

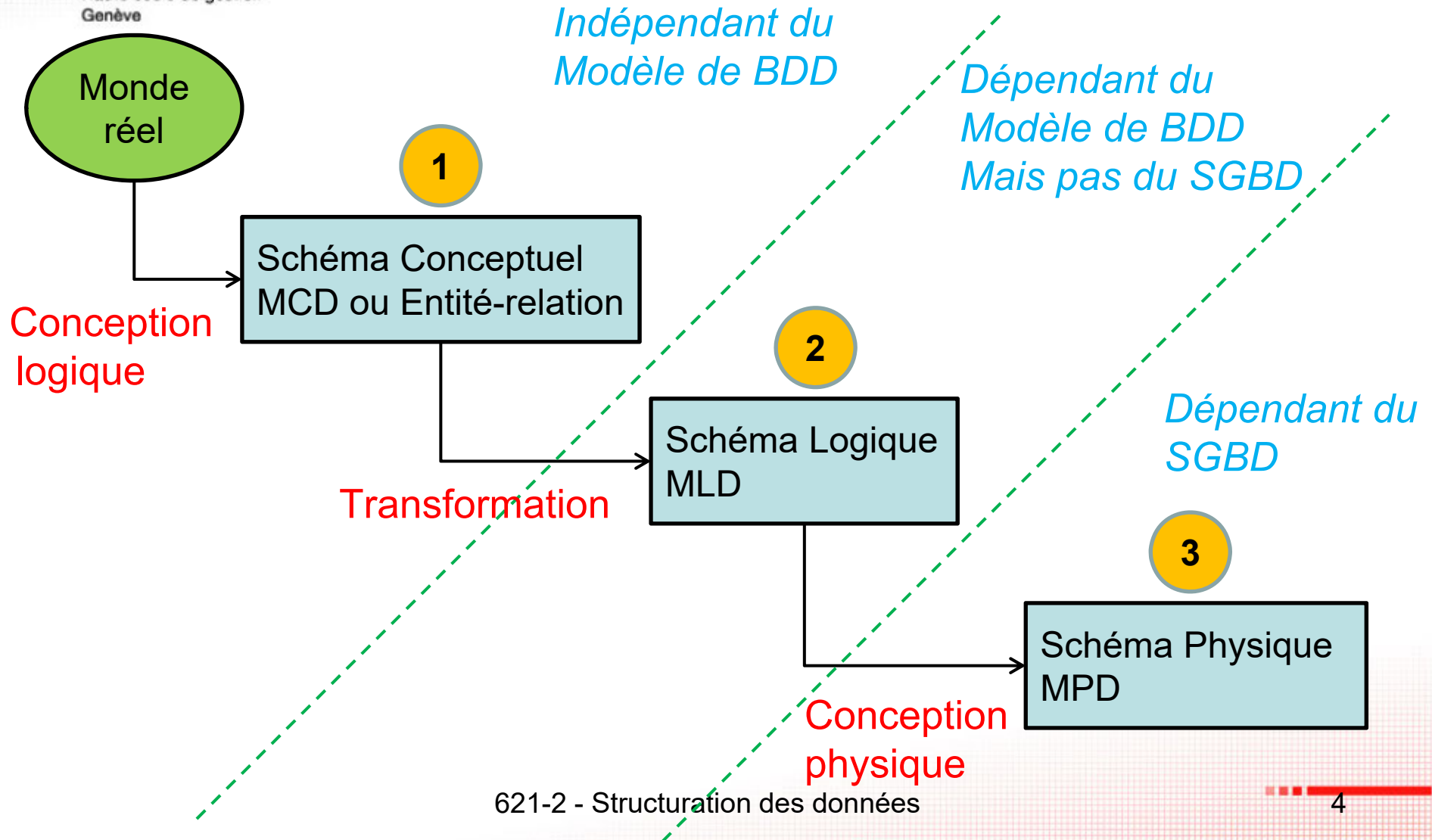
Le modèle relationnel

Christian Stettler

- Rappels (BDD / SGBD / CRUD)
- **Processus de Conception de BDD**
- **Origine du modèle relationnel**
- **Langage SQL**
- **Algèbre relationnel**
- **Création de table**

- BDD
- SGBD
- Modèles de BDD
- Opérations sur BDD (CRUD)

Processus de Conception d'une BDD



- SQL (**S**tructured **Q**uery **L**anguage) est un langage (pseudo-langage) non procédural
- C'est un langage de requêtes
- Permet de gérer et de manipuler un SGBD**R**

- SQL est un langage normalisé

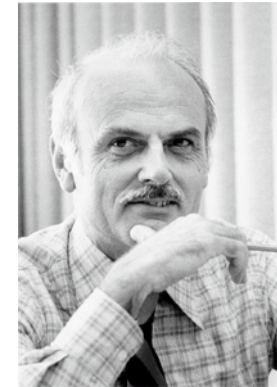
Année	Appellation	Commentaires
1986	SQL-86 ou SQL-87	Édité par l'ANSI puis adopté par l'ISO en 87
1989	SQL-89 ou SQL-1	Révision mineure.
1992	SQL-92 ou SQL2	Révision mineure.
1999	SQL-99 ou SQL3	Expressions rationnelles, requêtes récursives, déclencheurs, types non-scalaires
2003	SQL:2003	Introduction de fonctions pour la manipulation XML, « window functions », ordres standardisés et colonnes avec valeurs auto-produites (y compris colonnes d'identité)
2008	SQL:2008	Ajout de quelques fonctions de fenêtrage, limitation du nombre de ligne, amélioration mineure sur les types distincts, curseurs et mécanismes d'auto-incréments.

- Il y a une différence entre les normes SQL et la réalité ...
 - Les SGBD implémentent +- en profondeur les différentes normes SQL
 - Les SGBD implémentent de façon +- correcte les normes SQL
 - Les SGBD ajoutent souvent leur propres fonctionnalités qui ne sont pas normées

- SQL n'est pas sensible à la casse
- Le caractère point-virgule (;) est utilisé pour séparer des instructions SQL
- La mise en commentaire d'une ligne se fait par le double tiret --
- La mise en commentaire d'un bloc de lignes est faite par /* et */

Origines du modèle relationnel

- Proposé par Codd en 1970
- Objectifs
 - Haut degré d'indépendance
 - Index, chemins d'accès, représentation interne des données
 - Base solide pour traiter la cohérence et redondance
 - Langage de manipulation non procédural basé sur une théorie solide



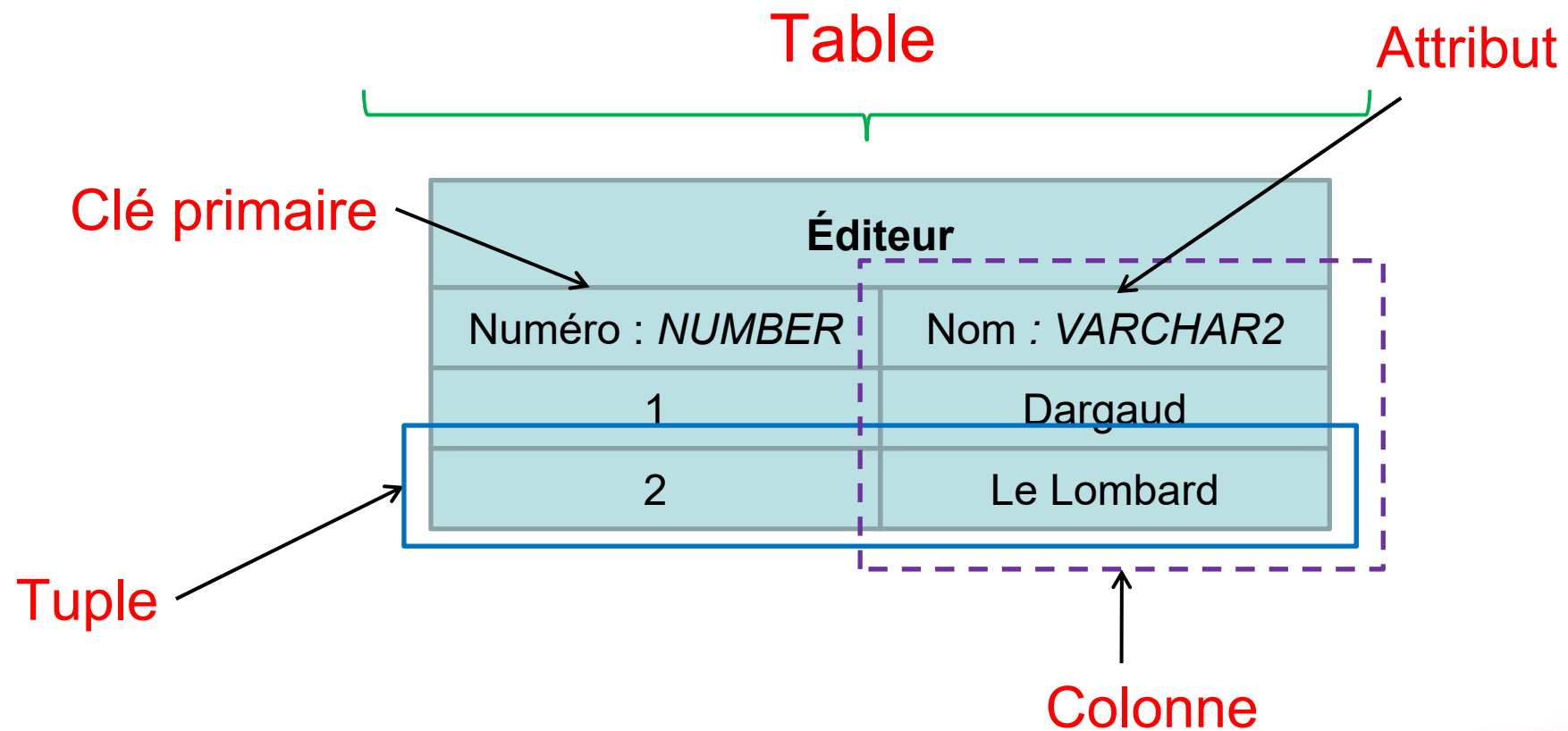
- Concepts structurels
 - Les structures logiques sont des **relations** basées sur des **domaines**
- Ensemble d'opérateurs de manipulation sur les relations
 - Exemple : Relation X UNION Relation Y = Relation Z
- Règles d'intégrité des données (contraintes)
 - Garantir la cohérence de la BDD sur la base de règles de gestion

- **Tout en 1 seul langage**

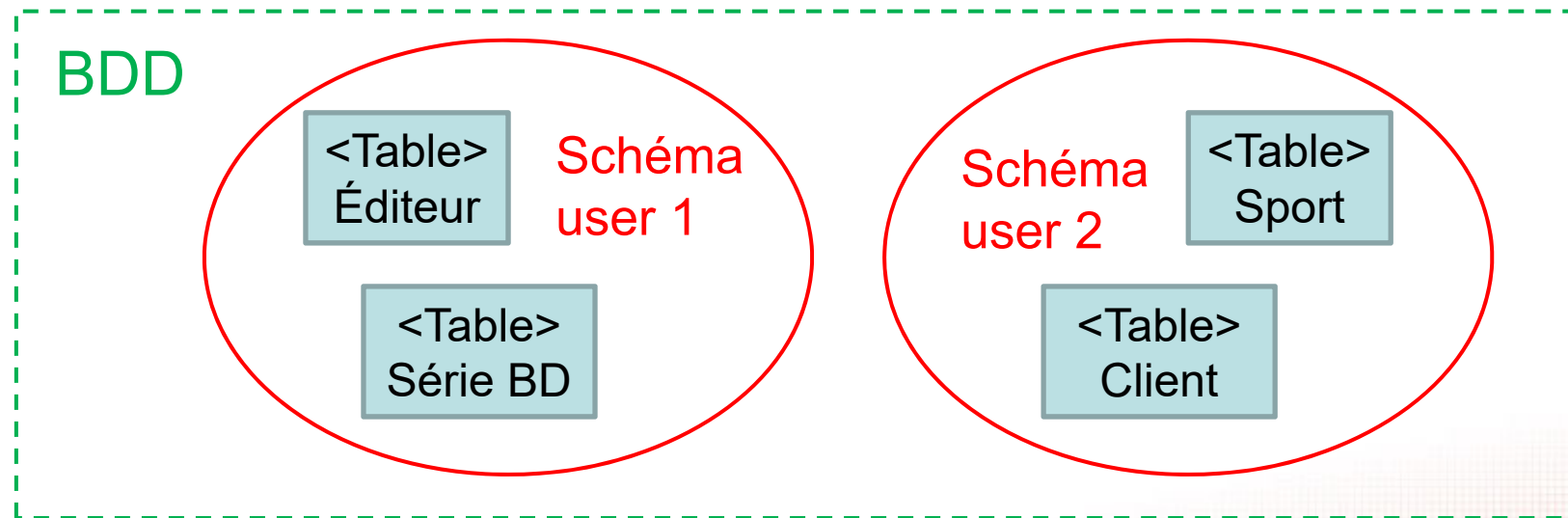
DDL	Data Definition Language	CREATE, ALTER, DROP
DML	Data Manipulation Language	INSERT, SELECT, UPDATE, DELETE
DCL	Data Control Language	GRANT, REVOKE
TCL	Transaction Control Language	SET, TRANSACTION, COMMIT, ROLLBACK
Embedded SQL	SQL intégré	SET, DECLARE, CURSOR, OPEN, FETCH

- **Table**
 - Représente une entité (MCD) au niveau du SGBD
 - Structure à deux dimensions
 - Chaque table a une clé primaire qui identifie de manière unique chaque tuple

- **Table**
 - Toutes les données sont enregistrées dans des entités appelées tables
 - La table matérialise la relation
 - Elle porte un nom identifiant
 - Les attributs de la relation sont matérialisés par les colonnes typées
 - Le nom d'une colonne est unique dans une table
 - En général, la table comporte une clé primaire



- Schéma
 - Ensemble des objets (tables ...) appartenant à un même propriétaire



- Définir un nom de table unique dans votre schéma
- Définir la structure de découpage d'enregistrements
 - Le nom de chaque colonne
 - Le type de chaque colonne
- Cette réflexion a normalement été faite lors de la conception de la BDD -> **MCD, MLD et MPD**

Syntaxe

```
CREATE TABLE nomTable (  
    nomColonne TYPE [DEFAULT exp]  
    [, nomColonne TYPE [DEFAULT exp]  
    , ...  
    , nomColonne TYPE DEFAULT exp]  
);
```

Exemple

```
CREATE TABLE bd_editeur (  
    edi_no    NUMBER(5),  
    edi_nom   VARCHAR2(30)  
);
```