

C'est quoi ton SIG ?

exercice- book

100

A4 (210 x 297mm)

70g/m²



UK Où vous apprendrez à :

F

— rien sur les SIG,

D

— un peu de vocabulaire

ES

de base de donnée,

NL

I


P

Mes carnets QGIS

C'est quoi ton SIG ?

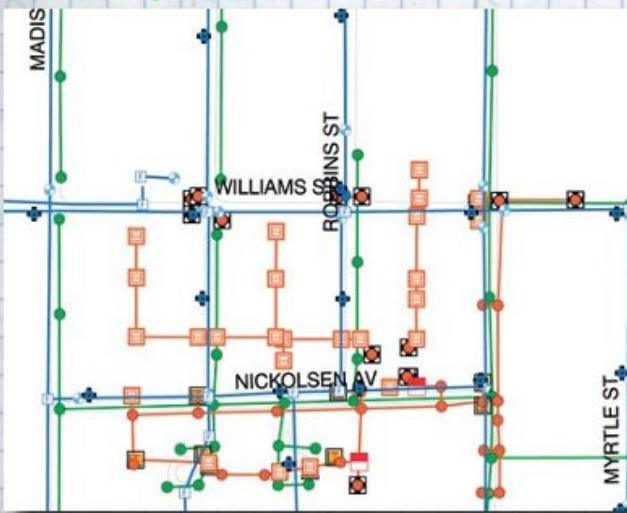
1

Je ne vais pas m'étaler sur le sujet car il existe des dizaines de sites, de tutoriels qui vous expliquent en long, en large et en 3D ce qu'est un SIG ( S.I.G => Solde Intermediaire de Gestion ??? mais non : Systeme d'Information Geographique ou GIS ^{in ENGLISH, dit}).

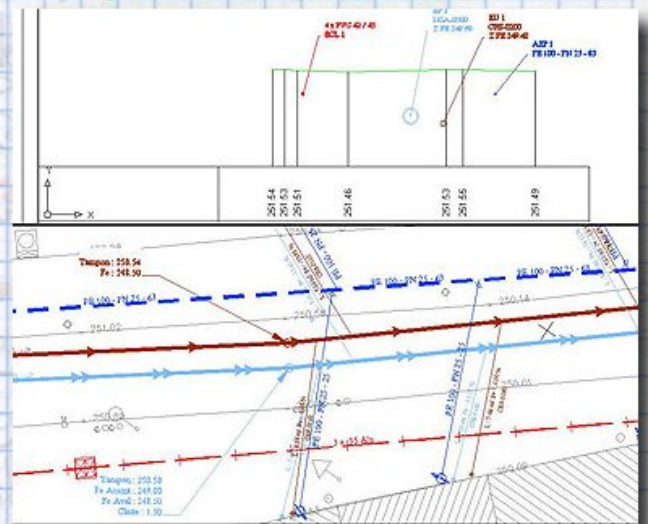
En gros, un SIG, c'est une base de données qui se voit.  w
 lien ↗

J'aimerais toutefois rappeler la différence entre un SIG et un logiciel de dessin technique style QCAD (non je ^{je dirais pas AutoCAD}) car ils sont souvent confondus.

EXEMPLE SIG



EXEMPLE DESSIN



Un SIG se base sur une représentation symbolique, "sous" laquelle des données sont enregistrées. L'outil de dessin ne fait que du dessin. ^{mais il le fait bien le logiciel!}

Les 2 outils sont confondus car ils gèrent des couches (LAYERS), des nœuds, des primitives, des attributs

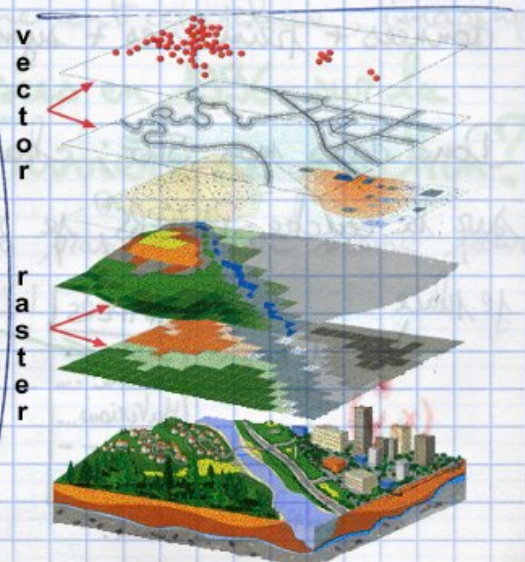
ceci est un point

ceci est une polygône

ceci est un polygone

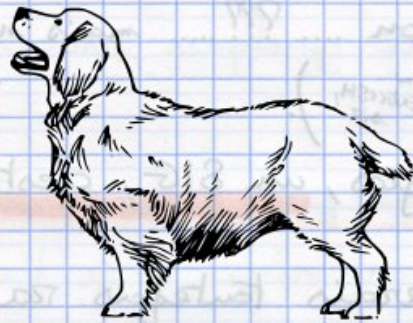
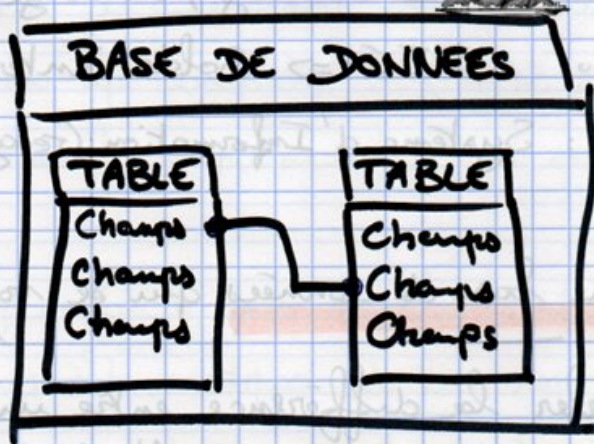
ceci est une tâche mais ce n'est pas une primitive

Un SIG n'est pas un prog de dessin et vice-versa.



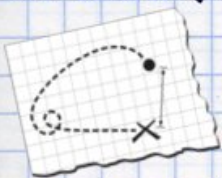
Donc un SIG, c'est avant tout une base de donnée. (2)

Je ne sais^{pas} si c'est nécessaire mais je vais préciser un peu le vocabulaire et fixer 2,3 notions pour bien se comprendre.



Une base est donc un ensemble de **TABLES** composées de **CHAMPS** ou **COLONNES** ou (surtout en SIG) des **ATTRIBUTS** qui contiennent les **DONNEES**. Les données peuvent avoir plusieurs types (ou format): **INTEGER** (entiers sans virgule), **FLOAT** (avec virgule), **CHAR**, **TEXT**... etc. avec pleins d'autres subtilités).

Les tables sont reliées entre ^{elles} par des **JOINTURES** ou associations ou relations... (d'où le terme base relationnelle)

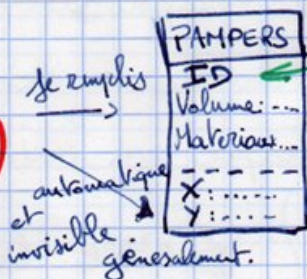


Un SIG "stocke" donc des données attributaires et des données **SPATIALES** ou **GEOMETRIQUES**. Les primitives sont stockées sous forme **X, Y** et parfois **Z**. Chaque point est appelé **NOEUD** ou **SOMMET**. (et ce sont eux qui sont stockés!).

Sur ces primitives sont placés des **SYMBLES**. L'ensemble : données + primitives + symbole = **OBJET** ou **ENTITE**

Donc dans mon SIG, chaque **COUCHE** est une table. Lorsque je suis sur la couche **PAMPERS**, je renseigne la table **PAMPERS**

je place
↓
(x,y)



TRUC IMPORTANT LA! ceci est l'identifiant unique (PRIMARY KEY ou CLE PRIMAIRE). Elle permet de ne pas confondre les données entre elles. Un peu comme les numéros de page de ces carnets.

En cliquant sur mon objet, je retire donc les données stockées dans les attributs (affichage de la table attributaire ou du **FORMULAIRE** ou **FICHE**). (3)



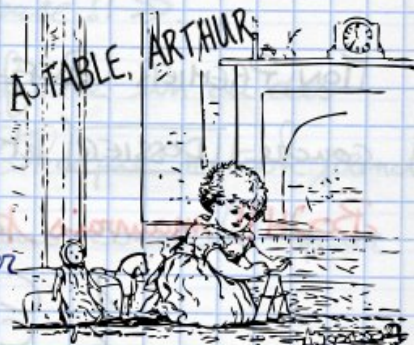
↳ ça signifie mon côté féminin.

Je peux retrouver mes entités en interrogeant directement les tables par des questions appelées : **REQUÊTES**

Je peux donc me poser légitimement la question suivante :

Où sont toutes les fosses d'un volume de 3 m³ ?

Bon, évidemment, le gars qui a inventé la base de données, tout seul, dans sa chambre, alors que sa mère faisait à manger et donc qu'il était bientôt l'heure de passer à table (jeux de mots), a cru bon de simplifier par un langage plus ou moins évident et surtout en anglais. Il a donc inventé le **SQL** traduisant ma première question : en un truc du genre **SELECT * FROM fosse WHERE volume=3**



mais maintenant, j'ai pas fini mon SQL !!

Ah, c'est donc ça une requête SQL WHERE !!! Bin oui

4. Les REQUÊTES SPATIALES.

← c'est quoi ça ? C'est juste que je viens de me rendre compte que je n'ai pas fais de plan alors je mets un 4 pour faire sérieux

La requête précédente était donc une requête ... attributaire, bravo mais on peut attaquer le côté géographique ou géométrique de la chose.

Toutes les fosses dans un rayon de 50 m de ce puits.

Mais bon, on verra peut-être plus tard tout ça.

Les requêtes peuvent donc être extrêmement complexe, imaginez **Toutes les fosses d'un volume de 3 m³ posées en 1996 sur la commune de machin à 500 m de la rivière de machin ?**

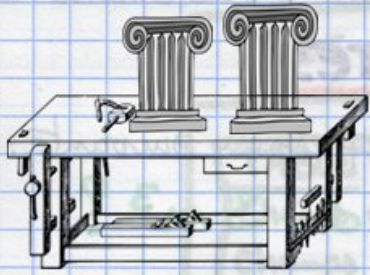
Non, je le fais pas en SQL même pour l'exemple NA!



④ Courage bientôt la fin!

2 mots sur les **JOINTURES** ou association ou relation ou...

Il arrive parfois, non, souvent que l'on ait besoin de données qui existe déjà dans une autre table.



EXEMPLE: Dans une table **FOSSE**, j'ai une colonne ANNEE DE POSE que j'ai rempli consciencieusement.

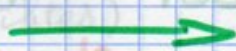
Dans une couche **DOSSIER** dans laquelle j'aimerais avoir l'année de pose des fosses.

MON PREMIER REFLEXE: je crée un attribut ANNEE DE POSE dans ma couche **DOSSIER** et je recopie mes années 1 par 1 pendant 300 heures.

BOUH!! mauvais point:

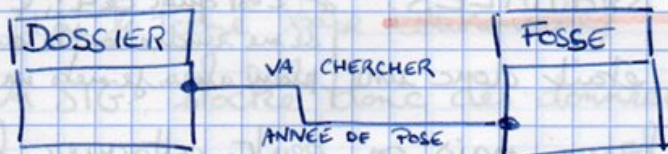
- 1 - C'est nul.
- 2 - C'est laborieux
- 3 - Je surcharge mes fichiers d'info redondantes qui ont une galère à mettre à jour et qui finissent par peser des tonnes.....
- 4 - C'est nul.

LA BONNE IDÉE



JOINTURE

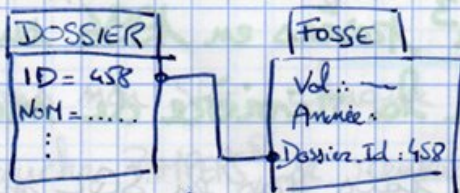
Je dis à ma couche:



Va chercher la PK!

Mais! Comment ma jointure sait quelle fosse va avec quel dossier? non, c'est vrai, c'est pas con comme question! Il faut avoir une **CLÉ**, un champs commun qui permet d'identifier l'un, l'autre.

Dans ce cas, il faut que **FOSSE** ai un attribut: **DOSSIER_ID** Par exemple



Je vous fais cours sur les relations 1→1, 1→n, n→n etc. 1 table vers plusieurs, etc voir les sites dédiés.

On peut très bien imaginer relier 2 fichiers un peu exotique. Exemple pas souvent: une couche **DEPARTEMENT** et un fichier .XLS de la population. La clé de jointure peut alors être le numéro de département (puisque il est unique).

Comme les requêtes, il existe des jointures complexes et (5)
je ne saurais trop vous conseiller de lire d'autres sources plus
complètes.



STOP, j'arrête là de vous fatiguer, en plus, je
dérive du sujet.

Je vois déjà certains grincheux me dire: « C'est nul ce truc, les notions
sont trop simplifiées ou que parfois voire même limite fausses!! ».

Oui c'est vrai! mais l'intention n'est pas de faire un ou des
cours magistraux sur les bases de données. Chaque notion prend
des livres entiers. J'ai donc survolé, je vous conseille pour
approfondir:

- site DEVELOPPEZ.COM rubrique SGBD 🤖
- site WIKIPEDIA 🤖
- 👉 - site SITE DU ZERO (+tar MySQL) 🤖

