉFINITION D'UN ATTRIBUT

- Pour créer la table « livre » – équivalente à une relation – il faut définir l'ensemble de ses attributs
- Cela implique deux choses :
 - Faire la liste de tout ce qu'il est nécessaire de savoir sur les livres dans l'application utilisant la base de données
 - Pour chaque attribut, définir ses caractéristiques

Dans phpMyAdmin, ces caractéristiques peuvent être affichées dans l'onglet « Structure » de la table

LES ATTRIBUTS

- Un attribut (que l'on appelle souvent une colonne) se caractérise *a minima* par :
 - un type
 - une taille
 - une valeur par défaut
 - si une valeur est requise ou non



LES TYPES

- SQL définit un nombre limité (et non extensible) de types de données :
 - chaînes de caractères : VARCHAR, TEXT, LONGTEXT, etc.
 - nombres : INTEGER, FLOAT, DOUBLE, etc.
 - dates : DATE, TIME, DATETIME
 - géo-spatiales : GEOMETRY, POINT, POLYGON, etc.
 - composites : JSON
 - binaires : BLOB, LONGBLOB

CRÉER UNE TABLE : AJOUTER LES ATTRIBUTS

```
CREATE TABLE livre (  
  'cote' VARCHAR(6) NOT NULL,  
  'titre' VARCHAR(50) NOT NULL,  
  'isbn' VARCHAR(13) NOT NULL,  
  'resume' TEXT,  
  'genre' ENUM('roman', 'essai') NOT NULL,  
  'langue' INT(2) NOT NULL DEFAULT 1,  
)
```

*VARCHAR : chaîne limitée
à 255 caractères*

*TEXT : Texte long, passed
longueur spécifiée*

*ENUM : Valeurs à choisir
dans une liste*

INT : Entiers (ici limité à 2 chiffres)

CRÉER UNE TABLE : CLEF PRIMAIRE

```
CREATE TABLE livre (  
    'cote' VARCHAR(6) NOT NULL,  
    'titre' VARCHAR(50) NOT NULL,  
    'isbn' VARCHAR(13) NOT NULL,  
    'resume' TEXT,  
    'genre' ENUM('roman', 'essai') NOT NULL,  
    'langue' INT(2) NOT NULL DEFAULT 1,  
    PRIMARY KEY ('cote')  
)
```

PRIMARY KEY : Attribut (ou groupe d'attributs) dont la valeur détermine de manière unique un enregistrement (p.e. le numéro de Sécurité Sociale pour une personne)

CRÉER UNE TABLE : OPTIONS

```
CREATE TABLE livre (  
    'cote' VARCHAR(6) NOT NULL,  
    'titre' VARCHAR(50) NOT NULL,  
    'isbn' VARCHAR(13) NOT NULL,  
    'resume' TEXT,  
    'genre' ENUM('roman', 'essai') NOT NULL,  
    'langue' INT(2) NOT NULL DEFAULT 1,  
    PRIMARY KEY ('cote')  
)
```

ENGINE=InnoDB

ENGINE : Moteur d'exécution de la table

DEFAULT CHARSET=utf8

*DEFAULT CHARSET : Jeu de caractères
de la table*

SUPPRIMER DES ÉLÉMENTS DE BASE DE DONNÉES

Supprimer une base de données entière

DROP DATABASE bibliothèque

*Supprimer une table d'une base de données
entière (attention ! peut souvent amener
des problèmes logiques)*

DROP TABLE livre

AJOUTER UN ATTRIBUT À UNE TABLE

La commande ALTER TABLE permet de modifier la structure d'une table après sa création.

```
ALTER TABLE `document`
```

```
ADD `language` VARCHAR(5) NOT NULL DEFAULT 'fr-FR'
```

```
COMMENT 'Langue du document' AFTER `publisher`,
```

```
ADD INDEX (`language`)
```

SCHÉMA DE BASE DE DONNÉES

Le schéma d'une base de données relationnelle est constitué par l'ensemble des tables (munies de leurs attributs) et des associations entre ces tables

MODÈLE ENTITÉ/ASSOCIATION

Les modèles de données (modèles de domaine) sont très souvent décrits par un ensemble d'entités (les tables) et des associations entre ces entités.

Par exemple, pour une bibliothèque, on pourra définir les entités LIVRE et AUTEUR, ce qui se traduira par deux tables différentes dans la base de données. Il devient alors nécessaire de pouvoir spécifier que tel livre « a été écrit par » tel auteur. C'est le sens de l'association.

TYPES D'ASSOCIATIONS

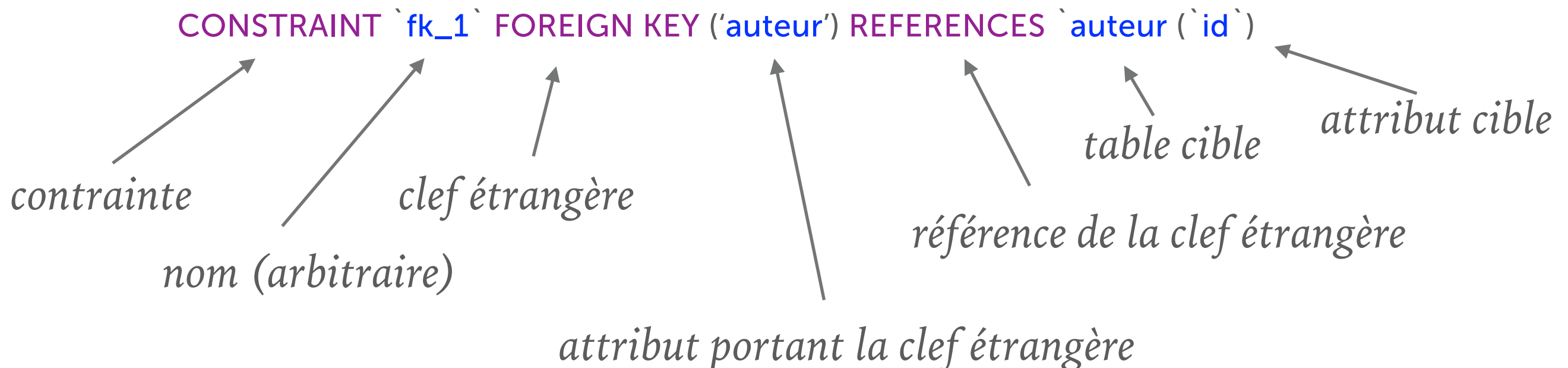
Globalement, il existe trois types d'associations :

- One-to-one (1-1) : Un objet d'une entité E1 est lié à un objet d'une entité E2 et réciproquement (p.e. un panier et un client), c'est l'association la moins fréquente
- Many-to-one (1-n) : Plusieurs objets de l'entité E1 ont liés à un objet de l'entité E2 (p.e. des achats effectués par un client)
- Many-to-many (n-n) : Plusieurs objets de l'entité E1 ont liés à plusieurs objets de l'entité E2 (p.e. dans les réseaux sociaux, une personne connaît plusieurs personnes, et est connue de plusieurs personnes)

ASSOCIATIONS ET CLEFS ÉTRANGÈRES

Les bases de données relationnelles implémentent des associations de deux manières différentes, basées toutes deux sur la notion de clef étrangère

Une clef étrangère est un attribut d'une table dont la valeur correspond à une valeur unique dans une autre table (généralement une clef primaire)



EXEMPLE

Nous voulons lier les entités LIVRE et AUTEUR, de manière à ce que chaque livre puisse se référer à l'auteur qui l'a écrit.

```
CREATE TABLE livre (
```

```
  'cote' VARCHAR(6) NOT NULL,
```

```
  'titre' VARCHAR(50) NOT NULL,
```

```
  'isbn' VARCHAR(13) NOT NULL,
```

```
  'resume' TEXT,
```

```
  'genre' ENUM('roman', 'essai') NOT NULL,
```

```
  'langue' INT(2) NOT NULL DEFAULT 1,
```

```
  'auteur' INT(11),
```

```
  PRIMARY KEY ('cote')
```

```
  CONSTRAINT `fk_1` FOREIGN KEY ('auteur') REFERENCES `auteur` (`id`)
```

```
)
```

```
ENGINE=InnoDB
```

```
DEFAULT CHARSET=utf8
```

```
CREATE TABLE auteur (
```

```
  'id' INT(11) NOT NULL AUTO_INCREMENT,
```

```
  'prenom' VARCHAR(50) NOT NULL,
```

```
  'nom' VARCHAR(13) NOT NULL,
```

```
  'biographie' TEXT,
```

```
  'naissance' DATETIME,
```

```
  'deces' DATETIME ,
```

```
  PRIMARY KEY ('id')
```

```
)
```

```
ENGINE=InnoDB
```

```
DEFAULT CHARSET=utf8
```

IMPLÉMENTATION DES ASSOCIATIONS MANY-TO-MANY

Les clefs étrangères, par définition ne peuvent représenter que des associations Many-to-one (la valeur cible étant unique).

Pour implémenter une association Many-to-many dans un modèle relationnel, on est obligé de la décomposer en deux associations Many-to-one (antagonistes) en ajoutant une table utilitaire nommée « table de jointure »

Une table de jointure ne contient que deux clefs étrangères.

EXEMPLE

*Nous voulons exprimer le fait qu'un livre peut avoir plusieurs auteurs
(en plus du fait que auteur a écrit plusieurs livres).*

```
CREATE TABLE livre (
```

```
  'cote' VARCHAR(6) NOT NULL,
```

```
  ... autres attributs
```

```
  PRIMARY KEY ('cote')
```

```
)
```

```
CREATE TABLE auteur (
```

```
  'id' INT(11) NOT NULL AUTO_INCREMENT,
```

```
  ... autres attributs
```

```
  PRIMARY KEY ('id')
```

```
)
```

```
CREATE TABLE a_écrit (
```

```
  'auteur_id' INT(11),
```

```
  'livre_id' VARCHAR(6),
```

```
  ... autres attributs
```

```
  PRIMARY KEY ('auteur_id', 'livre_id')
```

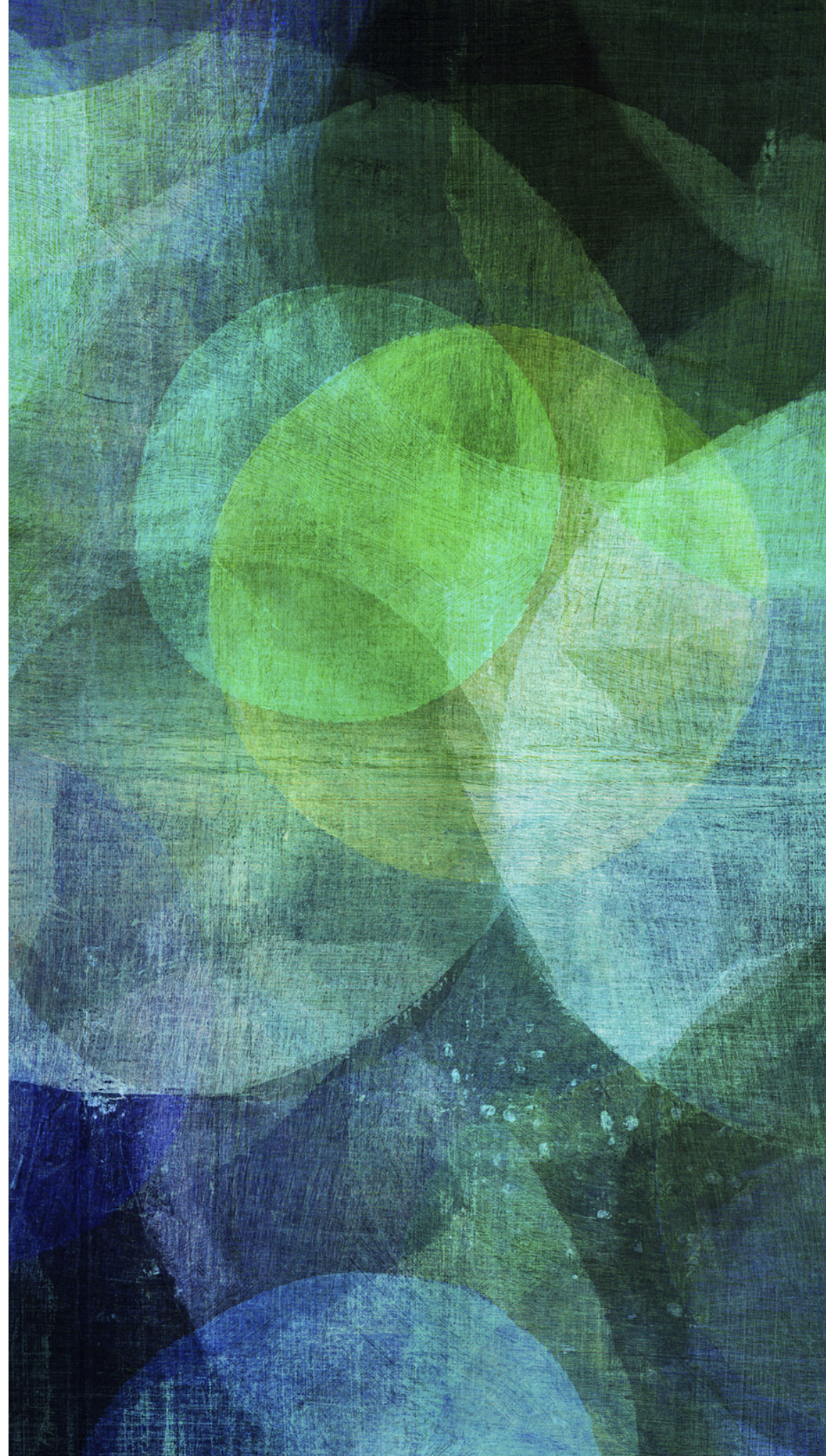
```
  CONSTRAINT `fk_1` FOREIGN KEY ('auteur_id') REFERENCES `auteur` (`id`)
```

```
  CONSTRAINT `fk_1` FOREIGN KEY ('livre_id') REFERENCES `livre` (`cote`)
```

```
)
```

```
graph TD; livre -- "cote" --> a_écrit; auteur -- "id" --> a_écrit;
```


LANGAGE DE REQUÊTES



INTERACTIONS FONDAMENTALES

Les bases de données sont fondées sur 4 interactions de base, souvent nommées sous l'acronyme CRUD.

- Insertion [C] : Ajout d'un nouvel enregistrement dans la base de données
- Lecture [R] : Extraction de données selon certains critères de recherche
- Modification [U] : Modification des valeurs d'enregistrements de la base de données
- Suppression [D] : Effacement d'enregistrements de la base de données

INSERTION

INSERT INTO TABLE livre

(liste des attributs)

VALUES

(liste des valeurs)

EXEMPLE D'INSERTION

id n'est pas spécifié car ajouté automatiquement (auto-incrémenté)

Seuls les attributs obligatoires (NOT NULL) sont requis avec INSERT

INSERT INTO TABLE auteur

(nom, prenom, naissance, deces)

VALUES

('Hugo', 'Victor', '1802-02-26', '1885-05-22')

Dates au format international

INSERTION EN MASSE

INSERT INTO TABLE auteur

(nom, prenom, naissance, deces)

VALUES

('Hugo', 'Victor', '1802-02-26', '1885-05-22'),

('Beckett', 'Samuel', '1906-04-13', '1989-12-22'),

('Kundera', 'Milan', '1929-04-01', NULL)



Un nombre arbitraire de tuples peut être inséré dans une table

MISE À JOUR

UPDATE table

SET attribut_1 = valeur, attribut_2 = valeur, attribut_3 = valeur

EXEMPLE DE MISE À JOUR

UPDATE auteur

SET prenom = 'Albert'

Aucune condition n'est spécifiée :

Tous les auteurs verront leur nom modifié en 'Albert'

AJOUT DE CONDITION

```
UPDATE auteur  
SET prenom = 'Albert'  
WHERE id = 57
```

En ajoutant une condition, seuls les enregistrements correspondants, ici une personne dont l'id vaut 57, seront modifiés

CAS D'ÉQUIVALENCE ENTRE INSERT ET UPDATE

Si l'on essaie d'insérer un tuple dont la valeur de clef primaire existe déjà dans la table, il est possible de basculer en modification

INSERT INTO TABLE auteur

(id, nom, prenom, naissance, deces)

VALUES

(57, 'Hugo', 'Victor', '1802-02-26', '1885-05-22')

ON DUPLICATE KEY UPDATE

prenom = 'Albert', nom = 'Camus'

UPDATE auteur

(id, nom, prenom, naissance, deces)

SET

prenom = 'Albert', nom = 'Camus'

WHERE

id = 57

Effets identiques



SUPPRESSION

DELETE FROM table

*Sans condition, DELETE supprime
tous les enregistrements de la table*

SUPPRESSION CIBLÉE

DELETE FROM table
WHERE <conditions>

*DELETE supprime tous les
enregistrements qui répondent aux
conditions énoncés*

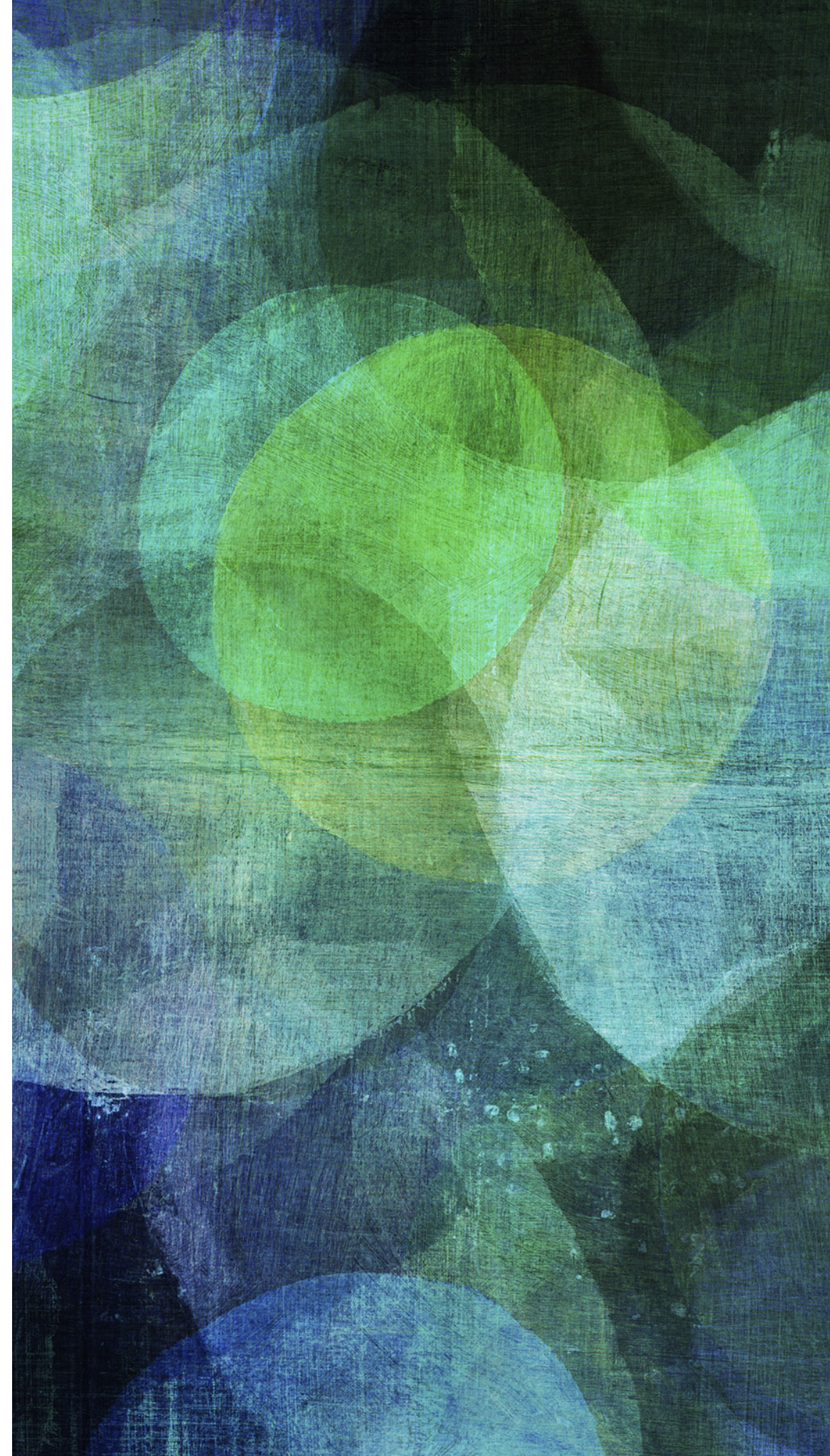
EXEMPLE

```
DELETE FROM task  
WHERE done = 1
```

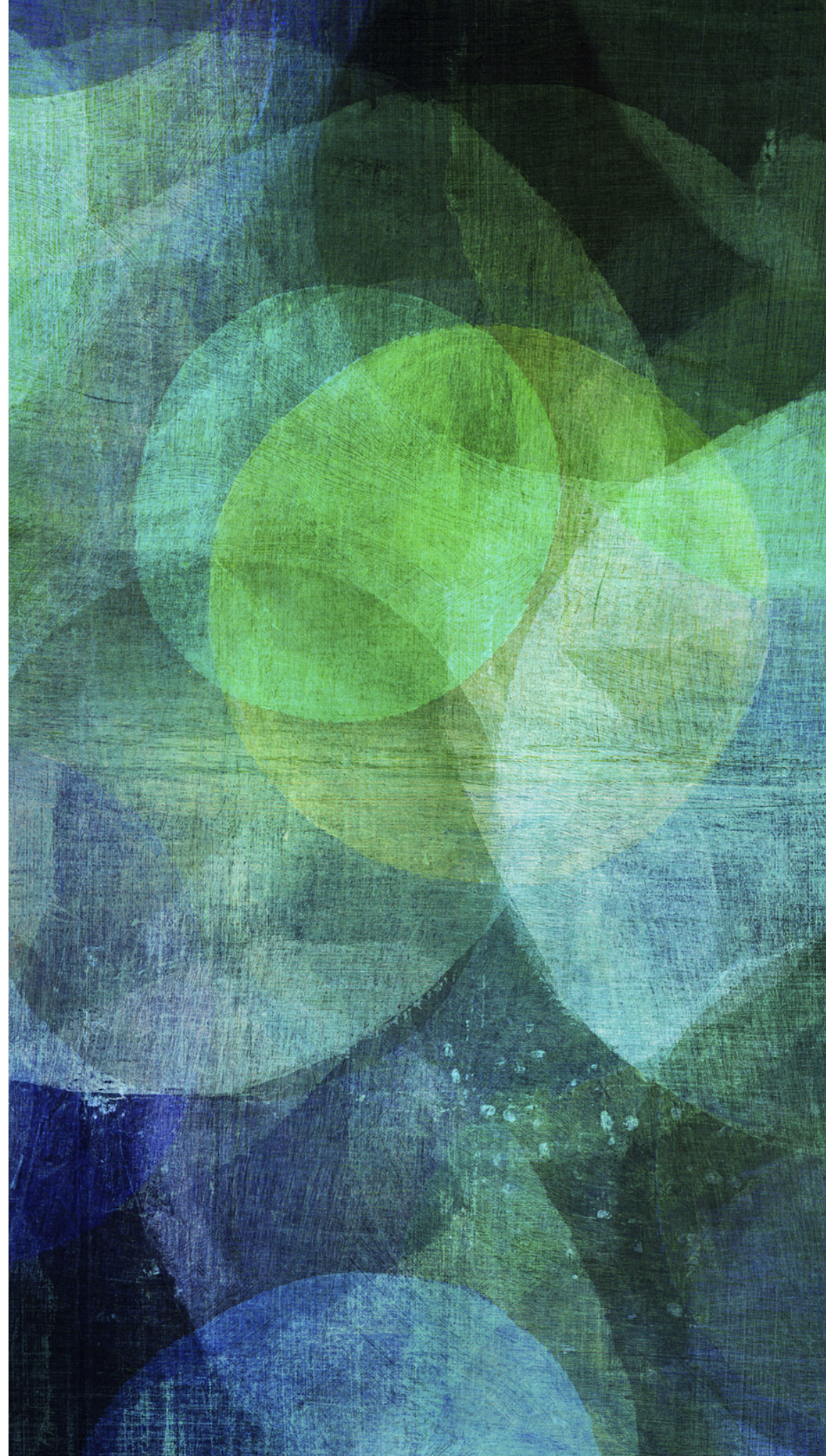
*On supprime toutes les tâches
terminées de la base de données*

SELECT

.....



DÉPENDANCES FONCTIONNELLES



DÉPENDANCE FONCTIONNELLE

Soient $R(A_1, A_2, \dots, A_n)$ un schéma de relation, X et Y des sous-ensembles de A_1, A_2, \dots, A_n . On dit que X détermine Y , ou que Y dépend fonctionnellement de X , si et seulement s'il existe une fonction qui à partir de toute valeur de X détermine une valeur unique de Y .

Plus formellement on pose que X détermine Y pour une relation R si et seulement si quelle que soit l'instance r de R , alors pour tous tuples t_1 et t_2 de r on a :

$$\text{PROJECTION}(T_1, X) = \text{PROJECTION}(T_2, X)$$

$$\Rightarrow \text{PROJECTION}(T_1, Y) = \text{PROJECTION}(T_2, Y)$$

DÉPENDANCE FONCTIONNELLE

Si X détermine Y , on note : $X \rightarrow Y$

DÉPENDANCE FONCTIONNELLE

Une DF est définie sur l'intension du schéma et non son extension. Une DF traduit une certaine perception de la réalité. Ainsi la DF (NSS, Type, Date) → Prix signifie que personne n'achète deux voitures du même type à la même date.

La seule manière de déterminer une DF est donc de regarder soigneusement ce que signifient les attributs et de trouver les contraintes qui les lient dans le monde réel.

DÉPENDANCE FONCTIONNELLE

Les DF font partie du schéma d'une BD, en conséquence, elles doivent être déclarées par les administrateurs de la BD et être contrôlées par le SGBD.

De plus l'identification des DF est la base indispensable pour déterminer dans quelle forme normale est une relation et comment en diminuer la redondance.

AXIOMES : RÉFLEXIVITÉ

Tout groupe d'attributs se détermine lui même et détermine chacun de ses attributs (ou sous groupe de ses attributs).

$$XY \rightarrow XY \Rightarrow XY \rightarrow X \text{ et } XY \rightarrow Y$$

AXIOMES : AUGMENTATION

Si un attribut X détermine un attribut Y , alors tout groupe composé de X enrichi avec d'autres attributs détermine un groupe composé de Y et enrichi des mêmes autres attributs.

$$X \rightarrow Y \Rightarrow XZ \rightarrow YZ$$

AXIOMES : TRANSITIVITÉ

Si un attribut X détermine un attribut Y et que cet attribut Y détermine un autre attribut Z , alors X détermine Z .

$$X \rightarrow Y \text{ et } Y \rightarrow Z \Rightarrow X \rightarrow Z$$

PROPRIÉTÉS DÉDUITES : PSEUDO-TRANSITIVITÉ

Si un attribut X détermine un autre attribut Y , et que Y appartient à un groupe G qui détermine un troisième attribut Z , alors le groupe G' obtenu en substituant Y par X dans G détermine également Z .

$$X \rightarrow Y \text{ et } WY \rightarrow Z \Rightarrow WX \rightarrow Z$$

$$X \rightarrow Y \text{ et } WY \rightarrow Z \Rightarrow WX \rightarrow WY \text{ et } WY \rightarrow Z \Rightarrow WX \rightarrow Z$$

PROPRIÉTÉS DÉDUITES : UNION

Si un attribut détermine plusieurs autres attributs, alors il détermine tout groupe composé de ces attributs.

$$X \rightarrow Y \text{ et } X \rightarrow Z \Rightarrow X \rightarrow YZ$$

$$X \rightarrow Y \text{ et } X \rightarrow Z \Rightarrow X \rightarrow XX \text{ et } XX \rightarrow XY \text{ et } YX \rightarrow YZ \Rightarrow X \rightarrow YZ$$

PROPRIÉTÉS DÉDUITES : DÉCOMPOSITION

Si un attribut détermine un groupe d'attribut, alors il détermine chacun des attributs de ce groupe pris individuellement.

$$X \rightarrow YZ \Rightarrow X \rightarrow Z \text{ et } X \rightarrow Y$$

$$X \rightarrow YZ \Rightarrow X \rightarrow YZ \text{ et } YZ \rightarrow Z \Rightarrow X \rightarrow Z$$

DÉPENDANCE FONCTIONNELLE ÉLÉMENTAIRE

Soit G un groupe d'attributs et A un attribut, une DF $G \rightarrow A$ est élémentaire si A n'est pas incluse dans G et s'il n'existe pas d'attribut A' de G qui détermine A .

EXEMPLES

$AB \rightarrow C$

est élémentaire si ni A, ni B pris individuellement ne déterminent C.

$\text{Nom, DateNaissance, LieuNaissance} \rightarrow \text{Prénom}$
est élémentaire.

EXEMPLES

$AB \rightarrow A$

n'est pas élémentaire car A est incluse dans AB .

$AB \rightarrow CB$

n'est pas élémentaire car CB n'est pas un attribut, mais un groupe d'attributs.

FERMETURE TRANSITIVE

On appelle fermeture transitive F^+ d'un ensemble F de [DFE](#), l'ensemble de toutes les DFE qui peuvent être composées par transitivité à partir des DFE de F .

Soit l'ensemble $F = \{A \rightarrow B, B \rightarrow C, B \rightarrow D, A \rightarrow E\}$.

La fermeture transitive de F est

$F^+ = \{ A \rightarrow B, B \rightarrow C, B \rightarrow D, A \rightarrow E, A \rightarrow C, A \rightarrow D \}$

CLEF

Soient une relation $R(A_1, A_2, \dots, A_n)$ et K un sous-ensemble de A_1, A_2, \dots, A_n . K est une clé de R si et seulement si :

$$K \rightarrow A_1, A_2, \dots, A_n$$

$$\neg \exists X \subset K \mid X \rightarrow A_1, A_2, \dots, A_n$$

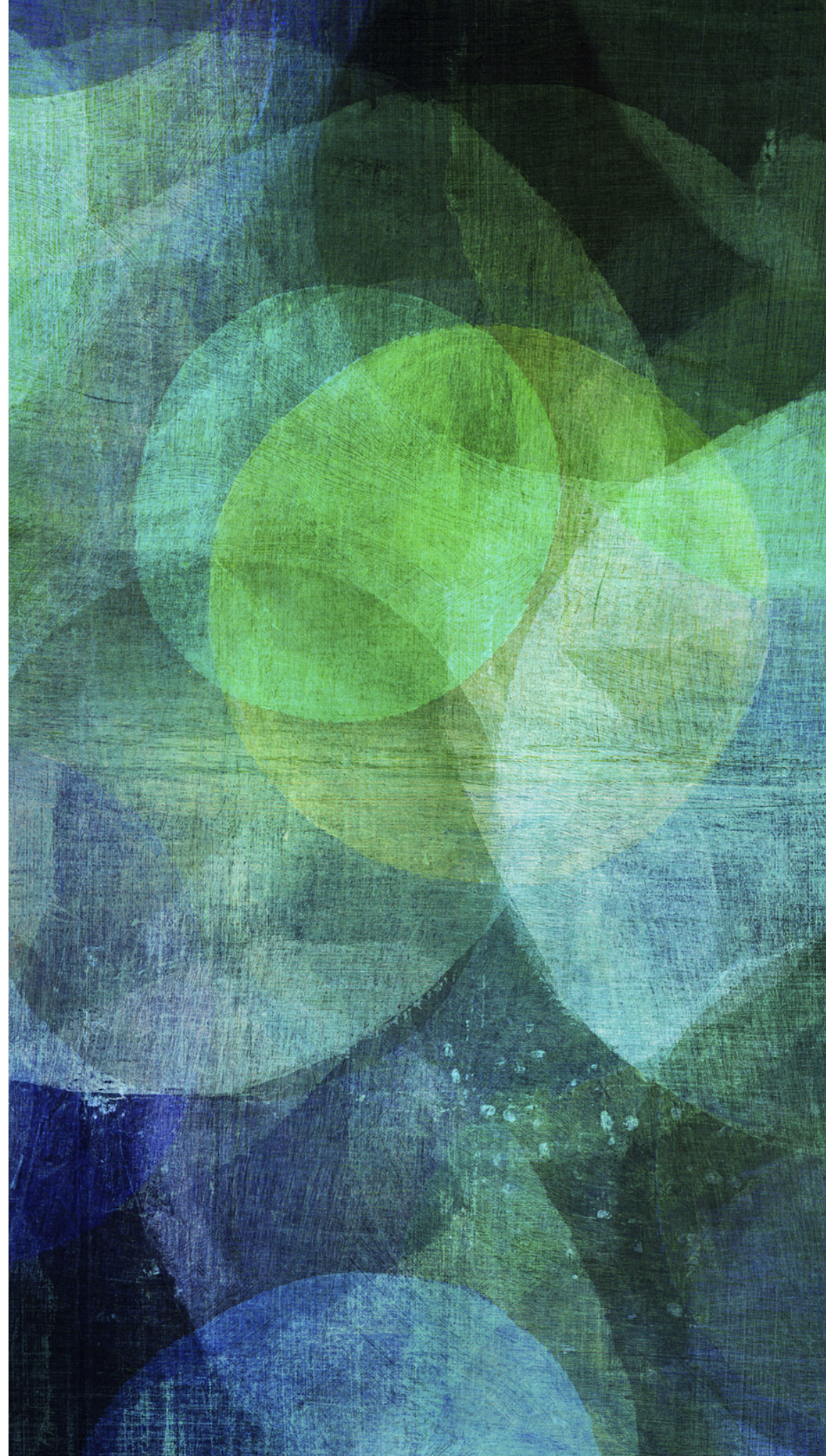
Une clé est donc un ensemble minimum d'attributs d'une relation qui détermine tous les autres.

REMARQUES

Il peut y avoir plusieurs clefs candidates. Toutes les clefs candidates se déterminent mutuellement. Il existe généralement une clef minimale.

Toute relation a nécessairement une clef. A minima, l'ensemble de tous les attributs constitue aussi une clef.

FORMES NORMALES



OBJECTIF

Les formes normales ont pour objectif de définir la décomposition des schémas relationnels, tout en préservant les DF et sans perdre d'informations, afin de représenter les objets et associations canoniques du monde réel de façon non redondante.

DÉCOMPOSITION

La décomposition d'un schéma de relation $R(A_1, A_2, \dots, A_n)$ est le processus de remplacement de ce schéma par une collection de schémas R_1, R_2, \dots, R_n telle qu'il est possible de reconstruire R par des opérations relationnelles de jointure sur R_1, R_2, \dots, R_n .

Une décomposition d'une relation R en relations R_1, R_2, \dots, R_n préserve les [DF](#) si la fermeture transitive F^+ des DF de R est la même que la fermeture transitive F^+ de l'union des DF de R_1, R_2, \dots, R_n .

EXAMPLE

.....

1NF

Une relation est en 1NF si elle possède au moins une clé et si tous ses attributs sont atomiques.

La condition « possède au moins une clef » est superflue puisque toute relation possède au moins une clef.

2NF

Une relation est en 2NF si elle est en 1NF et si tout attribut n'appartenant à aucune clé candidate ne dépend pas d'une partie seulement d'une clé candidate.

EXEMPLE

**PERSONNE(
#ID,
#PROFESSION,
NOM,
PRÉNOM,
SALAIRE
)**

**ID, PROFESSION → NOM
ID, PROFESSION → PRENOM
PROFESSION → SALAIRE**

La relation n'est pas en 2NF car SALAIRE
ne dépend que d'une partie de la clef

3NF

Une relation est en 3NF si elle est en 2NF et si tout attribut n'appartenant à aucune clé candidate ne dépend directement que de clés candidates.

EXEMPLE

**PROFESSION(
#PROFESSION,
SALAIRE,
PRIME
)**

SALAIRE → PRIME

PROFESSION → PRIME

PROFESSION → SALAIRE

Cette relation n'est pas en 3NF car SALAIRE, qui n'est pas une clé, détermine PRIME.

BCNF

Une relation est en BCNF si elle est en 3NF et si les seules DFE existantes sont celles pour lesquelles une clé candidate détermine un attribut.

EXEMPLE

**PERSONNE(
#N°SS,
#PAYS,
NOM,
RÉGION
)**

N°SS, PAYS → NOM

N°SS, PAYS → REGION

REGION → PAYS

Il existe une DFE qui n'est pas issue d'une clé et qui détermine un attribut appartenant à une clé. Cette relation est en 3NF, mais pas en BCNF (car en BCNF toutes les DFE sont issues d'une clé).

4NF

Une relation est en 4NF si et seulement si, pour chacune de ses dépendances multivaluées non triviales $X \twoheadrightarrow Y$, X est une super-clef — c'est-à-dire X est une clef candidate ou un sur-ensemble de celle-ci.

Une dépendance multivaluée $X \twoheadrightarrow Y$ est triviale si :

- soit Y est un sous-ensemble de X ,*
- soit X et Y constituent l'ensemble des attributs de la relation.*

EXEMPLE

| Vendeur | Plat | Région |
|-------------|---------------|--------------|
| A1 Pizza | Thick Crust | Springfield |
| A1 Pizza | Thick Crust | Shelbyville |
| A1 Pizza | Thick Crust | Capital City |
| A1 Pizza | Stuffed Crust | Springfield |
| A1 Pizza | Stuffed Crust | Shelbyville |
| A1 Pizza | Stuffed Crust | Capital City |
| Elite Pizza | Thin Crust | Capital City |
| Elite Pizza | Stuffed Crust | Capital City |
| Vincenzo's | Thick Crust | Springfield |
| Vincenzo's | Thick Crust | Shelbyville |
| Vincenzo's | Thick Crust | Springfield |
| Vincenzo's | Thick Crust | Shelbyville |

*Les trois attributs
constituent la clef
primaire, ce qui fait que
la table est en BCNF*

*Vendeur → Plat
Vendeur → Région,*

SOLUTION (?)

{Restaurant} n'est pas une super-clef. Mais les dépendances multivaluées indiquent que les variétés de pizzas sont indépendantes de l'aire de distribution.

Cela conduit à une redondance de l'information. Si A1 Pizza offre une nouvelle variété Ananas Pizza, nous serons obligés de dupliquer des lignes, ce qui est source d'erreurs.

Nous devrions donc décomposer la relation en deux tables pour respecter la 4NF

Néanmoins, ce ne serait pas suffisant, car on remarque que de l'information se perdrait dans cette transformation.