Knowledge Graph with Grakn

By Siddharth Khillon

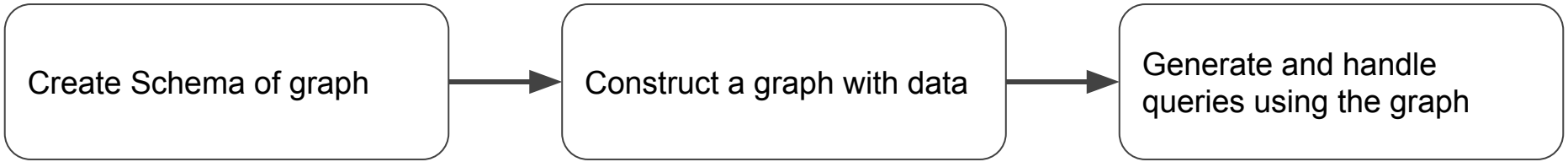
Uses of Grakn

- Relational database that is easy to set up and query
- Fast query speeds due to relational system
- Easier to set up with a simple schema system

Quickstart summary

---

1. Download Grakn from the Grakn docs into the location where the graph will be stored
2. Create a schema containing the description of the graph
3. Run the schema.gql into a keyspace which will contain your data
4. You can then access the knowledge graph and insert/query data



Setup Overview

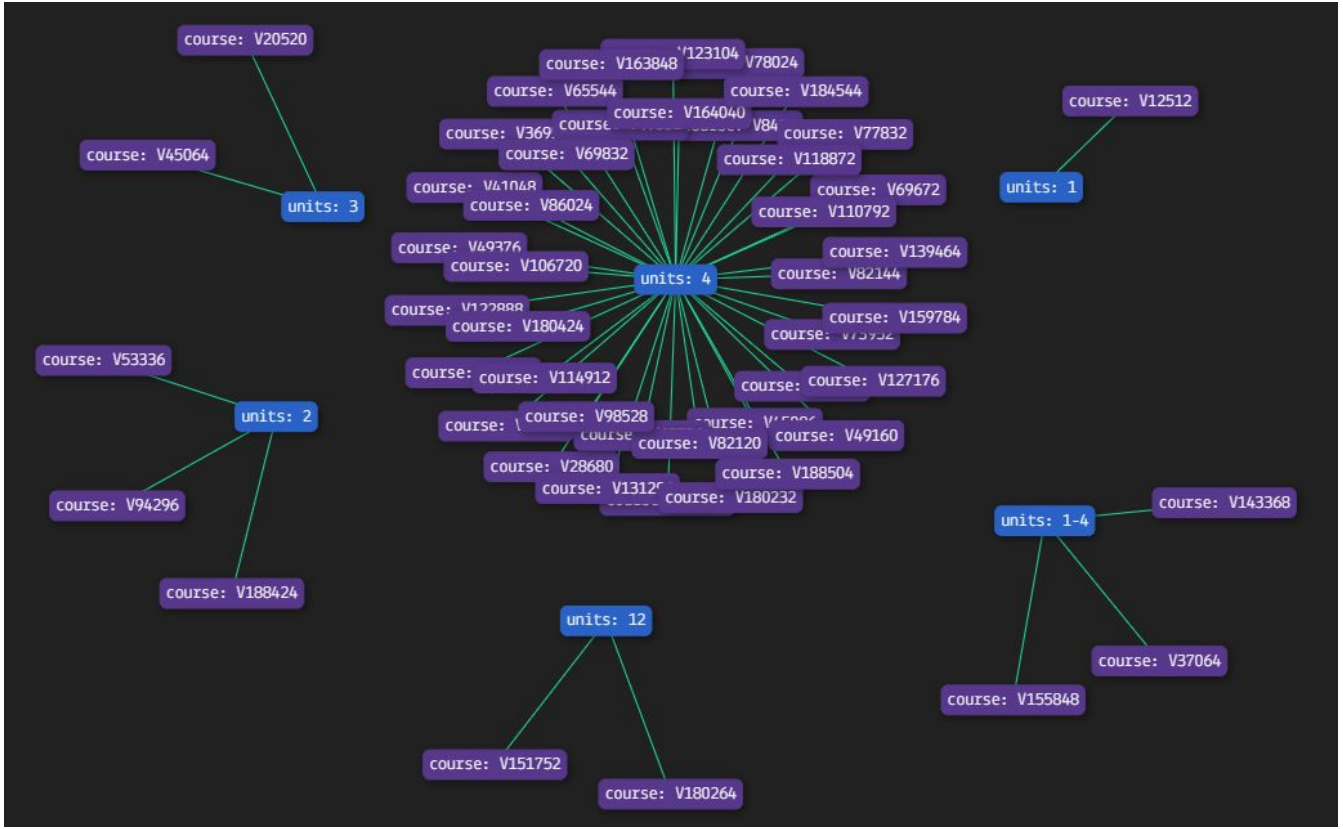
Specific Schema

```
course sub entity,
    has department,
    has course_num,
    has terms_offered,
    has units,
    has course_name,
    has concurrent,
    has corequisites,
    has recommended,
    has prerequisites,
    has ge_areas,
    has description,

    plays isCourseOfSection;
```

```
professorOffice sub relation,
    relates officeOfProfessor,
    relates professorOfOffice;


sectionOfCourse sub relation,
    relates isCourseOfSection,
    relates isSectionOfCourse;
```

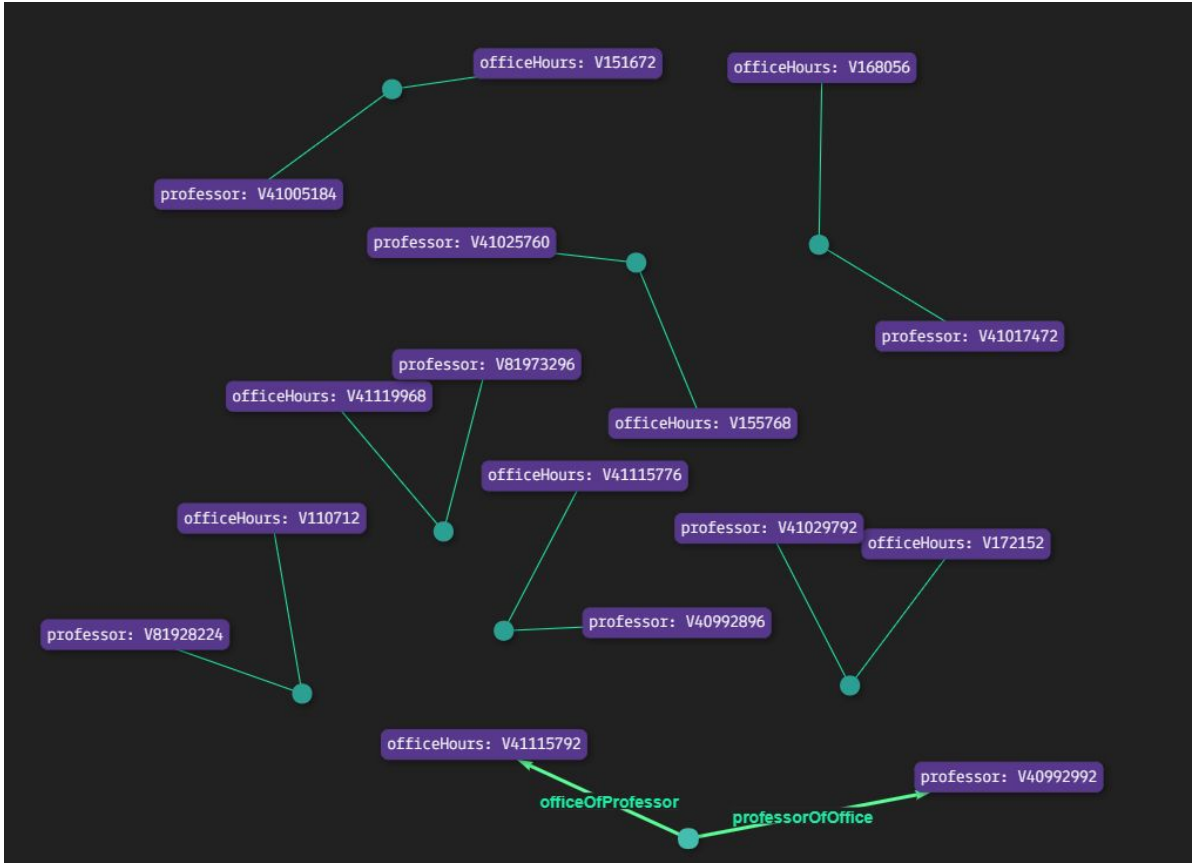
```
department sub attribute,
    value string;

course_num sub attribute,
    value string;

terms_offered sub attribute,
    value string;

units sub attribute,
    value string;

course_name sub attribute,
    value string;
```

```
match $x isa course, has units $u; get; offset 0; limit 50;
```



Knowledge Graph Visual - Attributes

```
match $x (officeOfProfessor: $o, professorOfOffice: $p) isa professorOffice; $o has platform "Zoom"; get; offset 0; limit 8;
```



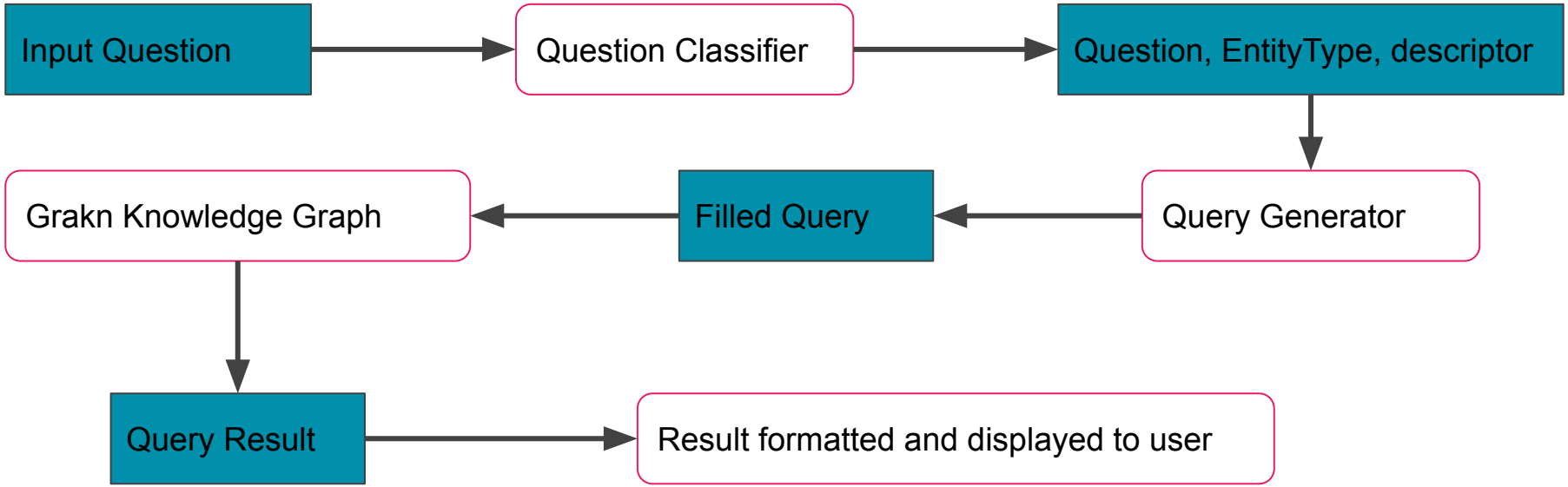
Knowledge Graph Visual - Relations

Question Descriptor Classifier

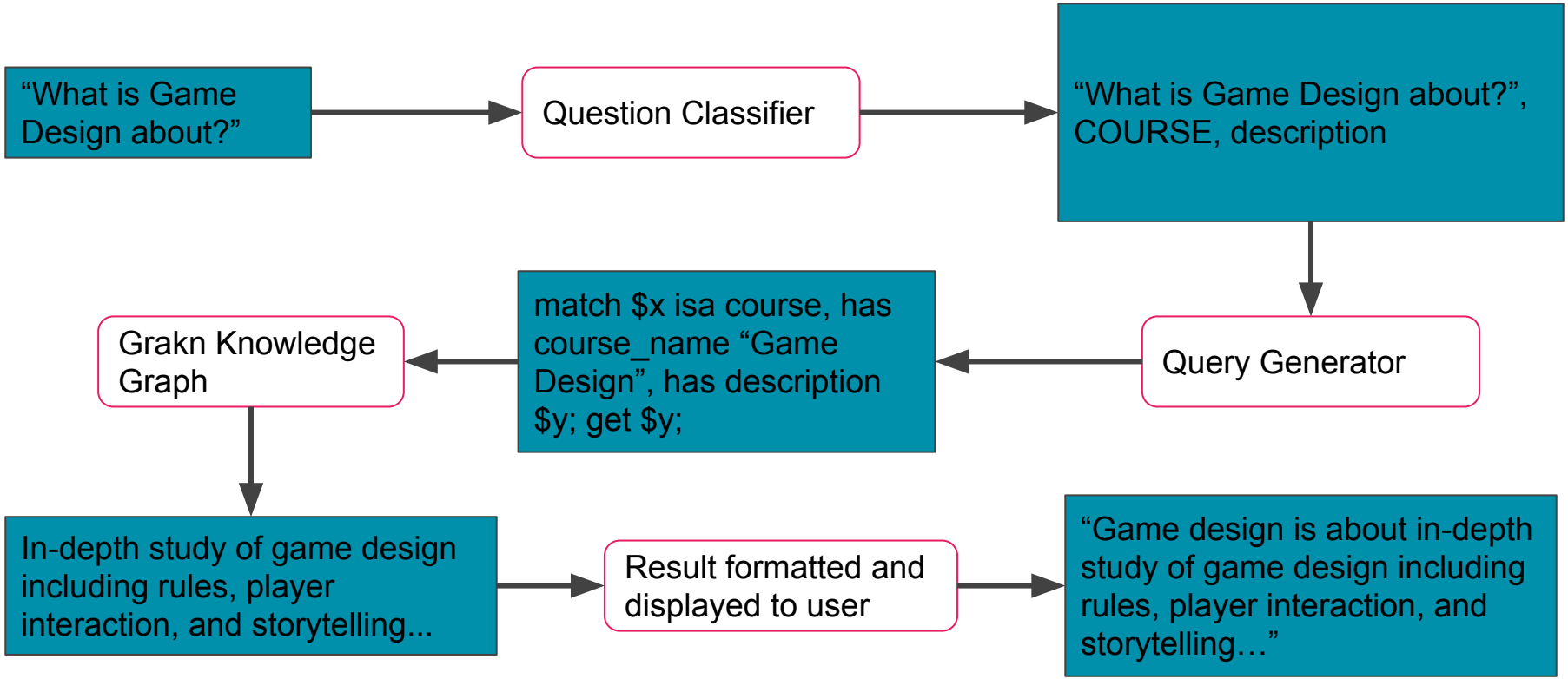
- Using a LinearSVC, a model was trained to determine the answer type of a question
- 'prerequisites', 'title', 'email', 'advisor', 'office'
- “What is [COURSE] about?” would return “description”
- Accuracy: 73%

Querying

- Questions are matched to an answer pair which has a type (course, section, etc) and a descriptor (email, prereqs, etc)
- This data is used to generate a query in a robust way using the Grakn querying system
- The query result is inserted into the answer format and returned


Input Question
Question Classifier
Question, EntityType, descriptor
Query Generator
Filled Query
Grakn Knowledge Graph
Query Result
Result formatted and displayed to user

Workflow



Workflow - Example

Issues

- Error messages are often vague and not helpful
  - There is an active forum on discord that can help with troubleshooting
- Abstracting queries can cause performance issues in certain situations
  - match $x isa COURSE, has **attribute** “Game Design”... (~16 seconds)
  - match $x isa COURSE, has **course_name** “Game Design”... (~0.1 seconds)

Questions?