D-014

埋め込み型 SuperSQL における infinite-scroll の導入

田嶋 将大† 五嶋 研人† 遠山 元道†
Tajima Masahiro Goto Kento Toyama Motomichi

1. はじめに

SuperSQL は，独自のクエリを記述することによって関係データベースの出力結果を構造化し，多様なレイアウト表現を可能とする SQL の拡張言語である．通常の SQL クエリでは，シンプルでフラットな表しか再現できないが、SuperSQL を用いると，HTML と PHP などを用いたサーバーサイドプログラミングなどによる一般的な方法に比べてはるかに少ない行数で Web ページを生成することができる．また，近年では，Web で容易に入手することが出来るフレームワーク等を利用したリッチなデザインの HTML テンプレートを含む既存コードに対して，SuperSQL クエリの PHP 経由での埋め込みを可能とする埋め込み型 SuperSQL[4]の実現がなされた．

これまでの研究によって，SuperSQL による動的 Web ページの生成が行えるようになってきたいい方で，従来の SuperSQL ではページロード時にクエリにマッチするデータを全て取得し，Web コンテンツの生成が行われる為，膨大なサイズのデータを扱う場合には，SuperSQL の処理時間が内部で生成される SQL クエリの実行時間に大きく影響を受けてしまう．また，SuperSQL によって巨大な Web ページが生成されてしまい，画面の小さなタブレット端末上での閲覧には不向きな Web ページとなってしまうことや，ユーザが見る必要の無い Web コンテンツも生成されてしまうといった問題点が挙げられる．巨大 Web ページ生成の問題点の対策として、従来の SuperSQL では，複合反復子によって実現するページネーションを採用することで対策が可能であるが，この手法もページアクセス時にデータベースからクエリに一致するデータを全て取得し Web コンテンツの生成を行なっている為，SQL クエリ実行による遅延の問題解決には至っていない．

また，近年では Ajax を用いた非同期通信をベースとした infinite-scroll と呼ばれる Web ページ開発手法が注目を集めており，Twitter や Facebook などのソーシャルメディアサイトの多くが導入を行なっている．これは，ユーザのスクロール操作によってデータロードを行い，Web コンテンツを分割して表示を行う為，追加コンテンツの取得の遅延が少なく済み，ユーザが必要としていない分までの Web コンテンツの生成が行われずに済む．このような Infinite-scroll の実現のためには近年では多くのプラグインなどが公開されているが，追加コンテンツ取得用のサーバサイドプログラムは開発者自身が実装しなければならず，また，プラグインの機能に合わせたベースとなる HTML のコーディングの必要があり，複雑な構造を持った Web ページに対して infinite-scroll を実現する場合や，1 ページ内に複数の infinite-scroll を適用する場合にそのコーディングが煩雑になってしまうといった問題点が挙げられる．

こういった背景から，本研究では結合子や反復子などを用いて多彩なレイアウト構造を直感的に記述できるという，SuperSQL の特徴を活かし，クエリ内に infinite-scroll の適用を明確かつ容易に記述を行うことで，infinite-scroll を用いた Web ページの制作におけるコーディングコストの削減を目指す．また，従来の SuperSQL ではクエリ実行時にクエリに一致するデータを全て取得し Web コンテンツの生成を行なっているため，無駄なデータ検索，データ転送が生じてしまう問題に対し，提案手法の導入によってユーザが要求する，必要な情報量のみを取得することができるようになり，SuperSQL 実行時の無駄なデータ転送を削減することを目指す．

以降，本稿では第 2 章で SuperSQL について，第 3 章で infinite-scroll について，第 4 章で埋め込み型 SuperSQL における infinite-scroll の導入について，第 5 章で実験・評価について述べ，第 6 章でまとめを記述する．

† 慶應義塾大学大学院理工学研究科 Keio University

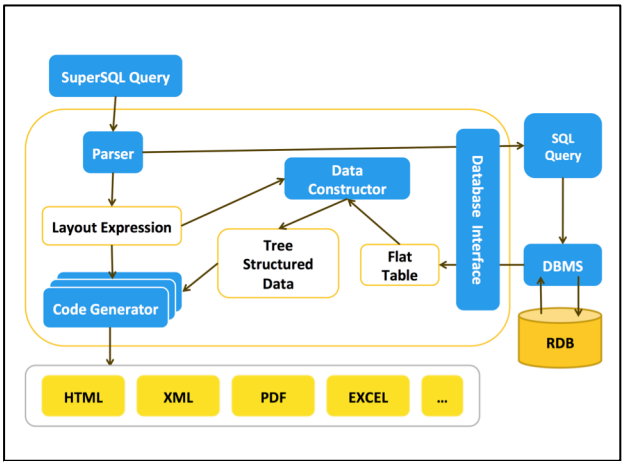

図 1 SuperSQL アーキテクチャ

2. SuperSQL とは

SuperSQL は関係データベースの出力結果を構造化し，多様なレイアウト表現を可能とする SQL の拡張言語であり，慶應義塾大学遠山研究室で開発されている[1, 2]．そのクエリは SQL の SELECT 句を GENERATE <media> <TFE>の構造を持った GENERATE 句で置き換えたものである．ここで<media>は出力媒体を示し，HTML，PDF，Mobile_HTML5[3]などの指定ができる．また<TFE>はターゲットリストの拡張である Target Form Expression を表し，結合子，反復子などのレイアウト指定演算子を持つ一種の式である．

2.1 SuperSQL アーキテクチャ

SuperSQL の内部処理について説明する．

まず，入力された SuperSQL クエリを構文解析部で解析し，「SQL クエリ」と「出力する HTML の構造や装飾情報を持ったレイアウト式」を生成する．次にデータベースインターフェース部を通して関係データベースに生成された

SQL クエリの問い合わせを行い, 結果をデータコンストラクタへ渡す. データコンストラクタは, 受け取ったフラットなデータをツリー構造のデータに変換しコード生成部へ受け渡す. 最後にコード生成部では, 受け渡された情報を基にして各出力ファイルの生成を行い, 出力する. SuperSQL のアーキテクチャを図 1 に示す.

2.2

2.2.1 結合子

結合子はデータベースから得られたデータをどの方向(次元)に結合するかを指定する演算子であり, 以下の 3 種類がある.

- 水平結合子(,)

データを横に結合して出力する.

例：title , year

マトリックス	1999

- 垂直結合子(!)

データを縦に結合して出力する.

例：title ! year

マトリックス
1999

- 深度結合子(%)

データを 3 次元方向へ結合する. 出力が HTML ならばリンクとなる.

例：title % year

マトリックス

↓

1999

2.2.2 反復子

反復子は指定する方向に, データベースの値があるだけ繰り返して表示する. また反復子はただ構造を指定するだけではなく, そのネストの関係によって属性間の関連を指定できる.例えば,

[部署名]! , [雇用者名]! , [給料]!

とした場合には各属性間に関係なく, 単に各々の一覧が表示されるだけである.一方, ネストを利用して

[部署名 ! [雇用者名 , 給料]!]!

とした場合にはその部署毎に雇用者名と給料の一覧が表示されると行ったように, 属性間の関連が指定される. 以下, その種類について述べる.

- 水平反復子

データインスタンスがある限り, その属性のデータを横に繰り返し表示する.

例：[title],

マトリックス	スターウォーズ	・・・

- 垂直反復子

データインスタンスがある限り, その属性のデータを縦に繰り返し表示する.

例：[title]!

マトリックス
スターウォーズ
・・・

2.2.3 装飾子

SuperSQL では関係データベースにより抽出された情報に文字サイズ, 文字スタイル, 横幅, 文字色, 背景, 高さ, 位置などの情報を付加することができる. これらは装飾演算子(@)によって指定する.

< TFE > @{ <装飾指定> }

装飾指定は”装飾子の名称 = その内容”として指定を行う. 複数指定をする場合は各々を” , ”で区切る.

[name, birthday]!@{width=200, color=”red”}

2.2.4 関数

- image 関数

image 関数を用いると画像を表示することが可能となる. 引数には属性値を用いて画像ファイルの存在するディレクトリにパスを指定する.

Image(“image/”|| pict)

- link 関数(出力メディアが HTML の場合のみ)

link 関数は, FOREACH 句と同時に用いる. これらを用いることで深度結合と同様にリンクを生成することができる.

link(name , “./menu.ssql” , place)

2.3 埋め込み型 SuperSQL

埋め込み型 SuperSQL[4]は, 動的 Web 開発言語である SuperSQL を PHP の関数群として実現することにより, 近年 Web で容易に入手することができるフレームワーク等を利用したリッチなデザインの HTML テンプレートを含む既存のコードに対して, SuperSQL クエリの PHP 経由での埋め込みを可能としている.

埋め込み型 SuperSQL では, データを XML で受け渡し, JavaScript で実装した SuperSQL レンダリングエンジンを用いて Web コンテンツのレンダリングを行うクライアントサイド処理機構により, 生成した Web コンテンツのメンテナンス性の向上を図っている.

SuperSQL の PHP を用いた埋め込み及び PHP 関数の処理の流れを図 2 に示す. ssql_config 関数および ssql_exec 関数により SuperSQL 実行に必要な引数の構築を行い, それらの引数を元にして SuperSQL が実行される.

受け渡された引数を元に, 図 1 で示した処理の流れに沿って SuperSQL が処理を行い, 最後のコード生成部において, 基盤となる HTML, スタイルを指定する CSS, データと構造を持つ XML, レンダリングエンジンの JavaScript が生成される. これらの出力ファイルを用いてクライアントサイドで Web コンテンツの構築が行われ, 構築された Web コンテンツは最終的に基盤となる HTML にアペンドされ既存コード内に埋め込まれる.

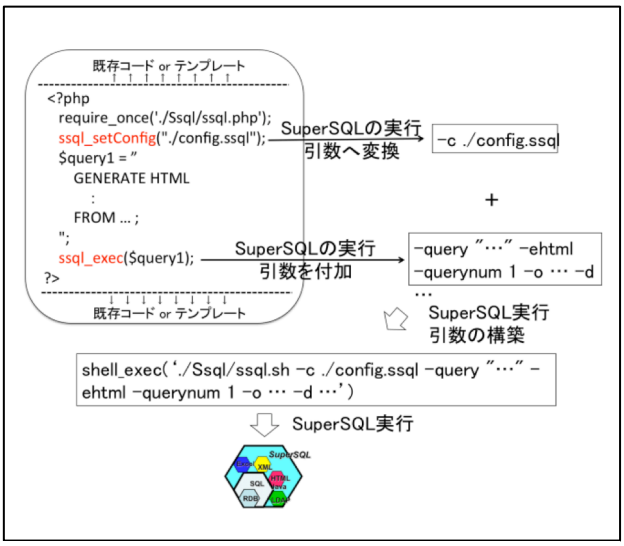

図 2　埋め込み型 SuperSQL の PHP 関数処理の流れ

3. infinite-scroll

ここでは，近年 Web ページ開発において注目を集めている infinite-scroll と呼ばれる Web ページ開発手法について述べる．infinite-scroll は，google map の登場から注目を集めてきた，JavaScript と DOM を用いた非同期通信を行う Ajax[5]技術をベースとしており，大量のデータを分割して処理を行う為，ユーザにとって比較的必要の無いデータ，例えば時系列的に古いデータなど，を取得せず，ユーザによって要求が行われた場合にのみ取得することができ，無駄なデータベースへの問い合わせデータ転送を削減することができる．また，この infinite-scroll を用いた Web ページでは，追加データの要求は，一般的にユーザによるスクロール操作によって行われる為，PC 上だけでなく，タブレット端末上でのユーザの操作を想定した Web ページにも適した手法であると言える．

また，コンテンツを分割して表示を行う手法にはページネーションが存在し，その端末による UI 面の違いや，Web サイトの種類による UI 面に関する研究も盛んに行われている[6, 7, 8, 9]．

現在では，infinite-scroll の実装のサポートを行う目的で様々なプラグインが提供されており，Infinite-scroll[10]や jScroll[11]などがある．これらは Web ページ開発者のスクロールイベント発生時の Web ページの動作実装をサポートすることに加え，スクロール操作における，追加コンテンツ取得タイミングやローディング中のインターフェースとしてローディング画像を指定できるなど，様々なオプションの設定を可能としている．

しかし，これらのプラグインは infinite-scroll を用いた Web ページの実装のサポートを行なってはいるが，それぞれのプラグインの機能に合わせて，Web 開発者が，追加コンテンツ取得用のスクリプトプログラムを記述する必要があり，サーバサイドプログラミングに関する知識が必要になることや，ベースとなる HTML のコーディングを行う必要があり，複雑なレイアウト構造を持った Web ページに対して infinite-scroll を用いる場合や 1 ページ内に複数の infinite-scroll を入れ子状に用いる場合に，そのコーディングが煩雑になってしまうといった問題点が挙げられる．

4. infinite-scroll の導入

4.1 言語仕様

本提案手法では，本来データインスンタンスがあるだけ繰り返し表示する意味を持つ反復子に対して装飾子を用いてデータコンテンツの分割数を指定する．

クエリ 1 のように，反復子に@{infinite-scroll=n}と記述することによって，インスタンスを n 個ずつに分割して表示を行うことができるようになる．また，SuperSQL の特徴である反復子のネスト構造に対しても@{infinite-scroll=n}をそれぞれに指定することができ，生成される Web ページにおいて複数箇所にわたって infinite-scroll を実現することができる．クエリ 1 の場合，各 TFE1 のインスタンスに関連した TFE2 のインスタンスを n2 個ずつに分割し，横方向に並んだ Web コンテンツが生成されるようになり，また TFE1 とそれに関連した TFE2 で構成されるインスタンスを n1 個ずつに分割し，縦方向に並んだ Web コンテンツが生成されるようになる．

```
GENERATE PHP
[ TFE1
    [ TFE2 ],@{infinite-scroll=n2}
]!@{infinite-scroll=n1}
FROM ・・・;
```

クエリ 1

```
GENERATE PHP
[ m.title! image(m.image)!
    [ image(r.rank)! r.comment ],@{infinite-scroll=4}
]!@{infinite-scroll=3}
FROM movie m, review r
WHERE m.id = r.m_id;
```

クエリ 2

具体的には，クエリ 2 によって生成される Web ページは図 3 のようになり，映画のタイトル，画像，レビューランクを示す画像，コメントの 4 つの属性で構成されるコンテンツを縦方向に 5 個表示し，縦方向にスクロールが行われ，コンテンツのロードが行われると追加で 3 個のコンテンツが表示されるようになる．また，各映画に対してのレビューランクを示す画像とコメントの 2 つの属性で構成されたコンテンツを 4 個表示し、横方向にスクロールが行われ，コンテンツのロードが行われると追加で 4 個のコンテンツが表示されるようになる．

4.2 既存の埋め込み型 SuperSQL と提案手法の違い

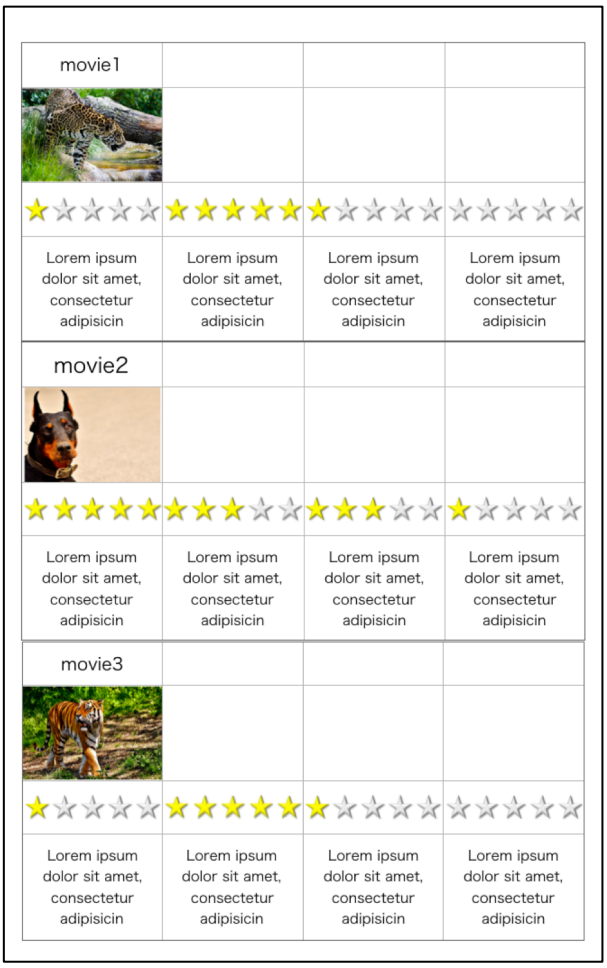

図 3　クエリ 2 による生成 Web ページ
※画像参照 URL: https://pixabay.com/

ここで, 既存の埋め込み型 SuperSQL と提案手法の違いについて述べる. 図 4 にその違いを表した図を示す. 既存の埋め込み型 SuperSQL と提案手法の大きな違いの一つは, SuperSQL の役割である. 既存の埋め込み型 SuperSQL では, HTML テンプレートや既存コードに対して, WEB ページ開発者が SuperSQL クエリを直接埋め込み, Web サーバへ配置する. そして, 該当ページにアクセスがされた際に SuperSQL が動的に実行され, Web コンテンツのみを生成して既存ページ内に埋め込みが行われる. 一方で提案手法では, Web ページ開発者は既存埋め込み型 SuperSQL と同様にして Web ページ開発を行う. そして, 該当ページに対して初めて閲覧者がアクセスした際に, SuperSQL が実行され PHP ファイルが生成される. これによって Web コンテンツが生成され既存ページ内に埋め込みが行われる. 以降も該当ページにアクセスがされた場合には, SuperSQL の実行は行われず, 生成された PHP によって Web コンテンツの構築及び埋め込みが行われる.

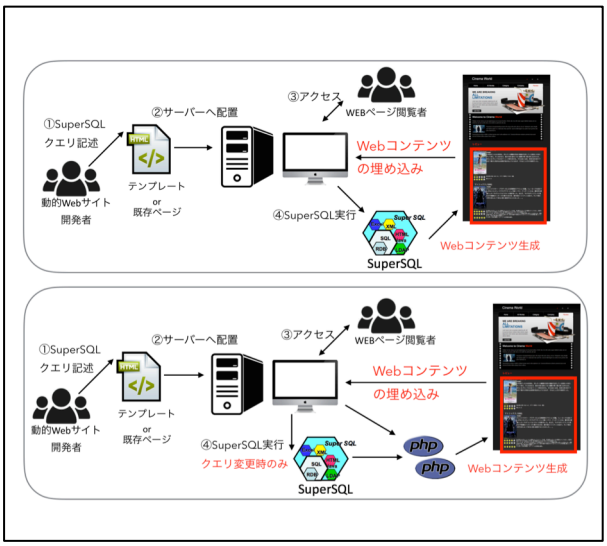

図 4　既存の埋め込み型 SuperSQL と提案手法の違い

また, もう一つの違いは, 既存の埋め込み型 SuperSQL では該当ページにアクセスが行われた際に, データベースからクエリに該当するデータを全て取得し, Web コンテンツの生成が行われるが, 提案手法では, ページアクセス時及び追加データロード時に, Web ページ開発者が指定したデータ分割数ずつ Web コンテンツの生成を行う.

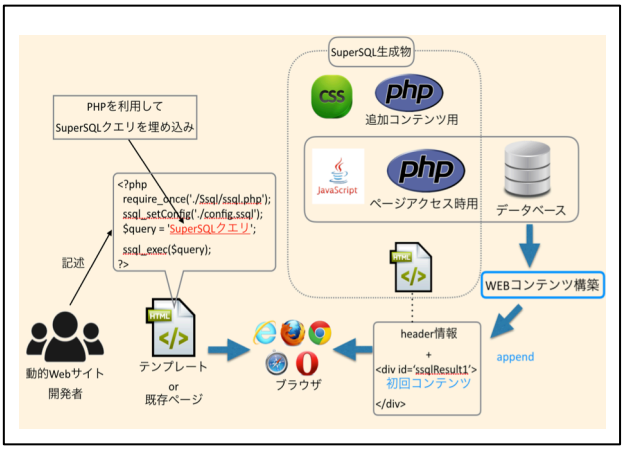

図 5　提案手法のアーキテクチャ(ユーザページアクセス時)

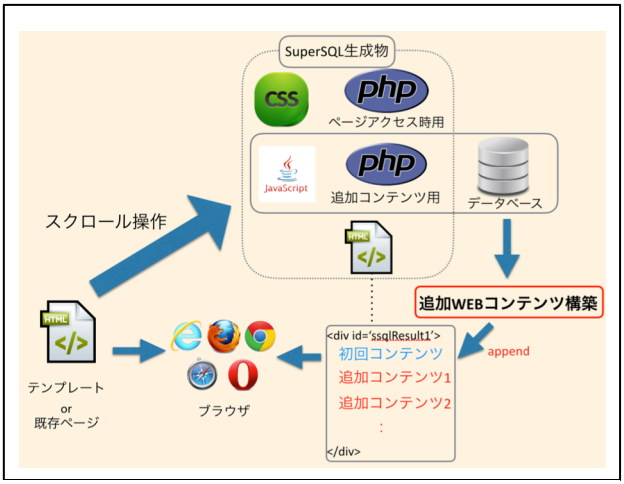

図 6　提案手法のアーキテクチャ(スクロールイベント検出時)

提案手法のアーキテクチャを図 5 と図 6 に示す. ユーザが該当ページにアクセスすると初回ページアクセス時用の PHP によって SQL クエリの実行及び Web コンテンツを SuperSQL クエリ内で指定された個数だけ構築を行い, 生成された Web コンテンツを該当ページ内に埋め込みが行われる. また, 該当ページ内において, infinite-scroll 指定箇所に対するスクロールイベントが検出され, 追加コンテンツのロード要求が行われると, SuperSQL によって生成された追加コンテンツ用の PHP によって SQL クエリの実行及び追加 Web コンテンツの構築が行われる.

```
GENERATE PHP
[g.name!
  [m.title! image(m.pict)]@{infinite-scroll=3}
],
FROM movie m, genre g
WHERE m.genre = g.id;
```

クエリ 3

```
$itemcount = 現在の子ノードの数;
$array_value = {関連が指定された値};
$limt = 3 //@{infinite-scroll=3}の右辺の整数
$SQL = “SELECT m.title, m.image FROM  movie m, genre
g WHERE m.genre = g.id AND g.name = $1;”;
$result = execute_sql(SQLquery, $array[0]);

for $i = $itemcount to $datacount + $limit – 1 do
   取得した値を利用して HTML コード生成;
end
```

図 7　データ分割取得アルゴリズム

4.3 処理系

ここで提案手法によって生成される PHP, JavaScript によって infinite-scroll の実現のための処理について説明をする.

クエリ 3 では, 映画のジャンル毎に映画のタイトルと画像を 3 個ずつに分割して取得を行う. 映画のタイトルと画像を含んだクラスにおいてスクロール操作が行われると, そのクラス内で現在表示されているコンテンツの数, そのクラス内の値の関連が指定されている属性の値(クエリ 3 では, 映画のジャンル)をそれぞれ取得し, 追加コンテンツの指定及び取得範囲の決定に使用する.

図 7 は,

5. 実験・評価

提案手法によるコーディングコスト削減を実証する為, 本研究では, 提案手法のみを用いた場合, jScroll プラグインのみを用いた場合とプラグインを使用せず, Ajax のみを用いた場合の 3 種類の実装方式によって, 同じ仕様の Web ページの作成を行い, それぞれの実装コストとしてコード量の計測を行なった.

今回, 図 2 に示した, 映画のタイトル, 画像, レビュー評価画像とコメントの 4 個の要素でできるコンテンツを縦方向に 3 個ずつ, レビュー評価画像とコメントの 2 個の要素でできるコンテンツを横方向に 4 個ずつ分割して取得および表示を行う Web ページの作成を行なった. 表 1 に計測結果を示す. 計測コストは, Web ページアクセス時用のページレイアウトおよびコンテンツ生成のコードを指す”HTML”, スクロール操作によって取得するコンテンツ生成のコードを指す”追加コンテンツ用 PHP”, スクロール方向やスクロール幅の指定および, スクロールイベント時の動作定義を行うために必要な”CSS + JavaScript”, それぞれの合計量として”全体”の 4 つ の計測を行なった. 今回の実験では, 既存のテンプレートに対してコーディングを行なったため, 計測値には, 実際に開発者自身が記述を行なったコードのみを対象とした.

表 1 に見られるように, 提案手法では, 既存ページに対して, PHP を用いて SuperSQL クエリの埋め込みを行うだけで, 追加コンテンツ用の PHP やページレイアウトに合わせた CSS, スクロール操作時の処理を行う JavaScript などが自動

表 1　コード量比較および実現可能性

	Ajax	jScroll	提案手法
HTML（行）	36	38	12
追加データ取得用 PHP（行）	48	66	0
CSS + JavaScript(行)	48	27	0
全体（行）	132	131	12
実現可能	○	×	○

で生成されるため，PHP や CSS，JavaScript のコーディングは必要なくなることがわかる．一方，Ajax や jScroll プラグインのみをそれぞれ用いた場合では，ベースとなる HTML の記述に加え，追加コンテンツ用の PHP なども全て Web ページ開発者が記述する必要があり，提案手法は全体を通して約 1/10 にもコーディングコストの削減を実現した．

また，表 1 の一番下の行にある実現可能から，Ajax と提案手法では，図 2 のような入れ子構造に infinite-scroll を用いた Web ページの実現が可能であったことがわかるが，jScroll のみを用いた場合では，実現ができなかったことがわかる．これは，jScroll ではスクロールイベント処理用の JavaScript のコーディングのサポートを行なっているものの，スクロール方向の検出に関しては，垂直方向に対してのみ対応しており，横方向への infinite-scroll に応用した場合，常に垂直方向へのスクロールイベントが行われたと判断されてしまい，常に追加コンテンツの取得が行われてしまっていた為，横方向への infinite-scroll を仕様した Web ページ開発は不可能であった．このことから，公開されているプラグインなどを用いた場合では infinite-scroll 適用に制限があるものの，提案手法では 2 次元方向自在に infinite-scroll を用いた Web ページの生成を行うことができ，より多彩なレイアウト構造を実現することがわかる．

6. まとめ

本研究では近年ソーシャルメディアサイトでの導入から注目を集めている，infinite-scroll の埋め込み型 SuperSQL への導入を行なった．これによって，SuperSQL では従来クエリに該当するデータを全て取得し，全てのコンテンツの生成を行なっていた為，ロングテールな特性を持ったデータベースなどにおいて，ユーザが比較的興味を持っているデータのみを優先的に提供できるようになる．

また，infinite-scroll を用いた Web ページの実装において，jScroll などのようなプラグインが多く公開されているが，それらを用いた実装にはスクロール方向に制限があったり，レイアウト構造が多様になれば，そのコーディングコストが膨大になり，煩雑になってしまう傾向にあるが，多彩なレイアウト構造を直感的に非手続き的に記述できる SuperSQL の特徴を活かし，複雑なレイアウト構造を持った Web ページにおいても，はるかに少ないコード量で infinite-scroll を用いた Web ページの実現を行うことができる．

謝辞

本研究を進めるにあたり，御多忙な中，最初から最後まで丁寧に御指導してくださった遠山准教授に深く感謝いたします．またミーティングを通じて本研究に様々な意見をくださった SuperSQL 班，研究室の皆様に感謝致します．最後にこれまで何不自由なく研究に集中できるよう支援してくれた両親に感謝致します．

参考文献

[1] SuperSQL: http://ssql.db.ics.keio.ac.jp
[2] . M. Toyama: “SuperSQL: An Extended SQL for Database Publisching and Presentation”, Proceedings of ACM SIGMOD ’98 International Conference on Management of Data, pp 584-586, 1998
[3] Kento Goto, Motomichi Toyama, “Mobile Web Application Generation Features for SuperSQL”, Proceedings of the 20th International Database Engineering & Applications Symposium , pp 308-315, 2016
[4] 五嶋研人, 木谷将人, 遠山元道, Web 開発フレームワークのための SuperSQL 埋め込み機構の開発, 情報処理学会論文誌:データベース 75 号, 2017.
[5] Ajax: https://pdfs.semanticscholar.org/c440/ae765ff19ddd3deda24a92ac39cef9570f1e.pdf, Jesse James Garrett, 2005
[6] Abdullah Natrah, Annamalai Muthukkaruppan, Rani Muhammad Khairul Amry Abd, “PAGING VS. SCROLLING: NAVIGATION STYLES FOR SELF-TRIAGE OF EPIDEMIC DISEASES”, Journal of Theoretical and Applied Information Technology, pp 262-272, 2016
[7] Jaewon Kim, Paul Thomas, Ramesh Sankaranarayana, Tom Gedeon, Hwan-Jin Yoon, “Pagination versus Scrolling in Mobile Web Search”, Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp 751-760, 2016
[8] Pingfei Wang, Qian Fei, Scrolling or Paging: The Impact of Interaction Style on the Search Result Page of Mobile Commerce Website, HCI International 2013-Posters’ Extended Abstracts pp 454-457, 2013
[9] Julie Campbell, Andrea Chin, and Alicia Lee, “The Effects of Pagination and Infinite Scrolling on Leisure Browsing”, https://andrealchindotcom.files.wordpress.com/2011/11/paginationvsscrolling.pdf
[10] Infinite-scroll: https://infinite-scroll.com/
[11] jScroll: http://jscroll.com/