DEIM Forum 2021 D33-3

公開型テーブルにおける集約可能データ推薦方法の提案

佃　陽平†　遠山　元道††

† 慶應義塾大学大学院理工学研究科　〒 223–8522 神奈川県横浜市港北区日吉 3-14-1
†† 慶應義塾大学理工学部情報工学科　〒 223–8522 神奈川県横浜市港北区日吉 3-14-1
E-mail: †tsukuda@db.ics.keio.ac.jp, ††toyama@ics.keio.ac.jp

あらまし　現在、政府や地方自治体は気象情報、郵便番号、その他統計データ等の様々なオープンデータを CSV や PDF などのダウンロード形式や Web サービスの API によって提供している。先行研究として、利用者側が自身の持つ RDB と合わせて, それらのデータを利用する際の、利便性の向上のために、オープンデータを RDB のまま利用できるアーキテクチャ (RTA:Remote Table Access) を開発した。これによりデータ公開者は自身が持つ RDB 形式のオープンデータを加工せずに、公開型テーブル (PTL:Public Table Library) に公開することができる。この PTL 上のオープンデータを用いてデータ分析を行う際、PTL 上には多くの公開テーブルがあるため、分析作業において有用であるテーブルを見逃してしまう可能性は大きい。特に、週単位で集計しているデータは集約すれば、月単位で集計しているデータと組み合わせて分析が行えるが、それに気づかないといった、集計単位が異なる事で見逃してしまうことが考えられる。本研究では大量のテーブルの中から、集約することで利用可能なテーブルをユーザーに推薦するアーキテクチャを提案し、RTA 及びオープンデータを活用したデータ分析の効率化を試みる。

キーワード　オープンデータ、データレイク、情報推薦

1 はじめに

現在オープンデータは各機関から提供されており、それらの提供方法として CSV、XML などのダウンロード形式、Web サービスの API による提供、LOD に基づく RDF による提供などがある。しかし、利用者側が自身の持つ RDB と合わせてそれらのデータを二次利用する際には、利便性が低いのが現状である。先行研究では，そのようなオープンデータに対して加工せずに、RDB のまま利用できるアーキテクチャ (RTA:Remote Table Access) を開発した。[1] 通常であれば 1 週間に 1 回、1 ヶ月に 1 回等しか更新されないオープンデータなどであっても、RDB のまま利用可能になることによって常に最新のデータにアクセスすることが可能になる。またこのアーキテクチャを利用する事によって、自身の持つ RDB 内のテーブル、複数の公開型テーブルなどと合わせて 1 つの SQL 内で処理を行う事も可能になる。

オープンデータを RDB のまま利用できるようにするために、RTA には公開型テーブル (PTL:Public Table Library) の機能がある。RTA を用いると、ユーザーはこの PTL 上のデータをあたかも自分のデータベースに置いているかのように扱うことができる。また、ユーザーは PTL 上のテーブルを、ユーザー自身のローカルテーブルと組み合わせて作業をすることができる。またデータ提供者側も、変更がある場合はテーブルを更新すれば良いだけなので、別の形式に変換するといった作業が不要になる。

PTL は性質上、時間が経つほど巨大なものになってしまう。そして PTL にテーブルが多くなれば多くなるほど、自分が利用したいテーブル及び、利用可能なテーブルを探すのが困難になる。現在の登録テーブル数は約 100 テーブルのため、多少時間を使えば人力で全てに目を通せるほどの数であるが、これが例えば 500 テーブルになれば 1 テーブル 1 分で目を通しても 6 時間以上かかってしまう。そのため、巨大な PTL の中からユーザーにとって有用なテーブルを推薦する機能が求められている。

関連する研究として、データレイクの中からローカルのテーブルと結合可能なテーブルを探し出す研究がある [8] [9] [15]。これらの研究は主にカラムや文字列が一致するテーブルを発見することを主眼に置いている。しかし、日単位で集計しているデータは集約すれば、月単位で集計しているデータと組み合わせることができるといった、集約によって結合可能になる集約可能データの発見の研究は進んでいない。例えば港区のコロナウイルス感染者数は以下の表 1 のような形式で公開されている。データは東京都のオープンデータカタログサイト [16] から引用した。集計開始日と集計終了日の期間内での感染者数と、累計数をまとめており、週単位で更新されている。また、港区の人口は以下の表 2 のような形式で公開されており、月単位で更新されている。これらのデータに対し、週単位で集計している港区のコロナウイルス感染者数を月単位で集約することで、港区の人口に対するコロナウイルス感染者数の割合を計算することができる。他の例としては、厚生労働省は毎日単位でコロナウイルス陽性者数のオープンデータを更新している。これを毎月単位に集約することができれば、毎月単位で更新される業種別の労働時間と比較、結合することができる。

オープンデータは基本的に一定の時間単位 (毎日、毎月の様な) で更新されることが多い。そのため、集約単位による結合可能性を検討できることは、組み合わせて使えるテーブル数が

表 1　港区のコロナ陽性者数に関する統計

集計開始日	集計終了日	集計期間の感染者数	累計
2020-11-23	2020-11-29	194	2738
2020-11-30	2020-12-06	183	2921
2020-12-07	2020-12-13	191	3112

表 2　港区の人口に関する統計

年月日	…	港区総数合計人口計	…
2020-11-01	…	259493	…
2020-12-01	…	259540	…
2021-01-01	…	259036	…

増えることに直結する。そこで、本研究では PTL に登録されているテーブルの中から、集約可能データを推薦するための方法を提案し、PTL の利便性の向上を図る。

以下、本文の構成を示す。2 章で関連研究について述べ、3 章で先行研究である RTA の概要について述べる。4 章で PTL における集約可能データの推薦方法について、5 章で評価指標を述べる予定である。

2　関 連 技 術

2.1　オープンデータ

オープンデータとは、政府、民間企業、個人などがそれぞれの保持するデータを、原則として二次利用を妨げないライセンスを基本として全ての人が利用できるように公開されたデータのことである。具体的には気象情報，郵便番号，株価など様々な分野で多岐に渡って公開されている。現在は、国や地方自治体ごとのオープンデータカタログサイトによる提供が主流となっていて、その数も膨大である。例えば米国の州や年が保有するデータを管理、公開するカタログサイトである「data.gov」[10] には 2020 年 12 月 24 日現在、217516 件ものデータセットが存在する。日本版である「data.go.jp」[11] には 28081 件のデータセットが公開されている。

このオープンデータを利用するための一般的な方法として、

- ダウンロード形式 (CSV、XML など) を利用して自身のテーブルに挿入する方法
- API などの Web サービスを利用する方法
- 既存のリモートアクセス技術を利用する方法

などが挙げられる。

しかし、ダウンロード形式を利用する方法は手間と時間がかかる上に、定期的に利用する場合はデータの更新をその都度行う必要があり、ダウンロードをやり直す必要が生じる。API を利用する方法も追加でプログラムを書く必要があるため、手間と時間がかかることはダウンロード形式と変わらない。それ以上に、API は利用する方法が提供者によって制限されるため、提供者の望む形でしかデータを取得できない。また、既存のリモートアクセス技術を使用する場合、ユーザーが既に持っているテーブルと結合できないものがあり、異種の DBMS 間で利用できないといった問題が発生する。

このような問題点を解決するために、先行研究では RTA システムを開発した。RTA ではデータ提供者側が Web アプリケーションを通じて公開データを RDB のまま登録。利用者側はそれをまるで自身の DBMS 内に存在するかのように SQL を記述することによって、上記の問題点を解決して利用者側、提供者側双方にとって円滑なオープンデータ利用を可能にしている。図 1 は RTA での利用方式と、ダウンロードでの利用形式の違いを表している。ダウンロード形式ではデータが更新されるたび、ダウンロードをし直さなくてはいけないが、RTA では、公開者側がテーブルを更新するだけで、利用者側もすぐに更新したデータを扱える。

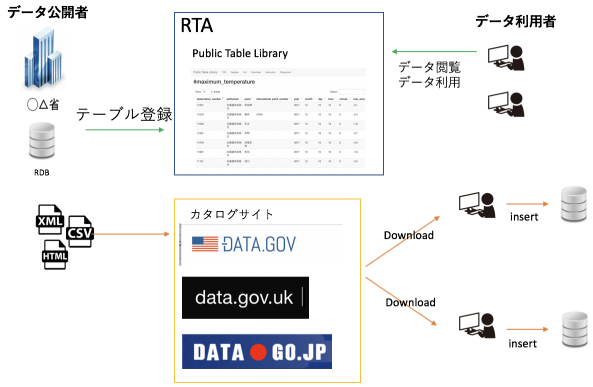

図 1　RTA とダウンロード形式の違い

2.2　リモートアクセス技術の関連研究

リモートのデータソースに対するアクセス手法は様々な企業などによって研究、開発が行なわれている。RDA とは、リモートデータベースへのデータベース操作の送信、操作結果のクライアントへの送信等について定めた ISO の国際標準規格である。現在、Microsoft などによって RDA 規格に基づき実装が行なわれている。[12]（SQL Server に実装が行なわれているが、これは将来のアップデートでは廃止予定となっている）MySQL では、リモートの MySQL データベースへのアクセスを可能にする FEDERATED ストレージエンジンを提供している。[13] これは，予めリモートテーブルと同じテーブル定義のテーブルを作成し，接続情報を登録しリンクしておくことで、ローカルの MySQL にクエリを発行するだけでリモートテーブルにアクセスすることが出来るようになるというものである。また PosgreSQL の postgres fdw [14] も同様の技術である。

Dennis Heimbigner らは、多様化する情報アクセスの手段に対する解決策として、Federated Database の概念ついての初期段階の論文を発表した [2]。この論文中で Federated Database とは、相互接続して互いのデータを利用できるような自立したデータベースの集合であると定義されている。また、Amit P.Sheth らは、Federated Database を、自立性があり不均一な協調システムの集合であると定義している [3]。この論文では、Federated Database System において重要な要素は自立性、不均一性、分散であると述べている。Laszlo Dobos らは、自身の持つデータベースとリモートデータベースを 1 箇所のリモートの SQL Server 上で管理することで、それらの相互利用を容易

にしようとする Graywulf Project という研究を行っている [?]。Graywulf では、リモートサーバーからコピーするデータ量を最小化するために、実行されたクエリを解析し、必要なカラムのみをコピーするようにしている。Youzhong らは、IoT 時代に大量に増え続けるデータを効率的に処理するために、更新とクエリの効率的なインデックスフレームワークである update and query efficient index framework(UQE-index) [5] を提案した。これは効率的な多次元クエリを同時に提供できる Key-Value 型のデータストアである。Jeff Shute(Google) らは、Bigtable のような NoSQL システムのスケーラビリティと、従来の関係データベースの一貫性と使いやすさを兼ね備えたハイブリッド・データベースである F1 [6] を開発した。F1 は同 Google 社の Spanner [7] 上に構築されており、クロスデータセンターレプリケーション (XDCR) と強力な一貫性を提供する。

2.3 データレイクの関連研究

データレイクから有用なデータやテーブルを抜き出す研究が行われている。この目的として、オープンデータのカタログサイトやデータレイクなどの大量に情報を持つ空間から、所望のデータを素早く抜き出したいというものがある。Fatemeh らはテーブルとテーブルが結合できるかどうかの評価指標を設計し、それを踏まえ、ユーザーが持っているテーブルと結合可能な部分を持つテーブルを、データレイクから検索する新しいアルゴリズムを提案した [8]。また、Erkang らはカラム一つ一つに順序づけをすることにより、データレイクの中からユーザーが使うテーブルと同じカラムを持つテーブルを検索する時間を削減することに成功した [9]。この JOSIE と呼ばれる手法はデータサイズやデータ数に関わらず、安定して従来手法よりも短時間で検索することができる。Ayman らはデータレイク中のメタデータに注目し、それらメタデータの管理プロセスを定義し、データレイク中の情報プロファイリングのためのフレームワーク作成を行った [15]。

3 RTA:Remote Table Access

この章では、RTA の概要、およびシステムがどのように構成され機能するかについて説明する。

3.1 RTA のアーキテクチャと機能

Remote Table Access(RTA) は、オープンリレーショナルデータの公開と利用を促進することを目的としたシステムである。リレーショナルデータベース形式で保存されているデータを持つデータ公開者について考えてみる。データ公開者は、RTA の Public Table Library(PTL) に公開したいテーブル (またはその列) を登録できる。PTL は、RTA を介して公開されたすべてのデータのインデックスであり、データ利用者はそれらを参照して、閲覧、利用したいデータセットのテーブルを見つけることができる。データ公開者は、Web アプリケーションを使用して、公開したいテーブルを PTL に簡単に登録できる。公開者がこの情報を提供すると、PTL は公開されたテーブルで使用可能な列名とタイプを自動的に取得し、発行者が公開する列を決定できるようにする。

ここで、データ利用者であるユーザーが、自身が好むリレーショナルデータベース管理システム (RDBMS) を使用して格納およびアクセスするデータで使用するオープンリレーショナルデータセットを探しているとする。ページ上の PTL、および各テーブルとその列の詳細を調べることで、RTA を通じて公開されたデータセットを参照できる。

自身が欲するテーブルが見つかったら、その *alias* を取得して RTA クエリで使用できる。RTA クエリは単純な SQL クエリで、#記号が手前についた RTA alias が RTA テーブルを参照するために使用され、RTA クライアントに送信される。RTA クエリを使用することにより、ユーザーは RTA を介して公開されたテーブルとそのデータに対してクエリを実行できる。公開されたデータベースからのデータの取得とすべてのリモートデータ (RTA テーブルからのデータ) の統合は、RTA クライアントによってシームレスに処理され処理される。

単一のクエリでリモートデータベースとローカルデータベースを使用するという考え方は、federated データベースの考え方と似ているが、第 2 章で紹介されたこれらの技術と RTA には以下のような異なる点がある。まず、RTA を通じて公開されたテーブルは、federation のメンバーだけでなく、どんな RTA ユーザーでも利用することができる。このように、RTA はユーザーがどんなデータベースにでもアクセスできるようにし、データが利用者の範囲を大幅に拡大する。さらに、RTA は異なる RDBMS によってバックアップされたデータソースのシームレスな統合を可能にし、エンドユーザーに読み取り権限のみを付与し、 federated データベースよりもメンテナンスコストを大幅に削減する。

3.2 Public Table Library

この節では、公開者側が自身の公開したいテーブルを登録する際に利用する Web アプリケーションである Public Table Library について説明する。

公開者側は、図 2 のように「データベース登録」から公開したいテーブルの公開情報を登録する。すると図 3 のように登録した情報が公開リストに追加される。そして、詳細ボタンを押すと図 4 のように公開テーブルの詳細が表示される。登録を行った段階で自動的にそのテーブルに存在するカラム名、型が検索、追加され、その後それぞれのカラムについての詳細説明を編集することが出来る。

利用者側は、「データベース一覧」から利用したいデータを探し、「アクセス名」に記述されているアクセス名をクエリ内に記述するだけでそのデータが自由に利用可能となる。

3.3 RTA の具体的な利用例

RTA の具体的な使用例を挙げる。まず、あるユーザーが 2 つのテーブルにいくつかのユーザー情報とそのユーザーが持つ株に関するデータを持っている (表 3 および表 4 にそのデータの一部を示す)。ユーザーが株における価値を計算したい場合、公開されている株価情報にアクセスする必要がある。PTL で

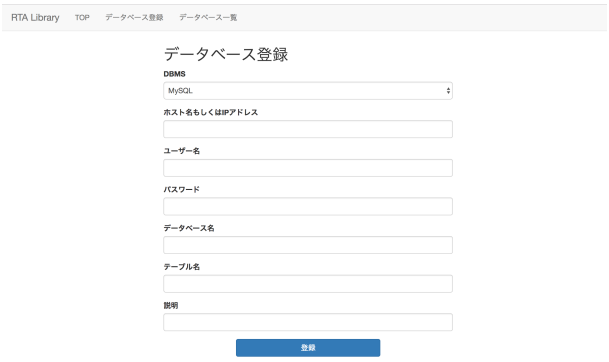

図 2　公開テーブル登録画面

図 3　公開テーブルリスト

図 4　公開テーブル詳細

stocks テーブル (表 5 に表示している) を見つけ、以下に示すクエリを使用して、RTA を介して公開されたデータを取得し、自身がもつデータと結合することができる。表 3、 表 4 および表 5 に示されているデータのみを考慮すると、 このクエリは表 6 に示された結果テーブルを返すことになる。

Query 1

```
SELECT u.id, u.name,
SUM (us.number * s.ending_price) as sum
FROM users u, user_stocks us, #stocks s
WHERE u.id = us.user_id AND us.code = s.code
GROUP BY u.id
```

表 3　users テーブル (ローカルテーブル)

id	name
1	田中
2	佐藤
3	中村

この例は、SQL クエリを記述することにより、RTA で公開されたデータをリレーショナルデータと簡単に統合できるという手法である。この例では、ユーザーは二つのローカルテーブルと一つのリモートテーブルのみを結合するが、クエリに記述すれば、複数のリモートテーブルを使用し、リモートテーブル間の結合を実行することもできる。RTA クライアントは、クエリに応答するために必要なデータを取得して処理する。

表 4　user_stocks テーブル (ローカルテーブル)

user_id	code	number
1	7203-T	100
1	6753-T	2000
1	4661-T	300

表 5　stocks テーブル (リモートテーブル)

code	brand	ending_price
7203-T	Toyota	7131
6753-T	シャープ	237
4661-T	OLC	6708

表 6　実 行 結 果

id	name	sum
1	田中	3199500
2	佐藤	1715100
3	中村	5910000

さらに、実際に RTA システムを利用する WEB アプリケーション開発者を想定する。開発者が図書館利用者が自分の貸出状況を把握できる「図書館の本貸出アプリケーション」を開発しようと考えた際には、図書館にある本の情報が格納されているデータ全てを保持することは現実的ではない。そこでオープンデータ側に本の情報 *book* テーブルがあり、アプリケーション側に利用者と貸出状況の情報 *user_rental* テーブルがあれば、RTA を利用することにより、以下に示す PHP プログラムを記述し、本貸出アプリを実現する。(図 5)

PHP programming

```
<?php
require_once('./rta/rta.php');
rtaSetConfig('config.rta');
$query = "
SELECT b.title, b.autor, ur.borrow, ur.return
FROM user_rental ur, #book b
WHERE ur.isbn = b.isbn AND ur.user_id = 1 ";
$result = rtaExec($query);
?>
```

4　集約可能データの推薦の提案

4.1　提案システムの概要

PTL はあらゆるオープンデータを一元的に管理しているため、提供者が増えれば増えるほど、PTL のサイズが大きくなる。そのためテーブルをより多く蓄えることを考えると、PTL には高度な検索機能、情報の推薦機能が求められる。特に集約

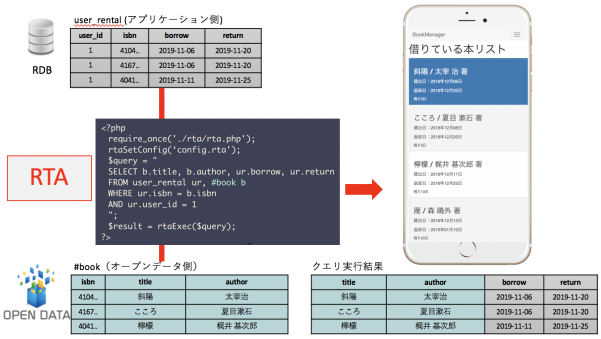

図 5　図書館の本貸出アプリケーション

可能データの推薦はオープンデータの性質により重要となる。

オープンデータは週単位や月単位などで、データを更新している。そのため週単位で更新しているデータは集約させることで、月単位のデータ、年単位のデータと同時に使うことができる。例えば東京都のオープンデータカタログサイト [16] では、港区の人口は月単位で更新されているが、コロナウイルス感染者数は週単位で更新されている。これらを月単位で揃えれば、港区の人口に対するコロナウイルス感染者数の割合を計算することができる。そのため集計単位に応じて PTL 登録テーブルのメタデータを管理し、ユーザーにとって有用かつ利用可能なテーブル推薦するフレームワークを作成することでさらなる利便性を高めることができる。図 6 は提案システムの概要図で、図 7 はメタデータの例である。

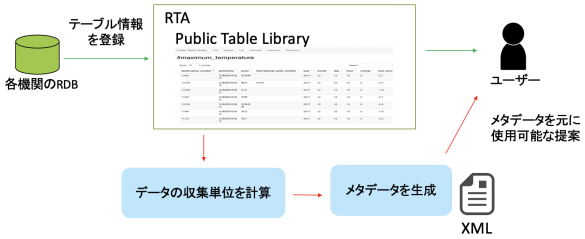

図 6　提案システムのアーキテクチャ

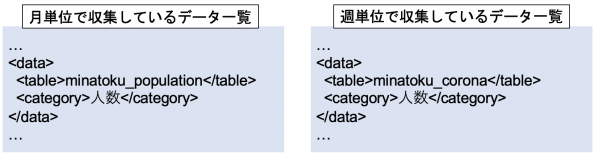

図 7　メタデータの例

登録されたテーブルに対して、日時カラムあるいはそれに相当するカラムがある場合、その中から数タプルを抜き出しその平均値を取ることで、毎週や毎月といった集計頻度を計算する。その集計単位を元にメタデータを更新し、ユーザーが使おうとしているデータの集計単位に合わせて、集約したデータを推薦する。図 7 の例を用いると、港区の人口のデータを使おうとしているユーザーに対して、週単位で集計している港区のコロナウイルス感染者数を月単位で集約して推薦するということである。

4.2　集約する上での問題点

集計単位を計算する際に、どのカラムが時系列情報を保持するカラムであるかを判断することに課題がある。例えば date 型や time 型であれば、そのカラムが時系列情報に関係していることは明白であるが、データによっては integer 型や varchar 型で登録されている場合もある。そのため、カラムの型のみでは判断できず、テーブルの内容から総合的に判断する必要がある。

4.3　メタデータの作成

メタデータの作成のフローを図 8 に示す。まず date 型や time 型では時系列データを扱っていることは明白である。integer 型や varchar 型では時系列データであるかはわからない。そこで、integer 型や varchar 型であった場合は、カラム名やデータの内容を自然言語処理を用いて時系列に関する情報であるかを判断する。その後集計単位の計算を行い、その情報をメタデータとして格納する。

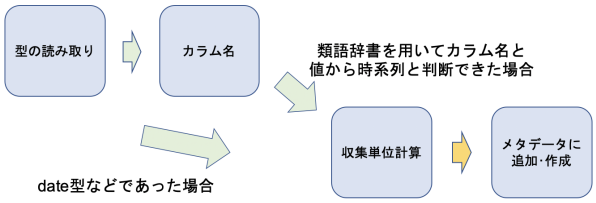

図 8　メタデータ作成の流れ

5　評 価 指 標

関連研究では検索に必要な時間や、成功率を評価指標としていた。今後は時間や成功率のみならず、意味的な類似度や、時系列カラムの判断の正当性を指標として評価をする予定である。

6　ま　と　め

6.1　結　　論

先行研究では，主にオープンデータの効率の良い二次利用という観点から、リモートデータへ RDB の形式のままアクセスすることが可能な RTA アーキテクチャの提案、実装を行った。RDB のまま利用可能という形式を取ることで、データ利用者がデータをダウンロードし、自身の DB に挿入するといった負担を軽減することが出来る。さらに、Public Table Library から利用可能なデータを収集するために、あるオープンデータを集約することで、別のデータと比較、結合といった利用が可能になることを示し、そのためのアーキテクチャを提案した。これにより RTA の利便性をより大きくできる。

6.2　今後の課題

データの集計を行うには、データごとによって集計ルールを

作る必要がある。本研究は時系列カラムがある前提での議論であったが、集約を行えるものは時系列データだけでない。例えば都道府県のテーブルを集約すれば、東北や関東のデータと比べられたり、全国のデータと結合できる。対応できるデータパターンを多くすることで、より有用性のあるシステムになる。

文　　献

[1] 村上 柊, 小坂 祐介, 五嶋研人, 遠山元道, "RTA: 公開型テーブルへの直接問い合わせ機構の提案", DEIM2017 第 9 回データ工学と情報マネジメントに関するフォーラム (第 15 回日本データベース学会年次大会), 高山グリーンホテル, 岐阜, 2017 年 3 月.

[2] Dennis Heimbigner，Boulder Dennis McLeod:“A federated architecture for information management”，Journal of ACM Transactions on Information Systems (TOIS)，1985

[3] Amit P. Sheth，James A. Larson:“Federated database systems for managing distributed, heterogeneous, and autonomous databases”，Journal of ACM Computing Surveys(CSUR)，1990

[4] Laszlo Dobos，Istvan Csabai:“Graywulf: A platform for federated scientific databases and services”，Proceedings of the 25th International Conference on Scientific and Statistical Database Management Article (SSDBM) No.30，2013

[5] Youzhong Ma，Jia Rao，Weisong Hu，Xiaofeng Meng，Xu Han，Yu Zhang，Yunpeng Chai，Chunqiu Liu:“An Efficient Index for Massive IOT Data in Cloud Environment”，Proceedings of the 21st ACM international conference on Information and knowledge management（CIKM）2012

[6] Jeff Shute，Radek Vingralek，Bart Samwel，Ben Handy，Chad Whipkey，Eric Rollins，Mircea Oancea，Kyle Littleeld，David Menestrina，Stephan Ellner，John Cieslewicz，Ian Rae，Traian Stancescu，Himani Apte:“F1: A Distributed SQL Database That Scales”，Proceedings of the VLDB Endowment 2013

[7] James C. Corbett，Jeffrey Dean，Michael Epstein，Andrew Fikes，Christopher Frost，J. J. Furman，Sanjay Ghemawat，Andrey Gubarev，Christopher Heiser，Peter Hochschild，Wilson Hsieh，Sebastian Kanthak，Eugene Kogan，Hongyi Li，Alexander Lloyd，Sergey Melnik，David Mwaura，David Nagle，Sean Quinlan，Rajesh Rao，Lindsay Rolig，Yasushi Saito，Michal Szymaniak，Christopher Taylor，Ruth Wang，Dale Woodford:“Spanner: Google's Globally Distributed Database”，ACM Transactions on Computer Systems 2013

[8] Fatemeh Nargesian,Erkang Zhu,Ken Q. Pu,Renée J. Miller:“Table Union Search on Open Data”Proc. VLDB Endowment11(7):813-825(2018)

[9] Erkang Zhu,Dong Deng,Fatemeh Nargesian,Renée J. Miller:“JOSIE: Overlap Set Similarity Search for Finding Joinable Tables in Data Lakes”ACM SIGMOD Conference2019:847-864

[10] Data.gov: http://data.gov/

[11] Data.go.jp: http://data.go.jp/

[12] Microsoft RDA: https://msdn.microsoft.com/ja-jp/library/cc414853.aspx

[13] FEDERATED ストレージエンジン: https://dev.mysql.com/doc/refman/5.6/ja/federated-storage-engine.html

[14] postgres fdw: https://www.postgresql.jp/document/9.3/html/postgres-fdw.html

[15] Ayman Alserafi, Alberto Abello, Oscar Romero, Toon Calders:"Towards Information Profiling: Data Lake Content Metadata Management" 2016 IEEE 16th International Conference on Data Mining Workshops

[16] 東京都オープンデータカタログサイト: https://portal.data.metro.tokyo.lg.jp/