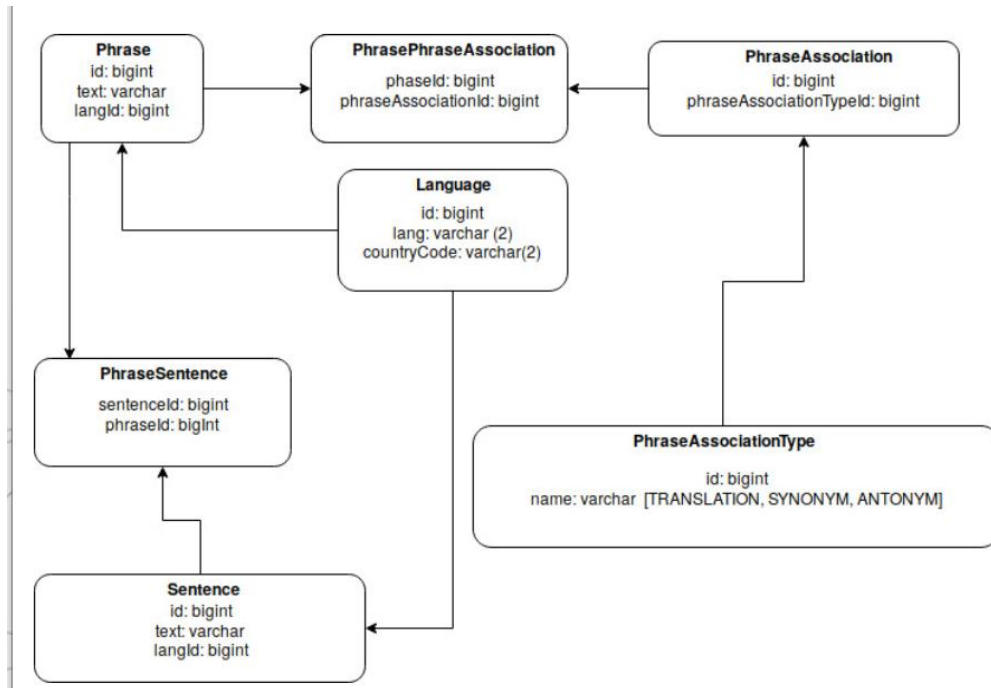
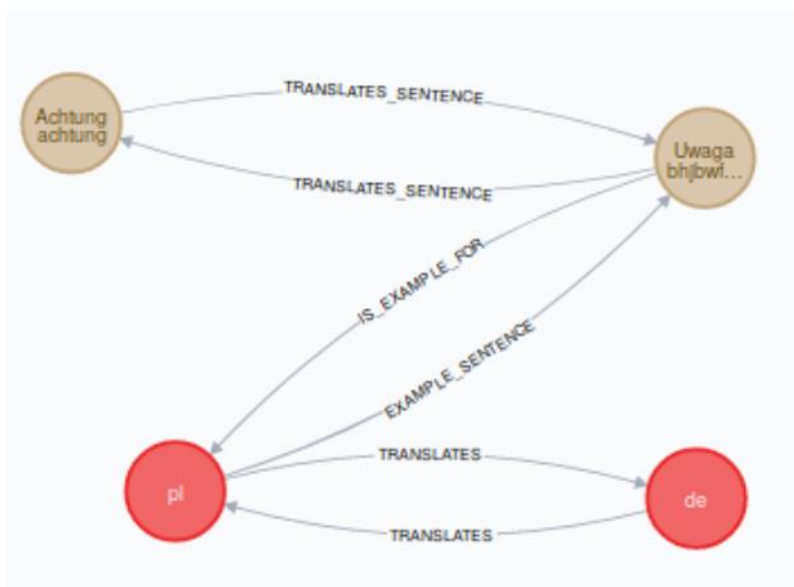


Migración de datos de PostgreSQL a Neo4j

Diagrama Relacional



Modelo Grafico



Migración de datos paso a paso

Nuestra aplicación, así como las bases de datos, se ejecutan en la ventana acoplable, por lo que los pasos a continuación incluyen las operaciones realizadas en ella.

1. Prepare una consulta SQL para seleccionar datos y exportarlos a un archivo CSV. *

La consulta de ejemplo que devuelve todas las traducciones tiene el siguiente aspecto.
Usaremos partes marcadas en color azul en pasos posteriores.

```
COPY (  
  SELECT p1.id      AS ID_FROM,  
         p1.text    as TEXT_FROM,  
         l1.lang     as LANG_FROM,  
         p2.id      AS ID_TO,  
         p2.text    as TEXT_TO,  
         l2.lang     as LANG_TO  
  FROM zettelchen_phrase p1  
       JOIN zettelchen_language l1 ON p1.lang_id = l1.id  
       JOIN zettelchen_phrase_phrase_association zppa1 on p1.id = zppa1.phrase_id  
       JOIN zettelchen_phrase_association zpa ON zppa1.phrase_association_id = zpa.id  
       JOIN zettelchen_association_type zat ON zat.id = zpa.association_type_id  
       JOIN zettelchen_phrase_phrase_association zppa2 ON zppa2.phrase_association_id = zpa.id  
       JOIN zettelchen_phrase p2 ON zppa2.phrase_id = p2.id  
       JOIN zettelchen_language l2 ON p2.lang_id = l2.id  
  WHERE p1.id <> p2.id  
        AND zat.name = 'TRANSLATION'  
        AND (p1.text = '') IS NOT TRUE  
        AND (p2.text = '') IS NOT TRUE  
  ) TO '/tmp/translations.csv' WITH CSV header;
```

Tenga en cuenta que cuando ejecutamos esta consulta en el archivo contenedor de la ventana acoplable, también se almacenará en el contenedor en la ubicación dada.

2. Prepare consultas CYPHER con el uso de la herramienta LOAD CSV para importar datos. *

```

USING PERIODIC COMMIT

LOAD CSV WITH HEADERS FROM 'file:/translations.csv' AS row

MERGE (from:Phrase {
    externalId: row.id_from
})

ON CREATE SET

from.uuid = randomUUID(),

from.text = row.text_from,

from.lang = row.lang_from

MERGE (to:Phrase {
    externalId: row.id_to
})

ON CREATE SET

to.uuid = randomUUID(),

to.text = row.text_to,

to.lang = row.lang_to

MERGE (from)-[r:TRANSLATES {
    code: row.lang_from + '-' + row.lang_to
}]->(to);

```

Como puede ver, obtenemos todos los valores del archivo csv y los leemos línea por línea creando nuevos nodos y relaciones. También coloreé algunos campos para mostrar que estamos usando los mismos campos que exportamos anteriormente. El archivo CSV debe colocarse en el directorio de importación . Puede nombrarlo import.cypher .

3. Ejecute el script desde el primer paso en postgres

Para poder exportar nuestros datos podemos usar psql. Este simple comando carga un archivo dado y lo ejecuta. Podemos llamarlo desde fuera del contenedor de la ventana acoplable, pero el archivo exportado se generará dentro.

```
psql -h $POSTGRES_HOST -p $POSTGRES_PORT -U $POSTGRES_USER -d $POSTGRES_DB < export_csv.sql
```

4 .Obtenga archivos csv exportados desde el contenedor de postgres.

```
docker cp $POSTGRES_CONTAINER_ID:/tmp/translations.csv translations.csv
```

```
docker cp Translations.csv $ NEO4J_CONTAINER_ID: /import/translations.csv
```

6. Mueva el script preparado en el paso 2 también a este contenedor de la ventana acoplable pero al directorio / tmp

```
docker cp import_csv.cypher $ NEO4J_CONTAINER_ID: /tmp/import_csv.cypher
```

7. Utilice la herramienta cypher-shell con un script preparado como entrada. *

Prepare el script de shell con el siguiente comando y luego cópielo de la misma manera que copiamos el archivo en el paso 6.

```
NOMBRE DE USUARIO = $ 1
```

```
CONTRASEÑA = $ 2
```

```
cat /tmp/import_csv.cypher | / var / lib / neo4j / bin / cypher-shell -u $ USERNAME -p $  
PASSWORD
```

Ahora podemos ejecutar el script en Docker usando el comando exec :

```
docker exec $ NEO4J_CONTAINER_NAME /tmp/$YOUR_SHELL_SCRIPT.sh $ USERNAME $  
PASSWORD
```

La operación debería terminar con un mensaje de éxito; en otro caso, verá un mensaje de error significativo en la consola. Para ver un mensaje más detallado Puede agregar indicadores `--debug --format verbose` a `cypher-shell`.

8. Elimine archivos temporales.

En este paso solo verificamos si el archivo existe y lo eliminamos.

```
if [ -f translations.csv ] ; then  
    rm translations.csv  
fi
```

* : Los pasos marcados con un asterisco son necesarios en el proceso de migración de datos. El resto de los pasos están conectados con la infraestructura de Docker que preparé en mi máquina. Significa que tenemos que hacer solo 3 simples pasos para que sea completamente funcional.