Base de données

Définition

Une base de données est une **collection de données** électroniques utilisée pour **stocker**, **gérer** et **récupérer** des **informations** de manière efficace.

BDD relationnelle

2 catégories

BDD non relationnelle

Schéma : structure commune à l'ensemble des données

Schéma

Schéma **fixe** et défini à l'avance, les **données** doivent **correspondre**.

```
CREATE TABLE Joueurs
(
    id INT PRIMARY KEY NOT NULL,
    nom VARCHAR(100),
    prenom VARCHAR(100),
)
```

Création relation avec définition du schéma

```
INSERT INTO Joueurs (nom, prénom)
VALUES ('Jelonch', 'Anthony')
```

OK

```
INSERT INTO Joueurs (nom, prénom, capitaine)
VALUES ('Dupont', 'Antoine', true);
```

ERREUR

Pas de schémas

Aucune restriction sur les types des données stockées

```
{
  "nom": "Dupont",
  "prenom": "Antoine"
  "capitaine": true,
}
```

```
{
  "nom": "Jelonch",
  "prenom": "Anthony"
}
```

"capitaine" présent sur un élément et absent sur l'autre

Cohérence

Propriétés ACID à respecter

Atomicité → transaction OK ou annulée

Cohérence → contraintes respectées, ou erreur

Isolation → transaction indépendante, gestion concurrence

Durabilité → transaction terminée = modif. permanente

Cas d'usages différents et spécifiques

Sacrifier la cohérence stricte au profit de la **disponibilité** et de la **tolérance** aux partitions, en utilisant le **théorème CAP** (Consistency, Availability, Partition tolerance).

Voir '**CAP Theorem**' pour plus d'infos.

Syntaxe

SQL

```
SELECT nom, prenom
FROM Joueurs
WHERE prenom = "Antoine"
AND nom = "Dupont";
```

```
MATCH ( j:Joueur )
WHERE j.prenom = "Antoine"
AND j.nom = "Dupont"
RETURN j.nom, j.prenom
```

Cypher Neo4j