

REX 06079

Contexte

Affichage des IRIS de Mandelieu-la-Napoule (06079) depuis PostGIS Alteris, avec population graduée, étiquettes, fond orthophoto IGN Géoplateforme et cadastre vecteur WFS — entièrement piloté depuis Claude Code via le MCP QGIS, sans interaction manuelle dans QGIS.

Données utilisées

Table / Source	Schéma / Provider	Contenu
contours_iris_2025	insee_raw (PostGIS Alteris)	Géométries IRIS France entière
iris_2025_population	insee_raw (PostGIS Alteris)	Population 2022 par IRIS (recensement)
Orthophoto IGN	Géoplateforme (WMTS → XYZ)	HR.ORTHOIMAGERY.ORTHOPHOTOS — sans clé API
Cadastre parcelles	Géoplateforme WFS CADASTRALPARCELS.PARCELLAIRE_EXPRESS:parcelle	Parcelles cadastrales vecteur — OGR driver

Colonnes clés — pièges à retenir

contours_iris_2025 : colonnes standard sans espace — code_insee, code_iris, nom_iris, type_iris, geom

iris_2025_population : colonnes avec **espace traînant** (import INSEE brut) :

- Clé IRIS : "IRIS " (avec espace)
- Commune : "COM " (avec espace)
- Population totale 2022 : "P22_POP " (avec espace)
- Toutes les colonnes sont de type text — caster en ::numeric avant usage

Résultats

9 IRIS pour 06079, population 2022 :

code_iris	nom_iris	population
060790101	Zone d'activités	24
060790102	IRIS 2	2 701
060790103	IRIS 3	2 000
060790104	IRIS 4	3 058
060790105	IRIS 5	2 197
060790106	IRIS 6	2 432
060790107	IRIS 7	3 188
060790108	IRIS 8	2 944
060790109	IRIS 9	2 656

Population min : **24 hab.** (zone d'activités/naturelle) — max : **3 188 hab.**

Cadastre : **26 439 parcelles** après reprojection EPSG:4326 → EPSG:3857 (COUNT=2000 limite la requête WFS initiale à 2000 features, mais la reprojection en mémoire conserve toutes les géométries valides).

Procédure complète

Prérequis

Tunnel SSH paramiko actif (voir page 229 — Option 2). À relancer à chaque redémarrage de QGIS :

```
import sys
sys.path.insert(0, r'C:\Users\eliob\AppData\Roaming\Python\Python312\site-packages')
import paramiko
import socket, threading, select

class SSHTunnel(threading.Thread):
    def __init__(self, ssh_host, ssh_user, ssh_password,
```

```

        remote_port, local_port=5433, remote_host='127.0.0.1'):
super().__init__(daemon=True)
self.ssh_host = ssh_host; self.ssh_user = ssh_user
self.ssh_password = ssh_password; self.remote_host = remote_host
self.remote_port = remote_port; self.local_port = local_port
self._stop = threading.Event(); self.transport = None; self.server_sock = None

def run(self):
    client = paramiko.SSHClient()
    client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    client.connect(self.ssh_host, username=self.ssh_user, password=self.ssh_password)
    self.transport = client.get_transport()
    self.server_sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    self.server_sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    self.server_sock.bind(('127.0.0.1', self.local_port))
    self.server_sock.listen(5); self.server_sock.settimeout(1.0)
    while not self._stop.is_set():
        try:
            conn, _ = self.server_sock.accept()
            threading.Thread(target=self._forward, args=(conn,), daemon=True).start()
        except socket.timeout:
            continue
    self.server_sock.close(); client.close()

def _forward(self, local_conn):
    try:
        chan = self.transport.open_channel('direct-tcpip',
            (self.remote_host, self.remote_port), local_conn.getpeername())
    except Exception:
        local_conn.close(); return
    while True:
        r, _, _ = select.select([local_conn, chan], [], [], 5)
        if local_conn in r:
            data = local_conn.recv(4096)
            if not data: break
            chan.send(data)
        if chan in r:
            data = chan.recv(4096)
            if not data: break

```

```

        local_conn.send(data)
    chan.close(); local_conn.close()

def stop(self):
    self._stop.set()

tunnel = SSHTunnel('79.137.14.202', 'debian', 'RAW+NEXTE!', remote_port=5432, local_port=5433)
tunnel.start()

```

1. Charger l'orthophoto IGN (fond de carte)

Bug QGIS 4.0.3 : `addMapLayer` déclenche `autoSelectAddedLayer` → `identifyMapTool` → access violation quand une couche raster est ajoutée. Deux contournements obligatoires :

1. Passer sur l'outil **Pan** avant d'ajouter la couche
2. Utiliser `addMapLayer(layer, False)` + `root.insertLayer()` (sans auto-sélection)

Format URI : le provider WMTS natif de QGIS (`crs=...&layers=...&url=...`) échoue sur la Géoplateforme (pas de capabilities). Solution : encoder l'URL WMTS en **XYZ tiles** avec `{z}/{y}/{x}`

```

from qgis.core import QgsProject, QgsRasterLayer
from qgis.gui import QgsMapToolPan
from qgis.utils import iface
import urllib.parse

# 1. Passer sur Pan AVANT d'ajouter la couche raster (évite le crash identifyMapTool)
pan_tool = QgsMapToolPan(iface.mapCanvas())
iface.mapCanvas().setMapTool(pan_tool)

# 2. URI XYZ (le provider WMTS natif échoue sur data.geopf.fr)
base_url = (
    "https://data.geopf.fr/wmts?SERVICE=WMTS&REQUEST=GetTile"
    "&VERSION=1.0.0&LAYER=HR.ORTHOIMAGERY.ORTHOPHOTOS"
    "&STYLE=normal&FORMAT=image/jpeg"
    "&TILEMATRIXSET=PM&TILEMATRIX={z}&TILEROW={y}&TILECOL={x}"
)
encoded = urllib.parse.quote(base_url, safe='')

```

```

uri = f"type=xyz&url={encoded}&zmin=0&zmax=19"

layer_ortho = QgsRasterLayer(uri, "Orthophoto IGN", "wms")
# 3. addToLegend=False + insertLayer manuel (évite autoSelectAddedLayer)
QgsProject.instance().addMapLayer(layer_ortho, False)
root = QgsProject.instance().layerTreeRoot()
root.insertLayer(-1, layer_ortho) # en bas de la pile

```

2. Créer la vue PostGIS et charger les IRIS

```

import psycopg2
conn = psycopg2.connect(host='localhost', port=5433, dbname='alteris_geo',
                        user='alteris_admin', password='Alteris2026')

cur = conn.cursor()
cur.execute("""
    CREATE OR REPLACE VIEW insee_raw.v_iris_mandelieu_pop AS
    SELECT c.fid, c.geom, c.nom_iris, c.code_iris,
           ROUND(p."P22_POP ">::numeric, 0)::integer AS population
    FROM insee_raw.contours_iris_2025 c
    LEFT JOIN insee_raw.iris_2025_population p ON p."IRIS " = c.code_iris
    WHERE c.code_insee = '06079'
""")
conn.commit(); conn.close()

from qgis.core import QgsVectorLayer
uri = (
    'host=localhost port=5433 dbname=alteris_geo '
    'user=alteris_admin password=Alteris2026 sslmode=disable '
    'table="insee_raw"."v_iris_mandelieu_pop" (geom) key=fid'
)
layer = QgsVectorLayer(uri, "IRIS Mandelieu-la-Napoule", "postgres")

```

3. Graduation par population (semi-transparent)

```

from qgis.core import QgsGraduatedSymbolRenderer, QgsRendererRange, QgsFillSymbol

colors = ['#EFF3FF', '#BDD7E7', '#6BAED6', '#2171B5', '#084594']
pmin, pmax = 24, 3188
step = (pmax - pmin) / 5
bounds = [pmin + i * step for i in range(6)]

ranges_def = []
for i in range(5):
    lo, hi = bounds[i], bounds[i+1]
    sym = QgsFillSymbol.createSimple({
        'color': colors[i], 'outline_color': '#555555', 'outline_width': '0.3'
    })
    sym.setOpacity(0.6) # semi-transparent pour laisser voir l'ortho
    ranges_def.append(QgsRendererRange(lo, hi, sym,
        f"{int(lo):,}-{int(hi):,} hab.".replace(',', ' ')))

layer.setRenderer(QgsGraduatedSymbolRenderer('population', ranges_def))

```

4. Étiquettes + ajout dans le bon ordre

```

from qgis.core import (QgsPalLayerSettings, QgsTextFormat,
    QgsTextBufferSettings, QgsVectorLayerSimpleLabeling)
from qgis.PyQt.QtGui import QColor, QFont

pal = QgsPalLayerSettings()
pal.fieldName = "concat(nom_iris, '\n', population, ' hab.')"
pal.isExpression = True
pal.placement = QgsPalLayerSettings.Placement.OverPoint
fmt = QgsTextFormat()
fmt.setFont(QFont("Arial", 8)); fmt.setSize(8); fmt.setColor(QColor('#1a1a1a'))
buf = QgsTextBufferSettings()
buf.setEnabled(True); buf.setSize(1); buf.setColor(QColor('white'))
fmt.setBuffer(buf); pal.setFormat(fmt)
layer.setLabelsEnabled(True)
layer.setLabeling(QgsVectorLayerSimpleLabeling(pal))

# IRIS en haut de la pile (index 0), ortho déjà en bas

```

```
QgsProject.instance().addMapLayer(layer, False)
root = QgsProject.instance().layerTreeRoot()
root.insertLayer(0, layer)
```

5. Zoomer sur Mandelieu

```
from qgis.core import QgsRectangle, QgsCoordinateReferenceSystem, QgsCoordinateTransform

src_crs = QgsCoordinateReferenceSystem("EPSG:4326")
dst_crs = QgsCoordinateReferenceSystem("EPSG:3857")
transform = QgsCoordinateTransform(src_crs, dst_crs, QgsProject.instance())
pt_min = transform.transform(6.87, 43.51)
pt_max = transform.transform(6.99, 43.60)
extent = QgsRectangle(pt_min.x(), pt_min.y(), pt_max.x(), pt_max.y())
iface.mapCanvas().setExtent(extent)
iface.mapCanvas().refresh()
```

6. Charger le cadastre (WFS vecteur)

Pièges :

- Le provider **WFS natif QGIS** est invalide sur `data.geopf.fr` pour ce typename → utiliser le driver **OGR WFS**
- L'**API Carto** (`geo.api.gouv.fr`) a timeout (appel synchrone dans le thread QGIS, 20 s) → abandonné
- Le WFS retourne les données en **EPSG:4326** mais le canevas est en **EPSG:3857** → la couche apparaît dans le panneau mais n'est pas visible sur la carte → reproj obligatoire

```
import processing
from qgis.core import (QgsVectorLayer, QgsCoordinateReferenceSystem,
                      QgsFillSymbol, QgsSingleSymbolRenderer, QgsProject)

# Driver OGR WFS avec BBOX en paramètre URL (le provider WFS natif QGIS échoue)
url = (
    "WFS:https://data.geopf.fr/wfs/ows?SERVICE=WFS&VERSION=2.0.0&REQUEST=GetFeature"
    "&TYPENAME=CADASTRALPARCELS.PARCELLAIRE_EXPRESS:parcelle"
    "&BBOX=6.87,43.51,6.99,43.60,EPSG:4326&COUNT=2000"
)
layer_wfs = QgsVectorLayer(url, "Cadastre Mandelieu", "ogr")
```

```
# → isValid() = True, 2000 features, EPSG:4326

# Reprojection en mémoire EPSG:3857 (sinon couche invisible dans le canevas)
result = processing.run("native:reprojectlayer", {
    'INPUT': layer_wfs,
    'TARGET_CRS': QgsCoordinateReferenceSystem('EPSG:3857'),
    'OUTPUT': 'memory:'
})
layer_cad = result['OUTPUT']
layer_cad.setName("Cadastre Mandelieu")
# → 26 439 features, EPSG:3857

# Symbolologie : contour orange, remplissage quasi-transparent
sym = QgsFillSymbol.createSimple({
    'color': '204,68,0,30',      # orange très transparent (alpha 30/255)
    'outline_color': '#CC4400',
    'outline_width': '0.5'
})
layer_cad.setRenderer(QgsSingleSymbolRenderer(sym))

# Ajouter au-dessus des IRIS (index 0)
QgsProject.instance().addMapLayer(layer_cad, False)
root = QgsProject.instance().layerTreeRoot()
root.insertLayer(0, layer_cad)
```

Ordre final des couches (haut → bas) :

1. Cadastre Mandelieu (vecteur WFS, EPSG:3857)
2. IRIS Mandelieu-la-Napoule (PostGIS, gradué population)
3. Orthophoto IGN (XYZ, fond)

Note cadastre : le BBOX rectangulaire déborde sur les communes voisines (Cannes à l'est, Pégomas au nord). Comportement attendu — pour restreindre au territoire communal strict, il faudrait une intersection post-chargement (pas de filtre `code_commune` disponible côté WFS Géoplateforme).

Points clés et pièges

Sujet	Constat
-------	---------

Bug QGIS 4.0.3 — raster + identifyMapTool	<code>addMapLayer</code> sur une couche raster → access violation si l'outil Identifier est actif. Fix : passer sur Pan + <code>addMapLayer(layer, False)</code> + <code>insertLayer()</code>
Ne pas cliquer sur la couche raster dans le panneau	Même après ajout réussi, cliquer sur la couche raster dans le panneau des couches déclenche <code>onActiveLayerChanged</code> → crash. Rester sur l'outil Pan.
Provider WMTS natif QGIS invalide	<code>crs=...&layers=...&url=https://data.geopf.fr/wmts</code> → couche invalide. Fix : encoder en XYZ avec <code>type=xyz&url=...{z}/{y}/{x}</code>
Canvas au zoom mondial au démarrage	Les tuiles ne se chargent pas. Toujours zoomer sur la zone cible après ajout.
Provider WFS natif QGIS invalide	<code>QgsVectorLayer(url, name, "WFS")</code> → couche invalide pour <code>CADASTRALPARCELS.PARCELLAIRE_EXPRESS:parcelle</code> . Fix : driver OGR WFS <code>QgsVectorLayer("WFS:https://...", name, "ogr")</code>
API Carto timeout	<code>geo.api.gouv.fr</code> : appel HTTP synchrone dans le thread QGIS → 20 s de blocage puis timeout. Driver OGR WFS Géoplateforme préférable.
CRS mismatch WFS	WFS retourne EPSG:4326, canevas EPSG:3857 → couche présente dans le panneau mais invisible sur la carte. Fix : <code>processing.run("native:reprojectlayer", ...)</code> vers EPSG:3857 en mémoire.
Colonnes INSEE avec espace	<code>"IRIS "</code> , <code>"COM "</code> , <code>"P22_POP "</code> — espace traînant partout dans <code>iris_2025_population</code>
Types text	Toutes les valeurs numériques INSEE sont en <code>text</code> — caster avec <code>::numeric</code>
Sous-requête URI QGIS	<code>QgsDataSourceUri.setDataSource()</code> sur-échappe les guillemets → couche invalide. Fix : créer une vue PostgreSQL
<code>sys.executable</code> QGIS	Pointe sur <code>qgis-bin.exe</code> — pip via subprocess inutilisable. Fix : <code>pip._internal.cli.main</code>
<code>sys.path</code> paramiko	QGIS n'inclut pas le dossier utilisateur Python. Ajouter <code>sys.path.insert(0, ...)</code> à chaque session

Généralisation à d'autres communes

```
code_insee = '06079' # remplacer par le code cible
cur.execute(f"""
    CREATE OR REPLACE VIEW insee_raw.v_iris_pop AS
```

```
SELECT c.fid, c.geom, c.nom_iris, c.code_iris, c.nom_commune,  
       ROUND(p."P22_POP ">::numeric, 0)::integer AS population  
FROM insee_raw.contours_iris_2025 c  
LEFT JOIN insee_raw.iris_2025_population p ON p."IRIS " = c.code_iris  
WHERE c.code_insee = '{code_insee}'  
""")
```

Pour le cadastre, adapter le BBOX à l'emprise de la commune cible.

REX — 2026-06-21 (second run)

Résultat : 0 crash — procédure validée reproductible.

Run complet depuis un projet QGIS vide, piloté intégralement via MCP Claude Code → QGIS :

Étape	Résultat
Tunnel SSH paramiko	OK — PostGIS Alteris accessible localhost:5433
Orthophoto IGN (XYZ)	Valide — rendu fond de carte immédiat
Vue PostGIS + couche IRIS	9 features, graduation 5 classes bleu, étiquettes population
Zoom sur Mandelieu	Cadrage correct EPSG:3857
Cadastre WFS OGR + reprojection	26 439 parcelles EPSG:3857, symbologie orange

Conclusion : la procédure est stable et reproductible. Base solide pour une routine généralisable à n'importe quelle commune (`code_insee` + BBOX comme seuls paramètres variables). Potentiel : automatiser la production de fiches communales IRIS + cadastre + fond ortho à la demande depuis Claude Code.