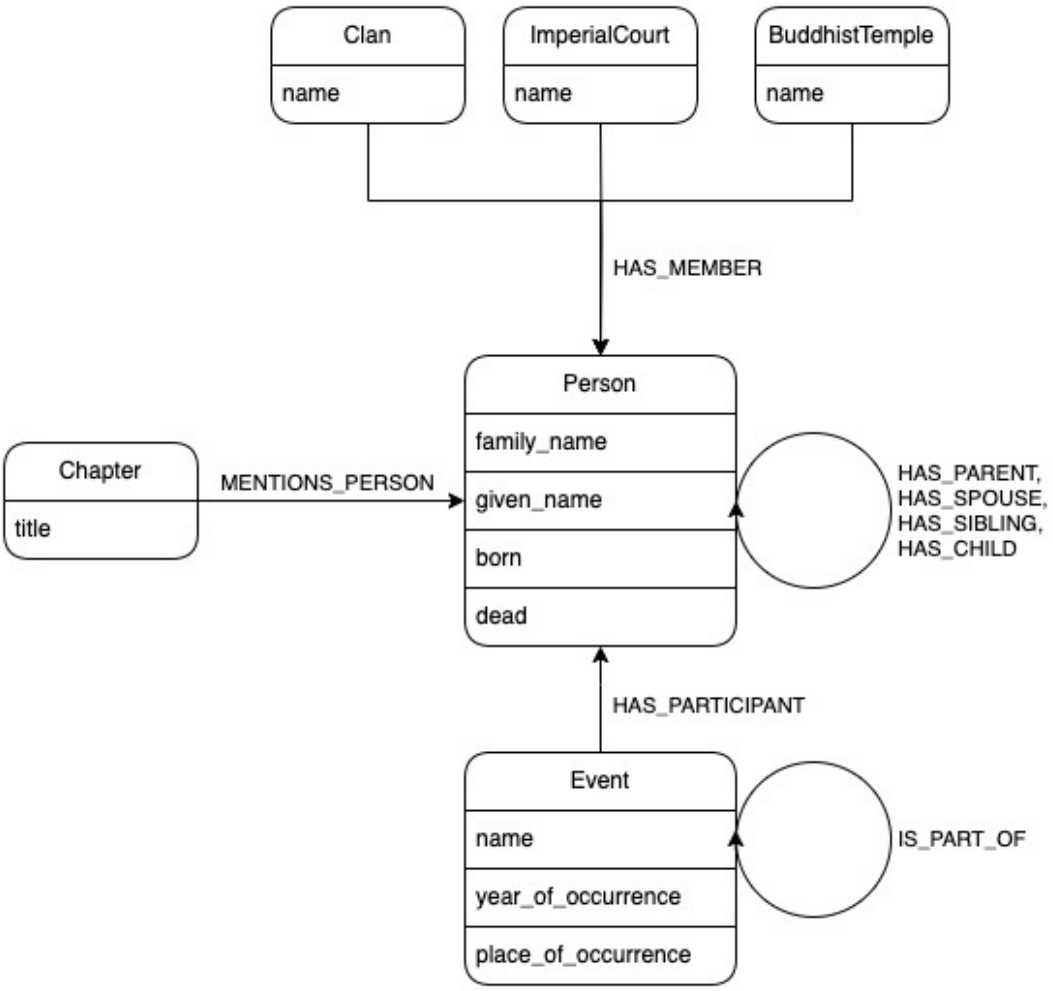
Appendix

Access `https://github.com/kinue00/KG_Eval` for data and code examples to reproduce the results.

A.1 KG schema in entity relationship diagram



A.2 LLM Prompt to generate KG

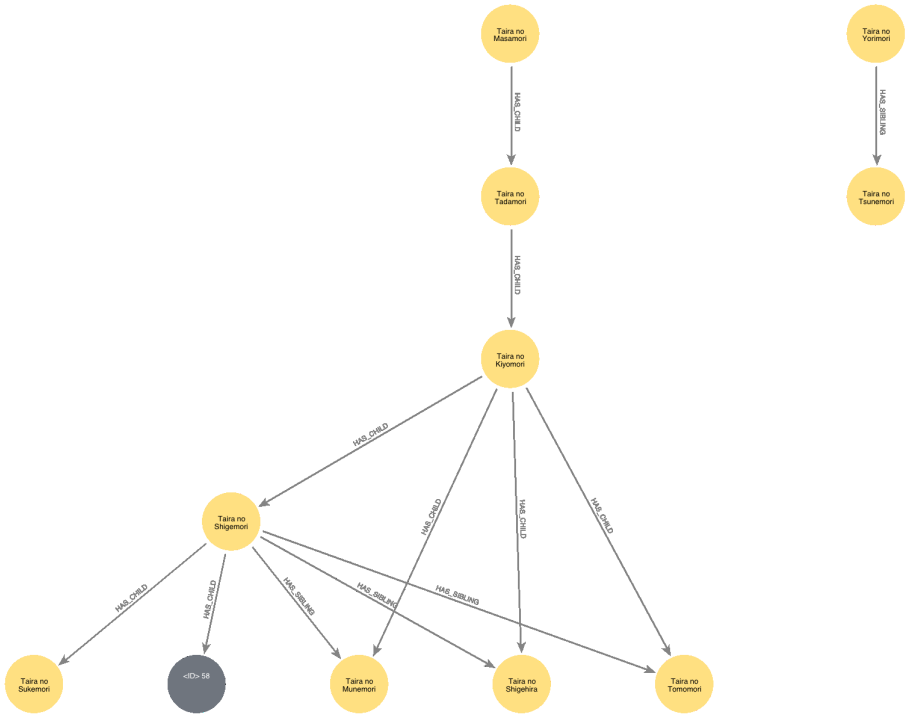
```
Generate Neo4j Cypher queries to create a comprehensive knowledge graph for The Tale of the Heike
(Heike Monogatari) following these specifications:

1. Create nodes with these labels and properties:
- Person: {name, family_name (optional), given_name (optional), born (optional), dead (optional)}
- Clan: {name}
- ImperialCourt: {name} (use Heian or Kamakura period)
- BuddhistTemple: {name}
- Chapter: {title}
- Event: {name, year_of_occurrence (optional), place_of_occurrence (optional)}

2. Create relationships following these patterns ONLY:
- (Clan)-[:HAS_MEMBER]->(Person)
- (ImperialCourt)-[:HAS_MEMBER]->(Person)
- (BuddhistTemple)-[:HAS_MEMBER]->(Person)
- (Chapter)-[:MENTIONS_PERSON]->(Person)
```

```
- (Event)-[:HAS_PARTICIPANT]->(Person)
- (Event)-[:IS_PART_OF]->(Event)
- (Person)-[:HAS_PARENT]->(Person)
- (Person)-[:HAS_CHILD]->(Person)
- (Person)-[:HAS_SIBLING]->(Person)
- (Person)-[:HAS_SPOUSE]->(Person)

3. Requirements:
- Generate queries in chronological order of the story
- Include all major characters, events, and relationships mentioned in the text
- Use historically accurate dates where available
- Add appropriate indexes and constraints
- Use MERGE instead of CREATE to avoid duplicates
- Include comments explaining each major section of queries

Generate clean, well-formatted Cypher queries that can be directly executed in Neo4j.
```

A.3 Cypher queries to extract genealogy for the Taira Clan

```
// Show Taira family tree with both parent-child and spouse relationships
MATCH (clan:Clan {name: "Taira Clan"})
MATCH (clan)-[:HAS_MEMBER]->(p1:Person)
MATCH (clan)-[:HAS_MEMBER]->(p2:Person)
MATCH path = (p1)-[:HAS_CHILD|HAS_SPOUSE*1..5]->(p2)
RETURN path;
```

A.4 Extracted Taira genealogy



A.5 Genealogy of the Heike from The Heike Monogatari