

Mes meilleures astuces QGIS pour consolider ses bases

21 astuces pour travailler vos fondamentaux à votre rythme.
Parce qu'avant d'accélérer avec l'IA, il faut un socle solide.

Introduction

Bonjour, je suis Emmanuel Ndofunsu.

Cela fait bientôt 10 ans que j'accompagne les géomaticiens et les professionnels qui manipulent de la donnée spatiale à monter en compétence, à travers la formation et le conseil. Cet été, j'ai créé la communauté des Géomagiens. Et depuis quelque temps, une question revient plus que toutes les autres : comment mettre de l'intelligence artificielle dans mon travail SIG ?

C'est le sujet de mon livre. Mais au cœur de cet ouvrage, je défends une idée qui va peut-être à contre-courant : ce qui compte le plus, à l'ère de l'IA, ce n'est pas l'IA elle-même. C'est votre expertise.

L'intelligence artificielle est un amplificateur. Elle démultiplie ce que vous savez déjà faire. Mais elle ne remplace pas le socle. Un géomaticien qui maîtrise ses fondamentaux et qui ajoute l'IA à sa panoplie devient redoutable. Celui qui espère que l'IA compensera des bases fragiles se retrouve à produire des résultats qu'il n'est même pas capable de vérifier.

Autrement dit : pour exploiter l'IA au mieux, il faut d'abord savoir faire soi-même.



« L'IA générative pour les géomaticiens » : dans mon livre, je défends une conviction simple — l'IA amplifie votre expertise, elle ne la remplace pas. Ce document en est la première marche : les fondamentaux d'abord.

C'est précisément la raison d'être de ce document. Avant de parler d'intelligence artificielle, j'ai voulu revenir aux fondamentaux de QGIS. Vingt-et-une astuces, testées et éprouvées, pour consolider les bases qui feront de vous un utilisateur solide. Celles qui vous permettront, ensuite, de tirer le meilleur de tous les outils que la rentrée vous réserve.

Ces astuces sont nées d'une série que j'ai publiée cet été, les « Séances de rattrapage ». Je les ai reprises, développées et détaillées ici pour que vous puissiez les travailler à votre rythme, seul, à tête reposée.

Prenez votre temps. Manipulez. Reproduisez chaque exemple de votre côté. C'est en faisant que l'on consolide.

Bonne lecture, et bon rattrapage.

Emmanuel

Sommaire

21 astuces, regroupées en 7 thématiques. Chaque titre est cliquable.

1 • EXPRESSIONS ET CALCULATRICE DE CHAMPS

- 01** L'expression CASE : des valeurs conditionnelles
- 02** aggregate() : comparer chaque entité au total
- 03** Les champs virtuels : des valeurs toujours à jour
- 04** Les champs dérivés : calculer à partir d'autres champs
- 05** Les variables : définir une fois, réutiliser partout
- 06** represent_value() : des libellés lisibles

2 • SÉLECTION ET FILTRAGE

- 07** Le filtre de couche : masquer sans supprimer
- 08** Les prédicats spatiaux : poser la bonne question

3 • JOINTURES ET RELATIONS

- 09** La jointure attributaire : relier deux tables
- 10** La jointure spatiale : la position comme clé
- 11** Les relations un à plusieurs

4 • TRAITEMENT

- 12** Vérifier et réparer les géométries
- 13** Le hub le plus proche : mesurer l'accessibilité
- 14** Le modeleur graphique : votre chaîne, une fois pour toutes
- 15** Le traitement par lot : préparer une fois, lancer une fois

5 • SYMBOLOGIE

- 16** La symbologie par règles : plusieurs messages, une carte
- 17** Le data-defined override : la donnée pilote le style

6 • MISE EN PAGE

- 18** L'atlas : une carte par entité, automatiquement

7 • ACTIONS

- 19** Les actions : rendre vos couches cliquables

BONUS

- 20** coalesce() : ne plus être trahi par un NULL
- 21** Les commentaires dans vos expressions

POUR FINIR

La communauté des Géomagiciens : la suite logique, des fondamentaux vers l'IA appliquée à la géomatique.

1

PARTIE 1

Expressions et calculatrice de champs

La calculatrice de champs est le cœur battant de QGIS : c'est elle qui transforme vos données brutes en information. Six astuces pour arrêter de remplir vos tables à la main et laisser les expressions travailler pour vous.

- 01 L'expression CASE
- 02 La fonction aggregate()
- 03 Les champs virtuels
- 04 Les champs dérivés
- 05 Les variables personnalisées
- 06 La fonction represent_value()

L'expression CASE : créer des valeurs conditionnelles sans saisie manuelle

Vous voulez créer un nouveau champ dont les valeurs dépendent d'une condition ? Le réflexe de beaucoup d'utilisateurs est de remplir la table attributaire à la main, ligne par ligne. C'est long, fastidieux, et surtout source d'erreurs : une faute de frappe, une ligne oubliée, et votre classification devient incohérente.

QGIS dispose d'un outil fait exactement pour cela : l'expression **CASE**, dans la calculatrice de champs. C'est le « si... alors... sinon » de QGIS : il évalue vos conditions une par une, dans l'ordre où vous les avez écrites, et renvoie le résultat de la première condition qui est vraie.

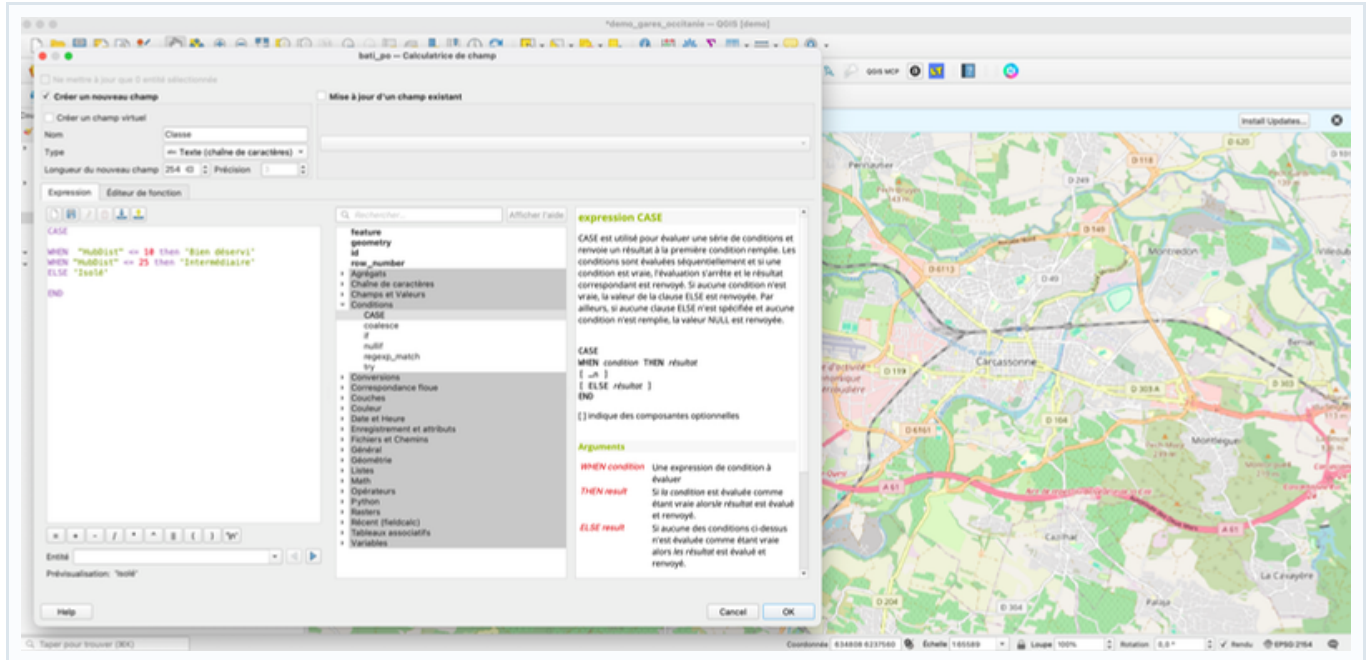
PAS À PAS

1. Ouvrez la table attributaire de votre couche, puis cliquez sur l'icône de la **calculatrice de champs** (le boulier).
2. Cochez **Créer un nouveau champ**, nommez-le (par exemple **Classe**) et choisissez le type **Texte (chaîne de caractères)**.
3. Dans la zone d'expression, écrivez votre bloc **CASE** (voir ci-dessous).
4. Cliquez sur **OK** : le champ se remplit tout seul, entité par entité.

Exemple concret : classer chaque bâtiment selon sa distance à la gare la plus proche, stockée dans un champ **HubDist** (vous verrez à l'[astuce 13](#) comment obtenir ce champ automatiquement).

```
CASE
  WHEN "HubDist" ≤ 10 THEN 'Bien desservi'
  WHEN "HubDist" ≤ 25 THEN 'Intermédiaire'
  ELSE 'Isolé'
END
```

Cinq lignes, un clic sur OK, et votre nouveau champ **Classe** est renseigné pour toute la couche. Notez la logique de lecture : une entité à 8 km valide la première condition et s'arrête là ; une entité à 18 km échoue à la première mais valide la deuxième ; au-delà de 25 km, c'est le **ELSE** qui prend le relais.



La calculatrice de champs avec l'expression CASE : le champ « Classe » se remplit selon la distance HubDist.

PIÈGE À CONNAÎTRE

Que se passe-t-il si aucune condition n'est remplie ? Le **ELSE** prend le relais. Mais si vous n'avez pas écrit de **ELSE**, QGIS renvoie **NULL** : une valeur vide, silencieuse, qui peut fausser vos analyses plus tard. Prenez l'habitude de toujours prévoir un **ELSE** (voir aussi l'[astuce 20](#) sur coalesce()).

La fonction `aggregate()` : comparer chaque entité au total de sa couche

Vous voulez comparer chaque entité au total de sa couche ? Par exemple, calculer la part de chaque ville dans la population totale ? Beaucoup d'utilisateurs sortent la calculatrice, ou pire : exportent leur table vers un tableur, calculent le total dans Excel, puis refont une jointure pour ramener le résultat dans QGIS. Tout cela est inutile.

La fonction `aggregate()`, c'est le tableau croisé dynamique de QGIS, en une seule ligne. Elle calcule une statistique (somme, moyenne, maximum...) sur toute une couche, et vous sert le résultat directement dans votre expression, entité par entité. Chaque ligne de votre table peut ainsi « voir » le total de l'ensemble.

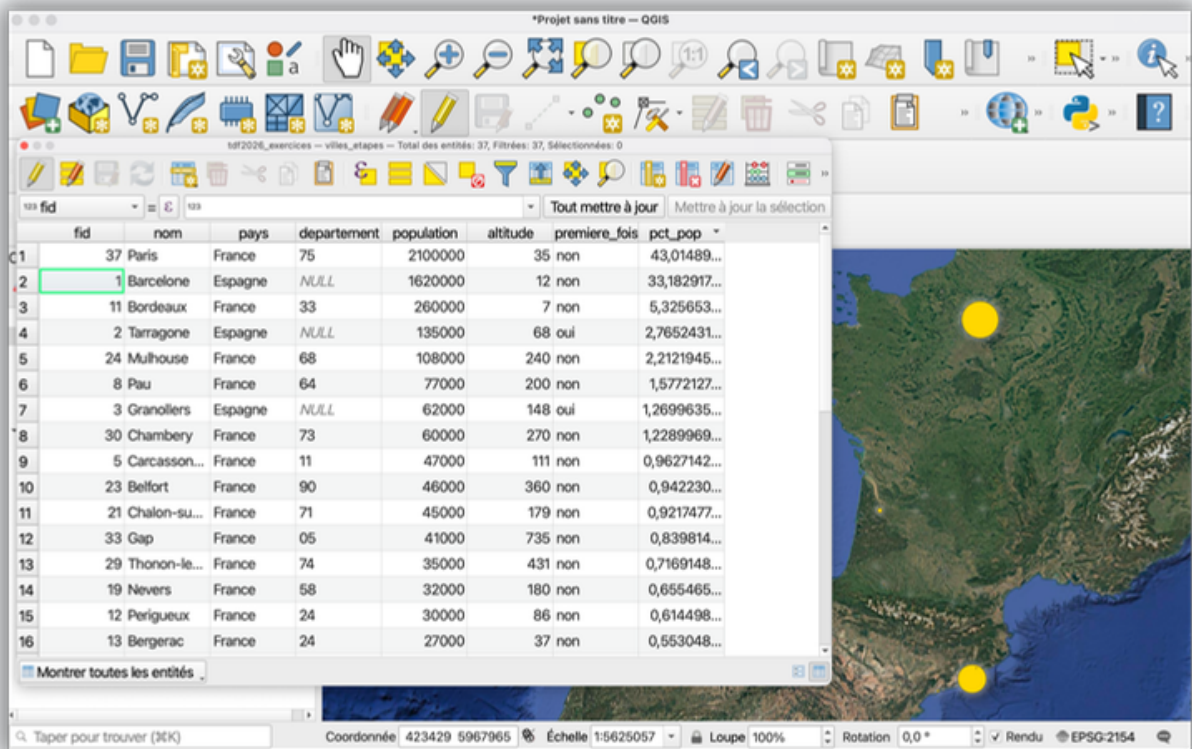
PAS À PAS

1. Ouvrez la **calculatrice de champs** sur votre couche.
2. Créez un nouveau champ décimal, par exemple `pct_pop`.
3. Écrivez l'expression ci-dessous, en remplaçant le nom de la couche et du champ par les vôtres.
4. Validez : chaque entité affiche sa part du total, sans aucune jointure.

Exemple concret, sur les villes-étapes du Tour de France 2026 : créer un champ `pct_pop` avec la part de chaque ville dans la population totale du parcours.

```
"population" / aggregate('villes_etapes', 'sum', "population") * 100
```

Une ligne, un clic sur OK : chaque ville affiche son pourcentage. La fonction prend trois arguments : le nom de la couche sur laquelle agréger, la statistique à calculer (`'sum'` , mais aussi `'mean'` , `'max'` ...), et le champ concerné. Au passage, vous découvrirez que Barcelone et Paris écrasent tout le monde.



L'expression `aggregate()` dans la calculatrice de champs : la part de chaque ville dans la population totale, sans jointure.

PIÈGE À CONNAÎTRE

Le nom de la **couche** s'écrit entre **apostrophes** (`'villes_etapes'`), le nom du **champ** entre **guillemets** (`"population"`). Inversez-les et rien ne fonctionne. C'est une convention générale des expressions QGIS : les apostrophes désignent du texte, les guillemets désignent un champ.

Les champs virtuels : des valeurs qui se mettent à jour toutes seules

Le scénario est classique : vous calculez un champ (une longueur, une surface), puis vous modifiez vos géométries quelques jours plus tard... et vous oubliez de recalculer. Votre table contient désormais des valeurs fausses, sans qu'aucun signal ne vous alerte.

La solution : dans la calculatrice de champs, cochez la case **Créer un champ virtuel**. Un champ virtuel ne stocke rien en dur. Il conserve l'expression elle-même, et la réévalue en continu, à chaque affichage. La valeur que vous voyez est donc toujours le reflet exact de l'état actuel de vos données.

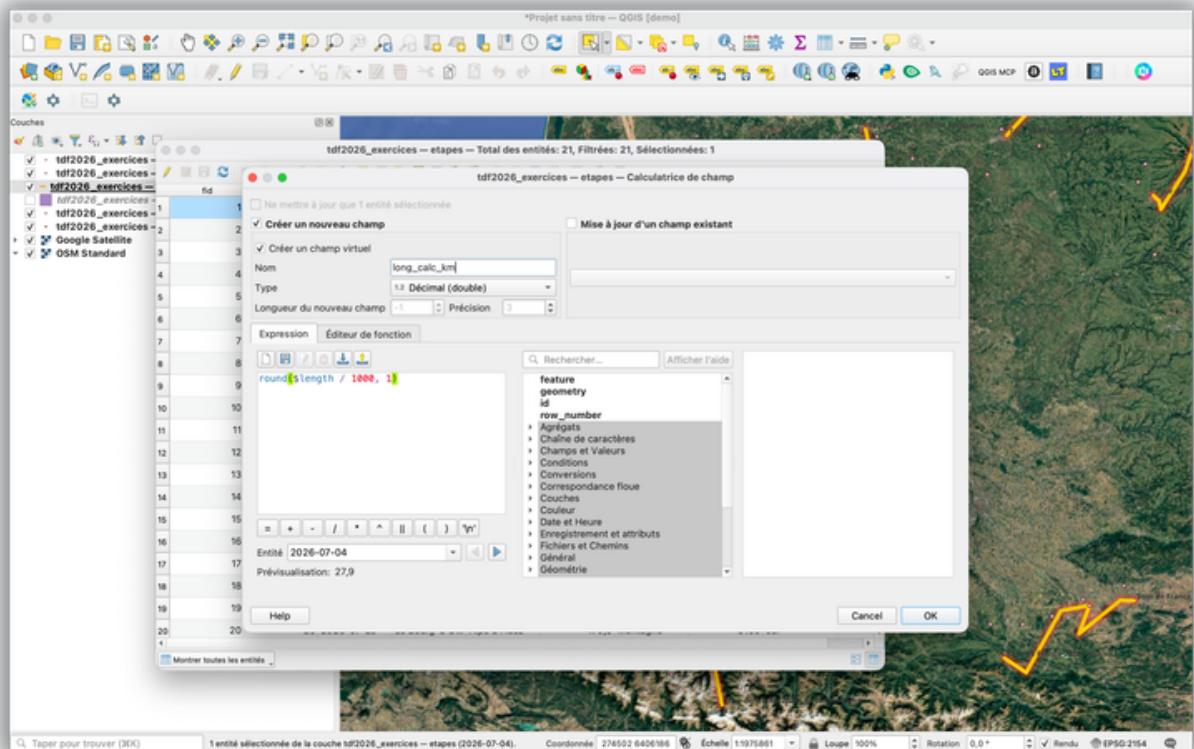
PAS À PAS

1. Ouvrez la **calculatrice de champs**.
2. Cochez **Créer un champ virtuel** (en haut de la fenêtre).
3. Nommez le champ (par exemple `Longueur_km`) et choisissez un type décimal.
4. Saisissez l'expression, puis validez.

Exemple concret : mesurer la longueur de chaque tracé directement depuis la géométrie, en kilomètres arrondis à une décimale.

```
round($length / 1000, 1)
```

Modifiez un tracé : la longueur se recalcule instantanément, sans que vous touchiez à quoi que ce soit. Un champ classique, lui, resterait bloqué sur l'ancienne valeur. Astuce dans l'astuce : si la valeur ne semble pas bouger à l'écran, ce n'est pas un bug — actualisez l'affichage et la table se rafraîchit avec la vraie valeur.



La case « Créer un champ virtuel » dans la calculatrice de champs : la longueur se recalcule à chaque modification du tracé.

PIÈGE À CONNAÎTRE

Un champ virtuel **n'est pas exporté** quand vous enregistrez la couche dans un autre fichier : il vit dans le projet, pas dans la donnée. Retenez la règle : *virtuel pour explorer, champ réel pour figer et exporter*. Avant un export, recalculez la valeur dans un champ classique si vous devez la transmettre.

Les champs dérivés : calculer à partir d'autres champs, sans valeurs périmées

« Un tableur ferait pareil. » Non. Un tableur ne recalcule pas quand vous modifiez la géométrie ou la donnée source. QGIS, si — à condition d'utiliser des champs calculés à partir d'autres champs, de la bonne manière.

Votre « vitesse », votre « densité », votre « ratio »... tout ce qui se déduit d'autres colonnes est un champ dérivé. Le réflexe habituel consiste à le calculer une fois, puis à espérer que personne ne modifiera jamais les données sources. Le bon réflexe : un **champ virtuel** (voir l'[astuce 03](#)) qui pointe vers vos autres champs. Vous changez une valeur source, le champ dérivé se met à jour dans la foulée.

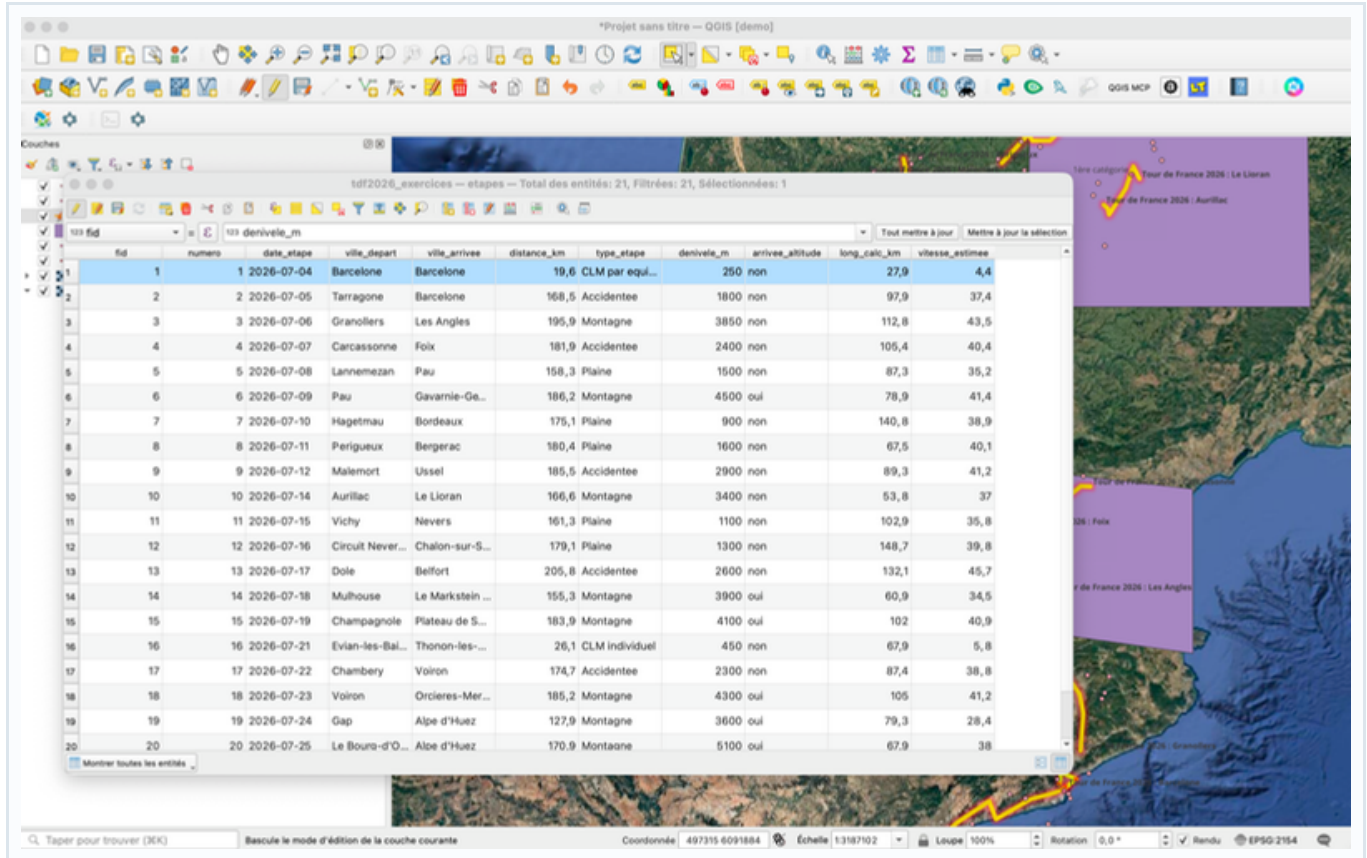
PAS À PAS

1. Ouvrez la **calculatrice de champs** et cochez **Créer un champ virtuel**.
2. Nommez le champ (par exemple `vitesse_estimee`).
3. Écrivez une expression qui référence vos autres champs, puis validez.
4. Testez : modifiez une valeur source dans la table, et observez le champ dérivé suivre.

Exemple concret : estimer une vitesse moyenne à partir de la distance de chaque étape, sur la base d'une durée de 4 h 30.

```
round("distance_km" / 4.5, 1)
```

Corrigez une distance : la vitesse suit, sans un clic de plus. Vos champs dérivés ne mentent jamais sur des données périmées. Posez-vous la question pour vos propres projets : vos colonnes calculées sont-elles vivantes, ou figées depuis six mois ?



Un champ dérivé qui pointe vers « distance_km » : corrigez la distance, la vitesse suit immédiatement.

LE POINT À RETENIR

Un champ dérivé n'est fiable que s'il se **recalcule**. Figé, c'est une bombe à retardement dans votre table : le jour où une donnée source bouge, toutes les valeurs qui en dépendent deviennent fausses en silence.

Une variable, c'est le mémo réutilisable de QGIS. Vous définissez la valeur une seule fois, et vous l'appellez partout avec son @ : dans les étiquettes, la symbologie, les mises en page, les expressions.

1. Menu **Projet > Propriétés > Variables**.
2. Cliquez sur le **+** pour ajouter une variable, par exemple `edition`, avec la valeur `Tour de France 2026`.
3. Dans n'importe quelle expression du projet, appelez-la avec `@edition`.

```
concat(@edition, ' : ', "nom")
```

The screenshot shows the 'Variables d'expression' dialog box in QGIS. The 'Global' tab is active, displaying a list of variables. The 'Projet' group contains the following variables and their values:

Variable	Valeur
layer_id	['OSM_Standard_0c85c6a0_3938_461d_b198_d6807463a684'; 'ca...']
layers	[<<couche cartographique>; <couche cartographique>; <couche...]
project_a...	''
project_ar...	'mètres carés'
project_a...	'd'atando'
project_b...	''
project_cr...	<dateheure : 2026-07-07 09:38:11 (CEST)>
project_cr...	'EPSG:2154'
project_cr...	'10e'
project_cr...	'4proj1ccc +lat_0=46.5 +lon_0=3 +lat_1=49 +lat_2=44 +x_0=7000...'
project_cr...	'RGF93 / Lambert-93'
project_cr...	'EPSG:7019'
project_cr...	'4proj1ccc +lat_0=46.5 +lon_0=3 +lat_1=49 +lat_2=44 +x_0=7000...'
project_cr...	'PROJCRS["RGF93 / Lambert-93", BASEGEOGCRS["RGF93", DATUM["Rese...'
project_di...	'mètres'
project_el...	'EPSG:7019'
project_fi...	''
project_fo...	''
project_h...	''
project_id...	''
project_k...	[]
project_la...	''
project_p...	''
project_s...	''
project_u...	'mètres'
edition	'1Tour de France 2026'

The dialog box includes 'Help', 'Apply', 'Cancel', and 'OK' buttons at the bottom.

L'onglet Variables des propriétés du projet : @edition définie une fois, appelée dans toutes les étiquettes.

PIÈGE À CONNAÎTRE

Une variable **de projet** suit votre fichier de projet, pas la couche. Si vous ouvrez la même couche dans un autre projet, la variable ne vous suit pas. Pour une valeur attachée à la donnée elle-même, définissez plutôt une variable de couche (Propriétés de la couche > Variables).

represent_value() : afficher des libellés lisibles à la place des codes

Vos étiquettes affichent des codes cryptiques comme « HC » ou « 1 » au lieu de vrais libellés ? C'est fréquent : les données métier stockent des codes, compacts et fiables, mais illisibles pour le lecteur d'une carte. La mauvaise solution : créer un champ en double juste pour traduire les codes. Vous alourdissez la donnée et créez un risque de désynchronisation.

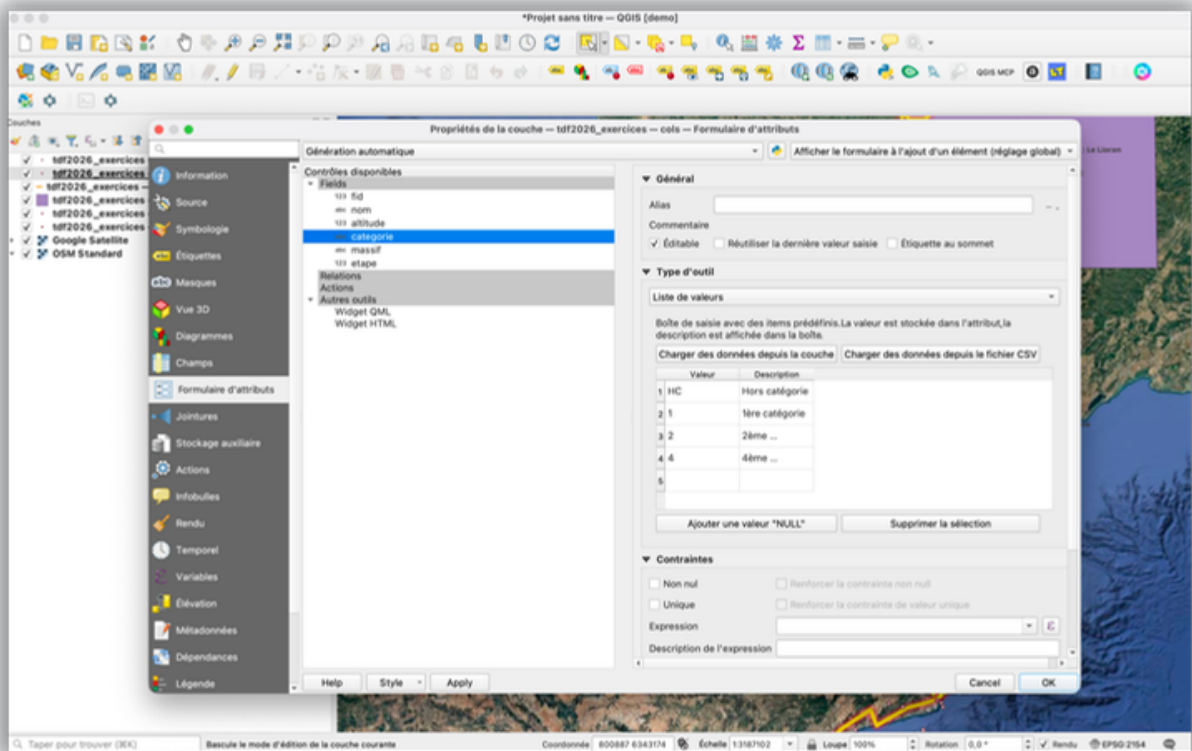
La bonne solution tient en deux temps : configurer un **widget de valeurs** sur le champ, puis appeler `represent_value()` dans vos étiquettes. C'est le traducteur intégré de QGIS : il affiche le libellé lisible défini dans le widget, tout en gardant la valeur brute stockée dans la donnée.

PAS À PAS

1. **Propriétés de la couche > Formulaire d'attributs**, sélectionnez le champ concerné (par exemple `categorie`).
2. Choisissez le type d'outil **Liste de valeurs**, et renseignez chaque paire code / libellé :
HC → Hors catégorie, 1 → 1re catégorie...
3. Dans les propriétés d'étiquetage, utilisez l'expression ci-dessous.

```
represent_value("categorie")
```

Vos étiquettes affichent désormais « Hors catégorie », « 1re catégorie », proprement — sans toucher à la donnée d'origine. Le principe à comprendre : la donnée reste brute et légère, seul l'*affichage* est traduit. Vous stockez des codes, vous montrez du français.



Le widget « Liste de valeurs » et `represent_value()` en action : les cols codés HC, 1, 2 s'affichent en toutes lettres.

PIÈGE À CONNAÎTRE

Sans widget configuré sur le champ, `represent_value()` renvoie simplement la valeur brute. Le libellé vit dans le **formulaire d'attributs**, pas dans la table : si vous partagez la couche sans son style, la traduction ne suit pas.

2

PARTIE 2

Sélection et filtrage

Travailler sur un sous-ensemble de ses données sans rien casser, et interroger ses géométries avec le bon vocabulaire : deux gestes fondamentaux qui évitent bien des sueurs froides.

07 Le filtre de couche

08 Les prédicats spatiaux

Le filtre de couche : travailler sur une partie de vos données sans rien casser

« J'ai supprimé la moitié de mes entités par erreur ! » Dans l'immense majorité des cas... non. Elles étaient simplement filtrées. Elles n'ont jamais bougé. Comprendre le filtre de couche, c'est s'épargner cette frayeur — et gagner un outil de travail quotidien.

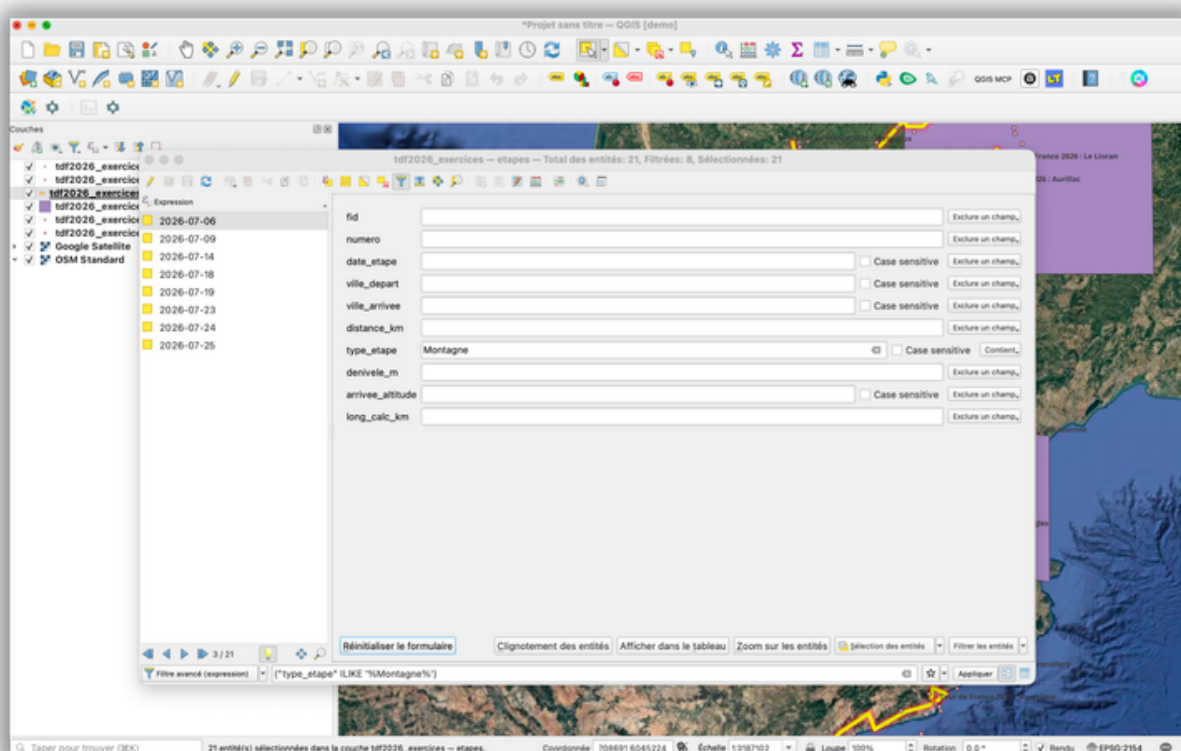
Vous voulez travailler sur une partie de vos données sans découper, sans exporter, sans rien casser ? Le filtre est fait pour ça. Vous écrivez une condition, et QGIS masque tout le reste : de la carte et de la table attributaire. Mais les entités masquées sont toujours là, intactes, dans le fichier. **Un filtre cache, il ne supprime jamais.**

PAS À PAS

1. **Clic droit sur la couche > Filtrer...**
2. Dans le constructeur de requête, écrivez votre condition (voir ci-dessous).
3. Validez : seules les entités qui répondent à la condition restent visibles.
4. Pour tout faire revenir : rouvrez le filtre et cliquez sur **Effacer**.

```
"type_etape" = 'Montagne'
```

Vos 8 étapes de montagne restent à l'écran, les autres disparaissent de la carte et de la table — le temps du filtre. Videz-le, tout revient, comme si de rien n'était.



Le filtre de couche en action : seules les étapes de montagne restent visibles ; l'entonnoir signale le filtre actif.

LE DÉTAIL QUI SAUVE

Quand un filtre est actif, une petite icône **entonnoir** apparaît à côté du nom de la couche dans le panneau. C'est votre rappel qu'une condition tourne en arrière-plan. Si un jour il vous « manque » des entités, vérifiez cet entonnoir avant de paniquer.

Les prédicats spatiaux : poser la bonne question à vos géométries

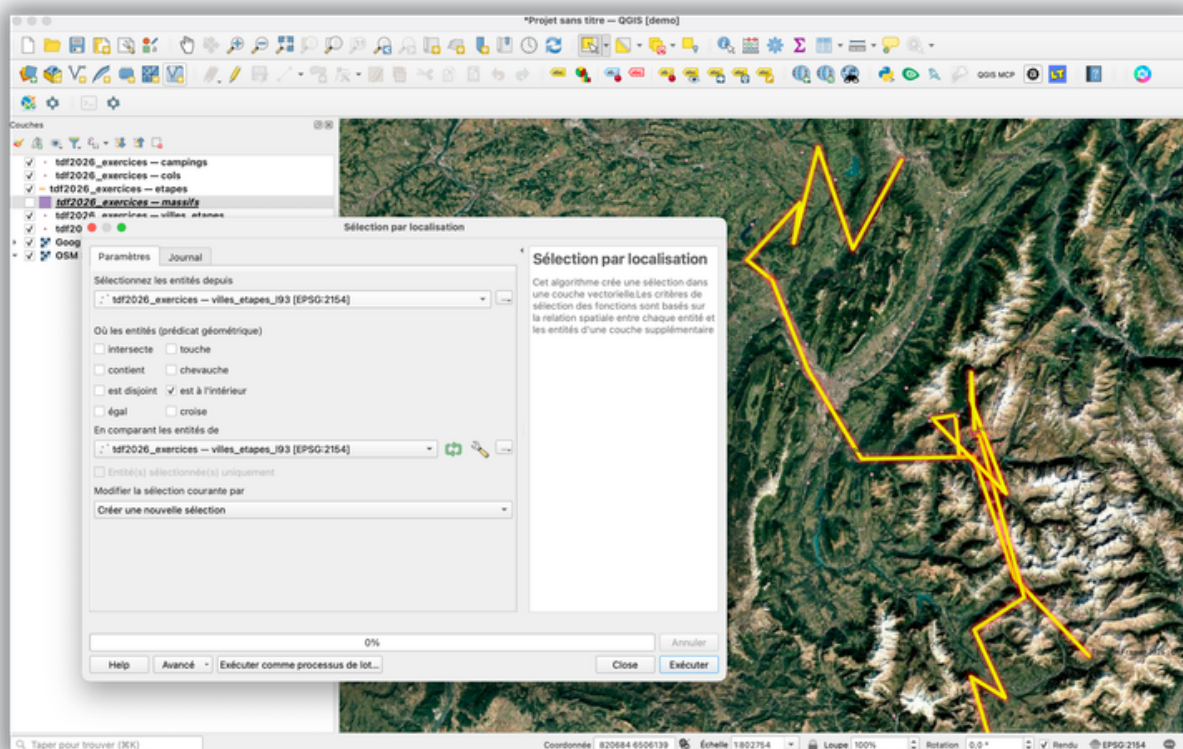
Votre sélection spatiale vous renvoie trop d'entités. Ou pas assez. Le problème n'est généralement pas l'outil : c'est le mot que vous avez choisi. Sélectionner par localisation (**Vecteur > Outils de recherche > Sélection par localisation**), tout le monde connaît. Mais entre « intersecte », « contient », « à l'intérieur de »... lequel choisir ?

Ce ne sont pas des synonymes. Chaque prédicat décrit une relation géométrique précise :

- **À l'intérieur de** : l'entité doit être *entièrement contenue* dans l'autre géométrie.
- **Intersecte** : le *moindre contact* suffit — un simple point de passage compte.
- **Contient** : la relation inverse d'« à l'intérieur de » : c'est votre entité qui englobe l'autre.

Exemple concret : sélectionner les étapes qui *intersectent* les Pyrénées. QGIS attrape aussi celles qui ne font que traverser le massif, sans y démarrer ni y finir. Changez « intersecte » en « à l'intérieur de », et ces étapes de passage disparaissent de la sélection. Même outil, mêmes couches, résultat différent : vous n'avez pas posé la même question.

Huit prédicats dans le menu déroulant, huit questions différentes posées à vos géométries. Prenez trente secondes, avant chaque sélection spatiale, pour formuler votre question en français : « je veux les entités qui... » — le bon prédicat en découle presque toujours.



La sélection par localisation : « intersecte » attrape les étapes de passage, « à l'intérieur de » les exclut.

PIÈGE À CONNAÎTRE

Aucun prédicat ne renvoie d'erreur. Le mauvais prédicat vous donne un résultat **parfaitement propre, et parfaitement faux**. C'est le pire type d'erreur en géomatique : celle qui ne se voit pas. Contrôlez toujours quelques entités sélectionnées à la main pour valider votre choix.

3

PARTIE 3

Jointures et relations

Vos données vivent rarement dans un seul fichier. Relier des tables par un champ commun, par la position, ou par une vraie relation un à plusieurs : c'est le passage du fichier isolé à la logique de base de données.

09 La jointure attributaire

10 La jointure spatiale

11 Les relations un à plusieurs

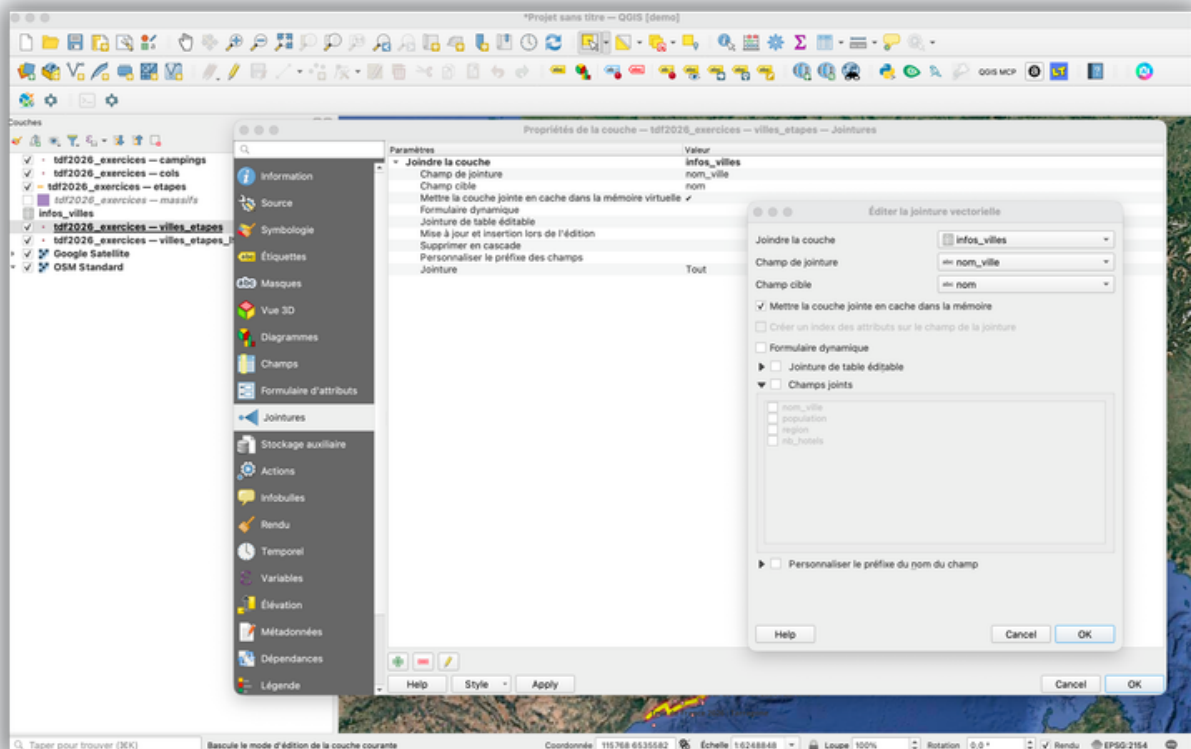
La jointure attributaire : relier deux tables par un champ commun

Votre couche a besoin d'informations qui vivent dans un autre fichier — un tableur de population, une liste de régions, un inventaire. Ouvrir les deux côte à côte et recopier à la main, ligne par ligne ? Il y a beaucoup mieux.

Deux tables, un champ en commun : c'est tout ce qu'il faut à QGIS pour les relier. Les colonnes de la table externe apparaissent alors d'un coup dans votre couche — population, région, nombre d'hôtels — sans un seul copier-coller. Et ce lien est **vivant** : si la table source change, votre couche suit.

PAS À PAS

1. Chargez votre table externe dans le projet (un CSV ou un tableur se glisse comme n'importe quelle couche).
2. **Propriétés de la couche > Jointures**, puis cliquez sur le +.
3. Désignez la **couche à joindre** (la table externe), le **champ de jointure** de son côté et le **champ cible** du côté de votre couche — par exemple `nom = nom_ville`.
4. Validez, puis ouvrez la table attributaire : les nouvelles colonnes sont là.



L'onglet Jointures des propriétés de la couche : le champ commun relie la couche et la table externe.

LE PIÈGE QUI FAIT PERDRE DES HEURES

L'appariement est **strict au caractère près**. « Gavarnie-Gedre » ne rejoindra jamais « Gavarnie Gèdre » : un tiret, un accent, une espace de trop, et vous récupérez des NULL — sans la moindre alerte. **Nettoyez vos clés avant de joindre. Toujours.** Et après la jointure, triez la nouvelle colonne pour repérer les NULL : ce sont vos correspondances ratées.

Une fois ce réflexe acquis, comptez le nombre de tableurs que vous recopiez à la main : la jointure attributaire est probablement l'astuce qui vous rendra le plus d'heures.

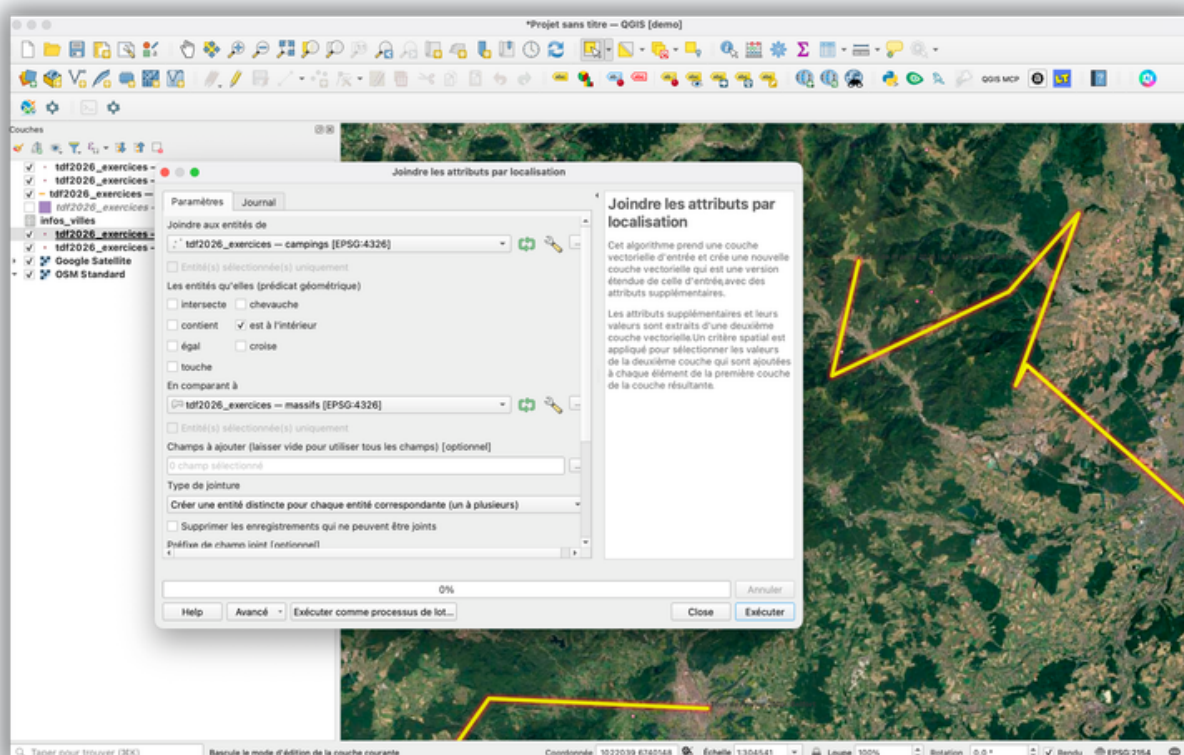
La jointure spatiale : quand la position devient la clé

La jointure classique ([astuce 09](#)) a besoin d'un champ commun entre vos deux tables. Mais quand il n'y en a aucun ? Vos campings n'ont pas de colonne « massif ». Vos massifs n'ont pas la liste de leurs campings. Aucun identifiant partagé. Et pourtant, QGIS va les relier : la **géométrie devient la clé**.

L'outil s'appelle **Joindre les attributs par localisation**. La question posée n'est plus « quelle valeur correspond ? », mais « *qui se trouve où ?* ». Chaque camping récupère le nom du massif qui le contient — sans champ commun, juste par sa position.

PAS À PAS

1. Ouvrez la **Boîte à outils de traitement** et cherchez **Joindre les attributs par localisation**.
2. **Couche d'entrée** : vos campings (celle qui doit recevoir les informations).
3. **Couche de jointure** : vos massifs (celle qui porte les informations).
4. Choisissez le prédicat géométrique — ici **est à l'intérieur** (revoyez l'[astuce 08](#) : le prédicat change le résultat).
5. Exécutez : une nouvelle couche de campings apparaît, enrichie du nom de leur massif.



Joindre les attributs par localisation : chaque camping hérite du nom du massif qui le contient.

La position comme clé de jointure : une fois qu'on l'a compris, on ne fait plus jamais de tableau de correspondance à la main. C'est l'un des gestes qui distinguent un utilisateur de SIG d'un utilisateur de tableur : votre donnée géométrique est une information de liaison, servez-vous-en.

BON À SAVOIR

Un camping en dehors de tout massif ressortira avec des valeurs **NULL** dans les champs joints. Ce n'est pas une erreur : c'est l'information « hors massif ». L'[astuce 20](#) (coalesce) vous montrera comment rendre cette information lisible.

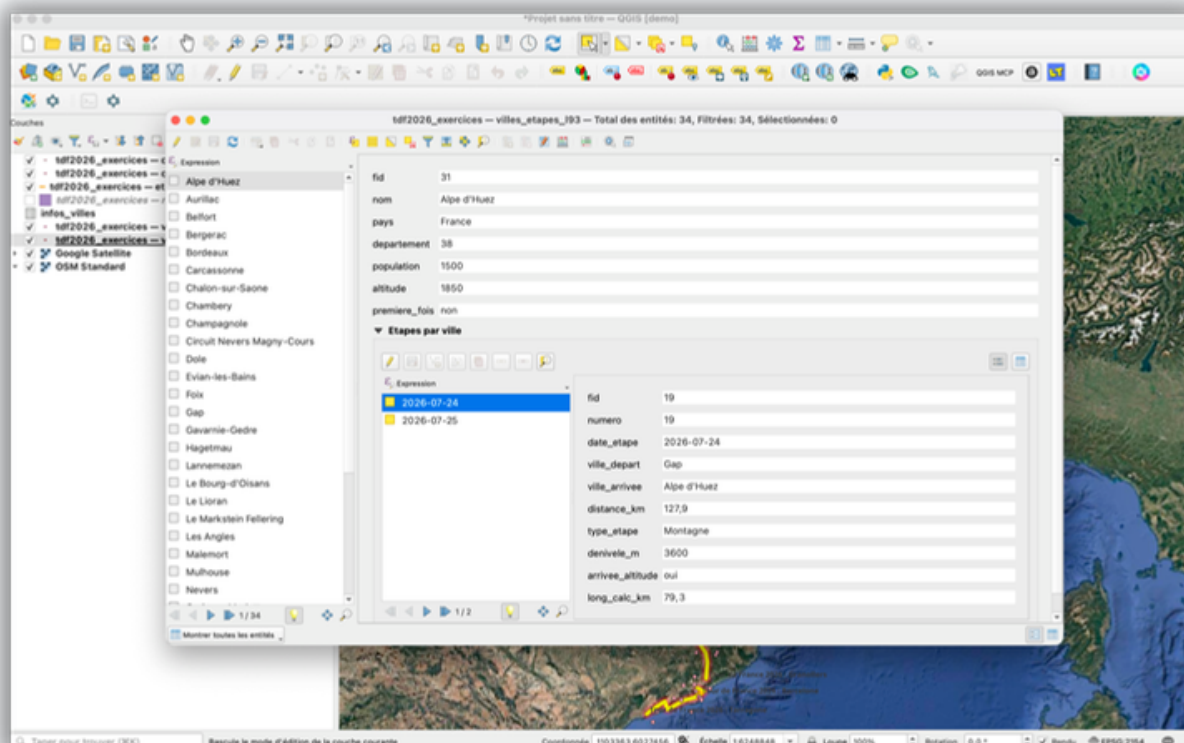
Les relations un à plusieurs : la logique base de données dans QGIS

Une ville, plusieurs étapes qui y arrivent. Essayez de gérer ça avec une jointure classique : vous allez dupliquer des lignes, mélanger des informations, vous perdre. La jointure attributaire relie *une ligne à une ligne*. Un parent, un enfant. Point.

Mais le réel ne fonctionne pas comme ça. Une ville reçoit plusieurs étapes. Un bassin contient plusieurs stations. Un chantier concerne plusieurs parcelles. **Un parent, plusieurs enfants** : c'est exactement ce que gère une *relation*.

PAS À PAS

1. Menu **Projet > Propriétés > Relations**, puis **Ajouter une relation**.
2. Désignez la **couche parente** (les villes) et la **couche enfant** (les étapes).
3. Indiquez le **champ qui les lie** de chaque côté (l'identifiant de la ville, présent dans les deux tables).
4. Ouvrez le formulaire d'une ville (outil **Identifier les entités**, puis affichage du formulaire) : toutes ses étapes s'affichent en dessous, listées.



Le formulaire d'une ville avec sa relation : toutes les étapes enfants listées, sans duplication.

Vos deux tables restent distinctes et propres : aucune ligne dupliquée, aucune colonne fourre-tout. QGIS gère le lien pour vous, et le formulaire devient une véritable fiche : la ville en haut, ses étapes en dessous.

C'est le passage du fichier isolé à la vraie logique de base de données. Et ça change tout dans un projet qui grossit : au lieu d'empiler vos données dans une seule table à rallonge, vous structurez — et chaque information n'existe qu'à un seul endroit.

LE POINT À RETENIR

Jointure = *une à une*, pour enrichir des colonnes. Relation = *une à plusieurs*, pour structurer des données liées. Si vous hésitez, posez la question : « combien de lignes enfants pour un parent ? » Plus d'une → relation.

4

PARTIE 4

Traitement

La boîte à outils de traitement est l'atelier de QGIS. Quatre astuces pour des géotraitements fiables — et pour arrêter de refaire à la main ce que la machine peut enchaîner pour vous.

- 12** Vérifier et réparer les géométries
- 13** Le hub le plus proche
- 14** Le modeleur graphique
- 15** Le traitement par lot

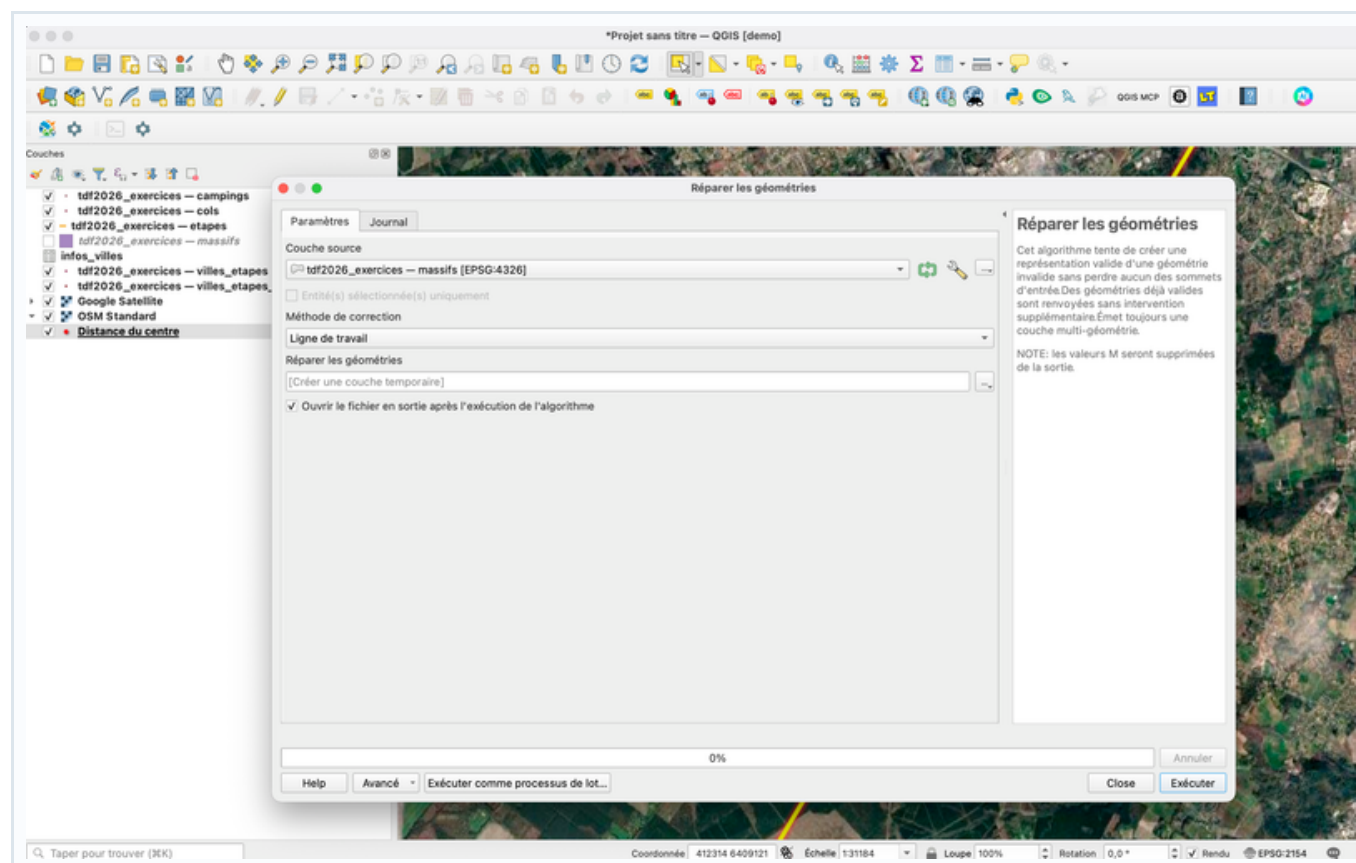
Vérifier et réparer les géométries : assainir avant de traiter

Votre géotraitement a planté sans raison apparente. Votre découpage laisse des trous inexplicables. Le coupable n'est pas QGIS : c'est une **géométrie cassée** que vous n'avez jamais vue. Un polygone qui se recoupe lui-même, un anneau mal fermé... Rien de visible à l'écran : tout est normal, en apparence.

Puis vient le tampon, le découpage, la jointure spatiale. Et là, ça casse. Ou pire : ça sort un résultat faux, sans une seule alerte. Deux outils de la boîte à traitement règlent le problème à la source.

PAS À PAS

1. Dans la boîte à outils, lancez **Vérifier la validité** sur votre couche.
2. QGIS produit trois sorties : les entités valides, les invalides, et surtout une couche de **points rouges** posés exactement là où ça coince.
3. Lancez ensuite **Réparer les géométries** : l'outil ferme, recoud et nettoie ce qui peut l'être.
4. Travaillez désormais sur la couche réparée.



Vérifier la validité : le point d'erreur localise précisément l'auto-intersection invisible à l'écran.

Le réflexe à graver : on vérifie **avant** de traiter, jamais après avoir passé une heure à comprendre pourquoi le résultat est bizarre. Une géométrie propre en entrée, c'est la moitié des bugs qui n'existent tout simplement pas.

PIÈGE À CONNAÎTRE

Une géométrie invalide ne fait pas toujours planter le traitement : parfois il aboutit... avec un résultat faux. C'est le cas le plus dangereux. Intégrez la vérification à votre routine d'import : toute couche reçue de l'extérieur passe par « Vérifier la validité » avant le premier géotraitement.

Le hub le plus proche : mesurer l'accessibilité en un traitement

« Quel est le point le plus proche de chacune de mes entités ? » Des campings et des villes. Des clients et des agences. Des capteurs et des stations. Mesurer à la règle, un par un ? QGIS répond pour vos milliers d'entités d'un seul coup, avec le calcul de **distance au hub le plus proche**.

Pour chaque entité de votre couche, l'outil trouve le point de référence le plus proche et mesure la distance. Deux champs apparaissent dans le résultat : **HubName** (le nom du plus proche) et **HubDist** (la distance jusqu'à lui). Aucune mesure manuelle, aucun tableur.

PAS À PAS

1. Dans la boîte à outils, cherchez **Distance au hub le plus proche**.
2. **Couche source** : vos entités (les campings). **Couche des hubs** : vos points de référence (les villes).
3. Choisissez l'unité de mesure, puis exécutez.
4. Ouvrez la table du résultat : **HubName** et **HubDist** sont renseignés pour toute la couche.

Et là, le vrai plaisir : ce **HubDist**, vous l'enchaînez avec le **CASE** de l'[astuce 01](#) : moins de 10 km, « bien desservi » ; au-delà, « isolé ». Un calcul spatial qui nourrit une expression logique : deux fondamentaux qui s'emboîtent, et vous obtenez une analyse d'accessibilité complète en cinq minutes.

id	nom	etoiles	capacite	piscine	HubName	HubDist
1	Camping Le ...	5	133	oui	Les Angles	126,5501529071483
2	Camping du ...	3	79	oui	Les Angles	136,63725088574074
3	Camping des...	3	26	non	Les Angles	124,49601412089494
4	Camping de L...	3	48	non	Les Angles	116,48929560078302
5	Camping du ...	4	191	oui	Les Angles	139,69012271487458
6	Camping Bell...	2	195	oui	Les Angles	104,69042532828229
7	Camping du ...	4	90	oui	Les Angles	156,86389790504837
8	Camping du ...	2	125	non	Les Angles	166,514742169808
9	Camping des...	4	51	oui	Les Angles	183,05380045476545
10	Camping Le ...	4	219	non	Les Angles	118,95286726561885
11	Camping des...	4	175	oui	Les Angles	140,8890648103741
12	Camping du ...	4	240	non	Les Angles	139,99998341203965
13	Camping des...	2	92	non	Les Angles	78,68085242983234
14	Camping du ...	4	216	non	Les Angles	67,66595996750678
15	Camping Le ...	3	163	oui	Les Angles	75,82417635785319
16	Camping des...	3	67	non	Les Angles	50,876600001116135

Le résultat du calcul : chaque entité connaît son hub le plus proche (HubName) et la distance (HubDist).

PIÈGE À CONNAÎTRE

La distance est **faussée si votre projet n'est pas dans un SCR métrique** : en degrés (WGS 84), les valeurs n'ont aucun sens terrain. Reprojectez d'abord vos couches dans un système projeté adapté (en France, le Lambert 93 / EPSG:2154), puis lancez le calcul.

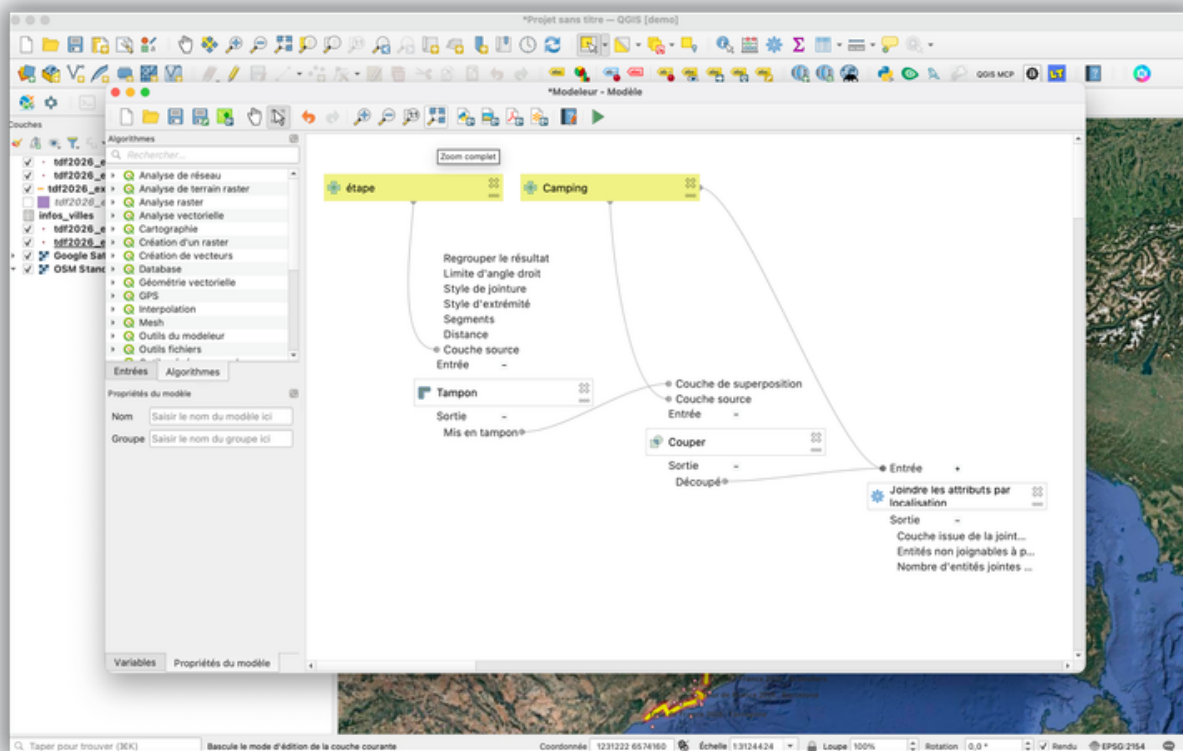
Le modelleur graphique : dessiner sa chaîne de traitements une fois pour toutes

Vous refaites le même enchaînement de traitements chaque semaine : un tampon sur vos étapes, un découpage de vos campings dans ce tampon, un comptage des campings par étape. Trois traitements, trois fenêtres, trois clics de lancement — à refaire à chaque nouveau jeu de données, dans le bon ordre, sans rien oublier.

Le **modelleur graphique** change la règle du jeu. Vous dessinez la chaîne une fois — les entrées, les algorithmes, leurs enchaînements — et vous obtenez un outil. Le vôtre. Rejouable en un clic sur n'importe quelles données.

PAS À PAS

1. Menu **Traitement > Modelleur graphique**.
2. Ajoutez vos **entrées** : une couche vectorielle « étapes », une couche « campings », éventuellement un paramètre « distance ».
3. Glissez les **algorithmes** depuis la liste : Tampon, puis Découpage, puis Compter les points dans les polygones — en connectant la sortie de l'un à l'entrée du suivant.
4. Enregistrez le modèle : il apparaît désormais dans votre boîte à outils, comme n'importe quel traitement natif.



Le modelleur graphique : tampon, découpage et comptage enchaînés en une seule chaîne visuelle.

Les bénéfices sont immédiats : plus d'étape oubliée, plus de couche temporaire égarée, plus d'ordre inversé par distraction. Et votre traitement devient **documenté, visuel, transmissible** : un collègue peut ouvrir le modèle, comprendre la chaîne d'un coup d'œil, et l'exécuter sur ses propres données.

POUR LA SUITE

C'est le premier pas vers l'automatisation dans QGIS. Le jour où vous passerez au scripting et à l'IA, cette logique de chaîne — des entrées, des étapes, des sorties — vous l'aurez déjà dans les mains.

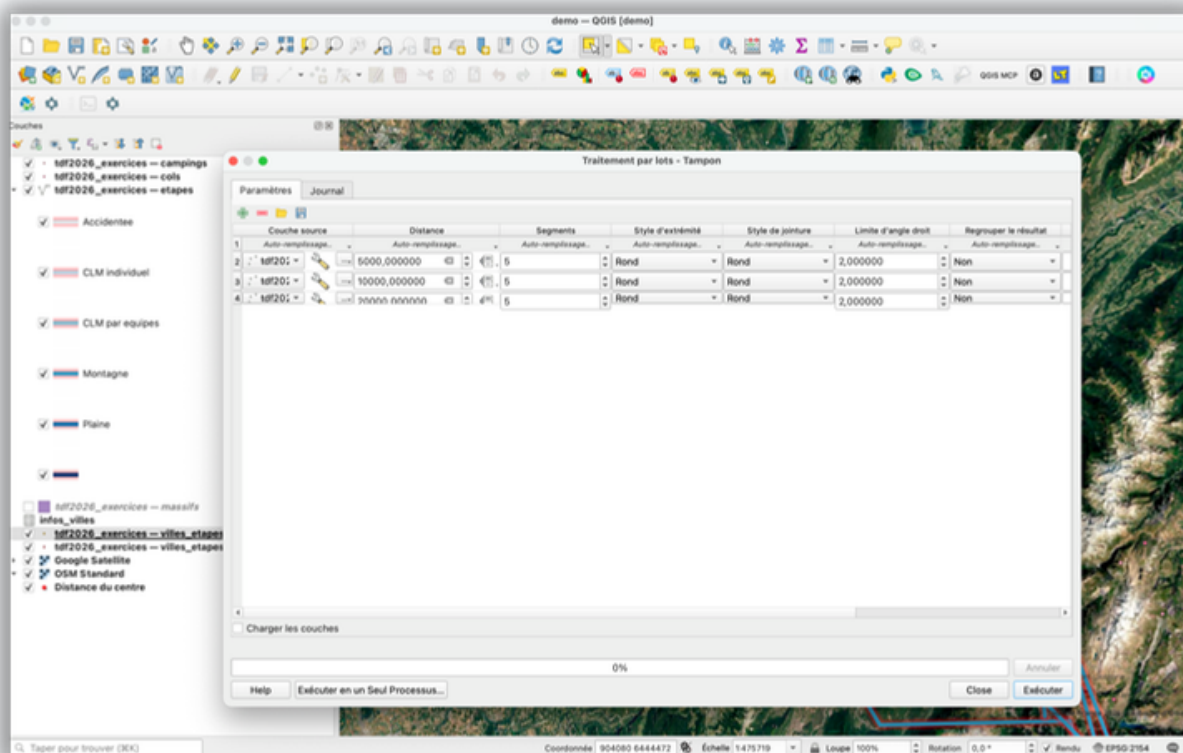
Le traitement par lot : préparer une fois, lancer une fois

Le même outil, relancé dix fois de suite : vous changez un paramètre, vous cliquez, vous attendez. Puis vous recommencez. Neuf fois de trop. N'importe quel outil de la boîte à traitement cache un second mode d'exécution : **Exécuter comme processus de lot**.

Ce mode ouvre une grille, façon tableur : chaque **ligne** est une exécution, chaque **colonne** un paramètre. Vous préparez toutes vos exécutions d'un coup, puis vous lancez une seule fois.

PAS À PAS

1. Ouvrez l'outil (par exemple **Tampon**), et cliquez sur **Exécuter comme processus de lot** en bas de la fenêtre.
2. Remplissez une ligne par exécution : la même couche d'étapes, et trois distances — 5, 10 et 20 km.
3. Donnez un nom de sortie à chaque ligne (le remplissage automatique sait numéroté les fichiers tout seul).
4. Un seul clic sur **Exécuter** : les trois tampons se génèrent à la suite.



La grille du traitement par lot : trois lignes, trois distances de tampon, un seul lancement.

Le remplissage automatique fait le gros du travail : une valeur constante copiée vers le bas, une distance qui s'incrémente, des noms de fichiers numérotés automatiquement. Et ce n'est pas réservé au tampon : reprojection, découpage, jointure... tout algorithme s'exécute en lot.

LE POINT À RETENIR

Entre le clic manuel répété et la grille du lot, il y a le même écart qu'entre saisir à la main et automatiser. Dès que vous vous surprenez à relancer un outil une troisième fois avec un paramètre différent, passez en mode lot.

5

PARTIE 5

Symbologie

Le moment où la symbologie cesse d'être de la décoration pour devenir de l'analyse : des règles qui raisonnent comme des expressions, et des propriétés de style pilotées directement par vos données.

16 La symbologie par règles

17 Le data-defined override

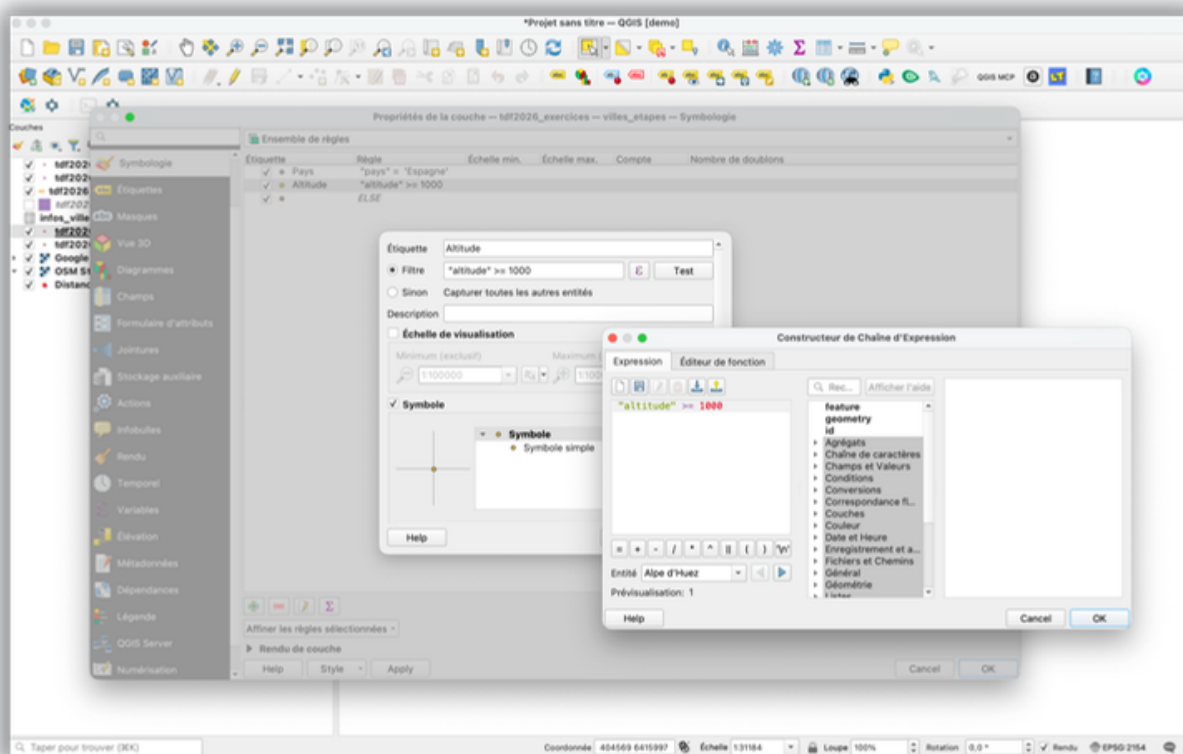
La symbologie par règles : plusieurs messages sur une seule carte

La symbologie catégorisée colore selon un champ. Un seul. C'est bien — mais vos données ont souvent plusieurs histoires à raconter en même temps : les villes espagnoles en rouge, les arrivées en altitude en violet, tout le reste en bleu. Trois critères, qui ne reposent pas sur le même champ.

C'est le domaine de la **symbologie par règles** (Propriétés de la couche > Symbologie > Ensemble de règles). Chaque règle porte sa condition — une expression QGIS complète — et son style. Vous les empilez comme des couches de logique : Si ceci, alors cette couleur.

PAS À PAS

1. **Propriétés de la couche > Symbologie**, choisissez **Ensemble de règles** dans le menu du haut.
2. Ajoutez une règle par critère, avec son expression : `"pays" = 'Espagne'`, puis `"altitude" ≥ 1000 ...`
3. Attribuez un style à chaque règle.
4. Ajoutez une dernière règle **ELSE** (case à cocher dans l'éditeur de règle) : elle ramasse tout ce qui n'entre dans aucune case. Personne n'est oublié, aucune entité ne reste invisible.



L'ensemble de règles : trois conditions, trois styles, et une règle ELSE pour le reste.

Résultat : une carte, plusieurs messages superposés, et **zéro champ intermédiaire à créer**. La logique vit dans le style, pas dans la donnée.

PIÈGE À CONNAÎTRE

L'**ordre et le recouvrement des règles** comptent : une entité peut cocher deux conditions à la fois (une ville espagnole *et* en altitude). À vous de décider ce qui prime — par l'ordre des règles, ou en rendant les conditions mutuellement exclusives.

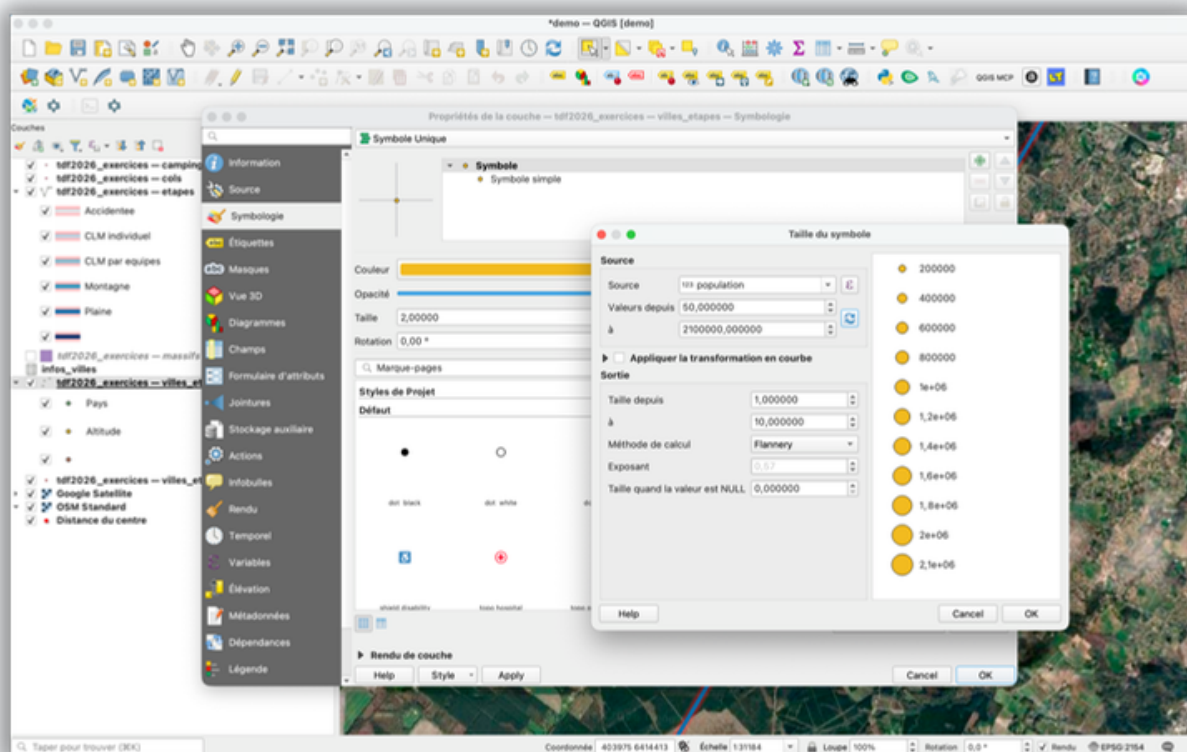
Le data-defined override : laisser la donnée piloter le style

Vous réglez la taille de vos symboles à la main ? Une taille pour tout le monde, ou trois catégories bricolées ? Pendant ce temps, vos données savent exactement quelle taille elles méritent.

À côté de chaque propriété de symbole — taille, couleur, rotation, opacité... — une petite icône passe inaperçue : le **data-defined override** (contrôle défini par les données). Cliquez dessus, et cette propriété n'est plus une valeur fixe : elle devient le reflet direct de vos données, via un champ ou une expression.

PAS À PAS

1. **Propriétés de la couche > Symbolologie**, sélectionnez votre symbole.
2. À droite du champ **Taille**, cliquez sur la petite icône de data-defined override.
3. Choisissez **Éditer...** et écrivez une expression basée sur vos données — par exemple une taille proportionnelle à la population : `"population" / 100000`.
4. Validez : chaque symbole prend la taille que sa donnée lui dicte.



La taille des villes pilotée par leur population : Barcelone et Paris dominent, sans aucune classe à créer.

Barcelone et Paris deviennent énormes, les petites communes minuscules — aucun champ intermédiaire, aucune classe à créer. Et ce principe ne s'arrête pas à la taille : une couleur qui suit une mesure, une flèche qui tourne selon un angle, une transparence qui traduit un indice de confiance.

Vous arrêtez de décrire vos données : vous les laissez se dessiner elles-mêmes. C'est le moment où la symbologie cesse d'être de la décoration pour devenir de l'analyse.

BON À SAVOIR

Quand un override est actif, l'icône change de couleur (jaune surligné) : c'est votre indice pour comprendre pourquoi une propriété « ne répond plus » aux réglages manuels. Pour revenir à une valeur fixe, désactivez l'override depuis ce même menu.

6

PARTIE 6

Mise en page

Produire des cartes, c'est bien. Laisser QGIS les produire à la chaîne pour vous, c'est mieux.

18 L'atlas

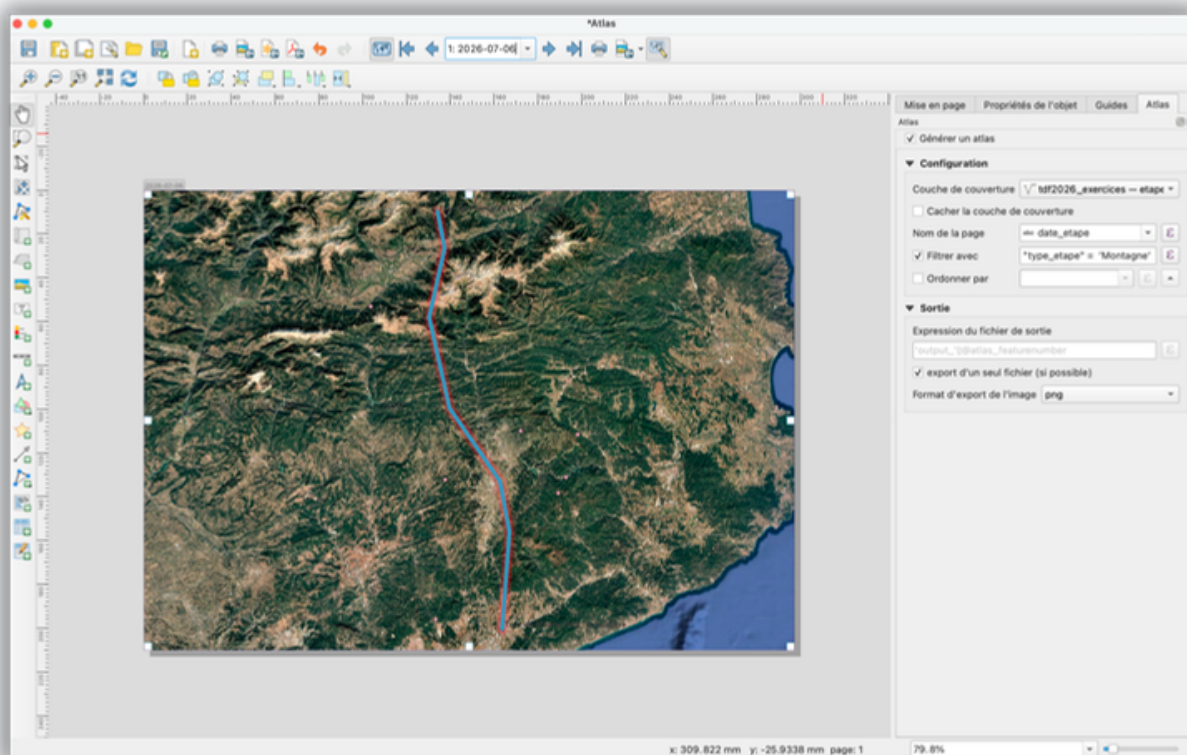
L'atlas : générer une carte par entité, automatiquement

On vous demande la même carte pour vos 21 secteurs. Vous ouvrez la mise en page, vous exportez, vous recadrez, vous recommencez... 21 fois ? Il existe un bouton qui fait tout cela pour vous : l'**atlas**.

Une mise en page sert à sortir une belle carte. Un atlas sort une belle carte *pour chaque entité de votre couche*, automatiquement. Vous concevez le gabarit une seule fois — titre, carte, légende, habillage — puis vous désignez la couche qui pilote l'atlas. QGIS décline le gabarit sur chaque entité : une étape, une page. 21 étapes, 21 pages — recentrées, recadrées, prêtes.

PAS À PAS

1. Créez votre mise en page (**Projet > Nouvelle mise en page**) et composez le gabarit : carte, titre, légende.
2. Dans le panneau **Atlas > Paramètres de l'atlas**, cochez **Générer un atlas** et désignez la **couche de couverture** (celle dont chaque entité fera une page).
3. Sur l'élément carte, cochez **Contrôlé par l'atlas** et réglez la marge autour de l'entité.
4. Pour un titre dynamique, utilisez une expression du type [% "nom" %] dans une étiquette.
5. **Atlas > Aperçu de l'atlas** pour vérifier, puis exportez : un clic, un PDF complet.



L'atlas dans la mise en page : un gabarit unique, décliné automatiquement sur chaque étape.

Le raffinement que peu de gens exploitent : le **filtre d'atlas**. Une expression, et vous ne générez que les pages qui vous intéressent — un carnet spécial montagne, par exemple, avec `"type_etape"` = `'Montagne'` . Ce qui vous prenait une après-midi de copier-coller devient un export de trente secondes.

LE POINT À RETENIR

C'est le jour où vous arrêtez de produire des cartes à la chaîne... et où vous laissez la chaîne les produire pour vous. Tout changement dans le gabarit se répercute sur toutes les pages : corrigez une fois, exportez de nouveau.

7

PARTIE 7

Actions

Toute cette série vous a montré comment lire, calculer et styliser vos données. Pour finir, une astuce qui change de nature : et si vous pouviez agir depuis votre carte ?

19 Les actions

Les actions : rendre vos couches cliquables

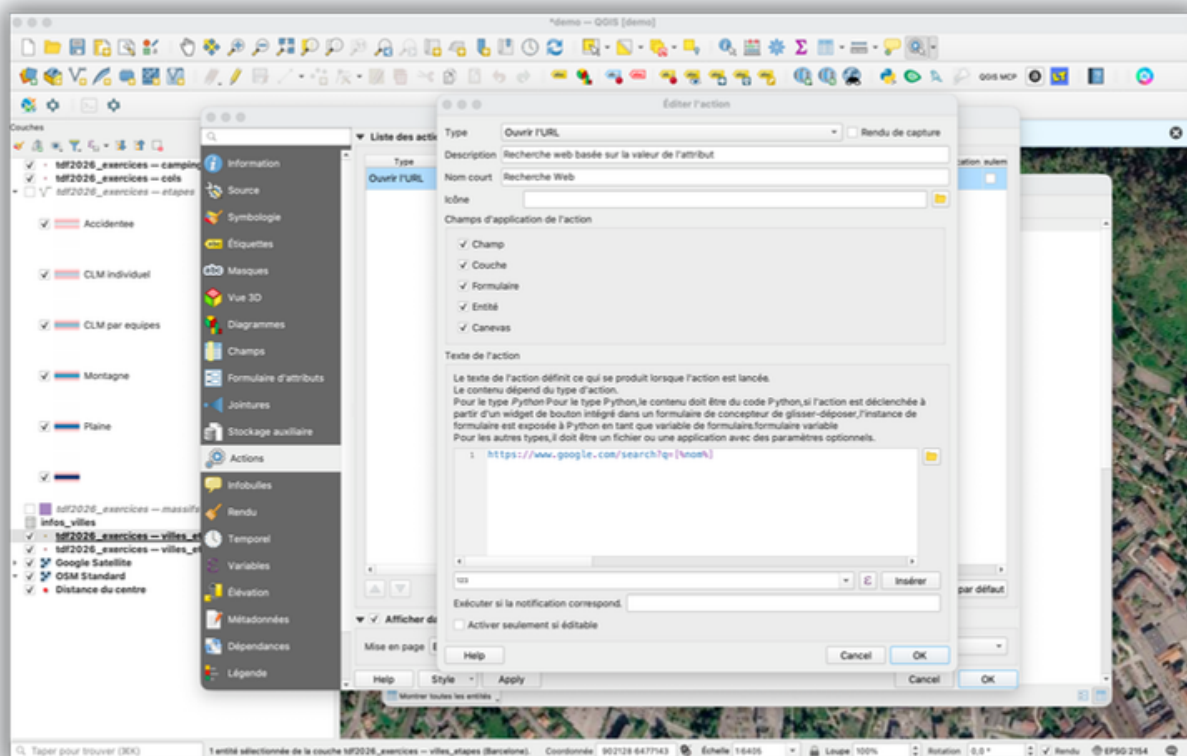
Une couche QGIS, ce n'est pas qu'un objet à regarder. On peut la rendre cliquable, vivante, connectée à l'extérieur. C'est le rôle des **actions** : associer à chaque entité un geste — ouvrir une page web, une photo de terrain, un dossier, ou même déclencher un script Python.

PAS À PAS

1. **Propriétés de la couche > Actions.**
2. Cliquez sur **Créer les actions par défaut** : QGIS vous en offre toute une série, prêtes à l'emploi.
3. Observez l'action « Recherche web » (voir ci-dessous) : la valeur de l'attribut est injectée dans l'URL entre les balises [% %] .
4. Sur la carte, activez l'outil **Exécuter l'action** (barre d'attributs) et cliquez sur une entité.

`https://www.google.com/search?q=[%nom%]`

Vous cliquez sur une ville : une recherche s'ouvre dans le navigateur avec son nom. La valeur de l'entité est injectée toute seule dans l'URL. Et ça va bien plus loin qu'une recherche : chaque entité devient un bouton vers ce que vous voulez — une fiche, une photo, un traitement.



L'onglet Actions : la recherche web par défaut, avec la valeur de l'attribut injectée dans l'URL.

LE POINT À RETENIR

QGIS n'est plus un visualiseur : c'est une **interface**. Et c'est exactement là que se termine ce rattrapage : vous avez les fondamentaux. La suite, c'est l'IA — sur des bases solides, elle devient redoutable.



BONUS

Deux astuces pour aller plus loin

Deux réflexes de qualité qui traversent toutes les astuces précédentes : neutraliser les NULL, et écrire des expressions que vous comprendrez encore dans six mois.

20 La fonction coalesce()

21 Les commentaires dans vos expressions

La fonction `coalesce()` : ne plus jamais être trahi par un NULL

Au fil de ce guide, vous avez croisé plusieurs fois les valeurs NULL : une jointure qui ne trouve pas de correspondance, un champ vide, un enregistrement sans donnée. Le NULL n'est pas un zéro, ni une chaîne vide. C'est l'**absence de valeur**, et il a une particularité redoutable : il contamine tout ce qu'il touche.

Additionnez un nombre avec un NULL, le résultat est NULL. Concaténez un texte avec un NULL, la chaîne entière devient NULL. Une seule valeur manquante peut ainsi faire échouer un calcul sur toute une colonne, sans le moindre message d'erreur.

La fonction `coalesce()` est votre filet de sécurité. Elle prend plusieurs arguments et renvoie le premier qui n'est pas NULL. Vous lui donnez une valeur de repli, et le vide est neutralisé.

```
coalesce("population", 0)
```

Ici, si le champ `population` est renseigné, sa valeur est utilisée. S'il est NULL, c'est 0 qui prend le relais. Votre somme ou votre moyenne ne sera plus cassée par une ligne incomplète. Le principe fonctionne avec tous les types ; pour du texte, vous pouvez renvoyer une mention par défaut :

```
coalesce("nom_massif", 'Hors massif')
```

Chaque camping sans massif affichera « Hors massif » au lieu d'un vide silencieux, ce qui rend l'information immédiatement lisible.

LE RÉFLEXE À PRENDRE

Dès que vous manipulez un champ susceptible de contenir des valeurs manquantes — dans une étiquette, un champ calculé ou une symbologie — enveloppez-le dans un `coalesce()`. Vous vous épargnerez des heures à chercher pourquoi un calcul « ne marche pas » alors que tout semble correct. Le coupable, neuf fois sur dix, est un NULL qui se propage en silence.

Les commentaires dans vos expressions : un code que vous comprendrez encore dans six mois

Vous écrivez une expression complexe dans la calculatrice de champs. Un CASE de quinze lignes, une condition imbriquée, un calcul de distance avec plusieurs seuils. Le jour même, tout est limpide. Six mois plus tard, vous rouvrez le projet et vous ne comprenez plus rien à votre propre logique.

QGIS accepte les **commentaires** dans les expressions, exactement comme un langage de programmation. C'est une fonctionnalité que presque personne n'utilise, alors qu'elle transforme un bloc obscur en un raisonnement qui s'explique tout seul.

Sur une seule ligne, deux tirets ouvrent un commentaire. Tout ce qui suit sur la ligne est ignoré par QGIS :

```
CASE
  WHEN "denivele_m" ≥ 3500 THEN 'Haute montagne' -- seuil des étapes reines
  WHEN "denivele_m" ≥ 2000 THEN 'Vallonné'        -- moyenne montagne
  ELSE 'Plat'
END
```

Pour un commentaire sur plusieurs lignes, ou pour désactiver temporairement une portion d'expression sans l'effacer, vous disposez des délimiteurs `/*` et `*/` :

```
/* Classement des étapes par difficulté.
   Seuils à réviser si le jeu de données change. */
```

Ce second usage est particulièrement précieux quand vous testez une expression : plutôt que de supprimer une condition pour voir ce que ça change, vous la mettez en commentaire, vous vérifiez le résultat, puis vous la réactivez. Vous ne perdez jamais votre travail.

LE RÉFLEXE À ADOPTER

Dès qu'une expression dépasse quelques lignes ou repose sur des seuils qui pourraient évoluer, commentez-la. Vous n'écrivez pas seulement pour QGIS : vous écrivez pour le vous du futur, et pour le collègue qui reprendra le projet. Une expression commentée, c'est une expression qui se transmet.

Vous avez consolidé vos bases. Et maintenant ?

Maintenant, c'est le moment de passer à la vitesse supérieure. Et vous n'êtes pas obligé de le faire seul.

Les Géomagiciens, c'est la communauté que j'ai construite pour les géomaticiens qui veulent progresser ensemble : échanger, poser leurs questions, partager leurs projets, et surtout ne pas rester bloqués seuls devant leur écran. C'est l'endroit où l'on transforme les fondamentaux que vous venez de travailler en véritable expertise appliquée.



Les administrateurs de la communauté des Géomagiciens.

À mes côtés pour animer cette communauté, vous retrouverez notamment Dr SIG, un espace où partager vos problèmes : nos administrateurs vous répondent.

Et parce que les fondamentaux ne sont qu'un point de départ, la communauté vous donne accès à bien plus. La formation « L'IA pour les Géomaticiens » est incluse dans votre adhésion. C'est la suite directe de ce document : là où vous apprenez à mettre l'intelligence artificielle au service de votre travail géospatial, sur les bases solides que vous venez de poser.

Les fondamentaux d'abord. L'IA ensuite. Et une communauté pour vous accompagner à chaque étape.

Rejoignez les Géomagiciens

<https://datando.fr/datando-geomagicien>



MES MEILLEURES ASTUCES QGIS

DATANDO · DATANDO.FR