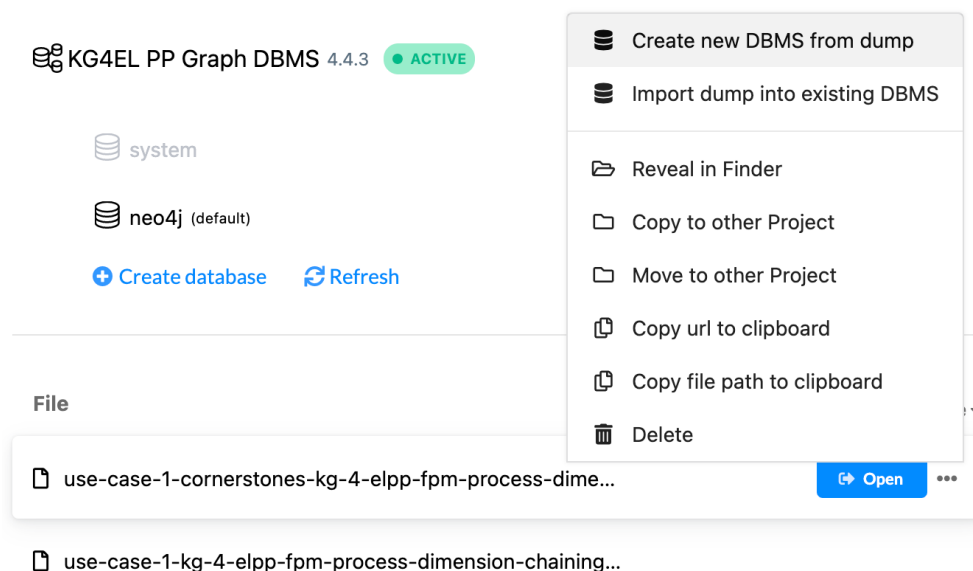


## Cypher Query Commands & Query Examples

Import the DUMP file to view the knowledge graph by creating a new graph database instance in Neo4j and importing the graph database DUMP as a new database.



### Knowledge Graph Requirements

R1: Understanding all processes event log class diagrams of the structure of the processes event log

R2: Understanding the event log class diagram of the target data model

R3: Visualise event log challenges on data quality as well as through a taxonomy (semantic nodes)

R4: Identify MIL which is contained in event data attribute nodes

R5: Identify duplicates which is contained in event data attribute nodes

R6: Identify interoperability problems – by identifying data types, differences in the timestamp format

R7: Percentage of MIL represent the NULLs in event data attribute nodes

R8: Script is run locally with Python and can be compared and viewed in Neo4j's KG

R9: Visualise and connect the imported the event logs and semantic nodes which represent Data Quality and Event Log Challenges through explicitly modelled relationships

R10: Added information nodes to the knowledge graphs

R11: Check how many data entities fit the target data model through relationships

R12: Not fulfilled. Event logs can be compared but the creation of a target data model which is to be constructed is still required.

### Queries R1 – R11

```
// Show the event log class diagram of BPIC Supplier 1
```

```
MATCH (n:BPI_C_2019_Supplier_1) RETURN n
```

```
// FIT and CHANGE Relationships overview
```

```
// Find all change relationships
```

```
MATCH p=()-[r:CHANGE]->() RETURN p
```

```
// Find all fit relationships
```

```
MATCH p=()-[r:FIT]->() RETURN p
```

```
// All of the CHANGE relationships between BPIC
```

```
// 2019_FEL and BPIC 2019 Supplier 1
```

```
MATCH p=(:BPI_C_2019_Federated_Event_Log)-[r:CHANGE]->(:BPI_C_2019_Supplier_1) RETURN p
```

```
// All of the CHANGE relationships between BPIC
```

```
// 2019_FEL and BPIC 2019 Supplier 2
```

```
// Revealing Datetime stamps do not fit
```

```
// requested in the requirements
```

```
MATCH p=(:BPI_C_2019_Federated_Event_Log)-[r:CHANGE]->(:BPI_C_2019_Supplier_1) RETURN p
```

```
// All of the CHANGE relationships between BPIC
```

```
// 2019_FEL and BPIC 2019 Supplier 3
```

```
MATCH p=(:BPI_C_2019_Federated_Event_Log)-[r:CHANGE]->(:BPI_C_2019_Supplier_3) RETURN p
```

```
// Count how many FIT and CHANGE relationships exist
```

```
MATCH (a:BPI_C_2019_Supplier_1)-[r:FIT|CHANGE]-
```

```
(c:BPI_C_2019_Federated_Event_Log)
```

```
RETURN type(r), count(r)
```

```
// or in more depth
```

```
MATCH (a:BPI_C_2019_Supplier_1)-[r:FIT|CHANGE]-
```

```
(c:BPI_C_2019_Federated_Event_Log)
```

```
RETURN type(r) as relationship_type, count(r) as relationship_count,
```

```
collect(distinct a) as start_nodes, collect(distinct c) as end_nodes
```

```
//Query the Data quality and Event log Challenges explicitly modelled
```

```
to the taxonomy the between the synthetic BPIC2019 supplier data set
```

```
and the discovered challenges from the Meta DataFrame
```

```
MATCH (a:BPI_C_2019_Supplier_3)-[r]-(c:`Semantic Node`)
```

```
RETURN type(r) as relationship_type, count(r) as relationship_count,
```

```
collect(distinct a) as start_nodes, collect(distinct c) as end_nodes
```