1. Time interval query
2. Time interval query

What is the temperature between 7am and 8am on October 23, 2016?（Object can be temperature", "humidity", "stock price of Google", etc.）

**Template1: [What is the <OBJECT> between <START\_TIME> and <END\_TIME> on <DATE>?]**

**Template2: [What is the <OBJECT> between <START\_TIME> and <END\_TIME>]**

Tell me ,where exactly were the trains between [8am, 10am]？（Find the location, excluding location information, just time information）（Represent queried objects, such as "trains", "buses", "ships", etc.）

**Template3: [Where exactly were the <OBJECTS> between <START\_TIME, END\_TIME>?]**

**Template4: [Where exactly were the <OBJECTS> from <START\_TIME> to <END\_TIME> ?]**

**Template5: [Which places did the <OBJECT> <OBJECT\_ID/OBJECT\_IDENTIFICATION> go between <START\_TIME> and <END\_TIME>?]**

1. Time point query

What is the temperature at 7am on October 23, 2016？

**Template6: [What is the <OBJECT> at <SPECIFIC\_TIME> on <DATE>?]**

**Template7: [What is the <OBJECT> at <SPECIFIC\_TIME>]**

**Template8: [Where exactly were the <OBJECTS> at <SPECIFIC\_TIME> ?]**

**Template9: [Which places did the <OBJECT> <OBJECT\_ID/OBJECT\_IDENTIFICATION> go at <SPECIFIC\_TIME>?]**

1. Range query
2. Spatial range query

Please tell me which kinos are located in thecenter？

**Template1: [Can you provide a list of <type\_of\_place> that are found in the <specific\_area> of the city?]**

**Template2: [I'm interested in visiting <type\_of\_place> around the <specific\_area>, could you show me the options available?]**

**Template3: [Could you tell me the <type\_of\_place> located at <specific\_area>?]**

（<type\_of\_place> can be "cinemas", "theaters", "restaurants", etc., and <specific\_area> can be "downtown", "suburban area", "xxx (e.g. Thielallee) street", etc.）

1. Temporal range query

What is the pressure, relative humidity, and temperature from October 1, 2016 to October 7, 2016?

**Template4: [Can you provide the <metrics> for <data\_type> between <START\_TIME> and <END\_TIME>?]**

**Template5: [What were the <metrics> of <data\_type> from <START\_TIME> to <END\_TIME>?]**

<metrics> is used for specific data metrics, such as "average values", "highs and lows", "trends", "fluctuations", etc.

<data\_type> represents the type of data, such as "weather conditions", "stock prices", "economic indicators", etc.

<START\_TIME> and <END\_TIME> represent the query time for start time and end time, respectively.

1. Moving objects rang query

Which trains pass through the park “Tiergarten”between 8:00 and 10:00？

Which trains pass the station Mehringdamm at 8:00 o’clock?

**Template6: ["Which <object> pass through <location\_type> “<specific\_location>” between <start\_time> to <end\_time>?"]**

**Template7: ["Which <object> pass through <specific\_location> from <start\_time> to <end\_time>]**

**Template8: ["Which <object> pass through <location\_type> “<specific\_location>” at <specific\_time>?"]**

<object> represents moving objects, such as "trains", "buses", "vehicles", etc.

<location\_type> represents the location type, such as "park", "station", "intersection", etc.

<specific\_location> represents the specific location name, such as "Tiergarten", "Mehringdamm", etc.

1. Nearest neighbor query
2. Spatial nearest neighbor query

Please tell me what are the 10 closest POIs to the Nanjing border line?

**Template1: ["Can you list the <number> nearest <objects> to the <specific\_location>?"]**

**Template2: ["Can you list the <number> closest <objects> to the <specific\_location>?"]**

**Template3: ["Can you list the nearest <objects> to the <specific\_location>?"]**

1. Moving objects nearest neighbor query

Which train is the nearest neighbor of the train 123 from 9am to 10am?

**Template4: [Which <object> is the closest to <object> <object\_id/object\_identification> between <start\_time> and <end\_time>?]**

**Template5: [Find the <number> continuous nearest <objects> to the <object> <object\_id/object\_identification> from <start\_time> to <end\_time>.]**

1. Join query
2. Spatial join query

The connection condition is the positional relation: What are the POIs in each district of Nanjing?

**Template1: [What are the <POI\_objects> in each <geographical\_unit> of <specific\_area/city>?]**

**Template2: [Please list the <POI\_objects> that intersect any <geographical\_unit> in <specific\_area/city>.]**

The connection condition is the distance relation: What are the POIs within 1.5 kilometers from the center of each district in Nanjing?

**Template3: [List all <POI\_objects> that are within <distance> <unit\_of\_distance> from the center of each district in <specific\_area/city>.]**

<distance> represents a distance threshold from a specific location.

<unit\_of\_distance> represents the unit of distance, such as "kilometers", "meters", "miles", etc.

<geographical\_unit> represents the unit of geography, such as "district", "neighborhood", "city", and so on.

1. Moving objects join query

Find the same trajectory between train O3 and train O4 during the time period [t10, t11].

**Template4: [Find the same trajectory between <object1> <object\_id1/object\_identification1> and <object2> <object\_id2/object\_identification2> during the time period [<start\_time>, <end\_time>].]**

**Template5: [Retrieve the matching trajectories for <object1> <object\_id1/object\_identification1> and <object2> <object\_id2/object\_identification2> from <start\_time> to <end\_time>.]**

1. Similarity query

What is the similarity between the temperature in January 2015 and January 2016?

**Template1: [How similar is the <object> data between <start\_time> and <end\_time>]**

**Template2: [What is the similarity <object> data from <start\_time> to <end\_time>]**

Which train has the similar trajectory of the train 17 between 17:00 and 19:00 o’clock?

**Template3: [Find the similar trajectory to the <object> <object\_id/object\_identification> during the time period [<start\_time>, <end\_time>].]**

**Template4: [Find the similar trajectory to the <object\_with\_adjectives> during the time period [<start\_time>, <end\_time>].]**

**Template5: [Could any one tell me which <object> is the most similar to the <object\_with\_adjectives> <object\_id/object\_identification> between <start\_time> and <end\_time>?]**

1. Basic spatial query

Query the distance: Returns the distance between mehringdamm and alexanderplatz

query distance(mehringdamm, alexanderplatz)

**Template1: [Calculate the direct distance from <location1> to <location2>.]**

Query the direction: Return to the direction from mehringdamm to alexanderplatz

query direction(mehringdamm, alexanderplatz)

**Template1: [Calculate the direction from <location1> to <location2>.]**

Query the area: Returns the area of thecenter

query area(thecenter)

**Template1: [Calculate the area of <specific\_area>.]**

Query the length: Returns the length of PotsdamLine

query size(PotsdamLine)

**Template2: [Calculate the length of <specific\_area>.]**

If the location is stored in one spatial relation: such as query the area of Treptower Park. Treptower Parkis stored in the relation Flaechen.

query area(Flaechen feed filter [.Name = "Treptower Park"] extract[GeoData])

1. Aggregation query
2. Temporal aggregation query

What was the average temperature in January 2016?

What was the highest temperature in January 2016?

What was the lowest temperature in January 2016?

**Template1: [Determine the <aggregation\_type> <metric> for <specific\_time>.]**

<aggregation\_type> represents the type of aggregation, e.g. average, highest, lowest etc.

<metric> represents the dimension of the metric, e.g. temperature, rainfall, humidity, etc.

1. Spatial aggregation query

How many POIs are there in each district of Nanjing?

**Template2: [Provide the count of <POI\_objects> within each <geographical\_unit> of <specific\_area/city>].**

1. Normal basic query

**Template1:Show me some trajectories of <objects>.**

**Template2: Show me some <objects>.**

**Template3: Show me the trajectory of the <object> <object\_id>.**

**Template4: Show me the path of the <object> <object\_id>.**

**Template5: List <objects> which <attributes> are <greater/less> than <specific\_content\_of\_attribute>.**

**Template6: List <objects> which <attributes> between <specific\_content\_of\_attribute> and <specific\_content\_of\_attribute>.**

**Template7: List <objects> which <attributes> are <greater/less> than <specific\_content\_of\_attribute> and sort them in <increasing/descending> order.**

**Template8: List <objects> which <attributes> between <specific\_content\_of\_attribute> and <specific\_content\_of\_attribute> and sort them in <increasing/descending> order.**