Understanding Native Projections in Neo4j

What is a Native Projection?

A Native Projection is a type of graph projection in Neo4j Graph Data Science (GDS) that uses the actual structure of the database for creating an in-memory graph. This approach directly maps nodes and relationships from the database to a projected graph, offering efficient and dynamic representation.

Key Characteristics of Native Projections

1. **Direct Mapping**: Uses nodes and relationships directly from the database.
2. **Customizable Properties**: Allows inclusion of specific properties for nodes and relationships.
3. **Dynamic**: Reflects changes in the database in real time.
4. **Fast Setup**: Leverages Neo4j's indexing for rapid creation of projections.

How to Create a Native Projection

1. **Create the Projection**:

```
CALL gds.graph.project(
    'nativeGraph',
    'Person',
    'FRIENDS_WITH',
    {
        nodeProperties: ['age'],
        relationshipProperties: ['strength']
    }
);
```

2. **View the Graph**:

```
CALL gds.graph.list('nativeGraph');
```

3. **Run an Algorithm** (e.g., Shortest Path):

```
CALL gds.shortestPath.stream('nativeGraph', {
    sourceNode: 1,
    targetNode: 2
})
YIELD index, sourceNode, targetNode, totalCost
RETURN index, sourceNode, targetNode, totalCost;
```

4. **Drop the Projection**:

```
CALL gds.graph.drop('nativeGraph');
```

Benefits of Native Projections

1. **Efficiency**: Faster computations due to direct mapping.
2. **Flexibility**: Allows tailored projections by selecting specific properties.
3. **Dynamic Updates**: Changes in the database are reflected in the graph.
4. **Minimal Transformation**: Avoids the need for complex data transformations.

Example Use Case

Data:

- **Nodes**: Person (with properties: age, name).
- **Relationships**: FRIENDS_WITH (with property: strength).

Steps:

1. **Create the Native Projection**:

```
CALL gds.graph.project('friendsGraph', 'Person', 'FRIENDS_WITH', {
    nodeProperties: ['age'],
    relationshipProperties: ['strength']
```

});

2. **Run an Algorithm** (e.g., PageRank):

```
CALL gds.pageRank.stream('friendsGraph')
YIELD nodeId, score
RETURN gds.util.asNode(nodeId).name AS name, score
ORDER BY score DESC;
```

3. **Drop the Projection**:

```
CALL gds.graph.drop('friendsGraph');
```

Conclusion

Native Projections in Neo4j GDS provide an efficient and dynamic way to create and analyze in-memory graphs. They are ideal for scenarios where minimal transformation and real-time updates are critical.