



Franz Girardin

Bases de données

NOTES DE COURS

Département d'informatique et de Recherche
Opérationnelle

Montréal 2025

11 janvier 2025

Table des matières

1	Introduction	1
1.1	Concepts généraux	1
1.2	Opérations Simples	1
1.3	Base de données	4
1.4	SGBD	5

1 Introduction

1.1 Concepts généraux

Définition 1.1 (Donnée). Une donnée est une **représentation d'un fait** à l'aide d'un code binaire stocké en mémoire.

Phénomène -----> 1011100 🍷 🍏 🐧

Les **types de données** représentent la nature du codage utilisé pour **représenter les données** ainsi que l'ensemble des opérations exécutables sur celles-ci.

Concept 1.2. Il faut savoir faire la **distinction** entre une donnée et une information :

$\underbrace{\text{Marie a 5 ans}}_{\text{Information}} : \underbrace{5}_{\text{Donnée (âge)}}$

1.2 Opérations Simples



Soit le tableau suivant, nous pouvons utiliser des commandes SQL pour manipuler les données.

MAT	NOM	FONCTION	COURS	AN_ENT
62945	Gilles	Prof_adj	MRT1111	1997
34560	Myriam	Prof_agr	MRT2221	1993
21539	Claudine	Prof_adj	MRT3331	1999
80200	Bernard	Prof_tit	MRT1112	1982
75902	Yida	Prof_agr	MRT1664	1990

Exemple 1 Sélection d'éléments via SQL

```
SELECT NOM, FONCTION
FROM PROFESSEUR
WHERE AN_ENT > 1992;
```

↓

NOM	FONCTION
Gilles	Prof_adj
Myriam	Prof_agr
Claudine	Prof_adj

SELECT spécifie la colonne à sélectionner
FROM spécifie la BD, soit le tableau PROFESSEUR
WHERE spécifie permet d'introduire une condition

Exemple 2 Insertion d'un élément via SQL

```
INSERT INTO PROFESSEUR (MAT, NOM, FONCTION, COURS, AN_ENT)
VALUES (662301, 'Jian', 'Prof_adj', 'MRT2323', 1996);
```

↓

MAT	NOM	FONCTION	COURS	AN_ENT
62945	Gilles	Prof_adj	MRT1111	1997
34560	Myriam	Prof_agr	MRT2221	1993
21539	Claudine	Prof_adj	MRT3331	1999
80200	Bernard	Prof_tit	MRT1112	1982
75902	Yida	Prof_agr	MRT1664	1990
662301	Jian	Prof_adj	MRT2323	1996

INSERT INTO insère des données dans la table
VALUES fournit les valeurs à insérer pour chaque colonne

Exemple 3 Mise à jour d'un élément via SQL

```
UPDATE PROFESSEUR
SET COURS = 'MRT2325'
WHERE MAT = 662301;
```

↓

MAT	NOM	FONCTION	COURS	AN_ENT
62945	Gilles	Prof_adj	MRT1111	1997
34560	Myriam	Prof_agr	MRT2221	1993
21539	Claudine	Prof_adj	MRT3331	1999
80200	Bernard	Prof_tit	MRT1112	1982
75902	Yida	Prof_agr	MRT1664	1990
662301	Jian	Prof_adj	MRT2325	1996

UPDATE modifie les données dans une table
SET spécifie les nouvelles valeurs
WHERE définit une condition pour la mise à jour

Exemple 4 Destruction d'un élément via SQL

```
DELETE FROM PROFESSEUR
WHERE MAT = 662301;
```

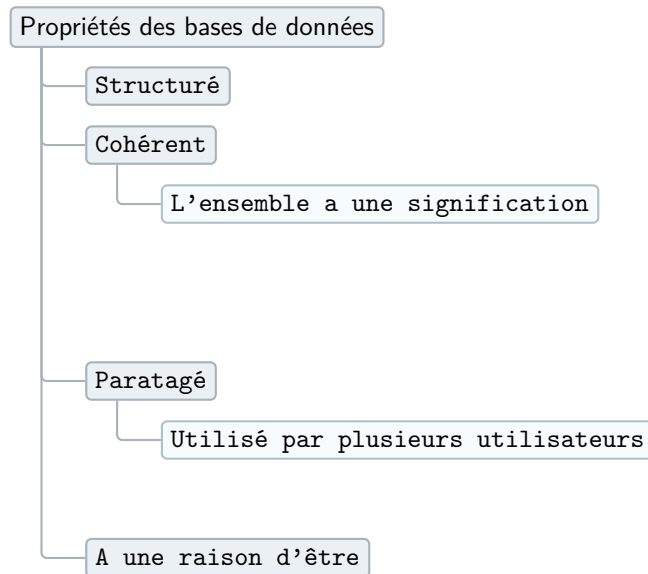
↓

MAT	NOM	FONCTION	COURS	AN_ENT
62945	Gilles	Prof_adj	MRT1111	1997
34560	Myriam	Prof_agr	MRT2221	1993
21539	Claudine	Prof_adj	MRT3331	1999
80200	Bernard	Prof_tit	MRT1112	1982
75902	Yida	Prof_agr	MRT1664	1990

DELETE FROM supprime les données d'une table
WHERE limite la suppression aux lignes qui respectent une condition

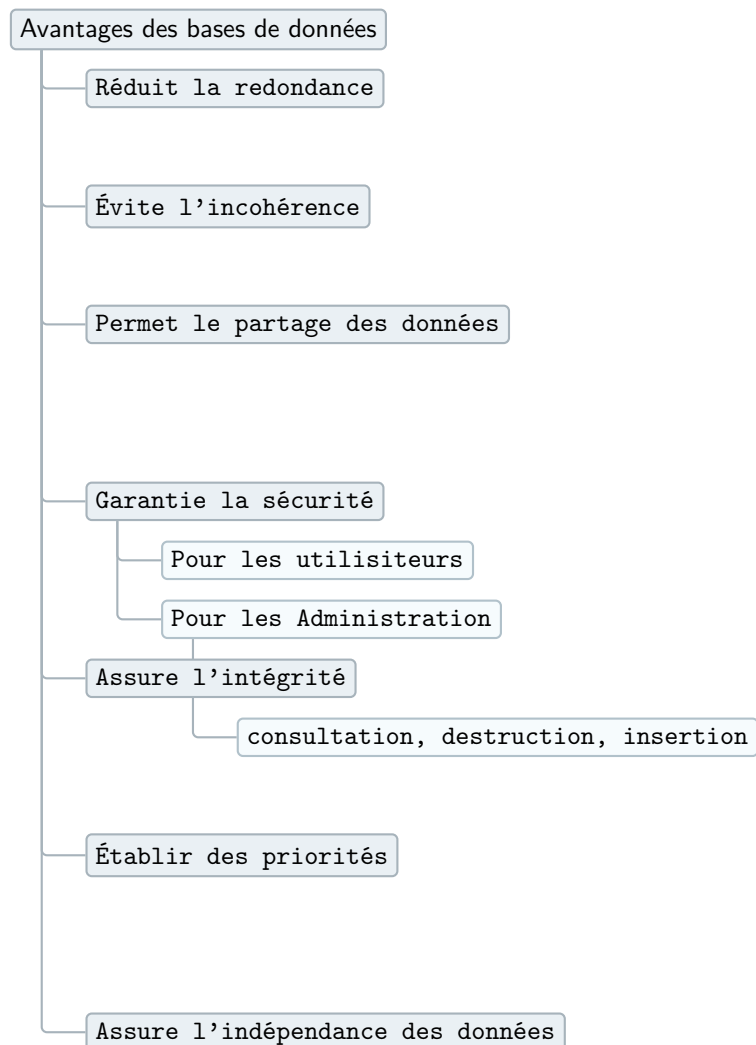
1.3 Base de données

Concept 1.3 (Bases de données). Les larges collections de données nécessitent des **systèmes**—les bases de données—et des **logiciels** pour être gérées.



Note

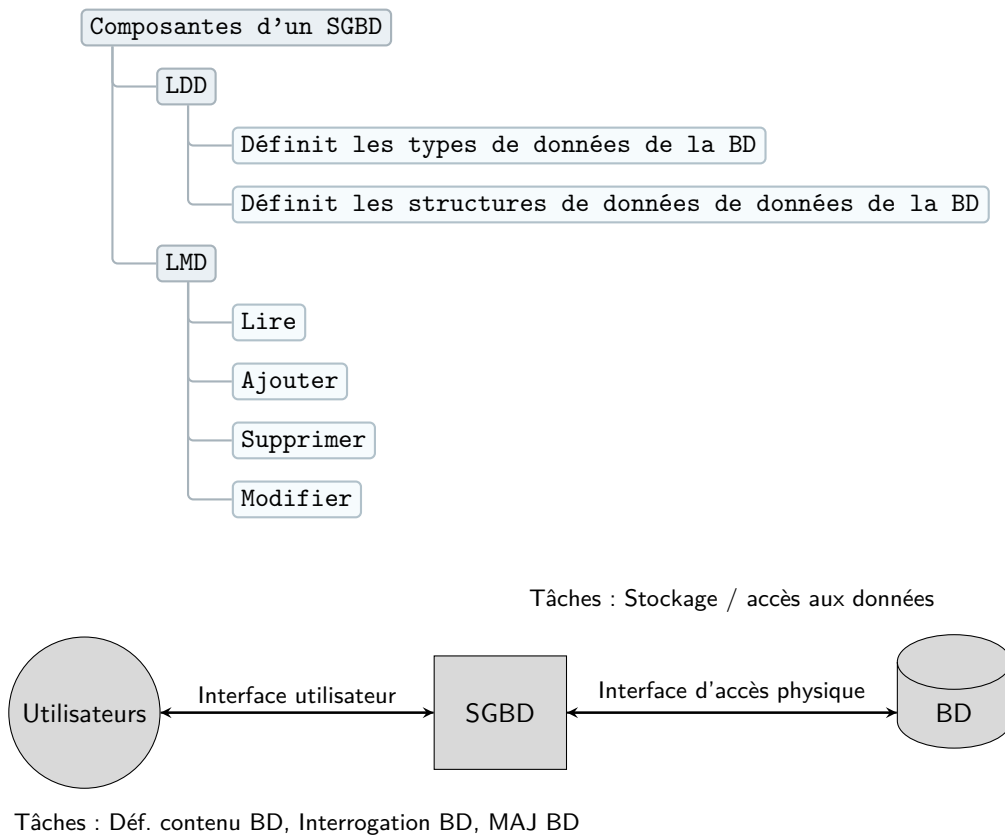
On retrouve **trois type d'acteurs** qui interagissent avec les BD, soit les concepteurs, les administrateurs et les utilisateurs finaux.



1.4 SGBD

Concept 1.4 (Système de gestion de base de données). Les SGBD sont des **logiciels** spécialisés qui permettent la définition, la construction, la manipulation et la maintenance des BD. Ils reposent sur les **langage de définition de données** (LDD) et les **langage de manipulation de données** (LMD) :

$$\text{SGBD} = \text{LDD} + \text{LMD}$$



Concept 1.5 (Architecture à 3 niveaux d'un SGBD). L'architecture d'un SGBD est organisée en trois niveaux distincts. Le **niveau externe** représente la vue des utilisateurs et des programmes qui interagissent avec le système, adaptée à leurs besoins spécifiques. Le **niveau conceptuel** offre une vue commune et intermédiaire qui est indépendante des applications et de l'implantation physique, permettant de se concentrer sur la structure logique des données. Enfin, le **niveau interne** se concentre sur la gestion des données telles qu'elles sont stockées physiquement, en prenant en compte les détails techniques de leur organisation. Cette architecture à trois niveaux garantit une abstraction efficace, une indépendance des données et une gestion optimisée.

Exemple 5 Illustration des niveaux conceptuel, interne et externe

Conceptuel : Définition logique des données.

Employee	
Num_emp	CHARACTER (6)
Num Dept	CHARACTER (46)
Salaire	NUMERIC

Interne : Définition physique et stockage des données.

STORED_EMP	LENGTH=20	
PREFIX	TYPE=BYTE(6)	OFFSET=0
EMP#	TYPE=BYTE(6)	OFFSET=6,
		INDEX=EMPX
DEPT#	TYPE=BYTE(4)	OFFSET=12
PAY	TYPE=FULLWORD	OFFSET=16

Externe 1 : Représentation spécifique pour une application.

```
typedef struct {
    char mat[6];
    float sal;
} employe;
```

Externe 2 : Une autre vue externe adaptée à une application.

```
typedef struct {
    char mat[6];
    char dept[46];
} employe2;
```

Définition 1.6 (Architecture client-serveur d'un SGBD). L'architecture client-serveur dans un système de gestion de base de données (SGBD) repose sur la séparation des rôles entre un client et un serveur. Le programme d'application agit comme client, fournissant une interface utilisateur graphique (GUI) et prenant en charge le traitement lié au domaine d'application. Le SGBD, quant à lui, joue le rôle de serveur, en fournissant l'accès aux données, souvent appelé "data server".