ToT for CSV: CSV データの公開型テーブルへの利用

土居　靖†　遠山元道††

† 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: †doi@db.ics.keio.ac.jp, ††toyama@ics.keio.ac.jp

あらまし　現在, 各国, 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, XML などのダウンロード形式や Web サービスの API, LOD に基づく RDF など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, 公開することができる. しかし, 現在のオープンデータには, そもそも RDB 形式で保持していないデータが様々な形式で多数存在する. そこで, そういったオープンデータを蓄積し, RDB に変換, そのまま RTA と連携するアーキテクチャ(ToT : Tables on Top) を提案する. 本論文では, 様々な形式の中でも, 表形式の状態が多い CSV を対象とした ToT for CSV を開発し, さらなるオープンデータの利便性の向上を図る.

キーワード　オープンデータ, SQL, CSV

1. はじめに

現在, 気象情報, 郵便番号, 株価その他統計データ等の様々なオープンデータが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．

しかし, 上記の通り, 現在のオープンデータには, RDB 形式で保持していないデータが様々な形式で多数存在する. そこで, そういったオープンデータを蓄積し, RDB 形式に変換し, そのまま RTA と連携するアーキテクチャ(ToT: Tables on Top) を提案する. 本論文では, 様々な形式の中でも, 表形式の状態が多い CSV を対象とした ToT for CSV を開発し, さらなるオープンデータの利便性の向上を図る.

以下，本稿の構成を示す．2 章では現在のオープンデータの概要，問題点について述べ，3 章では関連技術・関連研究について述べる．4 章では先行研究である RTA の概要，利用例を述べ，5 章ではオープンデータにおける CSV ファイルの概要, 利用上の問題点について述べる．そして 6 章では今回提案する ToT for CSV について述べ，7 章で評価, 8 章で結論を述べる．

2. オープンデータ

2.1 オープンデータの概要

この章では，オープンデータの概要と現在のオープンデータ利用の問題点について述べる．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. 以下に各国のオープンデータのデータセット数を表 1 に示す.

表 1　各国のデータセット数 (2018/06/01 調査)

国	カタログサイト	データセット数
アメリカ	data.gov	280613
イギリス	data.gov.uk	45911
日本	data.gov.jp	21647

ユーザーがオープンデータを二次利用する際には主に 3 つの方法がある．1 つ目は，ダウンロード形式であればそのファイルを一度ダウンロードし，自身の DBMS にテーブルを作成しデータを挿入してから利用する方法．2 つ目は．提供機関の提供する API を利用する方法．3 つ目は，第 3. 章で述べる，既存リモートアクセス技術を用いる方法である．

2.2 オープンデータ利用の問題点

オープンデータの現在の問題点として，前節で述べた 3 つの方法それぞれに以下のような問題点がある．

ダウンロード利用

- 一度 CSV 等をダウンロードし、自分のデータベースに挿入しなければいけない手間.
- 頻繁に更新されるデータ（毎日の気象情報、株価データ

等）では、データが更新されるたびに挿入しなければいけないという手間.

API 利用

- 追加でプログラムを記述する手間.
- 予め提供者側の想定する形でしかデータが取得できないため，柔軟性に欠ける.

既存リモートアクセス技術利用

- 利用者側のテーブルと JOIN 出来ないものがある，異種 DBMS 間で利用できない等の問題点（詳細は第 3. 章，第 7. 章で後述）.

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．

3. 関連技術・関連研究

3.1 関 連 技 術

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 [11]（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 [12]．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 [13] も同様の技術である．

3.2 関 連 研 究

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 [4]．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 Ma 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．

4. RTA の概要

この章では，RTA システムの概要について述べる．RTA は，まず各機関がオープンデータを RDB の形のまま公開型テーブルとして登録を行う．注意として本論文では，提供されるオープンデータは元々RDB の形式で保存されたものであるとする．そしてデータ利用者側は読み取り専用ユーザーとして，登録された公開型テーブルに SQL によって直接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，先述した関連技術では同一の DBMS 間でのデータアクセスのみが可能であったが，RTA では MySQL，PostgreSQL，Oracle 等の一般的に広く普及している RDBMS であればそれらの相互利用可能な点が大きな特徴である．関連研究で述べた Federated Database とは，他のデータベースのデータの利用を可能にするという点では共通しているが，RTA ではネットワーク上での相互接続は想定しておらず，公開されたデータに対して利用者側が一方的にアクセスするという点で異なる．

RTA によって実現可能になることとして，データ利用者側，データ公開者側それぞれに以下のようなものがある．

4.1 データ利用者側のメリット

- 公開情報を関係データベースとして扱えるため，他の形式よりデータが管理しやすくなる.
- 公開データベース（複数でも可）とローカルのデータベースとの間で 1 つの SQL で直接 JOIN 操作などを行うことが出来るようになり，データを利用しやすくなる
- 常に最新のデータを取得することができる

4.2 データ公開者側のメリット

- データを公開し，利用してもらうために専用の WEB サービスを作る必要が無くなる.
- 定期的に公開データの更新をする必要が無くなる.

4.3 RTA の具体的な利用例

RTA においてリモートテーブルを利用する際には，**RTA クエリ**と呼ばれるクエリを用いる．RTA クエリとは，以下のクエリ 1 のように，FROM 句にテーブル名を記述する際に，リモートテーブルであることを示す識別子#を付けた SQL のことである.

次に RTA の具体的な使用例について述べる．今回は例として、現在オープンデータとして公開されている東証の株価デー

タ [9] を利用して，ローカルの users テーブル（表 2）、ユーザーの所有している株式の証券コードのみが保存されている user_stocks テーブル（表 3）と JOIN する以下の操作を挙げる．

クエリ 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 2　利用者の持つテーブル (users)

id	name
1	村上
2	佐藤
3	中山

表 3　利用者の持つテーブル (user_stocks)

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

表 4　公開テーブル (stocks)

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

クエリ 1 を実行することで，表 5 のような実行結果が得られる．

表 5　実 行 結 果

id	name	sum
1	村上	3199500
2	佐藤	1715100
3	中山	5910000

また今回はローカルテーブルとリモートテーブルの JOIN を例に挙げたが，ただリモートテーブルのデータを取得してくること，複数のリモートテーブル同士の JOIN を行うことも出来る．

4.4　RTA 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，RTA Library について説明する．

RTA Library とは，PHP 言語により記述された，公開テーブル登録用 Web アプリケーションである．

公開者側は，図 1 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 2 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 3 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．

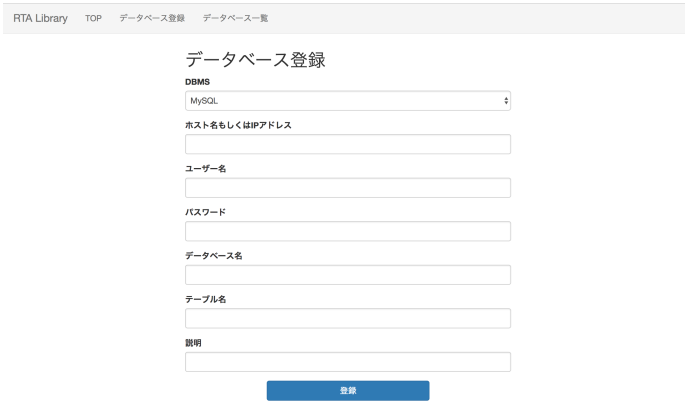


図 1　公開テーブル登録画面

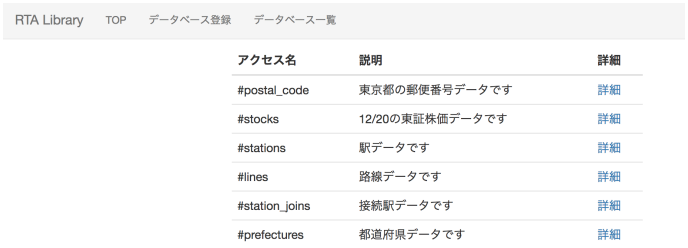


図 2　公開テーブルリスト

図 3　公開テーブル詳細

4.5 RTA 利用の改善

前節までで述べた通り, RTA を利用することにより, データ公開者側は自身のもつ RDB を別の形式に変換する必要なく, RTA Library に登録することができ, データ利用者側は RTA Library を通して, オープンデータを利用することができる. しかし, RTA はデータ公開者側がオープンデータを RDB 形式で保持している時にのみ有効であり, 表 1 で示した膨大なオープンデータがもともと RDB 形式ではない可能性もある. そのため現在のオープンデータ全てに有効であるとは言えないと考えられる. 以下, 図 4 に現在の RTA 利用の流れを示す.

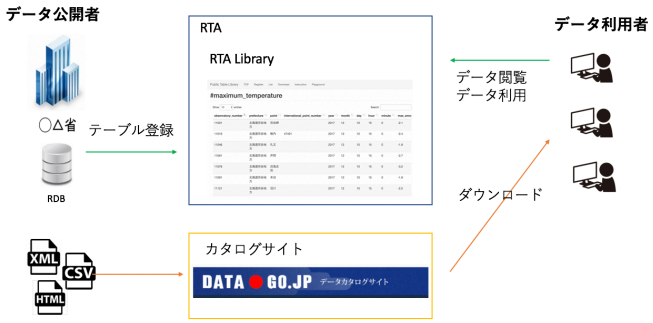


図 4　RTA とオープンデータ利用

以上で挙げた問題点を改善するために, データ公開者が持つ RDB 形式でないデータを, RDB 形式に変換するアーキテクチャを開発する必要がある. ただ変換するだけでなく, 変換可能な形へのファイルの整形や 複数データの集約, RDB 形式でのデータの保持, そして RTA Library との連携を可能にし, さらなるオープンデータの利便性の向上を図る. このアーキテクチャを ToT(Tables on Top) と称し, ToT を含めたオープンデータ利用の流れを図 5 に示す.

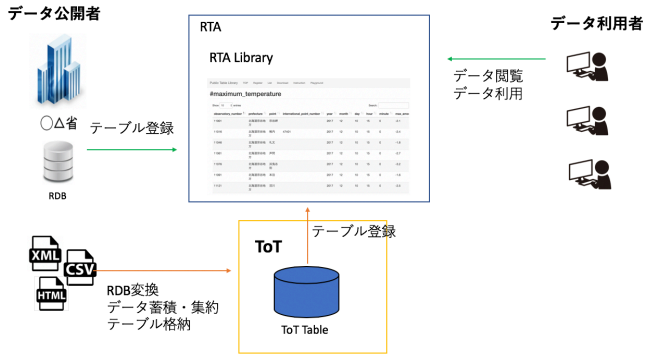


図 5　ToT とオープンデータ利用

5.　CSV 形式の問題点

この章では現在のオープンデータのデータ形式から, その中でも特に CSV 形式に着目して分析し, オープンデータの利用の問題点を述べていく.

5.1　オープンデータの CSV 形式

現在, 膨大なオープンデータが様々なデータ形式で存在しているが, 2014 年の ODI(Open Data Institute) のによると, data.gov.uk に公開されているデータの 90 %以上が表形式であり [17], また World Wide Web Found の調査ではオープンデータの中でも CSV は最も普及している形式の一つであるという結果がある. [18] また, 表 1 で示したカタログサイトから CSV データをダウンロードするとそれぞれ, data.gov には 16985 件, data.gov.uk には 39646 件, data.gov.jp には 5427 件あった. 以上のことから, オープンデータの利用の向上を図るために, まずは CSV 形式の分析をすべきであると考える.

5.2　CSV ファイルの形式

CSV 形式において, RFC4180 という CSV ファイルの一般書式が 2005 年 10 月に公開されている. [19] これは CSV ファイルについて最初にして唯一の公式な仕様なのだが, その内容はカンマ区切りやダブルクォテーションが正しく書かれているかなど, ヘッダの有無や各レコードのカラム数については判別していない. また, そもそも全ての CSV ファイルに対して, 守られていないのが現状である. [20] 各国のカタログサイトからダウンロードした CSV ファイルが RFC4180 に準拠しているかどうかを csvlint というツールを用いて, CSV ファイルをダブルクォテーションが正確に書かれていない (quoto-error), 各レコードのカラム数が一致していない (column-error), これらのエラーがない (valid) に分類したものが表 6 である.

表 6　CSV ファイルの分類

カタログサイト	CSV ファイル数	quato-error	column-error	valid
data.gov	16985	3305	288	13395
data.gov.uk	39646	5359	1378	32909
data.gov.jp	5427	13	2607	2807

また, ここで valid と分類されたものでも, 以下のよう条件に対しては分類できていない.

- ヘッダの有無が判別できない.
- メタ情報 (タイトルや表の説明) がファイル内に記述されているか判別できない.
- クロス表形式になっているか判別できない.

以上の点から, CSV ファイルには自由度が高いことがデータ利用の際の問題となると考えられる.

5.3　CSV ファイルの更新

オープンデータは継続的かつ反復的なデータ更新が行われる. CSV ファイルの更新においては, データを有効に利用, 蓄積するためには, 以下のようにファイルがどのような更新パターンなのか分析しなければならない.

- 年ごとや月ごとといった更新頻度がいつなのか.
- 更新の際にファイルの形が変わってしまうのか.
- ファイルを新規生成するのか.

特に, 更新の度に CSV ファイルが新規生成される場合は, 利用対象となるファイルが膨大になっていくという問題が生じ, 非常にオープンデータの利用効率が低下する.

6. ToT for CSV

この章では, 本研究で提案する ToT for CSV の機能についての概要と, 前章までで挙げた各問題点の解決策として, 具体的な例とともにそれぞれ述べていく. なお, 提案する機能の概要は表 7 に示す.

6.1 ToT for CSV の概要

図 5 で示したように ToT for CSV は CSV ファイルを対象とし, ToT Table に CSV ファイルを RDB 形式に変換し, 格納する. ToT Table を RTA と連携, つまり格納されているテーブルを RTA Library に登録することにより, CSV 形式を RTA で利用することが可能となる. しかし, そのままでは RDB 形式に変換できないものがあるので, ToT Table に格納する前に, 変換可能な形に整形を行う. 提案する機能の概要は表 7 に示す.

表 7　ToT for CSV の機能の概要

機能	問題点	解決策
Header Reading	ヘッダの有無	ヘッダの情報を読み取る
Table Cutting	メタ情報	メタ情報を省略したり, 情報を格納する
Cross Table Listing	クロス表	クロス表をリスト化する
CSV Aggregation	更新ファイルの新規生成	更新前後のファイルを集約する

6.2 ToT for CSV の機能

福岡県のオープンデータカタログサイトを対象に各機能について説明する. [22]

- Header Reading

図 6 のように, ヘッダの有無を確認し, ヘッダが存在した場合はその情報をテーブルの属性として格納する.

自治体コード	自治体名	地点名	年	月	週	患者数
401307	福岡市	中央区	2018	2	7	16.67
401307	福岡市	南区	2018	2	7	28.89
401307	福岡市	城南区	2018	2	7	53.8

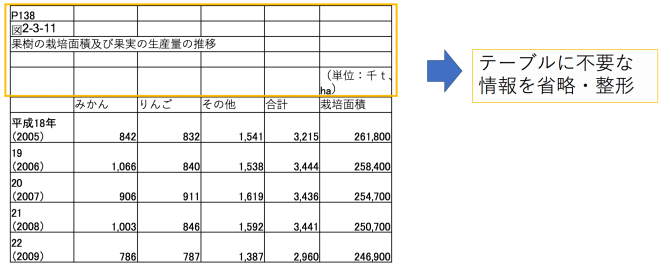
福岡市のインフルエンザ患者数

header情報をテーブルの属性として格納

図 6　Header Reading 機能

- Table Cutting

図 7 のように, RDB のテーブルに不要な情報, タイトルや単位などは省略し, ファイルを整形する.



果樹の栽培面積及び果実の生産量の推移

図 7　Table Cutting 機能

- Cross Table Listing

図 6 のように, オープンデータの CSV ファイルにはクロス表が多数存在した. このままでは RDB 形式への変換がうまくいかないため, 行のヘッダと列のヘッダの情報を格納し, リストの形に生成し直す.

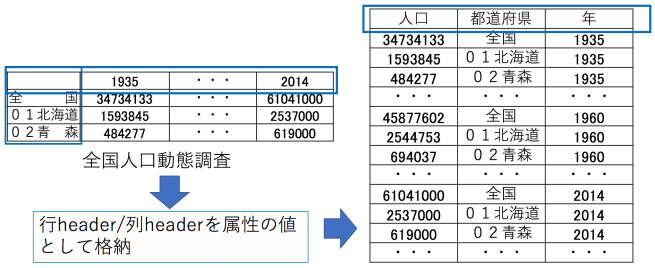


図 8　Cross Table Listing 機能

- CSV Aggregation

オープンデータの更新において, 月ごとに新規ファイルを生成するものもある. このようなファイルはオープンデータを利用する上で, ダウンロードするファイル数が膨大になるなどの問題が生じる. そこで図 9 のように, 更新情報から年や月の情報を属性としてテーブルに追加, 同一のデータセットのファイルを集約し, 一つの RDB に変換する.

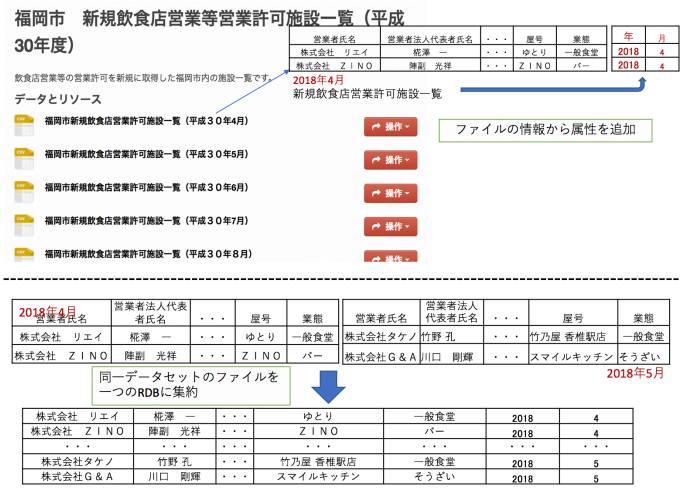


図 9　CSV Aggregation 機能

以上の機能により, 様々な CSV ファイルをデータ利用がしやすいように RDB 形式に変換可能となる.

7. 評　　価

7.1 既存リモートアクセス技術との RTA の機能比較

先行研究である RTA の性能を測るために，先述した既存技術・既存研究である RDA，FEDERATED DATABASE（FD）との機能比較を行った.

RTA と RDA，FD の機能比較を表 8 に示す. この表から，既存の技術に比べて，アクセスしたいリモートテーブルの情報をほとんど知らなくてもオープンデータが利用可能であるとい

うことが分かる．公開されているオープンテーブルの種類，そのテーブルのカラム情報などは全て先述した RTA Library 上に記述されているため，それを見れば一目で理解することができる．また，そのアクセス名をクエリ内に記述するだけで自動的にアクセスしてデータを取得するため，パスワード等のデータベースへの接続情報を利用者側が知っている必要も無い．

また，RDA，FD はそれぞれ同種の DBMS 間でのみ利用可能な技術である．しかし，RTA は例えば公開側の DBMS が PostgreSQL，利用者側の DBMS が MySQL 等である場合も，データを取得してきた際に自動的に利用者側の DBMS に合わせたデータ型にデータを変換し，テーブル作成，データ挿入等を行うため，利用者側が DBMS の違いを意識することなく自由にデータを利用することができるという優位性がある．

表 8　各リモートアクセス技術の機能比較表

機能	RDA	FD	RTA
互いの接続先情報が不要	○	×	○
自身で保存用テーブル作成不要	×	×	○
テーブル定義，カラム情報が不要	×	×	○
ローカルテーブルとの JOIN 操作	×	○	○
異なる DBMS 間でも利用可能	×	×	○

8. おわりに

8.1 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．また，RTA Library という Web アプリケーションを見ることで利用可能なデータの種類，カラム名，データ型などを容易に確認することが出来るので，利用者側がそれらを判断するといった負担を軽減することも出来る．

さらに本研究で提案した ToT for CSV により, オープンデータにおける CSV ファイルの様々な形式に対応して RDB 形式への変換が可能となっている．また ToT は RTA Library と連携することにより, 上記で挙げた RTA のメリットを活かしたまま利用可能となる．

8.2 今後の課題

ToT はまだ開発初期段階であり, 今回分析した CSV ファイルのデータ形式以外にも, XML や HTML といったオープンデータにおける主流なデータ形式がまだまだ存在する．今後は ToT をベースとして, 機能を拡張していき, RTA と連携すべきであると考えられる．

参 考 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] 郵便番号データ: http://www.post.japanpost.jp/zipcode/download.html

[9] 株価データ: http://k-db.com/

[10] JSQLParser: https://github.com/JSQLParser/JSqlParser

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

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

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

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

[15] Data.gov.uk: http://data.gov.uk/

[16] Data.gov.jp: http://www.data.go.jp/

[17] 2014: The Year of CSV — News, Open Data Institute. [Online]. Available: https://theodi.org/blog/2014-the-year-of-csv. [Accessed: 15- Jul-2016].

[18] T. Davies, R. M. Sharif, and J. M. Alonso, “Open Data Barometer Global Report,”World Wide Web Found., 2015.

[19] Y. Shafranovich, “Common format and MIME type for comma- separated values (CSV) files,”2005.

[20] Till Dhmen, Hannes Mhleisen, Peter Boncz “Multi-Hypothesis CSV Parsing”, SSDBM ’17 Proceedings of the 29th International Conference on Scientific and Statistical Database Management Article No. 16

[21] Wirawit Chaochaisit, Ken Sakamura, Noboru Koshizuka, Masahiro Bessho : CSV-X: A Linked Data Enabled Schema Language, Model, and Processing Engine for Non-Uniform CSV. iThings/GreenCom/CPSCom/SmartData 2016: 795-804

[22] 自治体オープンデータの CKAN: https://ckan.open-governmentdata.org/dataset/