Here's an explanation of the key components related to Neo4j:

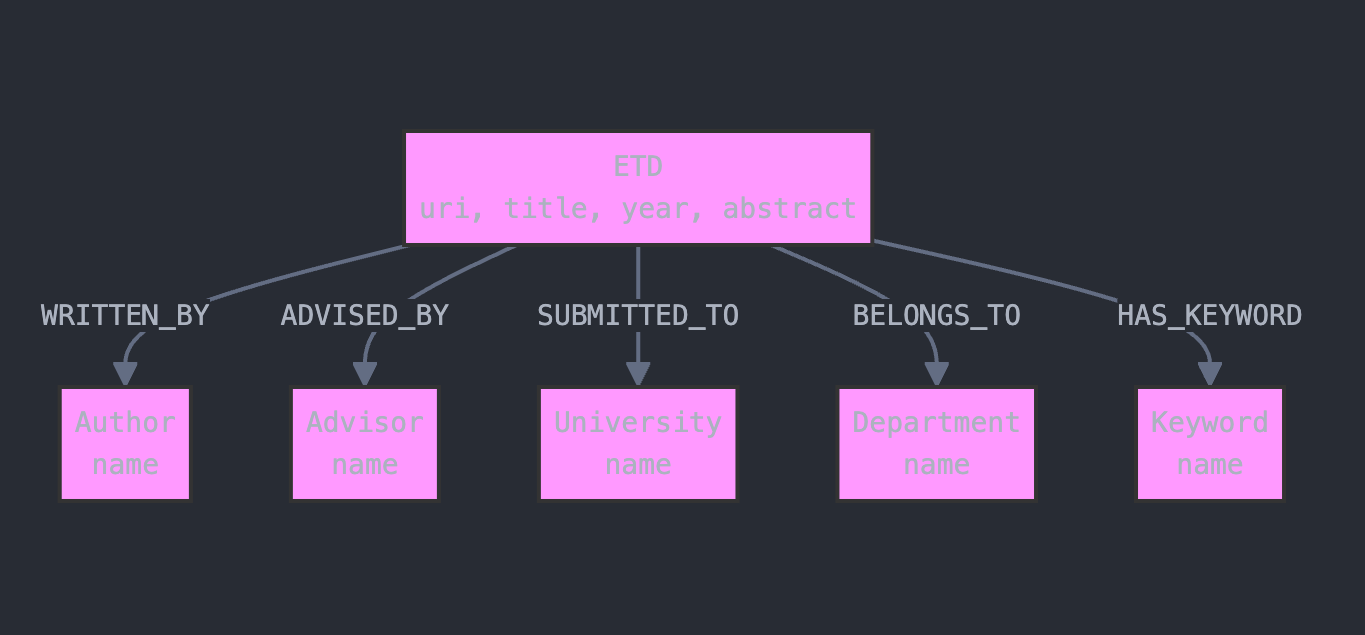
**Node Types**

1. **ETD** - The core entity representing a thesis/dissertation with properties:
   * uri (unique identifier)
   * title
   * year
   * abstract
2. **Author** - Person who wrote the ETD, with name as the primary property
3. **Advisor** - Faculty member who supervised the research
4. **University** - Institution where the ETD was submitted
5. **Department** - Academic department associated with the ETD
6. **Keyword** - Subject terms associated with the ETD

**Relationship Types**

1. **WRITTEN\_BY** - Connects ETD to Author
2. **ADVISED\_BY** - Connects ETD to Advisor
3. **SUBMITTED\_TO** - Connects ETD to University
4. **BELONGS\_TO** - Connects ETD to Department
5. **HAS\_KEYWORD** - Connects ETD to Keyword

The schema includes all the uniqueness constraints defined in your neo4j\_browser\_instructions.txt file and matches the query patterns used in Neo4jaccess.py and Neo4j\_loader.py.



More Details:

// ETD Neo4j Database Schema (Updated)

// ======================================================

// Node Labels with Properties

// ======================================================

// ETD - The electronic thesis or dissertation document

// Properties:

// uri: Unique identifier for the ETD (constraint enforced)

// id: Numeric identifier

// title: Title of the thesis/dissertation

// year: Publication year

// abstract: Text summary of the thesis content

// language: Language of the ETD (e.g., "eng")

// schooltype: Type of school (e.g., "REGULAR")

// oadsclassifier: Open Access Digital Scholarship classifier

// borndigital: Flag indicating if the ETD was born digital

// number: Sequential number in dataset

// Author - Person who wrote the ETD

// Properties:

// name: Full name of the author (constraint enforced)

// Advisor - Faculty member who supervised the ETD

// Properties:

// name: Full name of the advisor (constraint enforced)

// University - Institution where the ETD was submitted

// Properties:

// name: Name of the university (constraint enforced)

// code: Short code for the university (e.g., "ucb")

// Department - Academic department associated with the ETD

// Properties:

// name: Name of the department (constraint enforced)

// Keyword - Subject terms associated with the ETD

// Properties:

// name: Term representing a subject or topic (constraint enforced)

// ======================================================

// Relationships

// ======================================================

// WRITTEN\_BY: Connects ETD to Author

// ETD -[WRITTEN\_BY]-> Author

// ADVISED\_BY: Connects ETD to Advisor

// ETD -[ADVISED\_BY]-> Advisor

// SUBMITTED\_TO: Connects ETD to University

// ETD -[SUBMITTED\_TO]-> University

// BELONGS\_TO: Connects ETD to Department

// ETD -[BELONGS\_TO]-> Department

// HAS\_KEYWORD: Connects ETD to Keyword

// ETD -[HAS\_KEYWORD]-> Keyword

// ======================================================

// Schema Constraints

// ======================================================

// Create uniqueness constraints to ensure data integrity

CREATE CONSTRAINT etd\_uri\_unique IF NOT EXISTS

FOR (e:ETD) REQUIRE e.uri IS UNIQUE;

CREATE CONSTRAINT etd\_id\_unique IF NOT EXISTS

FOR (e:ETD) REQUIRE e.id IS UNIQUE;

CREATE CONSTRAINT keyword\_name\_unique IF NOT EXISTS

FOR (k:Keyword) REQUIRE k.name IS UNIQUE;

CREATE CONSTRAINT author\_name\_unique IF NOT EXISTS

FOR (a:Author) REQUIRE a.name IS UNIQUE;

CREATE CONSTRAINT advisor\_name\_unique IF NOT EXISTS

FOR (a:Advisor) REQUIRE a.name IS UNIQUE;

CREATE CONSTRAINT university\_name\_unique IF NOT EXISTS

FOR (u:University) REQUIRE u.name IS UNIQUE;

CREATE CONSTRAINT dept\_name\_unique IF NOT EXISTS

FOR (d:Department) REQUIRE d.name IS UNIQUE;

// ======================================================

// Example Data Creation with Real Data

// ======================================================

// Example ETD Record Creation from sample data

MERGE (e:ETD {uri: "https://escholarship.org/uc/item/0hn5z4f1"})

SET e.id = "1436",

e.number = 0,

e.title = "Plasma Diagnostics and Plasma-Surface Interactions in Inductively Coupled Plasmas",

e.year = "2010",

e.abstract = "The semiconductor industry's continued trend of manufacturing device features on the nanometer scale requires increased plasma processing control and improved understanding of plasma characteristics and plasma-surface interactions. This dissertation presents a series of experimental results for focus studies conducted in an inductively coupled plasma (ICP) system...",

e.language = "eng",

e.schooltype = "REGULAR",

e.oadsclassifier = "0",

e.borndigital = "0"

// Create Author and relationship to ETD

MERGE (a:Author {name: "Titus, Monica Joy"})

WITH a

MATCH (e:ETD {uri: "https://escholarship.org/uc/item/0hn5z4f1"})

MERGE (e)-[:WRITTEN\_BY]->(a)

// Create Advisor and relationship to ETD

MERGE (adv:Advisor {name: "Graves, David B"})

WITH adv

MATCH (e:ETD {uri: "https://escholarship.org/uc/item/0hn5z4f1"})

MERGE (e)-[:ADVISED\_BY]->(adv)

// Create University and relationship to ETD

MERGE (u:University {code: "ucb", name: "University of California, Berkeley"})

WITH u

MATCH (e:ETD {uri: "https://escholarship.org/uc/item/0hn5z4f1"})

MERGE (e)-[:SUBMITTED\_TO]->(u)

// Example of a second ETD from sample data

MERGE (e2:ETD {uri: "https://escholarship.org/uc/item/0sn1r9st"})

SET e2.id = "1437",

e2.number = 1,

e2.title = "Declarative Systems",

e2.year = "2011",

e2.abstract = "Building system software is a notoriously complex and arduous endeavor. Developing tools and methodologies for practical system software engineering has long been an active area of research...",

e2.language = "eng",

e2.schooltype = "REGULAR",

e2.oadsclassifier = "0",

e2.borndigital = "0"

// Create Author for second ETD

MERGE (a2:Author {name: "Condie, Tyson"})

WITH a2

MATCH (e2:ETD {uri: "https://escholarship.org/uc/item/0sn1r9st"})

MERGE (e2)-[:WRITTEN\_BY]->(a2)

// Create Advisor for second ETD

MERGE (adv2:Advisor {name: "Hellerstein, Joseph M"})

WITH adv2

MATCH (e2:ETD {uri: "https://escholarship.org/uc/item/0sn1r9st"})

MERGE (e2)-[:ADVISED\_BY]->(adv2)

// Link second ETD to same university

MATCH (u:University {code: "ucb"})

MATCH (e2:ETD {uri: "https://escholarship.org/uc/item/0sn1r9st"})

MERGE (e2)-[:SUBMITTED\_TO]->(u)

// Create Keywords and relationships to ETDs

FOREACH (keyword IN ["Plasma", "Semiconductor", "Diagnostics"] |

MERGE (k:Keyword {name: keyword})

WITH k

MATCH (e:ETD {uri: "https://escholarship.org/uc/item/0hn5z4f1"})

MERGE (e)-[:HAS\_KEYWORD]->(k)

)

FOREACH (keyword IN ["Declarative Programming", "System Software", "Cloud Computing"] |

MERGE (k:Keyword {name: keyword})

WITH k

MATCH (e2:ETD {uri: "https://escholarship.org/uc/item/0sn1r9st"})

MERGE (e2)-[:HAS\_KEYWORD]->(k)

)

// ======================================================

// Example Queries

// ======================================================

// Get all ETDs with metadata

MATCH (e:ETD)

OPTIONAL MATCH (e)-[:WRITTEN\_BY]->(author:Author)

OPTIONAL MATCH (e)-[:ADVISED\_BY]->(advisor:Advisor)

OPTIONAL MATCH (e)-[:SUBMITTED\_TO]->(u:University)

OPTIONAL MATCH (e)-[:BELONGS\_TO]->(d:Department)

RETURN

e.id,

e.uri,

e.title,

author.name AS Author,

advisor.name AS Advisor,

e.year AS Year,

u.name AS University,

e.language,

e.schooltype,

d.name AS Department,

e.abstract AS Abstract

LIMIT 100;

// Find ETDs by keyword

MATCH (k:Keyword {name: "Cloud Computing"})<-[:HAS\_KEYWORD]-(e:ETD)

RETURN e.title, e.year, e.uri;

// Find ETDs by year

MATCH (e:ETD)

WHERE e.year = "2010"

RETURN e.title, e.uri;

// Get ETDs by university code

MATCH (u:University {code: "ucb"})<-[:SUBMITTED\_TO]-(e:ETD)

RETURN e.title, e.year, e.uri;

// Get ETD count

MATCH (e:ETD)

RETURN count(e) AS ETDCount;

// Get all advisors and their ETD counts

MATCH (a:Advisor)<-[:ADVISED\_BY]-(e:ETD)

RETURN a.name, count(e) as ETDCount

ORDER BY ETDCount DESC;

// Find ETDs by language

MATCH (e:ETD)

WHERE e.language = "eng"

RETURN e.title, e.year, e.uri;