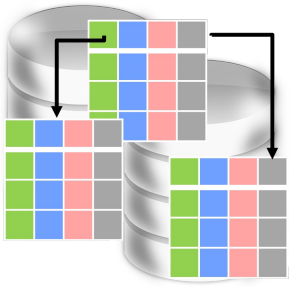


**UNIVERSITE DE CORSE**  
**Master informatique 1<sup>ère</sup> année**  
**Parcours DFS et DE**  
**2025-2026**

**BD Partie 1**  
**Bases de données**  
**relationnelles et Optimisation**



Evelyne VITTORI  
vittori\_e@univ-corse.fr



# Objectifs Partie 1

## Bases de données et Optimisation

- Renforcer et enrichir ses connaissances dans la manipulation de bases de données relationnelles et les outils associés.
- Apprendre à optimiser les performances d'une Base de données et en particulier l'exécution des requêtes.
- Découvrir les outils de statistiques et d'optimisation fournis par un SGBD pour le « tuning SQL »



Mise en pratique sur le SGDB PostgreSQL

# Pourquoi ce cours?

- Développeur « Full stack » : rôle multiple (concepteur/développeur/DBA)
- Data Engineer : Rôle fondamental de la gestion des données
- Sans être un spécialiste, vous devez acquérir des compétences dans l'optimisation de BD
- Trois niveaux de compétences à renforcer et/ou acquérir:
  - Modèle et données
  - Requêtes et Programmation
  - Outils et démarches d'optimisation (Indexation)

Recul technique



Force de proposition

# Préambule

- Commençons par quelques rappels pour se situer dans le contexte des BD en général
  - Notions fondamentales de l'univers BD
- Puis réfléchissons à l'optimisation
  - Qu'est-ce que l'optimisation?
  - Qui optimise? Quand? Comment?
- Comment allons-nous aborder cette problématique dans ce cours?
  - Plan du cours , TP







# Les notions fondamentales de l'univers des bases de données



UNIVERSITÀ DI CORSICA  
PASQUALE PAOLI



# Les données sont partout et il faut les stocker!!

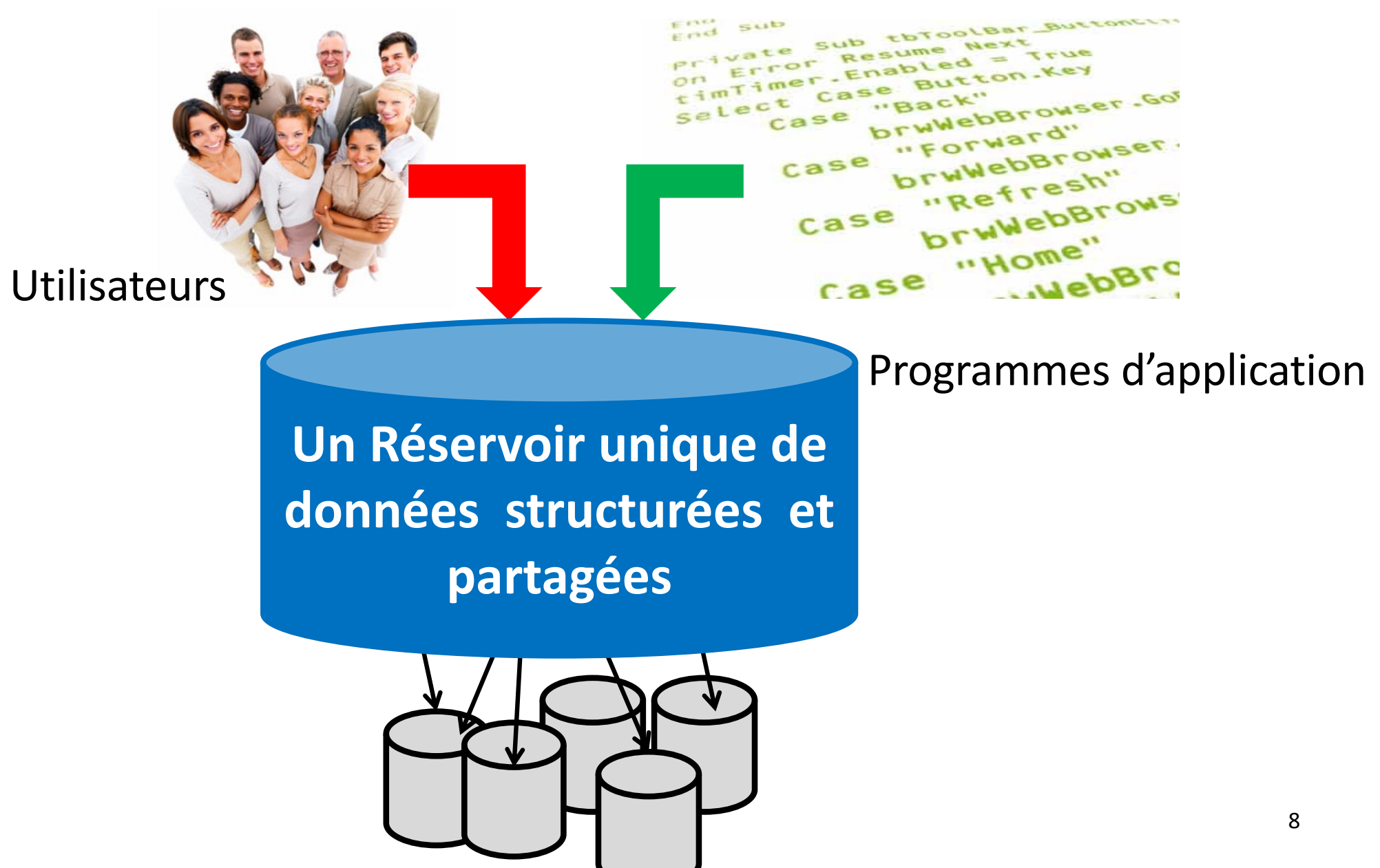
Tous les programmes ont besoin de conserver  
des données de manière persistante.





Pourquoi les fichiers ne  
suffisent pas?

# Qu'est-ce qu'une base de données?





# Que proposent les Systèmes de Gestion de Bases de Données?



SGBD = ensemble d' outils pour

- La **définition** de la structure des données
- La **mise à jour** des données
- La **recherche** de données
- Le **contrôle** des données

Langages textuels  
ou  
Interfaces  
graphiques

# Pourquoi construire la Base de données avant de commencer à coder?

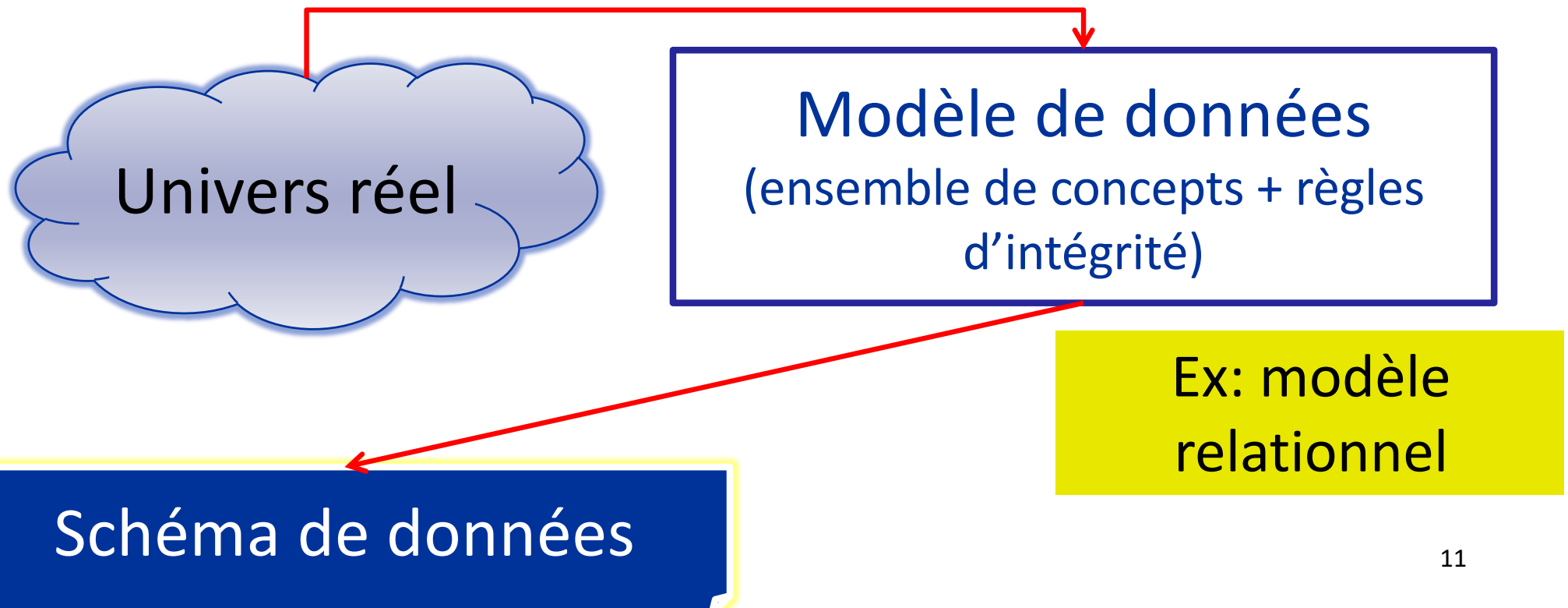


De mauvaises fondations peuvent conduire à l'effondrement !!

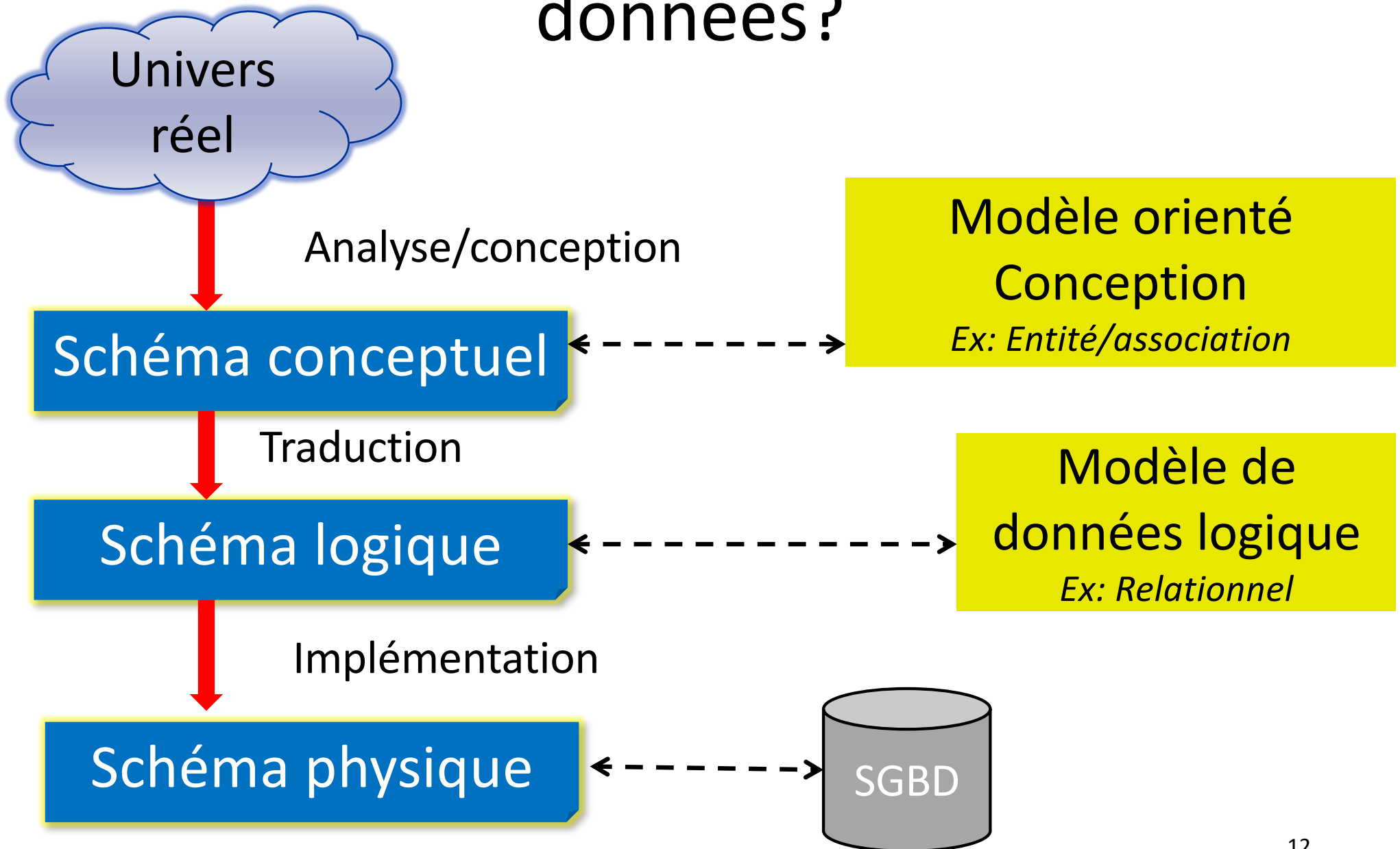
# Pourquoi a-t-on besoin d'un modèle de données?

- Pour savoir en quels termes décrire la structure des données:

Tables? Entités? Attributs? Champs? .....



# Pourquoi utiliser plusieurs modèles de données?



L'indépendance **Logique-Physique**  
a fait le succès des SGBD  
relationnels.

Qu'est-ce que cela signifie?



# Architecture Ansi/Sparc d'un SGBD



Développeurs d'application



Utilisateurs interactifs

Vue  
utilisateur

Vue  
utilisateur

Vue  
utilisateur

Niveau EXTERNE

**SCHEMA LOGIQUE**

Niveau LOGIQUE



ADMINISTRATEUR DBA

fichiers



Niveau PHYSIQUE/ INTERNE



ADMINISTRATEUR  
DBA



Quel est le rôle de l'administrateur ?

**DBA: DataBase Administrator**

# Que font les utilisateurs et qui sont-ils?



- **Mise à jour** des données  
(à travers des vues)
- **Consultation** des données

DML  
Data Manipulation Language

- Recherche à partir d'une interface: Formulaires, Web...
- Langages graphiques (QBE) (éventuellement)

UTILISATEURS interactifs

- Développent des interfaces pour les utilisateurs interactifs
- Utilisent **SQL intégré** dans un langage de programmation

PROGRAMMEURS d'application

# De l'origine aux SGBDr

- Années 1960-1970 :  
SGBD Hiérarchiques (IMS)  
SGBD en Réseau (CODASYL, SOCRATE)
  - Structures très contraignantes
  - forte dépendance logique/physique
  
- Années 1980 - aujourd'hui  
**SGBD Relationnels**
  - Vision tabulaire
  - Manipulation non-procédurale
  - Indépendance physique/logique



Leaders du marché  
actuel

# Principaux SGBD relationnels

	BD petites à moyennes <1Go	BD moyennes à grandes Client/Serveur
SGBD propriétaires	Access (Microsoft)	<ul style="list-style-type: none"><li>■ Oracle</li><li>■ SQLServer (Microsoft)</li><li>■ DB2 (IBM)</li><li>■ SyBase</li></ul>
SGBD openSource	<ul style="list-style-type: none"><li>■ OOBBase</li></ul>	<ul style="list-style-type: none"><li>■ PostgreSQL</li><li>■ MySQL (Oracle)</li></ul>



# SGBD ORACLE



- Leader mondial des SGBD relationnels
- Société *Oracle Corporation* (1977)
- Dernière Version : Oracle Database 21c (2021)
- Version gratuite Oracle XE (Express Edition) (12c depuis 2019)
- Nombreux outils d'administration et de développement:
  - définition/manipulation des données
  - cohérence, confidentialité, intégrité
  - sauvegarde, restauration, concurrence

# SGBD PostGreSQL



- Issu du projet PostGres de l'université de Berkley (1986) faisant suite au SGBD pionnier INGRES (1973)
- 1995 : 1<sup>ère</sup> version de PostgreSQL
- Gratuit et **Open source**
- Supporte SQL avancé
- Multiplateformes, Sécurité, performances
- Communauté très active
- **Extensible**: Autres types de données (JSON, ...)



Michael Stonebraker  
Professeur MIT

Notre choix cette année

# Le post-relationnel

Les **concurrents** et **extensions** du relationnel

- SGBD Objets (90-99): une impasse!
  - développement des SGBD relationnels-objets

- **SGBD NoSQL**

(BigTable, MongoDB , Dynamo,,...)

- BD géantes sur le Web
- Langage de requêtes spécifiques
- Simplicité, gestion des montées en charge




























# Etat des lieux en 2025...

## Comparaison de la popularité des SGDB du marché

Source: <https://db-engines.com/en/ranking>

424 systems in ranking, August 2025

Rank			DBMS	Database Model	Score		
Aug 2025	Jul 2025	Aug 2024			Aug 2025	Jul 2025	Aug 2024
1.	1.	1.	Oracle	Relational, Multi-model 	1220.70	+3.64	-37.78
2.	2.	2.	MySQL	Relational, Multi-model 	915.46	-25.26	-111.40
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model 	754.15	-16.99	-61.02
4.	4.	4.	PostgreSQL	Relational, Multi-model 	671.25	-9.63	+33.87
5.	5.	5.	MongoDB 	Document, Multi-model 	395.58	-8.25	-25.40
6.	6.	 7.	Snowflake	Relational	178.90	+2.73	+42.93
7.	7.	 6.	Redis	Key-value, Multi-model 	147.19	-2.53	-5.52
8.	8.	 9.	IBM Db2	Relational, Multi-model 	127.31	-0.20	+4.30
9.	 12.	 15.	Databricks	Multi-model 	115.82	+7.78	+31.36
10.	 9.	 8.	Elasticsearch	Multi-model 	114.27	-4.56	-15.56
11.	 10.	 10.	SQLite	Relational	112.59	-2.84	+7.80
12.	 11.	 11.	Apache Cassandra	Wide column, Multi-model 	108.51	-0.24	+11.51
13.	13.	 14.	MariaDB 	Relational, Multi-model 	93.59	-1.85	+7.02



# Problématique de l'optimisation BD



UNIVERSITÀ DI CORSICA  
PASQUALE PAOLI





# Qu'est-ce que l'optimisation d'une base de données?

- Optimiser une base de données consiste à améliorer les performances de son fonctionnement:
  - Interrogations, Mises à jour
- Performance= rapidité mais aussi fiabilité
  - La problématique de l'optimisation consiste à améliorer les performances en assurant l'intégrité.

En particulier lors des montées en charge

Qui sont les acteurs concernés?  
Sur quoi peuvent-ils agir?  
Quels sont les outils pour les aider?

# Qui sont les acteurs concernés?

## Sur quoi peuvent-ils agir?

- **Conception** du schéma relationnel
  - Construire un **schéma** relationnel «**de qualité** »

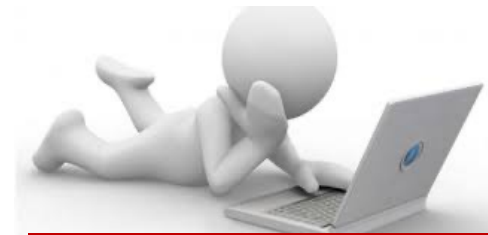
Normalisation ou dénormalisation étudiée et choisie de manière raisonnée

- **Développement**

- Concevoir des **requêtes** et procédures stockées qui évitent la pénalisation des performances



Concepteur



Developpeur

«SQL Tuning »

# Qui sont les acteurs concernés?

## Sur quoi peuvent-ils agir?

### ■ **Administration** de la BD

Rôle du DBA (Data Base Administrator)

Optimiser les performances de l'exécution des requêtes en agissant à deux niveaux:



#### ■ Modifier l'organisation physique de la BD :

- Paramètres d'initialisation
- Indexation et organisation du stockage physique

«Database  
Tuning »

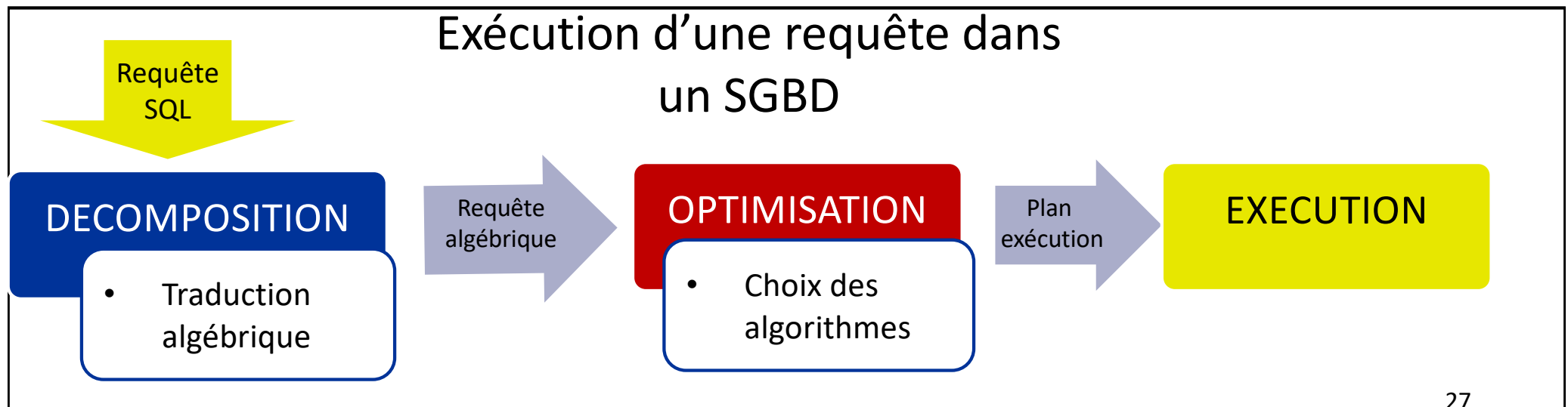
#### ■ Influencer le SGBD dans le choix des algorithmes d'exécution des requêtes

«SQL Tuning »

# Qui sont les acteurs concernés?

Le principal: LE **SGBD** lui même

- Tous les SGBD possèdent un **optimiseur** automatique intégré:
  - Les optimiseurs d'ORACLE et de PostgreSQL sont de type **CBO (Cost Based Optimizer)**
    - Ils déterminent le meilleur algorithme d'exécution d'une requête en fonction des coûts associés



# Quels outils pour aider l'optimisation?

- Le SGBD fournit des **outils d'évaluation** (*plus ou moins élaborés!*) qui permettent de guider l'optimisation:
  - Générateur des Plans d'exécution des requêtes
  - Outils statistiques portant sur les objets de la base de données
    - Nombre de tuples dans les tables
    - Temps d'exécution des requêtes
    - ...



# Démarche d'optimisation des requêtes

La démarche d'optimisation d'une requête est essentiellement empirique :

- **Evaluer** les performances de la requête
  - Analyse des plans d'exécution et des mesures statistiques
- **Appliquer** une ou plusieurs solutions techniques
  - Indexation et stockage physique des tables
- **Evaluer** l'amélioration



# Bases de données et Optimisation

## Organisation du cours



UNIVERSITÀ DI CORSICA  
PASQUALE PAOLI



# Organisation du cours

- Trois niveaux de compétences à renforcer et/ou acquérir:
  - Modèle et données
  - Requêtes et Programmation
  - Outils et démarches d'optimisation (Indexation)

# Modèle et données



## 1. Maîtriser les concepts du modèle relationnel

- Principes de l'intégrité des données

## 2. *Maîtriser la normalisation*

- Comprendre l'objectif du processus
- Savoir l'appliquer
- Savoir décider de ne pas normaliser

partie abordée en  
partie 2

Concilier  
Performance et  
maintien de  
l'intégrité

## 3. Savoir implémenter un schéma relationnel dans un SGBD en favorisant ses performances

- Choix des types de données
- Utilisation du DDL (**SQL-Data Definition Language**)
  - Contraintes (check)
- Définition de **triggers**

# Requêtage et programmation BD

1. Maîtriser le langage SQL DML (**Data Manipulation Language**)
  - Requêtes de mise à jour
  - Requêtes d'interrogation
2. Savoir faire des choix pertinents lors de la définition des requêtes

« Tuning SQL »
3. Savoir définir des procédures stockées  
Langage procédural (PL/PGSL (PostgreSQL))
  - Existe aussi dans d'autres SGBDr :
    - ex: PL/SQL ORACLE , lang. procédural MySQL

# Index et Outils d'optimisation



## Prerequis:

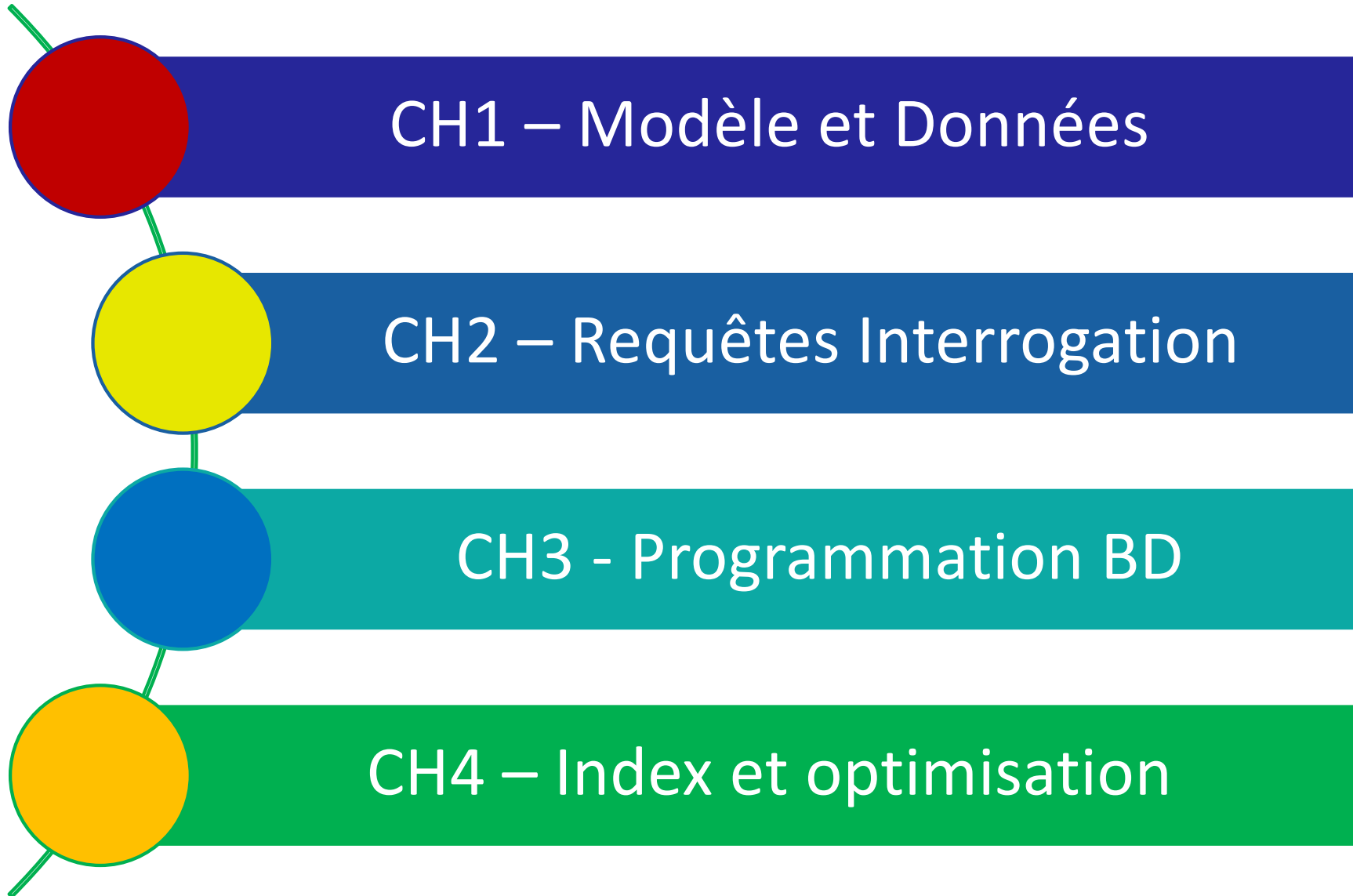
Maîtriser les concepts relationnels et le langage SQL (DDL et DML)

1. Comprendre la notion d'index et son utilité
2. Savoir utiliser les outils d'optimisation du SGBD
  - Comprendre les plans d'exécution fournis par l'optimiseur algorithmes d'exécution d'une requête en particulier jointures
  - Outils Statistiques sur les tables et objets de la BD
3. Savoir influencer l'exécution des requêtes
  - Utiliser les outils pour identifier les **index** et structures physiques **efficaces**
  - Savoir influencer l'optimiseur par des conseils (hints)  
*(ORACLE uniquement)*

« Tuning SQL »



# Optimisation et Bases de données



# Déroulement indicatif des séances

Cours	Thèmes	TPCours	Projet
CH1 – Modèle et données : définition et manipulation	<ul style="list-style-type: none"> <li>•Révision concepts BD</li> <li>•Implémentation</li> </ul> Et Mises à jour <ul style="list-style-type: none"> <li>•Vues</li> <li>•Privilèges</li> <li>•Transactions</li> </ul>	Questions et exercices corrigés en séance	Parties 1 - 2 - 3
CH2 – Requêtes Interrogation	<ul style="list-style-type: none"> <li>•Algèbre relationnelle et optimisation</li> <li>•Requêtes SQL complexes</li> </ul>		Partie 4
CH3 – Programmation BD	<ul style="list-style-type: none"> <li>•Procédures stockées</li> <li>•Triggers</li> </ul>		Partie 5
CH4 - Optimisation	<ul style="list-style-type: none"> <li>•Index</li> <li>•Principes tuning SQL</li> <li>•Outils d'optimisation</li> </ul>		Partie 3

# Projet Conception Assistée et optimisation

## Modéliser l'écosystème de la recherche



### Travail à faire

Travail en groupe de 2 à 3 étudiants

Créer une base de données à partir d'un cahier des charges indicatif, y insérer des données fictives de manière automatique et mettre en pratique les notions abordées dans le cadre du cours

### Objectifs

- Tester les possibilités des outils d'IA générative pour faciliter et optimiser la phase de conception, en adoptant une posture critique.
- Générer automatiquement des données fictives et réalistes par définition de scripts python (bibliothèque Faker).
- Gestion Sécurisée des Accès par définition de rôles, vues et privilèges pertinents
- Tester les outils d'optimisation de postgreSQL pour la création de requêtes complexes
- Acquérir des compétences en PL-SQL (triggers et procédures stockées)

# Projet Conception Assistée et optimisation

## Modéliser l'écosystème de la recherche



BD avec minimum 7 tables  
Solution originale

### Cahier des charges indicatif (minimal)

- Institutions : Université, organisme de recherche, partenaire privé), adresse.
- Laboratoires (UMR) : identifiant, nom, rattachement institutionnel.
- Projets de recherche structurants : identifiant, titre, description, discipline, ...
- Contrats de financement : type (ANR, H2020, Région...), financeur, intitulé, montant, durée, date de début et de fin.
- Chercheurs : identité, ... , discipline, laboratoire de rattachement. A titre de simplification on supposera qu'un chercheur n'est impliqué que dans un seul projet structurant.
- Publications : On stocke dans la base de données, les métadonnées de ces publications uniquement (titre, auteurs, taille, DOI, date, ...).
- Un jeu de données est un ensemble de données (dataset) associées à un contrat de recherche (résultats de recherche). On stocke ici uniquement les descriptifs de ces jeux de données : les Métadonnées des jeux de données.
- Plan de Gestion des Données (DMP) : . Le DMP décrit la manière dont les données issues des projets financés par un contrat seront produites, documentées, partagées et archivées. Dans la base de données, on conservera pour chaque contrat : le statut du DMP (brouillon, soumis, validé), la date de validation, un lien vers le document complet (stocké sur un support externe).
- Les jeux de données financés par un contrat ne peuvent être officiellement déposés que si le DMP associé au contrat est validé

# Projet Conception Assistée et optimisation

## Modéliser l'écosystème de la recherche



### Critères d'évaluation

- originalité de la solution proposée et justification des choix
- qualité du schéma défini par rapport aux besoins précisés
- analyse critique de l'aide apportée par les outils d'IA générative
- pertinence des rôles, vues et privilèges
- maîtrise des solutions proposées et compréhension des choix de l'optimiseur pour les requêtes
- qualité et pertinence des procédures stockées et des triggers

### Documents à rendre

- Scripts SQL et Python
- Rapport synthétique contenant :
  - Description détaillée de la conception de la base.
  - Méthodologie pour l'utilisation des IA génératives et Python Faker.
  - Comparaison des requêtes complexes avec captures d'écran des outils de diagnostic

# Projet Conception Assistée et optimisation

Modéliser l'écosystème de la recherche



## Soutenances 12 novembre

- Présentation orale finale de 20 minutes décrivant la démarche et les résultats obtenus :
  - Schéma, données, vues, requetes, procédures, triggers
- Démonstration
- Proposition d'une méthodologie sur l'utilisation d'IA génératives



Définition des groupes le 23 septembre