

1) SQL (Structured Query Language)

- ↳ Language pour exploiter BDD
- ↳ Définition / Manipulation / Contrôle de données
- ↳ Créer des tables / ajout entrée (ligne) / interroger BDD / M à j / gérer les données
- ↳ Angl norme International
- ↳ Créé par : Donald D Chamberlin & Raymond F. Boyce
- ⇒ Language standard pour la gestion de BDD relationnels

2) → Utiliser pour le stockage

2) :- développement & administration → Facebook / Spotify / Banking (Revolut) /

- Réseaux Sociaux
- Banks
- TT site avec des utilisateurs
- Cloud storage
- Cloud computing (AWS)
- SaaS

N'importe quelle société qui a besoin de stocker de la donnée relationnel.

3) → Stockage données en Back-End) Facebook / Twitter / Insta

→ Traitement des données

→ Classification) Wikipédia

→ Distribution


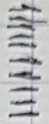

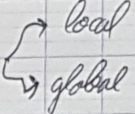
Utiliser pour apps comme Revolut (Bank)
SaaS (AWS).

4) → Type :

- Entier
- Caractère
- Valeur Monétaire
- date
- Temps
- Binaire

Possibilité de définir son propre type avec Transact SQL, Microsoft .NET Framework

- Exact Numerics
- Approximate
- Date and Time
- Character strings
- Unicode character strings
- Binary strings

- Structure:
- Stack:  FIFO: First in First out (Queue ligne) 
 - Tree: Structure hiérarchique (nœud parent/enfant / racine)
 - ↳ + populaire: Binary tree 
 - ↳ nœud associé à valeur clé (key value)
 - Wide Table (jusqu'à 30 000 colonnes): tableau avec "colonne de remplissage" pour stocker Null valeurs
 - ~~Diff~~ System Tables: Master DB (préfixe sys): accessible vers system
 - Partitioned Tables: données séparées en case répartis dans plusieurs fichiers (jusqu'à 15 000 partitions)
 - Temporary tables: tables temporaires pour partage: 
 - ↳ local
 - ↳ global
 - Hash table: paire clé - valeur ⇒ seulement en local
 - ↓
 - index
 - Heap table: pas d'index de groupement / les lignes sont stockées Random
 - Graph: 1 seul par DB: collection de tables
 - ↳ Réseau social (nœud et bord).

- 5) Avantages:
- Traitement rapide des requêtes (grande quantité d'infos rapidement)
 - Pas trop de compétences en codage
 - Standardisé
 - Portable (utilisable partout: serveur, PC, laptop, API etc...)
 - Plusieurs manières de voir les données.

- Désavantages:
- Interface complexe: inconfort
 - Parfois coûteux: donc difficile d'accès
 - Contrôle partiel: