

## 問い合わせ言語 PartiQL による統一された異種データベースアクセス： グラフデータベースへの問い合わせ変換

### Unified query access for heterogeneous databases using PartiQL: Query translation to graph databases

鈴木 源吾<sup>†</sup>  
Gengo Suzuki

#### 1. はじめに

交通網を含むグラフデータベースや、観光地や店舗などの POI (Point of interest) 情報のデータベースなどが異種で分散している環境で、旅行計画問合せのような応用的問合せを効率よく実行できる、異種分散時空間データベース技術を検討している。そのためには、それらのデータベースに共通にアクセスでき、開発者にとって利用しやすい共通問合せ言語があることが望ましい。その候補として、ネストや半構造に対応し記述能力の高い PartiQL は有望であると考えているが、情報源としてグラフデータベースに対してアクセスするためのデータモデルマッピングと問合せ変換方式を明らかにした。

#### 2. 本研究の背景

##### 2.1 異種分散 DB 環境での旅行計画問い合わせ

我々は、観光分野などへの応用を想定した、旅行者が（一般には複数の）興味のある場所 (POI) を経由する最適な経路を求める旅行計画問合せ技術[3][4] (Trip planning query) を異種分散データベース環境で実現する研究に着手している (図 1)。問合せに活用する情報が分散することが多いためである。対象とする情報源には地理情報を持つグラフデータベースがあり、グラフデータベースと他のデータベースを効率よく連携させる技術が必要である。

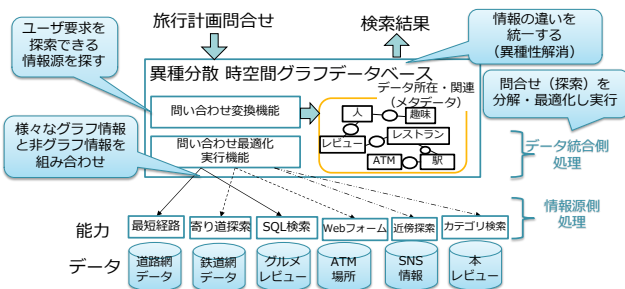


図1 研究の全体像

##### 2.2 共通問合せ言語

異種分散データベースを一元的に扱い、データベースを跨る問合せを適切に最適化するためには、共通の問合せ言語があることが望ましい[5][6][7]。共通問合せ言語には、データモデルが様々なモデルを包含できるよう十分豊かであること、利用者に親しみやすく使いやすいこと、等が要件としてあげられる。実用的な観点からは、実装が公開され利用しやすいことも望ましい。我々は本研究における共通問合せ言語として、PartiQL を有望と考えている。半構造・ネストに対応し、SQL 互換性が高く、実装も公開されている。ただし、我々が扱いたいグラフデータモデルを完全に扱うことはできないが、そこまでモデルを豊かにする

と開発者の使い勝手が悪くなる。グラフに本当に固有な処理 (任意回トラバース・経路探索等) は、ユーザ定義関数・グラフ固有処理のスルー等の手段を用いるほうがよいと考える。以下で、PartiQL を用いていかにグラフデータベースを利用するかを述べる。グラフデータベースとしては現在最も一般に普及している Neo4j を想定する。実現したい旅行計画問合せの簡易なイメージは以下である。

```
SELECT shortest_path_poi(start, end, "ラーメン屋")
FROM Loc AS start, Loc AS end
WHERE start.name = "大学" and end.name = "新潟駅"
```

#### 3. 問合せ言語 Cypher・PartiQL とその対応

##### 3.1 Cypher

Cypher は、グラフデータベース Neo4j のために開発された問合せ言語であり、グラフの作成・検索が行える。Neo4j 公式ページにあるチュートリアル[2]の例題で説明する (図 2)。

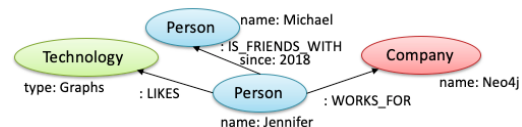


図2 サンプルのグラフデータ

グラフデータは、ノードと関連 (Neo4j におけるエッジを表す概念) からなる。それぞれ属性を付与できる。図中の丸はノードであるが、その中の Person はラベルと呼ばれ、ノード・関連双方に定義することができ、複数の構成要素をまとめる機能である。同じラベルであっても異なる属性を持つことは可能で、関係モデルのテーブルよりは緩い機能である。関連にある :IS\_FRIENDS\_WITH 等は関連タイプと呼ばれる概念で、関連を区別しトラバースの制御等に用いる。Cypher の問合せ例をあげる。

```
MATCH (p:Person) WHERE p.name = 'Jennifer'
RETURN p
```

MATCH 節は、グラフのパターンマッチを行なう節である。この例では、Person というラベルを持ったノードをマッチングし p という変数にバインドしている。ノードと関連を含むパス指定も可能であり、グラフデータベース特有の機能である。WHERE 句でパターンを絞り込む。RETURN 節は、返却する変数を指定する。ここでは、p のノード全体の情報を指定しているが、p.name のような属性値のみ指定、ノードと関連の組合せの指定等も可能である。

##### 3.2 PartiQL

PartiQL は、Amazon が開発した SQL 互換で半構造やネストされたデータへ容易にアクセスできる問合せ言語である。現在はデータ変更を追跡できる台帳データベースのサービスである Amazon Quantum Ledger Database (QLDB) で使われ

ており、仕様やコマンドラインインターフェースの実装がオープンに公開されている[1].

PartiQL は JSON に似たデータ構造 PartiQL データモデルを採用している (JSON との主な違いは bag のサポート). 例えば、図 2 の 2 つの Person データを PartiQL データモデル (<< >> は bag) で表し、それへの問合せ例は以下になる. オブジェクトをまとめて返却できる.

```
'GraphDB': {
  'Person': <<
    { 'id': 0, 'name': 'Jennifer'},
    { 'id': 1, 'name': 'Michael' }
  >>
}
```

```
SELECT p
FROM GraphDB.Person AS p
WHERE jenn.name = 'Jennifer'
```

### 3.3 PartiQL と Cypher マッピング

Cypher は SQL を意識して設計されており、PartiQL もオブジェクトを返却できるので、基本的にはノードと関連をオブジェクトの名前にマッピングすれば良い (表 1). PartiQL データモデルにテーブル・属性の明確な区別はないが、わかりやすさのためにカッコ書きしている. MATCH 節では、属性とその値を指定するパターンを記述できる. それは WHERE 句にマッピングする. ラベルをもたないノードは PartiQL に対応づけできない (制約事項).

表 1 Cypher と PartiQL の対応

Cypher	PartiQL
Node label	(table) name
Node without label	-
Relationship Type	(attribute)name
attribute	(attribute)name
MATCH	FROM + WHERE
WHERE	WHERE
RETURN	SELECT

## 4. PartiQL から Cypher への問合せ変換

### 4.1 PartiQL を用いたグラフトラバース

共通問合せ言語として PartiQL を用いるためには、利用者からの PartiQL を Cypher に変換し、Neo4j で検索した結果をさらに PartiQL データモデルに変換し利用者に返却する必要がある. 上記マッピングをそのまま適用すれば、シンプルにノード・関連のみを条件して検索することは容易である. さらに、我々は最低限実現すべき仕様としては、関連を陽に指定したトラバースと考えている. 以下の Cypher 問合せへの変換を考える.

```
MATCH (p:Person)-[:WORKS_FOR]->(company:Company)
WHERE p.name = 'Jennifer' RETURN company.name
```

Person から Company のグラフトラバースを PartiQL の仮想的なデータで考えれば、次のように Company への WORKS\_FOR の関連をネスト構造にマッピングすればよい.

```
{
  'GraphDB': {
    'Person': <<
      { 'id': 0, 'name': 'Jennifer', 'WORKS_FOR': [{ 'id': 3, 'name': 'Neo4j' }]},
      { 'id': 1, 'name': 'Michael' }
    >>
  }
}
```

このデータにアクセスするための上記 Cypher 問合せと等価な PartiQL 問合せは、以下となる.

```
SELECT company.name
FROM GraphDB.Person AS p, p.WORKS_FOR AS company
WHERE p.name = 'Jennifer'
```

“p.WORKS\_FOR” という変数を含むパス指定は PartiQL 独自の SQL 拡張であり、ネスト構造検索に利用できる. この問合せから上記 Cypher 問合せを生成するために、FROM 句の中のパス指定を解析し、ノードラベル間の関連が存在し整合がとれているかをチェックし、MATCH 句を生成する (関連が存在しない場合は直積と解釈する). この問合せには (ノードラベル) Company が陽に指定されないが、問合せ変換のためのメタデータ (関連とその両端のノードラベルを持つ) を管理しておき、補完する.

また、PartiQL では 3 段以上のパス指定も可能なので、同様のパス解析とノードラベルの補完を行って、3 つ以上のノードタイプをまたがる Match 節にマッピングすることができる.

### 4.2 問合せ変換の実装

上記の検討を元に PartiQL 問合せをパース・正規化 (Cypher で MATCH 句に変数指定が必須であるため AS 句追加等) ・パス解析し Cypher に変換し、Neo4j を検索し問合せ結果を返却するグラフデータベースアクセス機能を実装した (<https://github.com/ggszk/partiql-neo4j> で公開予定).

## 5. おわりに

PartiQL と Cypher のマッピングを定め、問合せ変換により Neo4j を PartiQL によって利用可能にした. 今後は、さらに旅行計画問合せの複数データベースへの問合せ分解や最適化を検討する.

### 参考文献

- [1] Amazon.com, Inc., “PartiQL,” <https://partiql.org> (accessed Jun. 02, 2020).
- [2] Neo4j, Inc., “Cypher basics.” <https://neo4j.com/developer/cypher-basics-i/> (accessed Jun. 02, 2020).
- [3] F. Li, D. Cheng, M. Hadjieleftheriou, G. Kollios, and S.-H. Teng, “On Trip Planning Queries in Spatial Databases,” in 9th International Symposium on Spatial and Temporal Databases (SSTD '05) (2005).
- [4] 鈴木 源吾, 榎本 俊文, 小林 伸幸, 山室 雅司, 鬼塚 真, “時間制約を持つ寄り道経路探索システムの実現と評価,” 情報処理学会論文誌, vol. 53, no. 2 (2012).
- [5] B. Kolev, P. Valduriez, C. Bondiombouy, R. Jiménez-Peris, R. Pau, and J. Pereira, “CloudMdsQL: querying heterogeneous cloud data stores with a common language,” *Distrib. Parallel Databases*, vol. 34, no. 4. (2016).
- [6] R. Hai, C. Quix, and C. Zhou, “Query Rewriting for Heterogeneous Data Lakes.” *ADBIS* (2018).
- [7] M. Stonebraker, “The Case for Polystores – ACM SIGMOD Blog.” <https://wp.sigmod.org/?p=1629> (accessed May 11, 2020).

† 開志専門職大学 Kaishi Professional University