Requisito BDN.01 – Modelagem de Dados – DSM – Profa. Lucineide

Banco de Dados: HypoOrbit

1. Criação do Banco de Dados

Scripts disponíveis em:

https://github.com/HypogramTechnologies/HypoOrbit/blob/develop/docs/banco_dados/hypoorbit_scripts.js

```
>_MONGOSH

> use hypoorbit;

< already on db hypoorbit
> db.createCollection("satellite");

< { ok: 1 }
> db.createCollection("satellite_item_assets");

< { ok: 1 }
hypoorbit>
```

```
>_MONGOSH
         "type": "image/tiff; application=geotiff; profile=cloud-optimized",
             "data"
         "title": "Land Cover Classification Image",
         satellite: db.satellite.findOne({ id: "LCC_L8_30_16D_STK_Cerrado-1" })._id
         "key": "thumbnail",
         "type": "image/png",
         "bands": {
             "red": "lcc",
             "blue": "lcc",
             "green": "lcc"
         "roles": [
             "thumbnail"
         "title": "Thumbnail",
         satellite: db.satellite.findOne({ id: "LCC_L8_30_16D_STK_Cerrado-1" })._id
     '0': ObjectId('68d72cb32c75f0ebbfd8efbd'),
     '1': ObjectId('68d72cb32c75f0ebbfd8efbe'),
     '2': ObjectId('68d72cb32c75f0ebbfd8efbf'),
     '3': ObjectId('68d72cb32c75f0ebbfd8efc0'),
     '4': ObjectId('68d72cb32c75f0ebbfd8efc1'),
     '5': ObjectId('68d72cb32c75f0ebbfd8efc2'),
     '6': ObjectId('68d72cb32c75f0ebbfd8efc3'),
     '7': ObjectId('68d72cb32c75f0ebbfd8efc4'),
     '8': ObjectId('68d72cb32c75f0ebbfd8efc5'),
     '9': ObjectId('68d72cb32c75f0ebbfd8efc6')
```

2. Modelagem de Relacionamentos

Scripts disponíveis em:

https://github.com/HypogramTechnologies/HypoOrbit/blob/develop/docs/banco_dados/hypoorbit_scripts.js

```
>_MONOOSH
> db.satellite.findOne({ id: "mosaic=cbers4a=paraiba=3m=1" });

c {
    _id: ObjectId('68d7zcb3zc7sf0ebbfd8efba'),
    id: 'mosaic=cbers4a=paraiba=3m=1',
    type: 'Collection',
    stac_version: 'l.0.0',
    stac_version: 'l.0.0',
    stac_version: [
        'https://stac=extensions.github.io/version/vl.0.0/schema.json',
        'https://stac=extensions.github.io/processing/vl.0.0/schema.json',
        'https://stac=extensions.github.io/eo/vl.0.0/schema.json']
        'inttps://stac=extensions.github.io/eo/vl.0.0/schema.json']
        'inttps://stac=extensions.github.io/eo/vl.0.0/schema.json']
        'inttps://stac=extensions.github.io/eo/vl.0.0/schema.json']
        'inttps://stac=extensions.github.io/eo/vl.0.0/schema.json']
        'inttps://stac=extensions.github.io/eo/vl.0.0/schema.json']
        'inttps://stac=extensions.github.io/eo/vl.0.0/schema.json']
        'inttps://
```

```
>_MONGOSH
db.satellite.aggregate([
         $lookup: {
             from: "satellite_item_assets",
             localField: "_id",
             foreignField: "satellite",
             as: "item_assets"
    _id: ObjectId('68d72cb32c75f0ebbfd8efba'),
   type: 'Collection',
   stac_extensions: [
     'https://stac-extensions.github.io/version/v1.0.0/schema.json',
     'https://stac-extensions.github.io/processing/v1.0.0/schema.json',
     'https://stac-extensions.github.io/item-assets/v1.0.0/schema.json',
     'https://stac-extensions.github.io/eo/v1.0.0/schema.json'
   title: 'CBERS-4A/WFI Image Mosaic of Brazil Paraiba State - 3 Months',
   deprecated: false.
   description: 'CBERS-4A/WFI image mosaic of Brazil Paraiba State with 55m of spatial resolution...',
      'mosaic',
     'cbers',
     'visible wavelengths',
     'ultraviolet wavelengths'.
     'cloud optimized geotiff',
     'earth observation',
```

Na primeira modelagem/consulta usamos **Embedding**, pois as informações principais do satélite (como properties, providers, keywords, extent) são sempre acessadas junto com o satélite. Já na segunda usamos **Referencing**, pois cada satélite pode ter muitos assets (imagens, bandas, thumbnails) que podem ser grandes e detalhados, além disso isso nos permite atualizar ou adicionar assets independentemente do satélite, sem precisar reescrever todo o documento principal.

3. Scripts MongoDB

Scripts disponíveis em:

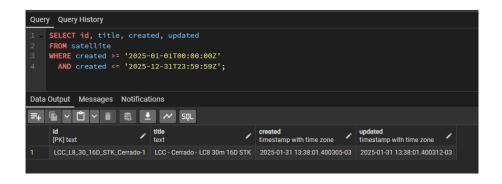
https://github.com/HypogramTechnologies/HypoOrbit/blob/develop/docs/banco_dados/hypoorbit_scripts.js

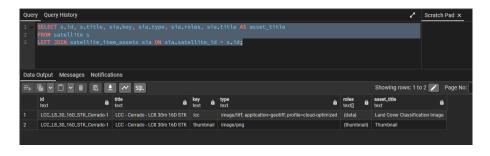
```
>_MONGOSH
> db.satellite.find(
     title: 1,
     "providers.processing:parameters.spectralIndices": 1,
   title: 'CBERS-4A/WFI Image Mosaic of Brazil Paraíba State - 3 Months',
   title: 'AMAZONIA-1/WFI - Level-4-SR - Cloud Optimized GeoTIFF',
       'processing:parameters': {
           'NDVI',
```

```
>_MONGOSH
> db.satellite.find(
     $and: [
         $or: [
          { id: "mosaic-cbers4a-paraiba-3m-1" },
           { id: "AMZ1-WFI-L4-SR-1" }
       { "properties.updated": { $type: "date" } }
     id: 1,
     title: 1,
     "properties.created": 1,
     "properties.updated": 1,
     _id: 0
   id: 'mosaic-cbers4a-paraiba-3m-1',
   title: 'CBERS-4A/WFI Image Mosaic of Brazil Paraíba State - 3 Months',
   properties: {
     created: 2024-05-03T17:58:15.432Z,
```

4. Comparação com SQL (PostgreSQL) Scripts disponíveis em:

https://github.com/HypogramTechnologies/HypoOrbit/blob/develop/docs/banco_dados/hypoorbit_scripts.sql





- **Tabela satellite**: armazena os metadados principais (id, título, datas de criação/atualização, descrição etc.).
- **Tabela satellite_provider**: armazena os provedores de cada satélite, vinculados pela chave estrangeira satellite_id.
- Tabela satellite_processing_parameters: armazena os parâmetros técnicos (bands, índices espectrais, etc.) de cada provider.
- Tabela satellite_item_assets: armazena os assets vinculados ao satélite.