

# L2 Informatique, UE INF403

## Gestion de données relationnelles et applications

Mario Cortés-Cornax

Université Grenoble Alpes – UFR IM<sup>2</sup>AG

2022/2023

## Equipe pédagogique

- Mario CORTES-CORNAX (MCF UGA, LIG)  
`mario.cortes-cornax@univ-grenoble-alpes.fr`
- Catherine VIGOUROUX (MCF UGA, VERIMAG)  
`catherine.vigouroux@univ-grenoble-alpes.fr`
- Quentin Nivon (PhD UGA, LIG)  
`quentin.nivon@inria.fr`
- Florent TALLERIE (PhD UGA, GSCOP)  
`florent.tallerie@univ-grenoble-alpes.fr`
- Philip SCALES (PhD UGA, LIG)  
`philip.scales@univ-grenoble-alpes.fr`

# Evaluation

CC1 = 2 comptes rendus	cr1 (12,5%), projet (12,5%)
CC2 = partiel	cc2 (25%)
Examen	ex (50%)

- Le **partiel** a lieu entre le lu. 13/03/2023 et le ve. 17/03/2023
- Les **comptes rendus de TP** sont à rendre sur caseine :  
<https://moodle.caseine.org/course/view.php?id=624>
- Les TPs sont faits **en binôme**, un seul compte rendu par binôme
- L'**examen** a lieu entre le lu. 15/05/2023 et le mer. 17/05/2023

# “Règles du jeu”

- Assiduité
- Retards
- CM pratique
- Inscription aux groupes sur Caseine  
(<https://moodle.caseine.org/course/view.php?id=624>)

# Objectifs

*Maîtriser le modèle relationnel de données et SQL.*

*Etre sensibilisé à la mise en œuvre d'une application, au dessus d'un Système de Gestion de Bases de Données.*

- Le modèle relationnel de données
- Un langage relationnel : SQL
- La conception d'une BD avec UML
- Les bases de la mise en œuvre d'une application de BD avec Python

# Notion de Bases de Données

Une **base de données** contient des informations (d'une fraction) de la réalité, conçues pour répondre à des besoins particuliers afin qu'elles soient interrogées, modifiées, éventuellement supprimées dans le futur.

## Exemples:

- *Un carnet d'adresses* : les noms, prénoms, adresses et téléphones de mes amis, sont enregistrés dans mon carnet d'adresses (quelques kilos octets)
- *World Data Centre for Climate* : 220 terabytes<sup>1</sup> de données disponibles sur le web 6 petabytes<sup>2</sup> d'autres données



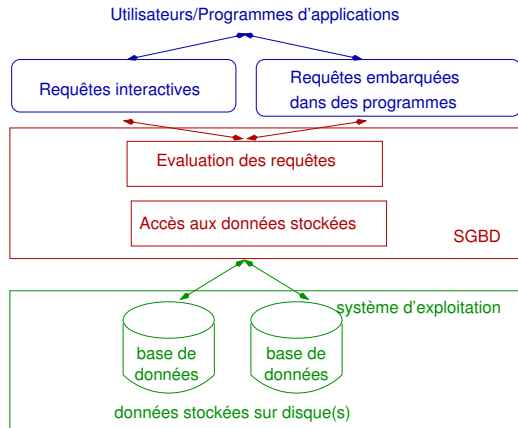
---

<sup>1</sup>1 tera=  $10^{12}$

<sup>2</sup>1 peta=  $10^{15}$

# Systèmes de Gestion de Bases de Données

Un Système de Gestion de Bases de Données (SGBD) est un ensemble de modules logiciels responsable du stockage et de l'accès à des informations.



# Un système de gestion de bases de données. A quoi ça sert?

Un système logiciel conçu pour être la base des applications de bases de données.



# Un système de gestion de bases de données. A quoi ça sert?

Un système logiciel conçu pour être la base des applications de bases de données.

- permettant un **accès aux données** indépendant de leur implantation (modèle relationnel, SQL)
- offrant un **stockage et une recherche efficace** (gestion de la mémoire et des disques)
- capable de **gérer de nombreux accès simultanés** (gestion des privilèges, des accès concurrents)
- capable d'**exécuter de très nombreuses opérations** (gestion des transactions)
- assurant une **gestion fiable** des données (sauvegarde, restauration)

Les SGBDs répondent à des problèmes similaires à ceux traités par les systèmes d'exploitation.

# Chapitre 2 – Le modèle relationnel de données

## 1 Le modèle relationnel

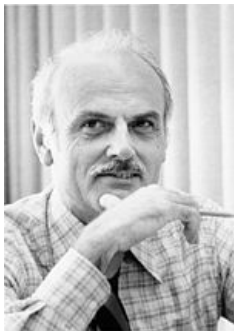
- Introduction
- Les relations
- Définitions et Notations
- Contraintes
- Spécification d'un schéma de relations

# Un peu de culture

## Le Modèle Relationnel de Données...

- A été introduit en 1970 par Ted Codd,
- Est attractif grâce à sa simplicité et ses fondements mathématiques,
- Utilise le concept de relation mathématique pour modéliser l'information,
- Ne peut pas être ignoré à cause de sa popularité.

## Ted Codd (1923 - 2003)



[https://en.wikipedia.org/wiki/Edgar\\_F.\\_Codd](https://en.wikipedia.org/wiki/Edgar_F._Codd)

# Origine de SQL



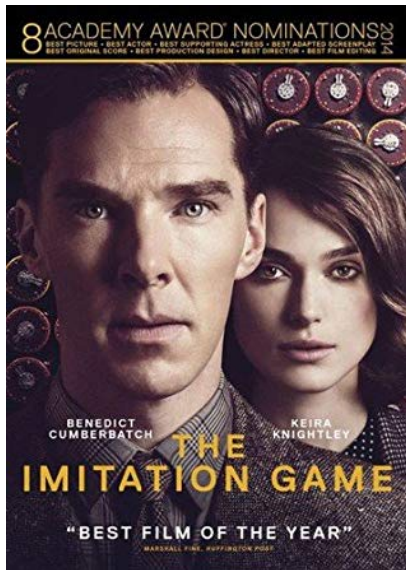
# Origine de SQL



# Ted Codd reçoit le Prix Turing en 1981

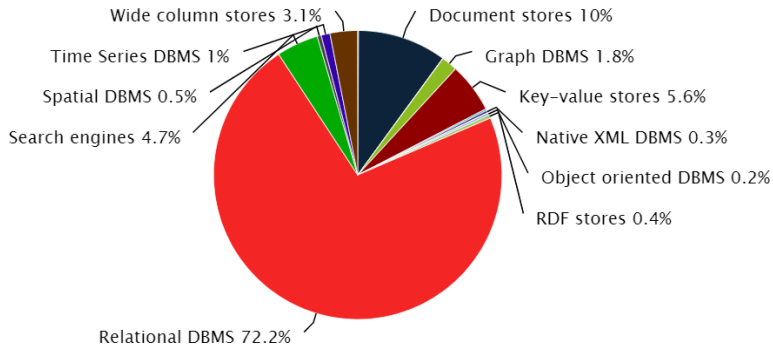


## Ted Codd reçoit le Prix Turing en 1981





# Polularité des Modèles de Bases de Données



© 2022, DB-Engines.com

source : [https://db-engines.com/en/ranking\\_categories](https://db-engines.com/en/ranking_categories)

# Vocabulaire

- Une TABLE = une RELATION
- On pourra faire référence à un attribut d'une relation en utilisant la notation : Relation.Attribut

**Colonne/Attribut**  
(nom et type)

Schéma de la relation

Extension de la relation

R1 (Cours, Etudiant, Professeur)		
COURS	ETUDIANT	PROFESSEUR
BD	Lulu	Michel
BD	Riri	Michel
BD	Fifi	Michel
PROG	Lulu	Roland
PROG	Toto	Roland
Droit	Lulu	Roland
Droit	Zoé	Roland
Droit	Toto	Roland

Ligne/Enregistrement/N-uplet

Valeur

# Premières Observations

- 1 Une base de données est constituée d'un **ensemble de relations**
- 2 Chaque **relation** contient les données relatives à des **entités de même nature**
- 3 Chaque **ligne** d'une relation reprend les **données relatives à une entité**
- 4 Les **lignes** d'une relation sont **distinctes**
- 5 Chaque **colonne** d'une relation décrit une **propriété commune des entités**
- 6 On évite de stocker les informations qui peuvent être calculées

# Opérations sur les relations

## Interroger des relations

*Employees*

firstname	salary	address	dept
John	120	Randwick	Toys
Mary	130	Wollongong	Furniture
Peter	110	Randwick	Garden
Tom	120	Botany Bay	Toys

- **Projection** : Quels sont les noms de rayon ?
- **Sélection** : Donner les employés qui gagnent plus ou égal à 120
- **Sélection + projection** : Quelle est l'adresse de Tom ?
- **Agrégation** : Combien d'employés travaillent au rayon Toys ?

# Opérations sur les relations

## Interroger des relations

*Employees*

firstname	salary	address	dept
John	120	Randwick	Toys
Mary	130	Wollongong	Furniture
Peter	110	Randwick	Garden
Tom	120	Botany Bay	Toys

- **Projection** : Quels sont les noms de rayon ?  
 $Employees[dept] \rightarrow \{ \langle Toys \rangle, \langle Furniture \rangle, \langle Garden \rangle \}$
- **Sélection** : Donner les employés qui gagnent plus ou égal à 120
- **Sélection + projection** : Quelle est l'adresse de Tom ?
- **Agrégation** : Combien d'employés travaillent au rayon Toys ?

# Opérations sur les relations

## Interroger des relations

*Employees*

firstname	salary	address	dept
John	120	Randwick	Toys
Mary	130	Wollongong	Furniture
Peter	110	Randwick	Garden
Tom	120	Botany Bay	Toys

- **Projection** : Quels sont les noms de rayon ?  
 $Employees[dept] \rightarrow \{ \langle \text{Toys} \rangle, \langle \text{Furniture} \rangle, \langle \text{Garden} \rangle \}$
- **Sélection** : Donner les employés qui gagnent plus ou égal à 120  
 $Employees:salary \geq 120 \rightarrow \{ \langle \text{John}, 120, \text{Randwick}, \text{Toys} \rangle, \langle \text{Mary}, 130, \text{Wollongong}, \text{Furniture} \rangle, \langle \text{Tom}, 120, \text{Botany Bay}, \text{Toys} \rangle \}$
- **Sélection + projection** : Quelle est l'adresse de Tom ?
- **Agrégation** : Combien d'employés travaillent au rayon Toys ?

# Opérations sur les relations

## Interroger des relations

<i>Employees</i>	firstname	salary	address	dept
	John	120	Randwick	Toys
	Mary	130	Wollongong	Furniture
	Peter	110	Randwick	Garden
	Tom	120	Botany Bay	Toys

- **Projection** : Quels sont les noms de rayon ?  
 $Employees[dept] \rightarrow \{ \langle \text{Toys} \rangle, \langle \text{Furniture} \rangle, \langle \text{Garden} \rangle \}$
- **Sélection** : Donner les employés qui gagnent plus ou égal à 120  
 $Employees:salary \geq 120 \rightarrow \{ \langle \text{John}, 120, \text{Randwick}, \text{Toys} \rangle, \langle \text{Mary}, 130, \text{Wollongong}, \text{Furniture} \rangle, \langle \text{Tom}, 120, \text{Botany Bay}, \text{Toys} \rangle \}$
- **Sélection + projection** : Quelle est l'adresse de Tom ?  
 $Employees:firstname = 'Tom'[address] \rightarrow \{ \langle \text{Botany Bay} \rangle \}$
- **Agrégation** : Combien d'employés travaillent au rayon Toys ?

# Opérations sur les relations

## Interroger des relations

*Employees*

firstname	salary	address	dept
John	120	Randwick	Toys
Mary	130	Wollongong	Furniture
Peter	110	Randwick	Garden
Tom	120	Botany Bay	Toys

- **Projection** : Quels sont les noms de rayon ?  
 $Employees[dept] \rightarrow \{ \langle \text{Toys} \rangle, \langle \text{Furniture} \rangle, \langle \text{Garden} \rangle \}$
- **Sélection** : Donner les employés qui gagnent plus ou égal à 120  
 $Employees:salary \geq 120 \rightarrow \{ \langle \text{John}, 120, \text{Randwick}, \text{Toys} \rangle, \langle \text{Mary}, 130, \text{Wollongong}, \text{Furniture} \rangle, \langle \text{Tom}, 120, \text{Botany Bay}, \text{Toys} \rangle \}$
- **Sélection + projection** : Quelle est l'adresse de Tom ?  
 $Employees:firstname = 'Tom'[address] \rightarrow \{ \langle \text{Botany Bay} \rangle \}$
- **Agrégation** : Combien d'employés travaillent au rayon Toys ?  $\rightarrow \{ \langle 2 \rangle \}$



### *Mettre à jour une relation*

<b>firstname</b>	<b>salary</b>	<b>address</b>	<b>dept</b>
John	120	Randwick	Toys
Mary	130	Wollongong	Furniture
Peter	110	Randwick	Garden
Tom	120	Botany Bay	Toys

- Le salaire de Mary a augmenté de 10%

### Mettre à jour une relation

firstname	salary	address	dept
John	120	Randwick	Toys
Mary	130	Wollongong	Furniture
Peter	110	Randwick	Garden
Tom	120	Botany Bay	Toys

- Le salaire de Mary a augmenté de 10%
- Phil est maintenant employé au magasin, son salaire est 140, il est affecté au rayon Furniture, son adresse est à Newtown.

firstname	salary	address	dept
John	120	Randwick	Toys
Mary	143	Wollongong	Furniture
Peter	110	Randwick	Garden
Tom	120	Botany Bay	Toys
Phil	140	Newtown	Furniture

*Ajouter un nouveau type d'information*

firstname	salary	address	dept
John	120	Randwick	Toys
Mary	143	Wollongong	Furniture
Peter	110	Randwick	Garden
Tom	120	Botany Bay	Toys
Phil	140	Newtown	Furniture

Chaque rayon est dirigé par un employé

## Ajouter un nouveau type d'information

firstname	salary	address	dept
John	120	Randwick	Toys
Mary	143	Wollongong	Furniture
Peter	110	Randwick	Garden
Tom	120	Botany Bay	Toys
Phil	140	Newtown	Furniture

Chaque rayon est dirigé par un employé

Deux solutions..... ajouter une(des) colonne(s)  
ajouter une(des) relation(s)

## Ajouter une colonne (#1) :

- Ajouter une marque pour chaque employé

firstname	boss	salary	address	dept
John	yes	120	Randwick	Toys
Mary	yes	143	Wollongong	Furniture
Peter	yes	110	Randwick	Garden
Tom	no	120	Botany Bay	Toys
Phil	no	140	Newtown	Furniture

*Dur à lire et à décoder ...*

## Ajouter une colonne (#2) :

- Ajouter un nom de chef à chaque rayon

firstname	salary	address	dept	boss
John	120	Randwick	Toys	John
Mary	143	Wollongong	Furniture	Mary
Peter	110	Randwick	Garden	Peter
Tom	120	Botany Bay	Toys	John
Phil	140	Newtown	Furniture	Mary

*Redondance des données...et problématique de maintien de la coherence  
"John est le chef du rayon Toys" est dit deux fois*

## Ajouter une relation

firstname	salary	address	dept
John	120	Randwick	Toys
Mary	130	Wollongong	Furniture
Peter	110	Randwick	Garden
Tom	120	Botany Bay	Toys

boss	dept
John	Toys
Mary	Furniture
Peter	Garden

*Les requêtes sont un peu plus compliquées ...  
"Donner le salaire du chef du rayon Toys"*

# Ensemble : définition et quelques opérations

- Un **ensemble** est une collection d'éléments (entre accolades) différents deux à deux et reliés à un domaine particulier.

$A = \{ \text{'Furniture'}, \text{'Toys'}, \text{'Garden'} \}$

$F = \{ p \in \text{ThePersons} : \text{Sexe}(p) = \text{'female'} \}$



# Ensemble : définition et quelques opérations

- Un **ensemble** est une collection d'éléments (entre accolades) différents deux à deux et reliés à un domaine particulier.  
 $A = \{ \text{'Furniture'}, \text{'Toys'}, \text{'Garden'} \}$   
 $F = \{ p \in \text{ThePersons} : \text{Sexe}(p) = \text{'female'} \}$
- **Appartenance** :  $3 \in \{1, 3, 5, 6\}, 8 \notin \{1, 3, 5, 6\}$

- **Produit cartésien** d'ensembles (noté  $\times$ ) :  
 $\{1, 6, 3, 5\} \times \{'Furniture', 'Toys', 'Garden'\} =$   
 $\{ \langle 1, 'Furniture' \rangle, \langle 1, 'Toys' \rangle, \langle 1, 'Garden' \rangle,$   
 $\langle 3, 'Furniture' \rangle, \langle 3, 'Toys' \rangle, \langle 3, 'Garden' \rangle,$   
 $\langle 5, 'Furniture' \rangle, \langle 5, 'Toys' \rangle, \langle 5, 'Garden' \rangle,$   
 $\langle 6, 'Furniture' \rangle, \langle 6, 'Toys' \rangle, \langle 6, 'Garden' \rangle \}$

- **Produit cartésien** d'ensembles (noté  $\times$ ) :  
 $\{1, 6, 3, 5\} \times \{'Furniture', 'Toys', 'Garden'\} =$   
 $\{ \langle 1, 'Furniture' \rangle, \langle 1, 'Toys' \rangle, \langle 1, 'Garden' \rangle,$   
 $\langle 3, 'Furniture' \rangle, \langle 3, 'Toys' \rangle, \langle 3, 'Garden' \rangle,$   
 $\langle 5, 'Furniture' \rangle, \langle 5, 'Toys' \rangle, \langle 5, 'Garden' \rangle,$   
 $\langle 6, 'Furniture' \rangle, \langle 6, 'Toys' \rangle, \langle 6, 'Garden' \rangle \}$
- **Intersection** :  $\{1, 3, 5, 6\} \cap \{10, 5, 3, 9\} = \{5, 3\}$

- **Produit cartésien** d'ensembles (noté  $\times$ ) :  
 $\{1, 6, 3, 5\} \times \{'Furniture', 'Toys', 'Garden'\} =$   
 $\{ \langle 1, 'Furniture' \rangle, \langle 1, 'Toys' \rangle, \langle 1, 'Garden' \rangle,$   
 $\langle 3, 'Furniture' \rangle, \langle 3, 'Toys' \rangle, \langle 3, 'Garden' \rangle,$   
 $\langle 5, 'Furniture' \rangle, \langle 5, 'Toys' \rangle, \langle 5, 'Garden' \rangle,$   
 $\langle 6, 'Furniture' \rangle, \langle 6, 'Toys' \rangle, \langle 6, 'Garden' \rangle \}$
- **Intersection** :  $\{1, 3, 5, 6\} \cap \{10, 5, 3, 9\} = \{5, 3\}$
- **Union** :  
 $\{1, 3, 5, 6\} \cup \{10, 5, 3, 9\} = \{1, 5, 3, 6, 10, 9\}$

- **Produit cartésien** d'ensembles (noté  $\times$ ) :  
 $\{1, 6, 3, 5\} \times \{'Furniture', 'Toys', 'Garden'\} =$   
 $\{ \langle 1, 'Furniture' \rangle, \langle 1, 'Toys' \rangle, \langle 1, 'Garden' \rangle,$   
 $\langle 3, 'Furniture' \rangle, \langle 3, 'Toys' \rangle, \langle 3, 'Garden' \rangle,$   
 $\langle 5, 'Furniture' \rangle, \langle 5, 'Toys' \rangle, \langle 5, 'Garden' \rangle,$   
 $\langle 6, 'Furniture' \rangle, \langle 6, 'Toys' \rangle, \langle 6, 'Garden' \rangle \}$
- **Intersection** :  $\{1, 3, 5, 6\} \cap \{10, 5, 3, 9\} = \{5, 3\}$
- **Union** :  
 $\{1, 3, 5, 6\} \cup \{10, 5, 3, 9\} = \{1, 5, 3, 6, 10, 9\}$
- **Différence** (asymétrique) :  
 $\{1, 3, 5, 6\} - \{10, 5, 3, 9\} = \{1, 6\}$   
 $\{10, 5, 3, 9\} - \{1, 3, 5, 6\} = \{10, 9\}$

- **Produit cartésien** d'ensembles (noté  $\times$ ) :  
 $\{1, 6, 3, 5\} \times \{'Furniture', 'Toys', 'Garden'\} =$   
 $\{ \langle 1, 'Furniture' \rangle, \langle 1, 'Toys' \rangle, \langle 1, 'Garden' \rangle,$   
 $\langle 3, 'Furniture' \rangle, \langle 3, 'Toys' \rangle, \langle 3, 'Garden' \rangle,$   
 $\langle 5, 'Furniture' \rangle, \langle 5, 'Toys' \rangle, \langle 5, 'Garden' \rangle,$   
 $\langle 6, 'Furniture' \rangle, \langle 6, 'Toys' \rangle, \langle 6, 'Garden' \rangle \}$
- **Intersection** :  $\{1, 3, 5, 6\} \cap \{10, 5, 3, 9\} = \{5, 3\}$
- **Union** :  
 $\{1, 3, 5, 6\} \cup \{10, 5, 3, 9\} = \{1, 5, 3, 6, 10, 9\}$
- **Différence** (asymétrique) :  
 $\{1, 3, 5, 6\} - \{10, 5, 3, 9\} = \{1, 6\}$   
 $\{10, 5, 3, 9\} - \{1, 3, 5, 6\} = \{10, 9\}$
- **Inclusion** :  
 $\{ \} \subseteq \{10, 5, 3, 9\}$  ( $\{ \}$  est aussi noté  $\emptyset$ )  
 $\{9, 10\} \subseteq \{10, 5, 3, 9\}$   
 $\{9, 10\} \subset \{10, 5, 3, 9\}$   
 $\{9, 10, 3, 5\} \not\subseteq \{10, 5, 3, 9\}$   
 $\{1, 9, 10, 3, 5, 7\} \not\subseteq \{10, 5, 3, 9\}$

# Question

Produit Cartésien:

$$\{1, 3\} \times \{2, 1\} =$$

- A.  $\{ \langle 1, 2 \rangle, \langle 1, 1 \rangle, \langle 3, 2 \rangle, \langle 3, 1 \rangle \}$
- B.  $\{ \langle 1, 2 \rangle, \langle 3, 1 \rangle \}$
- C.  $\{ \langle 1, 3, 2, 1 \rangle \}$

# Question

Produit Cartésien:

$$\{1, 3\} \times \{2, 1\} = \{ \langle 1, 2 \rangle, \langle 1, 1 \rangle, \langle 3, 2 \rangle, \langle 3, 1 \rangle \}$$

A.  $\{ \langle 1, 2 \rangle, \langle 1, 1 \rangle, \langle 3, 2 \rangle, \langle 3, 1 \rangle \}$

B.  $\{ \langle 1, 2 \rangle, \langle 3, 1 \rangle \}$

C.  $\{ \langle 1, 3, 2, 1 \rangle \}$



## Question

Intersection:  $\{1, 2, 5, 7\} \cap \{5, 3, 1\} =$

- A.  $\{1, 2, 5, 7, 3\}$
- B.  $\{2, 7\}$
- C.  $\{1, 5\}$

## Question

Intersection:  $\{1, 2, 5, 7\} \cap \{5, 3, 1\} = \{1, 5\}$

- A.  $\{1, 2, 5, 7, 3\}$
- B.  $\{2, 7\}$
- C.  $\{1, 5\}$

## Question

Union:  $\{2, 1, 4, 7\} \cup \{1, 5, 3\} =$

- A.  $\{2, 4, 7\}$
- B.  $\{1\}$
- C.  $\{2, 1, 4, 7, 5, 3\}$

## Question

Union:  $\{2, 1, 4, 7\} \cup \{1, 5, 3\} = \{2, 1, 4, 7, 5, 3\}$

- A.  $\{2, 4, 7\}$
- B.  $\{1\}$
- C.  $\{2, 1, 4, 7, 5, 3\}$

# Question

Minus (asymétrique):

$$\{1, 2, 5, 7\} - \{5, 3, 1\} =$$

A.  $\{2, 7, 3\}$

B.  $\{2, 7\}$

C.  $\{3\}$

# Question

Minus (asymétrique):

$$\{1, 2, 5, 7\} - \{5, 3, 1\} = \{2, 7\}$$

A.  $\{2, 7, 3\}$

B.  $\{2, 7\}$

C.  $\{3\}$

# Question

Inclusion:

$\{2, 1, 4, 7\} \subset \{7, 2, 1, 4\}$  (Vrais ou Faux?)

$\{4, 2\} \not\subset \{7, 2, 1, 4\}$  (Vrais ou Faux?)

- A. Vrais et Vrais
- B. Faux et Vrais
- C. Faux et Faux

# Question

Inclusion:

$\{2, 1, 4, 7\} \subset \{7, 2, 1, 4\}$  (Vrais ou Faux?) F

$\{4, 2\} \not\subset \{7, 2, 1, 4\}$  (Vrais ou Faux?) F

- A. Vrais et Vrais
- B. Faux et Vrais
- C. Faux et Faux



# Domaine, Relation, Attribut, Schéma

- Un **domaine** est un ensemble de valeurs atomiques (chaînes, nombres,...). {'Furniture', 'Toys', 'Garden'}, entiers  $> 100$

# Domaine, Relation, Attribut, Schéma

- Un **domaine** est un ensemble de valeurs atomiques (chaînes, nombres,...).  $\{ \text{'Furniture'}, \text{'Toys'}, \text{'Garden'} \}$ , entiers  $> 100$
- Une **relation** est un sous-ensemble du produit cartésien d'un ensemble de domaines.  
 $\{ \langle \text{'John'}, 120 \rangle, \langle \text{'Mary'}, 130 \rangle, \langle \text{'Peter'}, 110 \rangle, \langle \text{'Tom'}, 120 \rangle \} \subseteq \{ \text{'John'}, \text{'Mary'}, \text{'Peter'}, \text{'Tom'} \} \times \text{entiers} > 100$

# Domaine, Relation, Attribut, Schéma

- Un **domaine** est un ensemble de valeurs atomiques (chaînes, nombres,...).  $\{ \text{'Furniture'}, \text{'Toys'}, \text{'Garden'} \}$ , entiers  $> 100$
- Une **relation** est un sous-ensemble du produit cartésien d'un ensemble de domaines.  
 $\{ \langle \text{'John'}, 120 \rangle, \langle \text{'Mary'}, 130 \rangle, \langle \text{'Peter'}, 110 \rangle, \langle \text{'Tom'}, 120 \rangle \} \subseteq \{ \text{'John'}, \text{'Mary'}, \text{'Peter'}, \text{'Tom'} \} \times \text{entiers} > 100$
- Un **attribut** indique le rôle joué par un domaine dans une relation.  $\text{domaine}(\text{Salary}) = \text{entiers} > 100$

# Domaine, Relation, Attribut, Schéma

- Un **domaine** est un ensemble de valeurs atomiques (chaînes, nombres,...).  $\{ \text{'Furniture'}, \text{'Toys'}, \text{'Garden'} \}$ , entiers  $> 100$
- Une **relation** est un sous-ensemble du produit cartésien d'un ensemble de domaines.  
 $\{ \langle \text{'John'}, 120 \rangle, \langle \text{'Mary'}, 130 \rangle, \langle \text{'Peter'}, 110 \rangle, \langle \text{'Tom'}, 120 \rangle \} \subseteq \{ \text{'John'}, \text{'Mary'}, \text{'Peter'}, \text{'Tom'} \} \times \text{entiers} > 100$
- Un **attribut** indique le rôle joué par un domaine dans une relation.  $\text{domaine}(\text{Salary}) = \text{entiers} > 100$
- La structure de la relation (**schéma**) est donnée par son nom et par un ensemble d'attributs. *Employees (firstname, salary, address, dept)*

L'**interprétation** ou **spécification** d'une relation est un prédicat :

Employees (firstname, salary, address, dept)

*/\*  $\langle n, s, a, d \rangle \in \text{Employees} \iff$  l'employé identifié par son nom  $n$  gagne un salaire  $s$ . Il habite à l'adresse  $a$  et est affecté au rayon  $d$ . \*/*

Le prédicat est utile pour comprendre le schéma de la relation et le documenter.

# Contraintes Relationnelles

- Contraintes de domaine :  
 $domain(A)=T$  spécifie que les valeurs de A doivent être du type T.

# Contraintes Relationnelles

- Contraintes de domaine :  
 $domain(A)=T$  spécifie que les valeurs de A doivent être du type T.
- Contraintes d'identification :  
X spécifie une contrainte d'unicité telle que les n-uplets de la relation sont distincts deux à deux pour X (X un ensemble d'attributs).

# Contraintes Relationnelles

- Contraintes de domaine :  
 $domain(A)=T$  spécifie que les valeurs de A doivent être du type T.
- Contraintes d'identification :  
 $\underline{X}$  spécifie une contrainte d'unicité telle que les n-uplets de la relation sont distincts deux à deux pour X (X un ensemble d'attributs).
- Contraintes d'intégrité référentielle :  
*R projetée sur l'attribut X se réfère à S projetée sur l'attribut Y* : tous les n-uplets de R, restreints à X doivent avoir un n-uplet correspondant dans S restreinte à Y. Ce que l'on note  $R[X] \subseteq S[Y]$ .
- Autres



## Exemple

Employees (firstname, salary, address, dept) /\* *firstname est l'identifiant* \*/

*/\*  $\langle n, s, a, d \rangle \in \text{Employees} \iff$  l'employé identifié par son nom  $n$  gagne un salaire  $s$ . Il habite à l'adresse  $a$  et est affecté au rayon  $d$ . \*/*

Leaderships (boss, dept) /\* *2 identifiants : boss et dept* \*/

*/\*  $\langle b, d \rangle \in \text{Leaderships} \iff$  l'employé  $b$  est responsable du rayon  $d$ . \*/*

Contraintes d'intégrité référentielle :

Leaderships[boss, dept]  $\subseteq$  Employees[firstname, dept]

firstname	salary	address	dept
John	120	Randwick	Toys
Mary	130	Wollongong	Furniture
Peter	110	Randwick	Garden
Tom	120	Botany Bay	Toys

boss	dept
John	Toys
Mary	Furniture
Peter	Garden