Advance Data Bases – Project 2

Abhinav Bajaj (ab3900), Rafica Abdul Rahim (ra2688)

List of Files for submission -

mainFile.py - File with the code of the main framework of the system.

QuestionAnswer.py - Main file for Question Answering part

infoboxMain.py - Main file for Infobox part

Actor.py - File that contains the mapping for Actor and prints the actor information

Author.py - File that contains the mapping for Actor and prints the actor information

BussPerson.py - File that contains the mapping for Actor and prints the actor information

freebaseUtil.py - File that contains the mapping for Actor and prints the actor information

League.py - File that contains the mapping for Actor and prints the actor information

Person.py - File that contains the mapping for Actor and prints the actor information

SportsTeam.py - File that contains the mapping for Sports team and prints the Sports team information

printUtil.py - This file contains the methods that actually prints the output

README.pdf - This file

Makefile - File containing the steps to compile and run the code on CLIC machines

Transcript.txt - File containing the results of test cases

Running the program -

The project is in Python.

Run Command Options -

```
1.python mainFile.py --key  <Freebase API key> -q <query> -t
<infobox|question>
```

NOTE: In case when query has multiple words then they should be in double quotes. For example,

```
  python mainFile.py --key AIzaSyBuMq3W5wfLezCtWX9rIZXbGSXNtCCG7hY -q
                       "bill gates" -t infobox
```

```
2.python mainFile.py --key <Freebase API key> -f <file of queries> -t
<infobox|question>
```

NOTE: The question has multiple words so it should be in double quotes. For example,

```
python mainFile.py --key AIzaSyBuMq3W5wfLezCtWX9rIZXbGSXNtCCG7hY -q
                "Who created Microsoft?" -t infobox
```

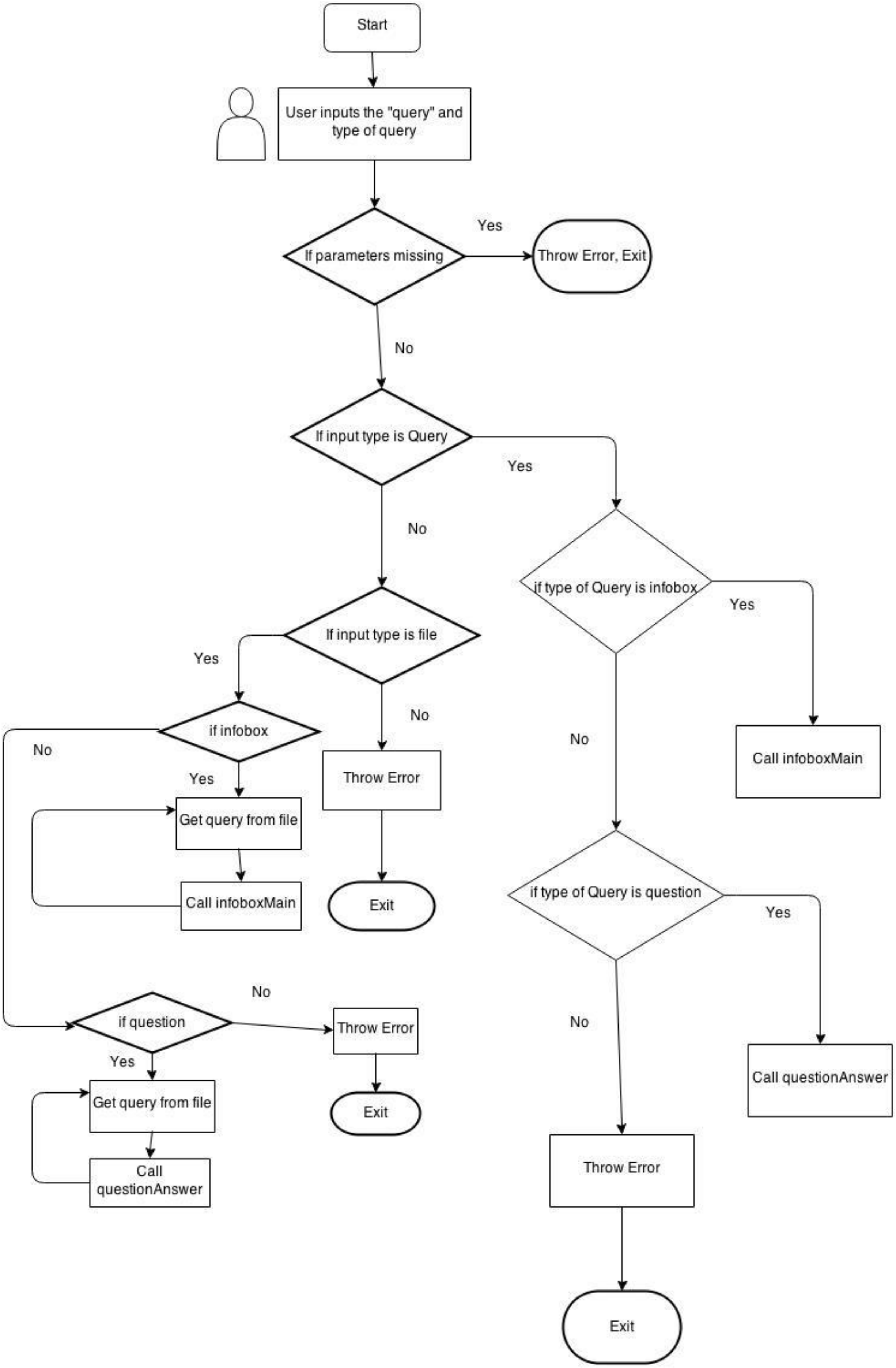
Freebase API Key

```
AIzaSyBuMq3W5wfLezCtWX9rIZXbGSXNtCCG7hY
```

Configured with 50 requests per second per user

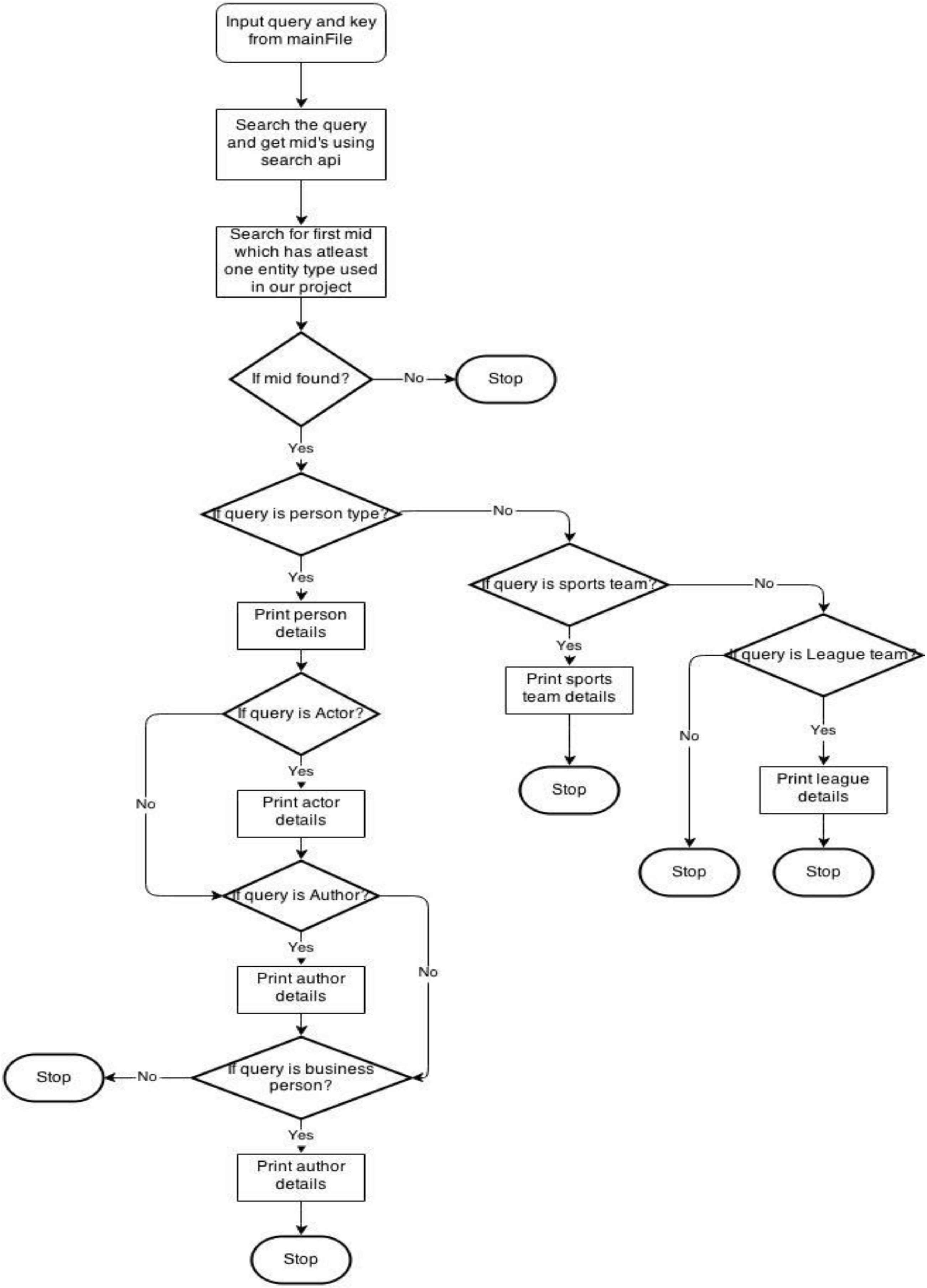
Design Description

The major work flow or design of the program is presented in the flow diagram below


Start
User inputs the "query" and type of query
If parameters missing
Yes
Throw Error, Exit
No
If input type is Query
Yes
No
if type of Query is infobox
Yes
Call infoboxMain
No
If input type is file
Yes
No
if infobox
No
Yes
Get query from file
Call infoboxMain
Throw Error
Exit
if type of Query is question
Yes
Call questionAnswer
No
Throw Error
Exit
if question
No
Throw Error
Exit
Yes
Get query from file
Call questionAnswer

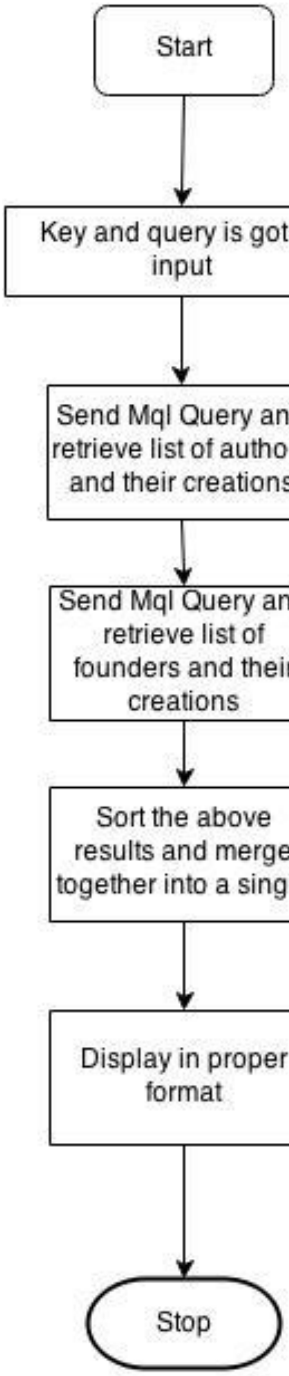
Design of Infobox (Part 1)

The major work flow or design of the Infobox function is presented in the flow diagram below -


Input query and key from mainFile
Search the query and get mid's using search api
Search for first mid which has atleast one entity type used in our project
If mid found?
No
Stop
Yes
If query is person type?
No
Yes
If query is sports team?
No
Print person details
If query is League team?
Yes
Print sports team details
If query is Actor?
No
Yes
Stop
Yes
No
Print actor details
Print league details
Stop
Stop
If query is Author?
Yes
No
Print author details
Stop
No
If query is business person?
Yes
Print author details
Stop

Design of Question Answering (Part 2)

The major work flow or design of the question answering function is presented in the flow diagram below -


Start
Key and query is got as input
Send Mql Query and retrieve list of authors and their creations
Send Mql Query and retrieve list of founders and their creations
Sort the above results and merge together into a single
Display in proper format
Stop

Entity and Property Mapping List

Part 1

The below table show the mapping list of all the entity types used in this part of the project -

Entity Type in the Project	Freebase Entity Type
Person	/people/person
Author	/book/author
Actor	/film/actor /tv/tv_actor
BusinessPerson	/organization/organization_founder /business/board_member
League	/sports/sports_league
SportsTeam	/sports/sports_team /sports/professional_sports_team

The below table shows the mapping list of all the Freebase property -

Entity Type/Property of interest	Freebase Property	Comments
Person		
Name	/type/object/name	Used only the first entry from the array of values provided by freebase json, as people have single name value.
Birthday	/people/person/date_of_birth	Used only the first entry from the array of values provided by freebase json, as people have single date of birth.
Place of Birth	/people/person/place_of_birth	Used only the first entry from the array of values provided by freebase json, as people have single place of birth.
Death.Place	/people/deceased_person/place_of_death	Used only the first entry from the array of values provided by freebase json as people have single place of death.
Death.Date	/people/deceased_person/date_of_death	Used only the first entry from the array of values provided by freebase json as people have single date of death.
Death.Cause	/people/deceased_person/cause_of_death	Used all the values provided by freebase json as people may have many reasons of death. For example "Jackson" query.
Siblings	/people/person/sibling_s:/people/sibling_relationship/sibling	Used all the values provided by freebase json for the first level property as people can have multiple siblings. Took the name from the second level property.
Spouses	/people/person/spouse_s:/people/marriage/spouse	Used all the values provided by freebase json for the first level property as people can have

		multiple spouses. Took the name from the second level property.
Spouses.from	/people/person/spouse_s:/people/marriage/from	Took the “from” the second level property. The property information is added only if it is available.
Spouses.to	/people/person/spouse_s:/people/marriage/to	Took the "to" the second level property. The property information is added only if it's available else it's assumed as "now"
Spouses.location_of_ceremony	/people/person/spouse_s:/people/marriage/location_of_ceremony	Took the location of ceremony the second level property. The property information is added only if it is available.
Description	/common/topic/description	Used only the first entry from the array of values provided by freebase json.
Author		
Books(Title)	/book/author/works_written	Used all the values provided by freebase json as author might have written multiple books.
Book About the Author(Title)	/book/book_subject/works	Used all the values provided by freebase json as there can be many books about the author.
Influenced	/influence/influence_node/influenced	Used all the values provided by freebase json as author might have influenced multiple persons
Influenced by	/influence/influence_node/influenced_by	Used all the values provided by freebase json as author might be influenced by multiple persons
Actor		
FilmsParticipated.Film Name	/film/actor/film:/film/performance/character	Used all the values provided by freebase json for the first level property as Actor can do multiple films. Took the film name from the second level property.
FilmsParticipated.Character	/film/actor/film:/film/performance/film	Used all the values provided by freebase json for the first level property as Actor can do multiple films. Took the character name from the second level property. Assumed there is only one character per film. Could not find any such example in reference implementation where an actor did multiple character in same film.
BusinessPerson		
Leadership.Organization	/business/board_member/leader_of:/organization/leadership/organization	Used all the values provided by freebase json for the first level property as business person can have leadership in multiple organization. Took the name from the second level property.
Leadership.To	/business/board_member/leader_of:/organization/leadership/to	Took the "to" the second level property. The property information is added only if it's available else it's assumed as "now"
Leadership.From	/business/board_member/leader_of:/organization/leadership/from	Took the “from” the second level property. The property information is added only if it is available.

Leadership.Role	/business/board_member/leader_of:/organization/leadership/role	Took the “from” the second level property. The property information is added only if it is available.
Leadership.Title	/business/board_member/leader_of:/organization/leadership/title	Took the “from” the second level property. The property information is added only if it is available.
BoardMember.Organization	/business/board_member/organization_board_memberships:/organization/organization_board_membership/organization	Used all the values provided by freebase json for the first level property as business person can be board member in multiple organization. Took the name from the second level property.
BoardMember.To	/business/board_member/organization_board_memberships:/organization/organization_board_membership/to	Took the "to" the second level property. The property information is added only if it's available else it's assumed as "now"
BoardMember.From	/business/board_member/organization_board_memberships:/organization/organization_board_membership/from	Took the “from” the second level property. The property information is added only if it is available.
BoardMember.Role	/business/board_member/organization_board_memberships:/organization/organization_board_membership/role	Took the “from” the second level property. The property information is added only if it is available.
BoardMember.Title	/business/board_member/organization_board_memberships:/organization/organization_board_membership/title	Took the “from” the second level property. The property information is added only if it is available.
Founded	/organization/organization_founder/organizations_founded	Used all the values provided by freebase json as business person might have founded multiple organizations.
League		
Name	/type/object/name	Used only the first entry from the array of values provided by freebase json, as League have single name value.
Championship	/sports/sports_league/championship	Used only the first entry from the array of values provided by freebase json, as League have single championship value.
Sport	/sports/sports_league/sport	Used only the first entry from the array of values provided by freebase json, as League have single sport value.
Slogan	/organization/organization/slogan	Used only the first entry from the array of values provided by freebase json, as League have single slogan value.
OfficialWebsite	/common/topic/official_website	Used only the first entry from the array of values provided by freebase json, as League have single official website value.
Description	/common/topic/description	Used only the first entry from the array of values provided by freebase json.

Teams	/sports/sports_league/teams:/sports/sports_league_participation/team	Used all the values provided by freebase json for the first level property as League can have multiple teams. Took the film name from the second level property.
SportsTeam		
Name	/type/object/name	Used only the first entry from the array of values provided by freebase json, as team have single name value.
Description	/common/topic/description	Used only the first entry from the array of values provided by freebase json.
Sport	/sports/sports_team/sport	Used only the first entry from the array of values provided by freebase json, as team will be for single sport.
Arena	/sports/sports_team/arena_stadium	Used only the first entry from the array of values provided by freebase json, as team will be for single main arena.
Championships	/sports/sports_team/championships	Used all the values provided by freebase json for the first level property as team can win multiple championships.
Coaches.Name	/sports/sports_team/coaches:/sports/sports_team_coach_tenure/coach	Used all the values provided by freebase json for the first level property as team can have multiple coaches. Took the name from the second level property.
Coaches.Position	/sports/sports_team/coaches:/sports/sports_team_coach_tenure/position	Used only the first entry from the array of values provided by freebase json, as coach will be for single position.
Coaches.From	/sports/sports_team/coaches:/sports/sports_team_coach_tenure/from	Took the “from” the second level property. The property information is added only if it is available.
Coaches.To	/sports/sports_team/coaches:/sports/sports_team_coach_tenure/to	Took the "to" the second level property. The property information is added only if it's available else it's assumed as "now"
Founded	/sports/sports_team/founded	Used only the first entry from the array of values provided by freebase json, as team have single founded value.
Leagues	/sports/sports_team/league:/sports/sports_league_participation/league	Used only the first entry from the array of values provided by freebase json, as team can belong to single league value.
Locations	/sports/sports_team/location	Used only the first entry from the array of values provided by freebase json, as team have single locations value.
PlayersRoster.Name	/sports/sports_team/roster:/sports/sports_team_roster/player	Used all the values provided by freebase json for the first level property as roster can have multiple players. Took the name from the second level property.

PlayersRoster.Position	/sports/sports_team/roster:/sports/sports_team_roster/position	Used all the values provided by freebase json for the this property as player can have multiple positions to play on.
PlayersRoster.Number	/sports/sports_team/roster:/sports/sports_team_roster/number	Used only the first entry from the array of values provided by freebase json, as player can have single number.
PlayersRoster.From	/sports/sports_team/roster:/sports/sports_team_roster/from	Took the from the second level property. The property information is added only if it is available.
PlayersRoster.To	/sports/sports_team/roster:/sports/sports_team_roster/to	Took the "to" the second level property. The property information is added only if it's available else it's assumed as "now"

Part 2

Creator	Creations
Author: /book/author	**Books written:** /book/author/works_written
Founder:/organization/organization_founder	**Organizations founded:** /organization/organization_founder/organizations_founded

The MQL Query used to get the authors is

```
[{
     "/book/author/works_written": [{ "a:name": null, "name~=": "Lord of the Rings" }],
     "id": null,
      "name": null,
      "type": "/book/author"
 }]
```

The MQL Query used to get the founders is

```
[{
     "/organization/organization_founder/organizations_founded": [{ "a:name": null, "name~=": "microsoft" }],
     "id": null,
      "name": null,
      "type": "/organization/organization_founder"
 }]
```