

Organisation des données de base

Géovisualisation dynamique et traitement de données

Christian Kaiser – Cours 2



Étapes d'un projet de géovis

1. Évaluer les possibilités **Analyse exploratoire** **+ Récolte des données**
2. Décrire et schématiser le produit final **Planification conceptuelle**
 - .. Réfléchir sur les possibilités de visualisation
 - .. Dessiner, discuter, réfléchir encore
3. Choix des technologies **Planification technique**
 - .. Évaluer plusieurs alternatives, choisir la meilleure option
3. Créer une maquette **Tests de réalisation**
 - .. Avoir une première version (non fonctionnelle) aussi vite que possible
 - .. Discuter, réfléchir, modifier, etc.
4. Progressivement implémenter le produit **Réalisation**
 - .. Améliorer le prototype progressivement
 - .. Tester, discuter, réfléchir, modifier, etc.
5. Tester, tester, tester! Documenter! **Finalisation**

Récolte des données

- .. Déterminer quelles données sont disponibles
 - .. Nos propres données? Partenariat?
 - .. Open data?
 - .. Données sous licence? Coûts? Droits?
- .. Nécessité d'acquisition de données?
 - .. Mesures, enquêtes, moissonnage, etc.
- .. Quelle **structure de données**?

Structure des données

- .. Tableau de données simple
 - .. Excel, fichier CSV ou TSV, SPSS, etc.
- .. Bases de données relationnelles (SQL)
 - .. Modèle de données, normalisation etc.
- .. Bases de données No-SQL

Tableau simple

- ⊕ Simple! Facile à échanger.
CSV ou TSV facile à lire dans un logiciel ou script. S'intègre avec VCS (p.ex. Git)
- ⊖ Difficile à représenter des relations ou données complexes.

→ Préférer format texte sur binaire (CSV, TSV)

CSV = comma-separated values; parfois point-virgule ou tabulateur comme séparateur

TSV = tab-separated values

VCS = version control system (on regardera ça plus tard)

BD relationnelle

- .. Tables liées, colonnes avec type fixe:
varchar, text, integer, double, decimal, date, datetime, blob
- ⊕ Structure des données consistante.
Normalisation de la structure assure une flexibilité des requêtes SQL.
- ⊖ Pas de colonnes flexibles / structure fixe
Types de données limitées
(p.ex. pas d'arrays ou objets par défaut)

BD relationnelle

- .. Exemples de SGBDR:
 - .. open-source: MySQL, SQLite, PostgreSQL
 - .. commercial: Oracle, MS SQL Server, MS Access

BD noSQL

- .. Structure de données flexible («schema-less», chaque «document» peut avoir des attributs différents)
- ..

```
{ "type": "Feature", "properties": { "id": 1, "name": "Lausanne", "pop": 127001 } }
```
- ⊕ Flexibilité. Rapidité d'insertion de données.
- ⊖ Possibilités de requêtes plus limitées.
Requêtes plus lentes. Pas de normalisation, transactions plus difficiles.

BD noSQL

- .. Exemples de BD noSQL:
 - .. Key-value databases: Redis, Memcached, Couchbase, Amazon DynamoDB
 - .. Document databases: MongoDB, CouchDB, (Firestore), ((PostgreSQL))
 - .. Graph databases: Neo4j, FlockDB, ...
 - .. Tabular databases (surtout pour big data): BigTable, HBase, Hypertable, ...

Fichier Shape dans BD?

- ou: comment stocker une géométrie dans une base de données?
- noSQL: GeoJSON
- BD relationnelle: extension géographique
 - p.ex. **PostGIS**, SQLite Spatial, **Oracle Spatial**, MySQL Spatial

PostgreSQL (+ PostGIS)

- .. Extensions par rapport au SQL standard
 - .. Géométrie = type de colonne
 - .. Array (plusieurs valeurs du même type dans un champ)
 - .. JSON / JSONB (document DB dans un champ)
- .. Opérations sur les géométries
 - .. UNION, INTERSECT, BUFFER, CONTAINS, etc.

PostGIS + logiciel SIG

- .. Intégration optimale dans QGIS
 - .. Charger des couches PostGIS (=table avec géométrie)
 - .. Éditer les géométries directement dans QGIS

Exercice PostgreSQL / PostGIS

1. Charger fichier Shape des communes suisses, et des noms de la carte 1:200k
2. Calculer la couche des cantons
3. Joindre une table de données (de l'OFS)
4. Déterminer le nombre de montagnes par canton

En même temps, nous prenons des notes comment nous procédons.