Generalitat de Catalunya
Centre de Telecomunicacions
i Tecnologies de la Informació

Next Generation
Catalunya

Generalitat
de Catalunya

RECOMANACIONS I BONES PRÀCTIQUES TRACTAMENT DADES GEOESPACIALS A DATABRICKS

Direcció de la Dada i Intel·ligència Artificial

Abril del 2026

Objectiu del document

Aquest document és una guia de recomanacions i bones pràctiques per l'ús de dades geoespacials a Databricks. Inclou enllaços a documentació addicional

Contingut

1. Emmagatzemar dades vectorials en tipus natiu GEOMETRY amb SRID explícit

Databricks ara admet tipus natius GEOMETRY i GEOGRAPHY (Public Preview des de DBR 17.1). Recomanacions:

- **Convertiu sempre** les dades a tipus natiu GEOMETRY en escriure en taules Delta — no deixeu les dades com a cadenes WKT o binaris WKB.
- **Especifiqueu sempre l'SRID**, p. ex. GEOMETRY(4326) per a WGS84. El valor per defecte és 0 (desconegut), cosa que fa perdre la informació de projecció. Databricks admet ~11.000 SRID.

Funcions de creació segons el format d'origen:

Format d'origen	Funció
WKT	ST_GeomFromWKT('POINT(1 2)', 4326)
WKB	ST_GeomFromWKB(col, srid)
GeoJSON	ST_GeomFromGeoJSON(col)
Coordenades	ST_Point(longitud, latitud)

Nota: les taules Iceberg encara no admeten columnes GEOMETRY — feu servir Delta. El tipus GEOGRAPHY té un suport creixent, però recomanem GEOMETRY com a tipus principal.

2. Materialitzar una columna H3 i aplicar Liquid Clustering

Encara que avui no feu servir H3 directament, recomanem precomputar una columna H3 i aplicar **Liquid Clustering** sobre aquesta columna. Això co-localitza les dades espacialment per habilitar *data-skipping* automàtic.

Funcions de creació segons el tipus de geometria:

Cas	Funció	Retorna
Punts (coordenades)	h3_longlatash3(lon, lat, res)	BIGINT
Punts (GEOGRAPHY)	h3_pointash3(geogExpr, res)	BIGINT
Polígons (tesel·lació)	h3_tessellateaswkb(geogExpr, res)	WKB chips
Cobertura (totes les cel·les)	h3_coverash3(geogExpr, res)	ARRAY<BIGINT>
Emplenat (centroides a dins)	h3_polyfillash3(geogExpr, res)	ARRAY<BIGINT>

Patró de "chipping" amb h3_tessellateaswkb: aquesta funció talla la geometria en peces al llarg de les vores H3. Cada *chip* és un **core chip** (totalment contingut — s'omet el costós ST_Contains) o un **boundary chip** (intersecció parcial — s'aplica el predicat només sobre aquest tros). Això redueix dràsticament el càlcul per a *joins* amb polígons grans.

> **Limitació actual:** les funcions H3 d'importació accepten expressions GEOGRAPHY però internament treballen amb WKT/WKB — encara no estan optimitzades per al tipus natiu GEOMETRY directament. Si les vostres dades són GEOMETRY, exporteu primer a WKB amb st_asbinary() o convertiu-les a GEOGRAPHY.

Emmagatzemeu els identificadors de cel·la com a **BIGINT** (no com a cadena) i apliqueu **Liquid Clustering** sobre aquesta columna. El Liquid Clustering directament sobre columnes GEOMETRY és al *roadmap* però encara no està disponible.
Més detall sobre el patró de *chipping*: Spatial Analytics at Any Scale with H3 and Photon

3. Spatial Joins — funcionen "out of the box"

Amb tipus natius a **DBR 17.3+**, els *spatial joins* amb ST_Contains, ST_Intersects, etc. s'optimitzen automàticament amb:

- **Indexació R-tree** automàtica
- ***Spatial joins* optimitzats a Photon**
- **Optimització intel·ligent de *range join***

Això dona fins a **17x millor rendiment** comparat amb llibreries externes. *Benchmarks*: 2.500M polígons × 450M punts → 200M+ coincidències. **No cal que canvieu el codi** — els tipus natius porten estadístiques de *bounding box* per a *data-skipping* automàtic.

Si en algun cas trobeu problemes de rendiment a molt gran escala, aviseu-nos — existeixen patrons addicionals com ara fer servir indexació H3 explícita (el patró de *chipping* del punt 2) per prefiltrar abans d'aplicar predicats geomètrics. Però l'expectativa és que tot hauria de funcionar directament i continuarà millorant amb el *roadmap*.
Més detall: Spatial Joins 17x Faster Out-of-the-Box

4. Dades Raster — encara sense tipus natiu, però amb una metodologia clara

Databricks encara no té un tipus raster natiu (és al *roadmap*). La metodologia recomanada:

- **Ingerir** fitxers raster (GeoTIFF, COG, etc.) i emmagatzemar-los a **Unity Catalog Volumes**
- **Dividir/trossejar** ràsters grans en *tiles* manejables mitjançant eines com GeoBrix RasterX (gbx_rst_clip, subdivisió per índex de graella). Manteniu una **taula Delta de metadades** amb data d'adquisició, resolució, bandes i *bounding boxes*.
- **Indexar** els *tiles* per cel·les H3 per a co-localització espacial i cerca eficient
- **Enriquir** retallant ràsters contra geometries vectorials d'alt valor
- Convertiu les **bounding boxes** a tipus natiu GEOMETRY (ST_GeomFromWKB('bbox', srid)) per a consultes espacials
- Per a **servir/visualitzar**, les sortides processades es poden carregar a **Lakebase** (PostgreSQL *serverless* de Databricks) per a renderització interactiva de mapes, o bé utilitzar connectors d'ESRI i eines de visualització

5. Arquitectura Medallion per a dades geoespacials

Capa	Què es fa
Bronze Raw	Ingerir tots els formats font (GeoJSON, Shapefile, GeoDB, OSM, GeoTIFF, COG, CSV, Parquet) amb transformació mínima. Escriure a **Delta Lake + Liquid Clustering**. Fer servir les API de DataFrame per abstraure les diferències de format.
Silver Refined	Validar i estandarditzar l'SRID/projecció. Construir taules base de geometria amb tipus natiu GEOMETRY. Afegir un **índex de límits** (xmin, ymin, xmax, ymax).
Silver+ Enriched	Calcular l'índex H3 global. Construir **taules de graella global** amb *chips* tesel·lats. Materialitzar columnes H3 com a BIGINT + Liquid Clustering.

Gold Analytics	API de Spatial SQL, API d'H3, models d'IA/ML, combinació amb dades no geoespacials, quadres de comandament. Els *joins* entre dominis es simplifiquen — dades espacials i de negoci en una sola capa.

El principi clau: tractar les **dades geoespacials com a ciutadanes de primera classe**. Tant el vector com el ràster aterren a la mateixa graella amb el CRS abstraït, de manera que els *joins* es converteixen en comprovacions d'igualtat d'índex en lloc d'operacions geomètriques costoses.

6. Recursos addicionals

Tutorial: Pipeline geoespacial con tipos nativos	Guia pas a pas *end-to-end*
Introducing Spatial SQL in Databricks	80+ funcions, tipus natius, *spatial joins*
Spatial Joins 17x Faster	Millores de rendiment a DBR 17.3
H3 + Photon Spatial Analytics	Patró de *chipping*/tesel·lació
Referencia: Funciones ST geoespaciales	80+ funcions SQL, Python, Scala
Referencia: Funciones H3	30+ funcions d'indexació global