

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6511438号
(P6511438)

(45) 発行日 令和1年5月15日(2019.5.15)

(24) 登録日 平成31年4月12日(2019.4.12)

(51) Int.Cl.

F I

G 0 6 F 16/00 (2019.01)

G 0 6 F 17/30 4 1 5

請求項の数 7 (全 33 頁)

(21) 出願番号 特願2016-516778 (P2016-516778)
 (86) (22) 出願日 平成26年5月28日 (2014.5.28)
 (65) 公表番号 特表2016-524756 (P2016-524756A)
 (43) 公表日 平成28年8月18日 (2016.8.18)
 (86) 国際出願番号 PCT/US2014/039771
 (87) 国際公開番号 W02014/193943
 (87) 国際公開日 平成26年12月4日 (2014.12.4)
 審査請求日 平成29年5月24日 (2017.5.24)
 (31) 優先権主張番号 13/906, 162
 (32) 優先日 平成25年5月30日 (2013.5.30)
 (33) 優先権主張国 米国 (US)

前置審査

(73) 特許権者 502303739
 オラクル・インターナショナル・コーポレ
 イション
 アメリカ合衆国カリフォルニア州9406
 5レッドウッド・シティ、オラクル・パ
 ークウェイ500
 (74) 代理人 110001195
 特許業務法人深見特許事務所
 (72) 発明者 ビシュノイ, サンディーブ
 インド、125055 ハリヤーナ、シル
 サ、バーナラ・ロード、コロニー、29-
 エム・アイ・ティ・シイ

最終頁に続く

(54) 【発明の名称】 連続データストリームにおけるリレーションに対する値ベースのウィンドウ

(57) 【特許請求の範囲】

【請求項 1】

コンピュータで実行される方法であって、
 アプリケーションに関連する連続入力データストリームを受信することと、
 前記連続入力データストリームから入力リレーションを生成することとを含み、
 前記入力リレーションは、前記連続入力データストリームのデータレコードの有界集合
 であり、方法はさらに、
 前記入力リレーションを識別する連続クエリを受信することと、
 前記入力リレーションの処理に関連付けられるコンフィギュラブルウィンドウ演算子を
 識別することと、
 前記入力リレーションの属性を前記コンフィギュラブルウィンドウ演算子に適用するこ
 とによって前記連続クエリを実行して出力リレーションを生成することとを含み、
 前記属性は、前記入力リレーションに関連付けられたイベントの特性を含み、前記方法
 は、さらに、
 前記コンフィギュラブルウィンドウ演算子によって定義された、前記入力リレーシ
 ョンに前記イベントが挿入された時間から特定時間範囲内において前記イベントが発生したか
 否かを判断することと、
 前記イベントが前記特定時間範囲内に発生したか否かの判断結果に応じて、前記出力リ
 レーションに対して前記イベントを挿入するか、または、前記出力リレーションから前記
 イベントを取り除くことと、

10

20

前記出力リレーションのデータレコードを提供することを含み、

前記出力リレーションは、前記イベントが前記特定時間範囲内に発生したことに基づいて挿入された当該イベントを含む、コンピュータで実行される方法。

【請求項 2】

前記入力リレーションは、過去データのデータベースに記憶される前記アプリケーションに関連する情報に少なくとも部分的に基づいて生成される外部データソースである、請求項 1 に記載のコンピュータで実行される方法。

【請求項 3】

前記入力リレーションは、前記アプリケーションに関連する 1 つ以上のアーカイブされたリレーションに少なくとも部分的に基づいて生成される外部データソースである、請求項 1 または 2 に記載のコンピュータで実行される方法。

【請求項 4】

前記データレコードを提供することは、前記出力リレーションを表示することを含み、前記出力リレーションは、前記コンフィギュラブルウィンドウ演算子によって定義される特定の範囲内に属性値がある前記入力リレーションからの前記データレコードの部分集合を含む、請求項 1 ~ 3 のいずれか 1 項に記載のコンピュータで実行される方法。

【請求項 5】

1 つ以上のプロセッサによって実行可能な複数の命令を含むコンピュータ読取可能プログラムであって、前記複数の命令は、

前記 1 つ以上のプロセッサに、アプリケーションに関連する連続入力データストリームを受信させる命令と、

前記 1 つ以上のプロセッサに、前記連続入力データストリームから入力リレーションを生成させる命令と、

前記 1 つ以上のプロセッサに、前記入力リレーションを識別する連続クエリを受信させる命令と、

前記 1 つ以上のプロセッサに、前記入力リレーションの処理に関連付けられるコンフィギュラブルウィンドウ演算子を識別させる命令と、

前記 1 つ以上のプロセッサに、前記入力リレーションの属性を前記コンフィギュラブルウィンドウ演算子に適用することによって前記連続クエリを実行して出力リレーションを生成させる命令とを含み、

前記属性は、前記入力リレーションに関連付けられたイベントの特性を含み、前記複数の命令は、さらに、

前記 1 つ以上のプロセッサに、前記コンフィギュラブルウィンドウ演算子によって定義された、前記入力リレーションに前記イベントが挿入された時間から特定時間範囲内において前記イベントが発生したか否かを判断させる命令と、

前記 1 つ以上のプロセッサに、前記イベントが前記特定時間範囲内に発生したか否かの判断結果に応じて、前記出力リレーションに対して前記イベントを挿入するか、または、前記出力リレーションから前記イベントを取り除くかを判断させる命令と、

前記 1 つ以上のプロセッサに、前記出力リレーションのデータレコードを提供させる命令とを含み、

前記出力リレーションは、前記イベントが前記特定時間範囲内に発生したことに基づいて挿入された当該イベントを含む、コンピュータ読取可能プログラム。

【請求項 6】

システムであって、

複数の命令を記憶するメモリーと、

前記メモリーにアクセスするように構成される 1 つ以上のプロセッサとを備え、前記 1 つ以上のプロセッサは、前記複数の命令を実行するようにさらに構成され、前記複数の命令は少なくとも、

アプリケーションに関連する連続入力データストリームを受信し、

連続入力データストリームから入力リレーションを生成し、

10

20

30

40

50

前記入力リレーションを識別する連続クエリを受信し、
前記入力リレーションの処理に関連付けられるコンフィギュラブルウィンドウ演算子を識別し、

前記入力リレーションの属性を前記コンフィギュラブルウィンドウ演算子に適用することによって前記連続クエリを実行して出力リレーションを生成する命令を含み、

前記属性は、前記入力リレーションに関連付けられたイベントの特性を含み、前記複数の命令は、さらに、

前記コンフィギュラブルウィンドウ演算子によって定義された、前記入力リレーションに前記イベントが挿入された時間から特定時間範囲内において前記イベントが発生したか否かを判断し、

10

前記イベントが前記特定時間範囲内に発生したか否かの判断結果に応じて、前記出力リレーションに対して前記イベントを挿入するか、または、前記出力リレーションから前記イベントを取り除き、

前記出力リレーションのデータレコードを提供する命令を含み、

前記出力リレーションは、前記イベントが前記特定時間範囲内に発生したことに基づいて挿入された当該イベントを含む、システム。

【請求項 7】

前記 1 つ以上のプロセッサは、前記出力リレーションを表示するようにさらに構成され、前記出力リレーションは、前記コンフィギュラブルウィンドウ演算子によって定義される特定の範囲内に属性値がある前記入力リレーションからの前記データレコードの部分集合を含む、請求項 6 に記載のシステム。

20

【発明の詳細な説明】

【背景技術】

【0001】

背景

従来のデータベースシステムにおいて、データは、通常はテーブルの形態である 1 つ以上のデータベースに記憶される。そして、記憶されるデータは、構造化照会言語 (SQL) などのデータ管理言語を使用して照会および操作される。たとえば、SQL クエリは、データベースに記憶されるデータから関連するデータを識別するために定義および実行され得る。したがって、SQL クエリは、データベースに記憶されるデータの有限集合に対して実行される。さらに、SQL クエリが実行される時、それはひとたび有限データ集合に対して実行され、有限の静的結果 (finite static result) を作成する。したがって、データベースは、有限の記憶されるデータ集合に対してクエリを実行するように最良に実装される。

30

【0002】

しかしながら、いくつかの最新のアプリケーションおよびシステムは、有限のデータ集合の代わりに、連続的なデータもしくはイベントのストリームの形態のデータを生成する。このようなアプリケーションの例としては、限定されるものではないが、センサーデータアプリケーション、株式相場表示装置、ネットワーク性能測定ツール (たとえば、ネットワーク監視およびトラフィック管理アプリケーション)、クリックストリーム分析ツール、自動車トラフィック監視などが含まれる。このようなアプリケーションは、データストリームを処理することができる新しい種類のアプリケーションの必要を生じさせた。たとえば、温度センサーは、温度測定値を発信するように構成され得る。

40

【0003】

これらのタイプのイベントストリームベースのアプリケーションについてのデータの管理および処理では、厳密な時間によるフォーカスを使用してデータ管理およびクエリ機能が作成される。連続的なデータの非有界集合に対する長期実行クエリ (long-running queries) を含む、異なる種類の照会メカニズムが必要である。現在、一部のベンダーはイベントストリーム処理を目的とした製品スーツを提供しているが、提供されるこれらの製品は、今日のイベント処理の需要に対処するために必要な処理の柔軟性に依然として欠けて

50

いる。

【発明の概要】

【課題を解決するための手段】

【0004】

簡単な概要

一部の例において、リレーションに対する値ベースのウィンドウを管理するための方法が提供される。方法は、入力リレーションを識別する連続クエリを受信することを含み得る。1つの例において、入力リレーションは、アプリケーションに関連するデータレコードの有界集合である。そして、方法は、入力リレーションの処理に関連付けられるコンフィギュラブルウィンドウ演算子 (configurable window operator) を識別することを含み得る。加えて、方法は、出力リレーションを生成するために少なくとも部分的にコンフィギュラブルウィンドウ演算子に基づいて連続クエリを実行することを含み得る。一部の局面において、方法は、連続クエリの実行に少なくとも部分的に基づいて出力リレーションのデータレコードを提供することを含み得る。

10

【0005】

一部の例において、入力リレーションは、アプリケーションに関連する受信中の連続入力データストリームに少なくとも部分的に基づいて生成される外部データソースであり得る。加えて、一部の例において、入力リレーションは、過去データのデータベースに記憶されるアプリケーションに関連する情報に少なくとも部分的に基づいて生成される外部データソースであり得る。入力リレーションは、アプリケーションに関連する1つ以上のアーカイブされたリレーションに少なくとも部分的に基づいて生成される外部データソースでもあり得る。

20

【0006】

一部の例において、コンフィギュラブルウィンドウ演算子は、入力リレーションに対して定義されるジェネリック値ウィンドウ演算子であり得て、方法は、出力リレーションを生成するために入力リレーションにおける属性に対してジェネリック値ウィンドウ演算子を適用することを含み得る。他の例において、コンフィギュラブルウィンドウ演算子は、入力リレーションに対して定義される現在時間値ウィンドウ演算子であり得て、方法は、出力リレーションを生成するために入力リレーションにおける属性に対して現在時間値ウィンドウ演算子を適用することを含み得る。加えて、一部の例において、コンフィギュラブルウィンドウ演算子は、入力リレーションに対して定義される現在期間値ウィンドウ演算子であり得て、方法は、出力リレーションを生成するために入力リレーションにおける属性に対して現在期間値ウィンドウ演算子を適用することを含み得る。

30

【0007】

一部の局面において、方法は、出力リレーションを表示することを含み得る。一部の例において、出力リレーションは、コンフィギュラブルウィンドウ演算子によって定義される特定の範囲内に属性値がある入力リレーションからのデータレコードの部分集合を含み得る。

【0008】

一部の例において、非一時的コンピューター読取可能媒体が提供され得る。媒体は、1つ以上のプロセッサによって実行可能な複数の命令を含み得る。命令は、一部の例において、入力リレーションを生成することと、入力リレーションを処理するように構成されるクエリを識別することとを含み得る。加えて、命令は、入力リレーションの処理に関連付けられるコンフィギュラブルウィンドウ演算子を識別することと、出力リレーションを生成するためにコンフィギュラブルウィンドウ演算子に少なくとも部分的に基づいてクエリを実行することとを含み得る。一部の局面において、命令は、出力リレーションを表示することによって連続クエリの実行に少なくとも部分的に基づいて出力リレーションのデータレコードを提供することを含み得る。一部の例において、出力リレーションは、コンフィギュラブルウィンドウ演算子によって定義される特定の範囲内に属性値がある入力リレーションからのデータレコードの部分集合を含み得る。

40

50

【 0 0 0 9 】

一部の例において、システムが提供され得る。システムは、メモリーと、メモリーにアクセスし、入力リレーションを生成する命令を実行するように構成される1つ以上のプロセッサとを含み得る。命令は、入力リレーションを処理するように構成されるクエリを識別し、入力リレーションの処理に関連付けられるコンフィギュラブルウィンドウ演算子を識別するために実行され得る。加えて、命令は、出力リレーションを生成するためにコンフィギュラブルなウィンドウ演算子に少なくとも部分的に基づいてクエリを実行するために実行され得る。

【 0 0 1 0 】

一部の例において、コンピューター実行装置が提供され得て、装置は、入力リレーションを生成する手段を含み、入力リレーションは、アプリケーションに関連するデータレコードの有界集合であり、装置はさらに、入力リレーションを識別する連続クエリを受信する手段と、入力リレーションの処理に関連付けられるコンフィギュラブルウィンドウ演算子を識別する手段と、出力リレーションを生成するためにコンフィギュラブルウィンドウ演算子に少なくとも部分的に基づいて連続クエリを実行する手段と、連続クエリの実行に少なくとも部分的に基づいて出力リレーションのデータレコードを提供する手段とを含む。

10

【 0 0 1 1 】

一部の例において、入力リレーションは、アプリケーションに関連する受信中の連続的な入力データストリームに少なくとも部分的に基づいて生成される外部データソースであり得る。

20

【 0 0 1 2 】

一部の例において、入力リレーションは、過去データのデータベースに記憶されるアプリケーションに関連する情報に少なくとも部分的に基づいて生成される外部データソースであり得る。

【 0 0 1 3 】

一部の例において、入力リレーションは、アプリケーションに関連する1つ以上のアーカイブされたリレーションに少なくとも部分的に基づいて生成される外部データソースであり得る。

【 0 0 1 4 】

一部の例において、コンフィギュラブルウィンドウ演算子は、入力リレーションに対して定義されるジェネリック値ウィンドウ演算子であり得て、連続クエリを実行する手段は、出力リレーションを生成するために入力リレーションにおける属性に対してジェネリック値ウィンドウ演算子を適用する手段を含む。

30

【 0 0 1 5 】

一部の例において、コンフィギュラブルウィンドウ演算子は、入力リレーションに対して定義される現在時間値ウィンドウ演算子であり得て、連続クエリを実行する手段は、出力リレーションを生成するために入力リレーションにおける属性に対して現在時間値ウィンドウ演算子を適用する手段を含む。

【 0 0 1 6 】

一部の例において、コンフィギュラブルウィンドウ演算子は、入力リレーションに対して定義される現在期間値ウィンドウ演算子であり得て、連続クエリを実行する手段は、出力リレーションを生成するために入力リレーションにおける属性に対して現在期間値ウィンドウ演算子を適用する手段を含む。

40

【 0 0 1 7 】

一部の例において、データレコードを提供する手段は、出力リレーションを表示する手段を含み、出力リレーションは、コンフィギュラブルウィンドウ演算子によって定義される特定の範囲内に属性値がある入力リレーションからのデータレコードの部分集合を含む。

【 0 0 1 8 】

50

一部の例において、サービスプロバイダーデバイス(1101)が提供され得て、サービスプロバイダーデバイス(1101)は、入力リレーションを識別する連続クエリを受信するように構成される入力リレーションユニット(1102)を含み、入力リレーションは、アプリケーションに関連するデータレコードの有界集合であり、サービスプロバイダーデバイス(1101)はさらに、入力リレーションの処理に関連付けられるコンフィギュラブルウィンドウ演算子を識別するように構成されるコンフィギュラブルウィンドウ演算子ユニット(1103)と、出力リレーションを生成するためにコンフィギュラブルウィンドウ演算子に少なくとも部分的に基づいて連続クエリを実行するように構成される出力リレーションユニット(1105)と、連続クエリの実行に少なくとも部分的に基づいて出力リレーションのデータレコードを提供するように構成される提供ユニット(1107)とを含む。

10

【0019】

一部の例において、入力リレーションは、アプリケーションに関連する受信中の連続的な入力データストリームに少なくとも部分的に基づいて生成される外部データソースであり得る。

【0020】

一部の例において、入力リレーションは、過去データのデータベースに記憶されるアプリケーションに関連する情報に少なくとも部分的に基づいて生成される外部データソースであり得る。

【0021】

20

一部の例において、入力リレーションは、アプリケーションに関連する1つ以上のアーカイブされたりリレーションに少なくとも部分的に基づいて生成される外部データソースであり得る。

【0022】

一部の例において、コンフィギュラブルウィンドウ演算子は、入力リレーションに対して定義されるジェネリック値ウィンドウ演算子であり得て、出力リレーションユニット(1105)は、出力リレーションを生成するために入力リレーションにおける属性に対してジェネリック値ウィンドウ演算子を適用するようにさらに構成され得る。

【0023】

一部の例において、コンフィギュラブルウィンドウ演算子は、入力リレーションに対して定義される現在時間値ウィンドウ演算子であり得て、出力リレーションユニット(1105)は、出力リレーションを生成するために入力リレーションにおける属性に対して現在時間値ウィンドウを適用するようにさらに構成され得る。

30

【0024】

一部の例において、コンフィギュラブルウィンドウ演算子は、入力リレーションに対して定義される現在期間値ウィンドウ演算子であり得て、出力リレーションユニット(1105)は、出力リレーションを生成するために入力リレーションにおける属性に対して現在期間値ウィンドウ演算子を適用するようにさらに構成され得る。

【0025】

一部の例において、提供ユニット(1107)は、出力リレーションを表示するようにさらに構成され得て、出力リレーションは、コンフィギュラブルウィンドウ演算子によって定義される特定の範囲内に属性値がある入力リレーションからのデータレコードの部分集合を含む。

40

【0026】

一部の例において、コンピューター実行装置が提供され得て、装置は、入力リレーションを生成する手段と、入力リレーションを処理するためにクエリを識別する手段と、入力リレーションの処理に関連付けられるコンフィギュラブルウィンドウ演算子を識別する手段と、出力リレーションを生成するためにコンフィギュラブルウィンドウ演算子に少なくとも部分的に基づいてクエリを実行する手段とを備える。

【0027】

50

一部の例において、入力リレーションは、アプリケーションに関連する受信中の連続的な入力データストリームに少なくとも部分的に基づいて生成される外部データソースであり得る。

【0028】

一部の例において、入力リレーションは、過去データのデータベースに記憶されるアプリケーションに関連する情報に少なくとも部分的に基づいて生成される外部データソースであり得る。

【0029】

一部の例において、入力リレーションは、アプリケーションに関連する1つ以上のアーカイブされたリレーションに少なくとも部分的に基づいて生成される外部データソースであり得る。

10

【0030】

一部の例において、コンフィギュラブルウィンドウ演算子は、入力リレーションに対して定義されるジェネリック値ウィンドウ演算子であり得て、連続クエリを実行する手段は、出力リレーションを生成するために入力リレーションにおける属性に対してジェネリック値ウィンドウ演算子を適用するための手段を含む。

【0031】

一部の例において、装置は、連続クエリの実行に少なくとも部分的に基づいて出力リレーションのデータレコードを提供するための手段をさらに含み得る。

【0032】

20

一部の例において、装置は、出力リレーションを表示するための手段をさらに含み得て、出力リレーションは、コンフィギュラブルウィンドウ演算子によって定義される特定の範囲内に属性値がある入力リレーションからのデータレコードの部分集合を含む。

【0033】

一部の例において、サービスプロバイダーデバイス(1201)が提供され得て、サービスプロバイダーデバイス(1201)は、入力リレーションを処理するように構成されるクエリを受信および識別するように構成される入力リレーションユニット(1202)と、入力リレーションの処理に関連付けられるコンフィギュラブルウィンドウ演算子を識別するように構成されるコンフィギュラブルウィンドウ演算子ユニット(1203)と、出力リレーションを生成するためにコンフィギュラブルウィンドウ演算子に少なくとも部分的に基づいてクエリを実行するように構成される出力リレーションユニット(1205)とを含む。

30

【0034】

一部の例において、入力リレーションは、アプリケーションに関連する受信中の連続入力データに少なくとも部分的に基づいて生成される外部データソースであり得る。

【0035】

一部の例において、入力リレーションは、過去データのデータベースに記憶されるアプリケーションに関連する情報に少なくとも部分的に基づいて生成される外部データソースであり得る。

【0036】

40

一部の例において、入力リレーションは、アプリケーションに関連する1つ以上のアーカイブされたリレーションに少なくとも部分的に基づいて生成される外部データソースであり得る。

【0037】

一部の例において、コンフィギュラブルウィンドウ演算子は、入力リレーションに対して定義されるジェネリック値ウィンドウ演算子であり得て、出力リレーションユニット(1205)は、出力リレーションを生成するために入力リレーションにおける属性に対してジェネリック値ウィンドウ演算子を適用するようにさらに構成され得る。

【0038】

一部の例において、サービスプロバイダーデバイス(1201)は、連続クエリの実行

50

に少なくとも部分的に基づいて出力リレーションのデータレコードを提供するように構成される提供ユニット(1207)をさらに含み得る。

【0039】

一実施形態において、提供ユニット(1207)は、出力リレーションを表示するようにさらに構成され得て、出力リレーションは、コンフィギュラブルウィンドウ演算子によって定義される特定の範囲内に属性値がある入力リレーションからのデータレコードの部分集合を含む。

【0040】

上記のものは、他の特徴および実施形態と合わせて、以下の明細書、請求項、および添付の図面を参照することによってより明らかとなる。

10

【0041】

詳細な説明が、添付の図面を参照して述べられる。図面において、参照番号の最も左の数字は、参照番号が最初に表われる図面を識別する。異なる図面において同じ参照番号が使用される場合、それは同様もしくは同一の要素を示す。

【図面の簡単な説明】

【0042】

【図1】リレーションに対する値ベースのウィンドウを管理するための技術が実施され得る、簡易化された例示的なシステムもしくは構成を示す図である。

【図2】リレーションに対する値ベースのウィンドウの管理のための特徴を説明するために用いられ得る簡易化されたブロック図である。

20

【図3】本開示の一実施形態に従う、入力リレーションに記憶される例示的な情報、およびコンフィギュラブルウィンドウ演算子を使用して入力リレーションが処理される場合における出力リレーションの生成を示す図である。

【図4】本開示の他の実施形態に従う、入力リレーションに記憶される例示的な情報、およびコンフィギュラブルウィンドウ演算子を使用して入力リレーションが処理される場合における出力リレーションの生成を示す図である。

【図5】本開示の他の実施形態に従う、入力リレーションに記憶される例示的な情報、およびコンフィギュラブルウィンドウ演算子を使用して入力リレーションが処理される場合における出力リレーションの生成を示す図である。

【図6】少なくとも1つの例に従う、本願明細書に記載されるリレーションに対する値ベースのウィンドウの管理の少なくとも一部の特徴を示す簡易化された処理フローを示す図である。

30

【図7】少なくとも1つの例に従う、本願明細書において記載されるリレーションに対する値ベースのウィンドウの管理の少なくとも一部の特徴を示す他の簡易化された処理フローを示す図である。

【図8】少なくとも1つの例に従う、本願明細書において記載されるリレーションに対する値ベースのウィンドウの管理の実施形態に従って使用され得るシステム環境の構成部分を示す簡易化されたブロック図である。

【図9】少なくとも1つの例に従う、本願明細書において記載されるリレーションに対する値ベースのウィンドウの管理の実施形態に従って使用され得るコンピューターシステムを示す簡易化されたブロック図である。

40

【図10】本発明の他の実施形態に従う、本願明細書において記載されるリレーションに対する値ベースのウィンドウの管理を実施するための処理を示す例示的なフロー図である。

【図11】本発明の特定の実施形態に従って使用され得るサービスプロバイダーデバイスを示す簡易化されたブロック図である。

【図12】本発明の特定の実施形態に従って使用され得るサービスプロバイダーデバイスを示す簡易化されたブロック図である。

【発明を実施するための形態】

【0043】

50

詳細な説明

以下の説明において、様々な実施形態が記載される。説明を目的として、実施形態についての完全な理解を提供するために、具体的な構成および詳細が述べられる。しかしながら、具体的な詳細がなくとも実施形態が実施され得ることは当業者にとって明らかである。さらにまた、周知の特徴は、記載される実施形態を不明瞭にしないために、省略もしくは簡易化され得る。

【0044】

一部のアプリケーションにおいて、データは、記憶される有限のデータセットではなく、連続的な非有界データストリームの形態を取り得る。このようなデータストリームの例には、金融アプリケーションにおける株式相場表示、ネットワーク監視およびトラフィック管理における性能測定、ウェブトラッキングおよびパーソナル化におけるログ記録もしくはクリックストリーム、センサーアプリケーションからのデータフィード、ファイアウォールベースのセキュリティにおけるネットワークパケットおよびメッセージ、ならびに電話通信におけるコール詳細記録などが含まれ得る。それらの連続的な特性により、これらのデータストリームは、典型的に、従来のワンタイムSQLクエリではなく連続クエリを使用して照会され得る。

【0045】

一般に、連続データストリーム（イベントストリームともいわれる）は、明示的な終点のない連続もしくは非有界の特性を有し得るデータもしくはイベントのストリームを含み得る。論理的に、イベントもしくはデータストリームは、データ要素（イベントともいわれる）のシーケンスであり得て、各データ要素は、関連付けられたタイムスタンプを有する。連続イベントストリームは、要素のバッグもしくはセット（ s , T ）として論理的に表わされ得て、ここで「 s 」はデータ部分を表わし、「 T 」は時間ドメインに属する。「 s 」部分は、概してタプルもしくはイベントといわれる。したがって、イベントストリームは、タイムスタンプを持つタプルもしくはイベントのシーケンスであり得る。

【0046】

一部の局面において、ストリームにおけるイベントに関連付けられるタイムスタンプは、クロックタイムと同等とされ得る。しかしながら、他の例において、イベントストリームにおけるイベントに関連付けられる時間は、アプリケーションドメインによって定義され得て、クロックタイムに対応しない場合があるが、たとえば、代わりにシーケンス番号によって表わされ得る。このため、イベントストリームにおけるイベントに関連付けられる時間情報は、数字、タイムスタンプ、または時間の概念を表わす他の情報によって表され得る。入力イベントストリームを受信するシステムについては、イベントは、タイムスタンプの増加順でシステムに到達する。同じタイムスタンプを有する2つ以上のイベントがあり得る。

【0047】

一部の例において、イベントストリームにおけるイベントは、いくぶん世俗的なイベント（たとえば、温度センサーが値を新しい値に変更した場合や、株式表示記号の価格が変化した場合など）の発生を表わし得て、イベントに関連付けられる時間情報は、データストリームイベントによって表わされる世俗的なイベントが発生した場合を示し得る。

【0048】

イベントストリームを介して受信したイベントについては、イベントストリームにおけるイベントが確実にタイムスタンプ値の増加順で到達するように、イベントに関連付けられる時間情報が使用され得る。これにより、イベントストリームにおいて受信されたイベントを、それらに関連付けられる時間情報に基づいて順序付けることができる。この順序付けを可能にするために、後に生成されるイベントが前に生成されるイベントよりも後のタイムスタンプを有するように、タイムスタンプがイベントストリームにおけるイベントに対して非減少の態様で関連付けられ得る。他の例として、シーケンス番号が時間情報として使用されている場合、後に生成されるイベントに関連付けられるシーケンス番号は、前に生成されるイベントに関連付けられるシーケンス番号よりも大きくなり得る。一部の

例において、たとえばデータストリームイベントによって表される世俗的なイベントが同時に発生した場合に、複数のイベントが、同じタイムスタンプもしくは同じシーケンス番号に関連付けられ得る。同じイベントストリームに属するイベントは、概して、関連付けられる時間情報によってイベントに対して加えられる順序で処理され得て、前のイベントは後のイベントの前に処理される。

【 0 0 4 9 】

イベントストリームにおけるイベントに関連付けられる時間情報（たとえば、タイムスタンプ）は、ストリームのソースによって設定され得る、または代替的にストリームを受信するシステムによって設定され得る。たとえば、特定の实施形態において、イベントストリームを受信するシステムに対してハートビートが維持され得て、イベントに関連付けられる時間は、ハートビートによって測定されるようにシステムへのイベントの到達の時間に基き得る。イベントストリームにおける2つのイベントが同じ時間情報を有することは可能である。なお、タイムスタンプ順序付け要件は1つのイベントストリームに対して特定のものであるが、異なるストリームのイベントが適宜インターリーブされ得る。

【 0 0 5 0 】

イベントストリームは、関連付けられるスキーマ「S」を有し、スキーマは、時間情報と、1つ以上の指定される名前付き属性の集合とを含む。特定のイベントストリームに属するすべてのイベントは、その特定のイベントストリームに関連付けられるスキーマに準拠する。このため、イベントストリーム（s, T）については、イベントストリームは、スキーマ「S」を(<time_stamp>, <attribute(s)>)として有し得て、ここで<attributes>は、スキーマのデータ部分を表わし、1つ以上の属性を含み得る。たとえば、株式相場表示装置イベントストリームについてのスキーマは、<stock symbol>および<stock price>の属性を含み得る。このようなストリームを介して受信される各イベントは、タイムスタンプと2つの属性とを有する。たとえば、株式相場表示装置イベントストリームは、以下のイベントおよび関連付けられるタイムスタンプを受け取り得る。

【 0 0 5 1 】

```
(<timestamp_N>, <NVDA, 4>)
(<timestamp_N+1>, <ORCL, 62>)
(<timestamp_N+2>, <PCAR, 38>)
(<timestamp_N+3>, <SPOT, 53>)
(<timestamp_N+4>, <PDCO, 44>)
(<timestamp_N+5>, <PTEN, 50>)
```

上記のストリームにおいて、ストリーム要素(<timestamp_N+1>, <ORCL, 62>)については、イベントは、「stock_symbol」および「stock_value」を伴う<ORCL, 62>である。ストリーム要素に関連付けられるタイムスタンプは、「timestamp_N+1」である。したがって、連続イベントストリームはイベントのフローであり、各イベントは同じ一連の属性を有する。

【 0 0 5 2 】

記載したように、ストリームはCQLクエリが作用し得るデータの主要なソースであり得る。ストリームSは、要素（s, T）のバッグ（「マルチセットともいわれる」）であり得て、ここで「s」はSのスキーマに属し、「T」は時間ドメインに属する。加えて、ストリーム要素は、タプルとタイムスタンプのペアであり得て、タイムスタンプを持つタプル挿入のシーケンスとして表わされ得る。言い換えると、ストリームは、タイムスタンプを持つタプルのシーケンスであり得る。一部の場において、同じタイムスタンプを有する2つ以上のタプルがあり得る。また、入力ストリームのタプルは、タイムスタンプの増加順でシステムに到達する必要がある。代替的に、リレーション（「時間で変化するリレーション」ともいわれ、リレーションアルデータベースからのデータを含み得る「リレーションアルデータ」と混同されない）は、時間ドメインからスキーマRのタプルの非有界バッグへのマッピングであり得る。一部の例において、リレーションは、順序付けされていない、時間で変化するタプルのバッグ（すなわち、瞬間的なリレーション）であり得る

。一部の例において、時間の各瞬間では、リレーションは有界集合であり得る。これは、挿入、消去、および/または更新を含んでリレーションの変化する状態を表わし得る、タイムスタンプを持つタプルのシーケンスとしても表わされ得る。ストリームと同様に、リレーションは、リレーションの各タプルが準拠し得る固定スキーマを有し得る。さらに、本願明細書で使用される連続クエリは、概してストリームおよび/またはリレーションのデータを処理する(すなわち、照会する)ことが可能であり得る。加えて、リレーションは、ストリームのデータを参照し得る。

【0053】

一部の例において、ビジネスインテリジェンス(BI)は、特定の間隔で(たとえば、一部の例においては毎日)事業実施の運営および最適化を補助し得る。このタイプのBIは、通常、オペレーショナルビジネスインテリジェンス、リアルタイムビジネスインテリジェンス、またはオペレーショナルインテリジェンス(OI)と呼ばれる。オペレーショナルインテリジェンスは、一部の例において、BIと事業活動の監視(BAM)との線引きを不明瞭にする。たとえば、BIは、過去データの周期的なクエリにフォーカスし得る。このようなことから、BIは、後向きフォーカスを有し得る。しかしながら、BIは、実施用途にも利用され得て、このため、単なる戦略的分析ツールから事業実施における前線へ拡大し得る。このようなことから、BIシステムは、イベントストリームを分析し、リアルタイムで集合を演算するようにも構成され得る。

【0054】

一部の例において、連続クエリ言語サービス(CQサービス)は、連続クエリを取り扱うとともにリアルタイムの警告を可能にするように、BI分析サーバーを拡張するように構成され得る。CQサービスは、一部の局面において、BI分析サーバーおよびCQLエンジンとの統合を提供し得る。例示のみであるが、BI分析サーバーは、連続クエリをCQサービスに転付し得て、CQサービスは、CQLエンジンのための論理データベース(DB)ゲートウェイとしても作用し得る。この方法により、CQLエンジンは、その分析能力およびセマンティックモデリングのために、BI分析サーバーを活用することが可能となり得る。

【0055】

一部の例において、CQサービスは、とりわけ、以下の機能を提供し得る。

・CQLエンジンゲートウェイとしてBI分析サーバーのためのリモーティングサービス

- ・イベントソース/シンクアダプター
- ・論理SQLにCQL拡張を加えたものからのデータ定義言語(DDL)の生成
- ・すべてのタイプの連続クエリおよび実施選択のための統一モデルの提供
- ・メタデータおよびサポート再開可能性の維持
- ・高い利用可能性およびスケーラビリティのサポート

加えて、一部の例において、OIは、事業実施に対して可視性および見識をもたらすリアルタイムの動的事業分析の形態である。大量の情報から理解可能とすることを助けるといふ点において、OIは、BIもしくはリアルタイムBIに対してリンクもしくは比較されることが多い。しかしながら、いくつかの基本的な違いがある。OIは、主に活動中心であり得て、BIは主にデータ中心であり得る。加えて、OIは、進展する状況(たとえば、傾向およびパターン)を検知する、およびこれに対応するのにより適し得て、これに対してBIは、事後ベースおよび報告ベースの手法として従来より使用され得る。

【0056】

一部の例において、事業イベント分析および監視(BEAM)システムは、インフライトデータを処理および/または受信するためにCQLエンジンを含み得る。たとえば、CQLエンジンは、受信中のリアルタイムの情報を照会もしくはそれ以外に処理するように構成されるインメモリーリアルタイムイベント処理エンジンであり得る(たとえば、BIまたはOI)。CQLエンジンは、一時セマンティクスを利用または理解し得るとともに、データのウィンドウの定義を処理することを可能にするように構成され得る。CQLエ

10

20

30

40

50

ンジンを利用することは、一部の例において、受信中のデータに対して常にクエリを実行することを伴う。

【 0 0 5 7 】

一部の局面において、CQLエンジンは、完全に発達したクエリ言語を含み得る。このようなことから、ユーザーは、クエリに関して演算を特定し得る。加えて、CQLエンジンは、メモリーの最適化、クエリ言語特性の利用、演算子の共有、リッチパターンマッチング、リッチ言語構成などのために設計され得る。加えて、一部の例において、CQLエンジンは、過去データおよびストリーミングデータの両方を処理し得る。たとえば、ユーザーは、カリフォルニアの販売が特定のターゲットを超えた時に警告を送るようにクエリを設定し得る。したがって、一部の例において、警告は、過去の販売データおよび受信中のライブ（すなわち、リアルタイム）販売データに少なくとも部分的に基づき得る。

10

【 0 0 5 8 】

一部の例において、CQLエンジンまたは以下に記載されるコンセプトの他の特徴は、リアルタイムで過去のコンテキスト（すなわち、ウェアハウスデータ）と受信中のデータとを結合するように構成され得る。したがって、一部の例において、本開示は、データベースに記憶される情報とインフライト情報との間の境界を記載し得る。データベースに記憶される情報およびインフライト情報の両方は、BIデータを含み得る。このようなことから、データベースは、一部の例において、BIサーバーであり得る、または任意のタイプのデータベースであり得る。さらに、一部の例において、本開示の特徴は、ユーザーがどのようにプログラムするのか、またはそれ以外にどのようにコードを書くのかを知らなくとも上記の特徴の実施を可能とし得る。言い換えると、特徴は、多機能ユーザーインターフェイス（UI）において提供され得る、または過去データとリアルタイムのデータとの結合を非開発者が実施することを可能とする態様で提供され得る。

20

【 0 0 5 9 】

一部の例において、上記のコンセプトは、複合イベント処理に関連付けられるリッチリアルタイム連続イベント処理能力を活用するために利用され得る。限定されないが、アーカイブされたリレーションなどのいくつかの特徴がサポートされ得る。このようなことから、このような特徴（たとえば、リッチリアルタイム連続イベント処理）を活用するために、システムは、リレーショナルデータのスタートアップ状態およびランタイム状態に透明に対処するように構成され得る。言い換えると、システムは、作成の瞬間においては空でないクエリ（すなわち、アーカイブされたリレーション）を管理するように構成され得る。

30

【 0 0 6 0 】

一部の例において、アーカイブされたリレーションが利用され得る。このようなことから、アーカイブされたリレーションに基づくことを示すクエリをCQLエンジンが確認した場合、そのアーカイブされたリレーションは、たとえば、過去のコンテキストについてクエリを呼び出すことができる特定のエンティティがあることを示し得る。一部の例において、データ定義言語（DDL）は、照会をどのように行なうか、テーブル内の重要なカラムは何か、および/または残りのデータをどこへ送信するかなど、アーカイブされたリレーションについての注釈を示し得るが、これらに限定されない。一部の例において、ひとたびクエリがCQLエンジンにおいて構築されると（たとえば、グラフとして）、システムはクエリグラフを分析し得る。加えて、一部の局面において、「distinct」、「group agg」、「pattern」、および/または「group by」のような、ステートフルな特定の演算子がある。しかしながら、ステートレスな演算子は、入力を取り、それをたとえば下流の演算子などの親に送るのみであり得る。このため、1つの手法は、このテーブル全体をここに記憶することである。しかしながら、アーカイブされたリレーションを利用することにより、システムは、クエリグラフを分析し、アーカイブを照会するために使用することができる最低のステートフル演算子がどれかを決定し得る。一部の例において、システム（または1つ以上のコンピューター実行方法）は、グラフを横断しながら到達した最低のステートフル演算子において状態を検索する。たとえば、クエリグラフは、ソースか

40

50

らのトポロジー順序において分析され得る。この第1のステートフル演算子に少なくとも部分的に基づき、CQLエンジンは、アーカイブされたりレーションに対して定義されるクエリについての演算子の状態を初期化するために、取得されるデータの最適量を判定し得る。

【0061】

少なくとも1つの非限定的な例において、リレーションおよび/またはソースのようなソース演算子は、トポロジー横断線において最初に置かれ得て、クエリ出力および/またはルートは最後に置かれ得る。たとえば、CQLクエリが、`select sum(c1) from R1 where c2>c25`のような場合、このクエリについてのプランは、RelationSource SELECT GroupAggrのようになり得る。したがって、トポロジー順序に従うとともに、RelationSource およびSELECTが両方ともステートレスであることから、最低のステートフル演算子はGroupAggrであり得る。この方法により、クエリのステートフル演算子（この例では、GroupAggr）は、クエリエンジンがストリーミングデータを受信する前にデータストアから過去データをクエリエンジンに集合させることを可能にし得る。これは、アーカイブされたりレーションをクエリが分析しており、そのアーカイブされたりレーションがそのように示しているという事実少なくとも部分的に基づいて可能となり得る。

【0062】

一部の例において、所与のアーカイブリレーションのウィンドウサイズは、ユーザーによって指定され得る。一部の局面において、ウィンドウは、アーカイブされたりレーションに関連して、受信中のアクティブデータを分析もしくはそれ以外に評価するクエリグラフにおけるノードを含み得る。言い換えると、ウィンドウは、クエリエンジンによって分析および/または処理され得るアクティブデータの量、および/またはアーカイブされたりレーションに含まれる過去データの量を定義し得る。

【0063】

高いレベルにおいて、ひとたびウィンドウがStreamに対して適用されると、それはRelationとなり、リレーショナルデータベースと同様に、通常のリレーショナルロジックが適用され得る。タプルが到着してウィンドウを離れると、考慮中のRelationは、それに対してコンパイルされるクエリによって変化し、同時に結果を発する。CQLは、RANGE（ナノ秒粒度に至る）、ROWS、PARTITION BY、および拡張可能ウィンドウをサポートし得る。これらのウィンドウは、ストリームからリレーションへの演算子（stream-to-relation operators）の例である。他方、ISTREAM（すなわち、挿入ストリーム）、DSTREAM（すなわち、削除ストリーム）、およびRSTREAM（すなわち、リレーションストリームは、リレーションからストリームへの演算子（relation-to-stream operators）である。一部の例において、ユーザー、開発者、および/または管理者は、クエリエンジンまたはクエリエンジンを操作もしくはホスティングする1つ以上のコンピューティングシステムによって提供されるウィンドウサイズを（たとえば、UIを介して）設定し得る。一部の例において、ストリーム上のウィンドウは、時間ベースの範囲ウィンドウであり得る。たとえば、アーカイブされたりレーション上のコンフィギュラブル値ウィンドウは、ウィンドウサイズおよびウィンドウが計算される属性を使用して特定され得る。アーカイブされたりレーションの上部に特定されるコンフィギュラブル値ウィンドウがある場合、スナップショットクエリが演算され得て、ウィンドウ限界内のスナップショットタプルが出力され得る。加えて、状態の初期化後、値ウィンドウは、受信中のアクティブデータに対して適用され得る。一部の例において、ウィンドウ属性の値がウィンドウサイズよりも小さい現在イベント時間とは異なるウィンドウには受信中のアクティブデータのみが挿入される。

【0064】

上述のように、一部の例において、連続クエリを使用した連続データストリームの処理には、ストリームに対してウィンドウを適用し、全ストリーム履歴の代わりに最近のストリームタプルの固定数が考慮されるように特定すること、または全ストリーム履歴の代わりに特定の時間内に到着したストリームタプルのみが考慮されるように特定することを伴い得る。上でさらに記載されるように、一部の例において、連続クエリ処理には、受信中

10

20

30

40

50

の連続ストリームをリレーションに変換するよう処理する様々な演算子、または入力リレーションを出力ストリームデータに変換する演算子を使用することを伴い得る。

【0065】

しかしながら、特定の状況においては、ユーザーは、アプリケーションに関連する受信中のリアルタイムデータをリレーション（たとえば、外部データソース）に記憶されるアプリケーションに関連するデータを用いて処理することが必要であり得るクエリを設定し得る。リレーションのサイズが非常に大きい場合、このデータを照会するためにメモリーに持ち込む必要のあるデータの量もまた非常に大きくなり得る。CQLエンジンは、典型的に、受信中のリアルタイム情報を照会もしくは処理するように構成されるインメモリーリアルタイムイベント処理エンジンであり得ることから、有限のメモリーにおいてこのような高いレートの入力ストリームを大きなサイズのリレーションと結合するようにスケール可能とはなり得ない。

10

【0066】

このため、一部の実施形態において、ウィンドウ演算子は、連続クエリ（たとえば、CQLクエリ）におけるリレーションを処理するために定義され得る。1つの例において、ウィンドウ演算子は、ユーザーコンフィギュラブルであり得る。ウィンドウ演算子は、特定の時間の範囲に対してリレーションにおけるデータレコードの部分集合の処理を可能とし得る。このため、リレーションのデータレコードの部分集合は、メモリーにおけるリレーションのデータレコードのセットの全体を処理するのではなく、有限のメモリーにおいて照会され得る。

20

【0067】

特定の実施形態において、入力リレーションが生成され得る。1つの例において、入力リレーションは、アプリケーションに関連するデータレコードの有界集合を含む外部データソースであり得る。1つの例において、入力リレーションは、連続入力データストリームから生成され得る。代替的に、入力リレーションは、アプリケーションに関連する過去データのデータベースに基づいて生成され得る。加えて、入力リレーションは、1つ以上のアーカイブされたリレーションから最初に生成され得る、入力リレーションの残りは、受信中のストリーミングデータに基づいて生成され得る。

【0068】

本開示の一実施形態に従えば、入力リレーションを識別する連続クエリ（たとえば、CQLクエリ）が受信され得る。そして、入力リレーションに対して定義されるウィンドウ演算子が識別され得る。クエリは、出力リレーションを生成するためにウィンドウ演算子に少なくとも部分的に基づいて実行され得る。一実施形態において、出力リレーションは、入力リレーションにおけるウィンドウ演算子によって定義される特定の範囲内に属性値がある入力リレーションからのデータレコードの部分集合を含み得る。

30

【0069】

上記および下記の技術は、いくつかの方法およびいくつかの状況において実施され得る。以下においてより詳細に記載されるように、いくつかの例示的な実施および状況が以下の図面を参照して提供される。しかしながら、以下の実施および状況は多くのうちの一部である。

40

【0070】

図1は、リレーションに対する値ベースのウィンドウを管理するための技術が実施され得る、簡易化された例示的なシステムもしくはアーキテクチャ100を示す。アーキテクチャ100において、1人以上のユーザー102（たとえば、アカウント保持者）が、コンピューティングデバイス104（1）～104（N）（まとめて、「ユーザーデバイス104」）を利用し、1つ以上のネットワーク108を介して1つ以上のサービスプロバイダーコンピューター106にアクセスし得る。一部の局面において、サービスプロバイダーコンピューター106は、ネットワーク108を介して、1つ以上のストリーミングデータソースコンピューター110および/または1つ以上のデータベース112と通信し得る。たとえば、ユーザー102は、サービスプロバイダーコンピューター106を利

50

用し、ストリーミングデータソースコンピューター 110 および / またはデータベース 112 のデータにアクセスし得る、またはそれ以外にデータを管理し得る（たとえば、110 および 112 のうちのいずれかもしくは両方に対してクエリが実行され得る）。データベース 112 は、リレーショナルデータベースまたは SQL サーバーなどであり得て、一部の例においては、過去データ、イベントデータ、リレーション、またはアーカイブされたリレーションなどをユーザー 102 の代わりに管理し得る。加えて、データベース 112 は、ストリーミングデータソースコンピューター 110 によって提供されるデータを受信もしくはそれ以外に記憶し得る。一部の例において、ユーザー 102 は、ユーザーデバイス 104 を利用し、データ（たとえば、過去のイベントデータ、ストリーミングイベントデータなど）についてのクエリ（「クエリ宣言」ともいう）または他のリクエストを提供することによってサービスプロバイダーコンピューター 106 と対話し得る。そして、このようなクエリまたはリクエストは、サービスプロバイダーコンピューター 106 によって実行され、データベース 112 のデータおよび / またはストリーミングデータソースコンピューター 110 からの受信中のデータが処理され得る。さらに、一部の例において、ストリーミングデータソースコンピューター 110 および / またはデータベース 112 は、サービスプロバイダーコンピューター 106 に関連付けられる、統合された分散環境の一部であり得る。

【0071】

一部の例において、ネットワーク 108 は、ケーブルネットワーク、インターネット、無線ネットワーク、携帯電話ネットワーク、イントラネットシステム、ならびに / または他の私的および / もしくは公共ネットワークなど、複数の異なるタイプのネットワークのうちの 1 つもしくはこれらの組み合わせを含み得る。示される例は、ユーザー 102 がネットワーク 108 を介してサービスプロバイダーコンピューター 106 にアクセスすることを表わしているが、記載の技術は、ユーザー 102 が固定電話で 1 つ以上のユーザーデバイス 104 を介して、キオスクを介して、または他の方法により 1 つ以上のサービスプロバイダーコンピューター 106 と対話する場合においても等しく適用され得る。また、記載の技術は、他のクライアント / サーバー構成（たとえば、セットトップボックスなど）および非クライアント / サーバー構成（たとえば、ローカルに記憶されるアプリケーションなど）においても適用され得る。

【0072】

ユーザーデバイス 104 は、携帯電話、スマートフォン、パーソナルデジタルアシスタント（PDA）、ラップトップコンピューター、デスクトップコンピューター、シンクライアントデバイス、タブレット PC などの任意のタイプのコンピューティングデバイスであり得るが、これらに限定されない。一部の例において、ユーザーデバイス 104 は、ネットワーク 108 を介して、または他のネットワーク接続を介して、サービスプロバイダーコンピューター 106 と通信し得る。さらに、ユーザーデバイス 104 は、データベース 112（または他のデータストア）の処理されるデータを要求するための 1 つ以上のクエリもしくはクエリ宣言を提供するようにも構成され得る。

【0073】

一部の局面において、サービスプロバイダーコンピューター 106 は、モバイル、デスクトップ、シンクライアント、および / またはサーバーのようなクラウドコンピューティングデバイスなど、任意のタイプのコンピューティングデバイスでもあり得るが、これらに限定されない。一部の例において、サービスプロバイダーコンピューター 106 は、ネットワーク 108 を介して、または他のネットワーク接続を介してユーザーデバイス 104 と通信し得る。サービスプロバイダーコンピューター 106 は、場合によってはクラスターにおいて、サーバーファームとして、または互いに関連付けられていない個別のサーバーとして構成される 1 つ以上のサーバーを含み得る。これらのサーバーは、本願明細書において記載される CQL リレーションの管理、入力リレーションの生成、入力リレーションに関連付けられるコンフィギュラブルウィンドウ演算子、および出力リレーションの生成を含む、本願明細書に記載される特徴を行なう、もしくはホスティングするように構

10

20

30

40

50

成され得る。加えて、一部の局面において、サービスプロバイダーコンピューター 106 は、ストリーミングデータソースコンピューター 110 および / またはデータベース 112 を含む、統合された分散コンピューティング環境の一部として構成され得る。

【0074】

1つの例示的な構成において、サービスプロバイダーコンピューター 106 は、少なくとも 1つのメモリー 136 と 1つ以上の処理ユニット (プロセッサ) 138 とを含み得る。プロセッサ 138 は、ハードウェア、コンピューター実行可能命令、ファームウェア、またはこれらの組み合わせにおいて適切に実施され得る。プロセッサ 138 のコンピューター実行可能命令またはファームウェアの実施は、記載される様々な機能を行なうために任意の適したプログラミング言語で書かれたコンピューター実行可能命令もしくは

10

【0075】

メモリー 136 は、プロセッサ 138 上でローディング可能および実行可能なプログラム命令ならびにこれらのプログラムの実行時に生成されるデータを記憶し得る。サービスプロバイダーコンピューター 106 の構成およびタイプに応じて、メモリー 136 は、揮発性 (ランダムアクセスメモリー (RAM) など) および / または不揮発性 (読み取り専用メモリー (ROM)、フラッシュメモリーなど) であり得る。サービスプロバイダーコンピューター 106 またはサーバーは、追加の記憶部 140 も含み得て、この記憶部 140 は、リムーバブル記憶部および / または非リムーバブル記憶部を含み得る。追加の記憶部 140 は、磁気記憶部、光学ディスク、および / またはテープ記憶部を含み得るが、これらに限定されない。ディスクドライブおよびこれに関連付けられるコンピューター読取可能媒体は、コンピューター読取可能命令、データ構造、プログラムモジュール、およびコンピューティングデバイスのための他のデータの非揮発性記憶部を提供し得る。一部の実施において、メモリー 136 は、スタティックランダムアクセスメモリー (SRAM)、ダイナミックランダムアクセスメモリー (DRAM)、または ROM などの複数の異なるタイプのメモリーを含み得る。

20

【0076】

メモリー 136 ならびにリムーバブルおよび非リムーバブルの両方の追加の記憶部 140 は、すべてコンピューター読取可能記憶媒体の例である。たとえば、コンピューター読取可能記憶媒体は、コンピューター読取可能命令、データ構造、プログラムモジュール、もしくは他のデータなどの情報の記憶のための任意の方法もしくは技術において実施される、揮発性もしくは不揮発性のリムーバブルもしくは非リムーバブル媒体を含み得る。メモリー 136 および追加の記憶部 140 は、すべてコンピューター記憶媒体の例である。

30

【0077】

サービスプロバイダーコンピューター 106 は、記憶されるデータベース、他のコンピューティングデバイスもしくはサーバー、ユーザー端末、および / またはネットワーク 108 上の他のデバイスとサービスプロバイダーコンピューター 106 が通信することを可能にする通信接続 142 も含み得る。サービスプロバイダーコンピューター 106 は、キーボード、マウス、ペン、音声入力デバイス、タッチ入力デバイス、ディスプレイ、1つ以上のスピーカー、プリンターなどの入力 / 出力 (I/O) デバイス 144 も含み得る。

40

【0078】

より詳細にメモリー 136 の内容を見ると、メモリー 136 は、オペレーティングシステム 146 と、少なくとも入力リレーションモジュール 148、コンフィギュラブルウィンドウ演算子モジュール 150、および出力リレーションモジュール 152 を含む、本願明細書に開示される特徴を実施するための 1つ以上のアプリケーションプログラムもしくはサービスとを含み得る。本願明細書において使用されるモジュールは、サービスの一部であるサーバーもしくはサーバーのクラスターによって実行されるプログラミングモジュールを言い得る。この特定の文脈において、モジュールは、サービスプロバイダーコンピューター 106 の一部であるサーバーもしくはサーバーのクラスターによって実行され得る。

50

【 0 0 7 9 】

一部の例において、入力リレーションモジュール 1 4 8 は、1 つ以上の入力リレーションを受信、識別、生成、もしくは提供するように構成され得る。一実施形態において、入力リレーション 1 5 4 は、アプリケーションに関連するデータレコードの有界集合（たとえば、リレーションまたはアーカイブされたリレーション）を含む外部データソースである。

【 0 0 8 0 】

1 つの例において、入力リレーション 1 5 4 は、アプリケーションに関連する（たとえば、データソースコンピューター 1 1 0 からの）データもしくはイベントのストリームを含む受信中の連続入力データストリームに基づいて生成され得る外部データソースであり得て、ここで入力リレーションは、連続データストリームにおけるイベントストリームエントリー $s_1, s_2 \dots s_N$ のうちの 1 つ以上を含むデータレコードの有界集合を含み得る。他の例において、入力リレーション 1 5 4 は、過去データのデータベース（たとえば、データベース 1 1 2）に記憶されるアプリケーションに関連する情報に基づいて生成され得る外部データソースであり得て、入力リレーションは、Persistence または過去データのデータベースから事前にローディングされる 1 つ以上のイベントストリームエントリー（たとえば、 s_1 および / もしくは s_2 、またはそれより多いもしくは少ない）を含むデータレコードの有界集合を含み得る。

【 0 0 8 1 】

他の例において、入力リレーション 1 5 4 は、アプリケーションに関連する 1 つ以上のイベントストリームエントリー $s_1, s_2 \dots s_N$ への参照を含み得る 1 つ以上のアーカイブされたリレーションに基づいて生成され得る外部データソースであり得る。1 つの例において、入力リレーション 1 5 4 は、1 つ以上のアーカイブされたリレーションから最初に生成され得て、入力リレーションの残りは、受信中のストリーミングデータに基づいて生成され得る。

【 0 0 8 2 】

1 つの例において、入力リレーション 1 5 4 におけるデータレコードは、入力リレーション 1 5 4 の変化する状態を表わすタイムスタンプを持つタプルもしくはデータレコードのシーケンスを含み得る。一実施形態において、入力リレーション 1 5 4 における各データレコードは、以下のスキーマ「S」：(<time stamp>, <attribute(s)>) によって表され得るイベントストリームエントリーを含み得て、ここで <attributes> は、スキーマのデータ部分を表わし、1 つ以上の属性を含み得る。

【 0 0 8 3 】

コンフィギュラブルウィンドウ演算子モジュール 1 5 0 は、連続クエリにおける入力リレーションを処理するための 1 つ以上のコンフィギュラブルウィンドウ演算子 1 5 6 を定義するように構成され得る。入力リレーションについてコンフィギュラブルウィンドウ演算子が定義され得る様々な方法がある。一実施形態において、コンフィギュラブルウィンドウ演算子は、ユーザーコンフィギュラブルであり得る。特定の実施形態において、および以下に詳細に記載されるように、コンフィギュラブルウィンドウ演算子は、入力リレーションにおけるデータレコードのタイムスタンプ t に対してコンフィギュラブルウィンドウ演算子によって特定される範囲内に属性値がある入力リレーションからのデータレコードの部分集合を含む出力リレーションを作り得る。

【 0 0 8 4 】

一実施形態において、第 1 のタイプのコンフィギュラブルウィンドウ演算子は、入力リレーションに対する「ジェネリック値」ウィンドウ演算子として定義され得る。1 つの例において、「ジェネリック値」ウィンドウ演算子は、以下のように入力リレーション 1 5 4 に対するサブクローズとして表現され得る。

【 0 0 8 5 】

RANGE range_val ON attr.

特定の実施形態において、「ジェネリック値」ウィンドウ演算子は、入力リレーション

10

20

30

40

50

における対応するデータレコードのタイムスタンプ t に対する「range val」である特定の範囲内に属性「attr」値がある入力リレーションからのデータレコードの部分集合を含む出力リレーションを作り得る。CQLクエリにおける入力リレーションに対して「ジェネリック値」ウィンドウオペレーターが定義され得る方法の例示的な動作は以下に詳細に記載される。

【0086】

他の実施形態において、第2のタイプのコンフィギュラブルウィンドウ演算子は、入力リレーションに対する「現在時間値」ウィンドウ演算子として定義され得る。1つの例において、「現在時間値」ウィンドウ演算子は、以下のように入力リレーションに対するサブクローズとして表現され得る。

【0087】

CurrentHour on attr

一実施形態において、「現在時間値」ウィンドウ演算子は、入力リレーションにおける対応するデータレコードのタイムスタンプ t に対する「CurrentHour」である現在時間に属性「attr」値がある入力リレーションからのデータレコードの部分集合を含む出力リレーションを作り得る。CQLクエリにおける入力リレーションに対して「現在時間値」ウィンドウ演算子が定義され得る方法の例示的な動作は以下に詳細に記載される。

【0088】

他の実施形態において、第3のタイプのコンフィギュラブルウィンドウ演算子は、入力リレーションに対する「現在期間値」ウィンドウ演算子として定義され得る。1つの例において、「現在期間値」ウィンドウ演算子は、以下のように入力リレーションに対するサブクローズとして表現され得る。

【0089】

CurrentPeriod("t1","t2") on attr

一実施形態において、「現在期間値」ウィンドウ演算子は、入力リレーションにおける対応するデータレコードのタイムスタンプ t に対する「CurrentPeriod」である現在時間期間内に属性「attr」値がある入力リレーションからのデータレコードの部分集合を含む出力リレーションを作り得る。CQLクエリにおける入力リレーションに対して「CurrentPeriod」ウィンドウ演算子が定義され得る方法の例示的な動作は、以下により詳細に記載される。

【0090】

上の記載は、CQLクエリにおける入力リレーションに対するコンフィギュラブルウィンドウ演算子の3つのタイプの定義づけに関連するが、少なくとも一部の実施形態においては、追加のタイプのウィンドウ演算子が、CQLクエリにおける入力リレーションに対してコンフィギュラブルウィンドウ演算子モジュール150によって定義され得ることが理解される。

【0091】

出力リレーションモジュール152は、1つ以上の出力リレーション158を生成するように構成され得る。一実施形態において、出力リレーションモジュール152は、CQLエンジンおよび/またはCQサービスからのCQLクエリを受信するように構成され得る。特定の実施形態において、出力リレーションモジュール152は、CQLクエリにおける入力リレーションを識別し、入力リレーションに対して定義されるコンフィギュラブルウィンドウ演算子を識別し、コンフィギュラブルウィンドウ演算子に少なくとも部分的に基づいてCQLクエリを実行し、出力リレーションを生成し得る。代替的に、一部の例において、入力リレーションを含むクエリ（たとえば、CQLクエリ）が識別もしくは受信される場合、CQLエンジンおよび/またはCQサービスは、クエリ206をパースすることで、入力リレーションを処理して出力リレーションを生成する。一実施形態において、および以下に詳細に記載されるように、出力リレーションは、入力リレーションにおけるコンフィギュラブルウィンドウ演算子によって定義される特定の範囲内に属性値がある入力リレーションからのデータレコードの部分集合を含む。特定の実施形態において、

10

20

30

40

50

CQLクエリが受信され、メモリー内の出力リレーションにおけるデータレコードの部分集合が処理される。

【0092】

図2は、リレーションに対する値ベースのウィンドウの管理のための特徴を記載するのに用いられる簡易化されたブロック図を示す。示されるように、図2は、入力リレーション154を管理するためのCQLエンジンおよび/またはCQサービス200の少なくとも1つの実施について記載する。CQLエンジンおよび/またはCQサービス200は、入力ソース202からの情報を最初に受け取り得る。1つの例において、入力ソース202は、アプリケーションに関連するデータもしくはイベントのストリームを含む受信中の連続入力データを受信するデータソースコンピューター110を含み得る。1つの例において、CQLエンジンおよび/またはCQサービス200は、入力ソース202からのデータの表現であり得る入力リレーション154を識別し得る。一部の例において、および上記のように、入力ソース202は、アプリケーションに関連するデータレコードの有界集合を含み得る。特定の実施形態において、入力リレーション154を含むクエリ（たとえば、連続クエリ）が識別もしくは受信されると、CQLエンジン200はクエリをパースし、クエリにおけるコンフィギュラブルウィンドウ演算子156を識別するとともにコンフィギュラブルウィンドウ演算子に少なくとも部分的に基づいてクエリを実行し、出力リレーション158を生成する。一実施形態において、CQLエンジンおよび/またはCQサービス200は、入力リレーションにおける属性に対してコンフィギュラブルウィンドウ演算子を適用することによってクエリを実行し、出力リレーションを生成し得る。一部の例において、出力リレーションは、アプリケーションに関連するデータ（たとえば、データレコードの部分集合）の表現であり得る。そして、CQLエンジンおよび/またはCQサービス200は、たとえば図1に示されるデータベースなどの出力先204に出力リレーションを記憶し得る。

【0093】

図3は、本開示の一実施形態に従う、入力リレーションに記憶される例示的な情報、ならびにコンフィギュラブルウィンドウ演算子を使用して入力リレーションが処理される場合の出力リレーションの生成を示す。例として、以下のように生成され得る入力リレーション「CallCentreRecords」について考慮する。

【0094】

```
CREATE RELATION CallCentreRecords(callerName char(50), callTime, timestamp)
```

この定義により、入力リレーション「CallCentreRecords」は、第1の属性「callerName」および第2の属性「callTime」という2つの属性と、タイムスタンプとを含む。1つの例において、入力リレーション「CallCentreRecords」は、コールセンターからのコール詳細記録に関連する連続データストリームを記憶し得る。ここで使用される「コールセンター」は、典型的に、カスタマーサービスの代表者が電話を介してカスタマーにサービスを提供し得るサービスネットワークをいう。コールセンターからの連続データストリームは、典型的に、短い時間間隔で要約されるコールに関連するコールバイコール（call-by-call）情報を含み得る。このようなコールバイコール情報は、たとえば、データベースにコールが記録もしくは受信される場合のためのタイムスタンプ、呼出元名、コールが行なわれた時間などを含み得る。

【0095】

図3に示される例を参照すると、一実施形態において、入力リレーション「CallCenterRecords」302は、データレコードが入力リレーション302に挿入される時の各タイムスタンプ、コールを行なった呼出元の名前を識別する「callerName」属性、およびコールが行なわれた日時を識別する「callTime」属性を識別するカラムを含む。

【0096】

特定の実施形態において、以下のように「CallCenterRecords」入力リレーション302を含むCQLクエリが識別もしくは受信され得る。

【0097】

10

20

30

40

50

SELECT * FROM CallCentreRecords [RANGE 1 HOUR ON callTime]

上に示されるCQLクエリにより、コンフィギュラブルウィンドウ演算子は、「ジェネリック値」ウィンドウ演算子303として識別される。そして、CQLクエリは、「CallCenterRecords」入力リレーション302における属性「callTime」に対して「RANGE 1 HOUR」ウィンドウ演算子を適用することによって「ジェネリック値」ウィンドウ演算子303に少なくとも部分的に基づいて実行され、出力リレーションが生成される。「RANGE 1 HOUR」ウィンドウ演算子303を適用した結果として生成される出力リレーション304も図3に示される。

【0098】

一実施形態において、および図3に示されるように、出力リレーション304は、データレコードのタイムスタンプtに対して特定の範囲「1 HOUR」に属性値「callTime」がある入力リレーション302からのデータレコードの部分集合を含む。図3に示される例において、一実施形態では、「CallCenterRecords」出力リレーション304は、各データレコードについて、入力リレーション302に対して「ジェネリック値」ウィンドウ演算子303を適用した結果として入力リレーション302からの特定のデータレコードが出力リレーション304に挿入されるか、もしくは出力リレーション304から取り除かれたかを示す「SIGN」属性を識別するカラムも含む。「CallCenterRecords」出力リレーション304が生成される方法は、以下に詳細に記載される。

【0099】

「CallCenterRecords」入力リレーション302における第1のデータレコードについて考慮する。「CallCenterRecords」入力リレーション302における情報により、このレコードは、21/01/2012 9:00:00 AMのタイムスタンプを有し、呼出元「Robin」によって21/01/2012 8:30:00 AMにコールが行なわれたことを示す。「RANGE 1 HOUR」ウィンドウ演算子303が「callTime」属性に対して適用される場合、「CallCenterRecords」出力リレーション304に第1のデータレコードが挿入されるかどうか判定される。この例において、「CallCenterRecords」入力リレーション302における第1のデータレコードは、「CallCenterRecords」出力リレーション304に挿入され、これはこのコールのコール時間(8:30:00AM)が21/01/2012 9:00:00 AMの現在タイムスタンプから1時間の範囲内に行なわれたためであり、ここで現在タイムスタンプからの1時間の範囲は、このデータレコードについては(8:00AMから9:00AM)であると定義される。一実施形態において、および図3Bに示されるように、「CallCenterRecords」出力リレーション304へのこのデータレコードの挿入は、データレコードに対応するSIGNカラムにおける+エントリによって示される。

【0100】

同様に、呼出元「Michael」によって21/01/2012 8:45:00 AMにコールが行なわれたことを示す「CallCenterRecords」入力リレーション302における第2のデータレコードもまた、このコールのコール時間(8:45:00AM)が21/01/2012 9:00:00 AMの現在タイムスタンプから1時間の範囲内に行なわれたことから、「CallCenterRecords」出力リレーション304に挿入され、ここで現在のタイムスタンプからの1時間の範囲は、このデータレコードについては(8:00:00 AMから9:00:00AM)であると定義される。

【0101】

「CallCenterRecords」入力リレーション302における次のデータレコードは、現在タイムスタンプが21/01/2012 10:00:00 AMに移ったことを示す。さらに、10:00:00 AMの現在タイムスタンプにおける入力リレーション302に2つのデータレコードが挿入されることがさらに観察され得る。これらのレコードは、21/01/2012 9:06:00 AMにコールを行なった呼出元「Sandeep」と、21/01/2012 9:30:00 AMにコールを行なった呼出元「Anand」を含む。これらのレコードは、21/01/2012 10:00:00 AMの現在タイムスタンプの1時間の範囲内にあり、ここで現在タイムスタンプからの1時間の範囲が(9:00:00 AMから10:00:00 AM)であると定義されることから、これらもまた、これらのデータレコードの隣のSIGNカラムにおける+演算子によって示されるように「CallCenterRecords」出力リレ

10

20

30

40

50

ーション304に挿入される。しかしながら、呼出元「Robin」および「Michael」を含むデータレコードはここでは現在タイムスタンプから1時間の範囲(9:00:00 AMから10:00:00 AM)内にはないことが観察され得る。このため、これらのレコードは、10:00:00 AMの現在タイムスタンプにおける「CallCenterRecords」から取り除かれる。一実施形態において、および図3に示されるように、これらのデータレコードが「CallCenterRecords」出力リレーション304から削除されることは、データレコードに対応するSIGNカラムにおける-エントリーによって示される。

【0102】

ここで、現在タイムスタンプは21/01/2012 3:00:00 PMに移った。このタイムスタンプでの「CallCenterRecords」入力リレーション302におけるデータレコードは、21/01/2012 11:00 AMにコールを行なった呼出元「Unmesh」と、21/01/2012 1:15:00 PMにコールを行なった呼出元「Alex」とを含む。しかしながら、これらのコールのコール時間は21/01/2012 3:00:00 PMの現在タイムスタンプから1時間の範囲内に行なわれず、現在タイムスタンプから1時間の範囲は(2:00:00 PMから3:00:00 PM)であると定義されることから、これらのレコードは、出力リレーション304に挿入されない。しかしながら、21/01/2012 2:30:00 PMにコールを行なった呼出元「Sundar」は、このコールが21/01/2012 3:00:00 PMの現在タイムスタンプの1時間の範囲内に行なわれたことから、「CallCenterRecords」出力リレーション304に挿入される。同様に、呼出元「Sandeep」および「Anand」を含むデータレコードがここでは21/01/2012 3:00:00 PMの現在タイムスタンプから1時間の範囲内にはないことが観察され得る。このため、これらのレコードは、データレコードに対応するSIGNカラムにおける-エントリーによって示されるように、「CallCenterRecords」出力リレーション304から取り除かれる。

【0103】

「CallCenterRecords」入力リレーション302および「CallCenterRecords」出力リレーション304に列挙される情報は単に例として提供されるものであって本開示の範囲を限定する意図がないことが理解される。当業者は、多くの変形、変更、および代替を認識し得る。

【0104】

図4は、本開示の他の実施形態に従う、入力リレーションに記憶される例示的な情報、ならびに入力リレーションがコンフィギュラブルウィンドウ演算子を使用して処理される場合の出力リレーションの生成を示す。図4は、本開示の一実施形態に従う、「CallCenterRecords」入力リレーション402に記憶される例示的な情報を示す。図4に示される例において、一実施形態では、入力リレーション「CallCenterRecords」402は、データレコードが入力リレーションに挿入される各タイムスタンプについて、コールを行なった呼出元の名前を識別する「callerName」属性およびコールが行なわれた日時を識別する「callTime」属性を識別するカラムを含む。

【0105】

特定の実施形態において、以下のように「CallCenterRecords」入力リレーション402を含むCQLクエリが識別もしくは受信され得る。

【0106】

```
SELECT callerName, callTime FROM CallCentreRecords[CurrentHour ON callTime]
```

上に示されるCQLクエリにより、コンフィギュラブルウィンドウ演算子は「現在時間」ウィンドウ演算子403として識別される。そして、CQLクエリは、「CallCenterRecords」入力リレーション402における属性「callTime」に対して「CurrentHour」ウィンドウ演算子403を適用することによって「現在時間」ウィンドウ演算子に少なくとも部分的に基づいて実行され、出力リレーションが生成される。「CurrentHour」ウィンドウ演算子403を適用した結果として生成される出力リレーション404は、図4に示される。

【0107】

一実施形態において、および図4に示されるように、出力リレーション404は、デー

10

20

30

40

50

タレコードのタイムスタンプ t に対する現在時間に属性値「callTime」がある入力リレーション 402 からのデータレコードの部分集合を含む。図 4 に示される例において、一実施形態では、「CallCenterRecords」出力リレーション 404 は、各データレコードについて「CurrentHour」ウィンドウ演算子 403 を入力リレーション 402 に適用した結果として入力リレーション 402 からの特定のデータレコードが出力リレーション 404 に挿入されるか、または出力リレーション 404 から取り除かれたかどうかを示す「SIGN」属性を識別するカラムも含む。

【0108】

図 5 は、本開示の他の実施形態に従う、入力リレーションに記憶される例示的な情報、ならびにコンフィギュラブルウィンドウ演算子を使用して入力リレーションが処理される場合の出力リレーションの生成を示す。図 5 は、本開示の一実施形態に従う、「CallCenterRecords」入力リレーション 402 に記憶される例示的な情報を示す。図 5 に示される例において、一実施形態では、入力リレーション「CallCenterRecords」502 は、データレコードが入力リレーションに挿入される時の各タイムスタンプについて、コールを行なった呼出元の名前を識別する「callerName」属性およびコールが行なわれた日時を識別する「callTime」属性を識別するカラムを含む。

【0109】

特定の実施形態において、以下のように「CallCenterRecords」入力リレーション 502 を含む CQL クエリが識別もしくは受信され得る。

【0110】

```
SELECT callerName, callTime FROM CallCentreRecords[CurrentPeriod("0900","1500"
ON callTime]
```

上に示される CQL クエリにより、コンフィギュラブルウィンドウ演算子は、「現在期間」ウィンドウ演算子 503 として識別される。そして、CQL クエリは、「CallCenterRecords」入力リレーション 502 における属性「callTime」に対して「CurrentPeriod」ウィンドウ演算子 503 を適用することによって「CurrentPeriod」ウィンドウ演算子 503 に少なくとも部分的に基づいて実行され、出力リレーションが生成される。「CurrentPeriod」ウィンドウ演算子 503 を適用した結果として生成される出力リレーション 504 も図 5 に示される。

【0111】

一実施形態において、および図 5 に示されるように、出力リレーション 504 は、データレコードのタイムスタンプ t に対する現在期間 $\text{CurrentPeriod}("0900","1500")$ に属性値「callTime」がある入力リレーションからのデータレコードの部分集合を含み得る。図 5 に示される例において、一実施形態では、「CallCenterRecords」出力リレーション 504 は、 $\text{CurrentPeriod}("0900","1500")$ ウィンドウ演算子 503 を入力リレーション 502 に適用した結果として入力リレーション 502 からの特定のデータレコードが出力リレーション 504 に挿入されるか、または出力リレーション 504 から取り除かれたかどうかを示す「SIGN」属性を各データレコードについて識別するカラムも含む。

【0112】

開示の技術により、連続クエリ（たとえば、CQL クエリ）を使用して、外部データソース（たとえば、リレーション）に記憶されるアプリケーションに関連する受信中のリアルタイム情報の処理が可能となる。この受信中のリアルタイム情報は、典型的に、インメモリーリアルタイムイベント処理エンジン（たとえば、CQL エンジン）を使用して処理される。しかしながら、リアルタイムで受信されるアプリケーションに関連する情報の量が増加すると、メモリーに記憶されるリレーションのサイズも増加する。連続クエリを使用して有限のメモリーにおいてこのような大きなサイズのリレーションを処理することは、有限のメモリーにおいて大きなサイズのリレーションに対して高いレートの入力ストリームを結合させることを伴い得ることから、スケーラブルではない。

【0113】

開示の技術により、連続クエリ（たとえば、CQL クエリ）におけるウィンドウ演算子

10

20

30

40

50

を定義することによって、メモリーにおけるリレーションのデータレコードのセットの全体を処理するのではなく限られたメモリーにおいてリレーションのデータレコードの部分集合を照会することが可能となる。連続クエリ（たとえば、CQLクエリ）は、ウィンドウ演算子の継続的な期間にわたって入力リレーションに対して適用され、出力リレーションが生成される。出力リレーションに関連するデータレコードのセットは、ユーザーに対して提供される。ウィンドウ演算子は、ユーザーによってコンフィギュラブルであり、特定の時間の範囲にわたってリレーションにおけるデータレコードの部分集合を処理することを可能とする。このため、リレーションのデータレコードの部分集合は、メモリーにおけるリレーションのデータレコードのセットのすべてを処理するのではなく、限られたメモリーにおいて照会され得る。

10

【0114】

開示の技術は、連続的な非有界データストリームの形態であり得るリアルワールドアプリケーションに関連するデータに適用され得る。このようなデータストリームの例としては、金融アプリケーションにおける株式相場表示装置、ネットワーク監視およびトラフィック管理における性能測定、ウェブトラッキングおよびパーソナリゼーションにおけるログ記録もしくはクリックストリーム、ファイアーウォールベースのセキュリティにおけるネットワークパケットおよびメッセージ、電話通信におけるコール詳細記録などが含まれ得る。

【0115】

図6および図7は、本願明細書において記載されるリレーションに対する値ベースのウィンドウの管理を実施するためのそれぞれの処理600および700を示す例示的なフロー図である。これらの処理600および700は、論理フロー図として示され、その各動作は、ハードウェア、コンピューター命令、またはそれらの組み合わせにおいて実施され得る動作のシーケンスを表わす。コンピューター命令の文脈において、動作は、1つ以上のプロセッサによって実行される場合に記載の動作を行なう、1つ以上のコンピューター読取可能記憶媒体に記憶されるコンピューター実行可能命令を表わす。概して、コンピューター実行可能命令には、特定の機能を行なう、もしくは特定のデータタイプを実施する、ルーチン、プログラム、オブジェクト、コンポーネント、データ構造などが含まれる。動作が記載される順序は、限定的に解釈されることを意図しておらず、任意の数の記載の動作が任意の順序で、および/または並列に組み合わせられ得て、処理が実施され得る。

20

【0116】

加えて、一部の、任意の、もしくはすべての処理は、実行可能命令を用いて構成される1つ以上のコンピューターシステムの制御下において行なわれ得る、および1つ以上のプロセッサ上でまとめて実行されるコード（たとえば、実行可能命令、1つ以上のコンピュータープログラム、または1つ以上のアプリケーション）として実施され得る、またはハードウェアによって実施され得る、またはこれらの組み合わせによって実施され得る。上記のように、コードは、たとえば1つ以上のプロセッサによって実行可能な複数の命令を含むコンピュータープログラムの形態でコンピューター読取可能記憶媒体上に記憶され得る。コンピューター読取可能記憶媒体は、非一時的であり得る。

30

【0117】

一部の例において、少なくとも図1（およびその他）に示される1つ以上のサービスプロバイダーコンピューター106（たとえば、少なくとも入力リレーションモジュール148、コンフィギュラブルウィンドウ演算子モジュール150、または出力リレーションモジュール152を利用する）は、図6の処理600を行ない得る。処理600は、入力リレーションの精製を含むことによってブロック602において開始され得る。ブロック604において、処理600は、入力リレーションを処理するクエリ（たとえば、CQLクエリ）を識別することを含み得る。ブロック606において、処理600は、入力リレーションの処理に関連付けられるコンフィギュラブルウィンドウ演算子を識別することを含み得る。さらに、一部の例において、処理600は、コンフィギュラブルウィンドウ演算子に少なくとも部分的に基づいてクエリを実行して出力リレーションを生成することを含むことにより、ブロック608において終了し得る。

40

50

【 0 1 1 8 】

図 7 は、本願明細書に記載されるリレーションに対する値ベースのウィンドウの管理を実施するための処理 7 0 0 を示す例示的なフロー図である。少なくとも図 1（およびその他）に示される 1 つ以上のサービスプロバイダーコンピューター 1 0 6（たとえば、少なくとも入力リレーションモジュール 1 4 8、コンフィギュラブルウィンドウ演算子モジュール 5 5 0、または出力リレーションモジュール 1 5 2 を利用する）は、図 7 の処理 7 0 0 を行ない得る。処理 7 0 0 は、入力リレーションの生成を含むことによってブロック 7 0 2 において開始され得る。一実施形態において、入力リレーションは、アプリケーションに関連するデータレコードの有界集合である。ブロック 7 0 4 において、処理 7 0 0 は、入力リレーションを識別する連続クエリ（たとえば、C Q L クエリ）を受信することを含み得る。ブロック 7 0 6 において、処理 7 0 0 は、入力リレーションの処理に関連付けられるコンフィギュラブルウィンドウ演算子を識別することを含み得る。さらに、一部の例において、処理 7 0 0 は、ブロック 7 0 8 において、コンフィギュラブルウィンドウ演算子に少なくとも部分的に基づいて連続クエリを実行して出力リレーションを生成することを含み得る。さらに、処理 7 0 0 は、ブロック 7 1 0 において、連続クエリの実行に少なくとも部分的に基づいて出力リレーションのデータレコードを提供することを含むことによって終了し得る。

10

【 0 1 1 9 】

図 8 は、本開示の実施形態に従って使用され得るシステム環境 8 0 0 の構成部分を示す簡易化されたブロック図である。示されるように、システム環境 8 0 0 は、1 つ以上のクライアントコンピューティングデバイス 8 0 2、8 0 4、8 0 6、8 0 8 を含み、これらは、1 つ以上のネットワーク 8 1 0（図 1 のネットワーク 1 0 8 と同様のネットワークなどであるが、これに限定されない）を通じて、ウェブブラウザ、専用クライアント（たとえば、Oracle Forms）などのクライアントアプリケーションを動作させるように構成される。様々な実施形態において、クライアントコンピューティングデバイス 8 0 2、8 0 4、8 0 6、8 0 8 は、ネットワーク 8 1 0 を通じてサーバー 8 1 2 と対話し得る。

20

【 0 1 2 0 】

クライアントコンピューティングデバイス 8 0 2、8 0 4、8 0 6、8 0 8 は、汎用パーソナルコンピューター（たとえば、Microsoft Windows（登録商標）および/または Apple Macintosh（登録商標）オペレーティングシステムの様々なバージョンを実行するパーソナルコンピューターおよび/またはラップトップコンピューターが含まれる）、携帯電話もしくは P D A（Microsoft Windows Mobile（登録商標）などのソフトウェアを実行するとともに、インターネット、電子メール、SMS、ブラックベリー、もしくは他の通信プロトコルに対応する）、および/または様々な商業的に利用可能な UNIX（登録商標）もしくは UNIX のようなオペレーティングシステム（様々な GNU / Linux（登録商標）オペレーティングシステムを含み、限定されない）のいずれかを実行するワークステーションコンピューターであり得る。代替的に、クライアントコンピューティングデバイス 8 0 2、8 0 4、8 0 6、8 0 8 は、ネットワーク（たとえば、以下に記載のネットワーク 8 1 0）を通じて通信可能な、シンクライアントコンピューター、インターネット対応ゲーミングシステム、および/またはパーソナルメッセージングデバイスなどの任意の他の電子デバイスであり得る。4 つのクライアントコンピューティングデバイスを用いて例示的なシステム環境 8 0 0 が示されているが、任意の数のクライアントコンピューティングデバイスが支持され得る。センサーを有するデバイスなどの他のデバイスが、サーバー 8 1 2 と対話し得る。

30

40

【 0 1 2 1 】

システム環境 8 0 0 は、ネットワーク 8 1 0 を含み得る。ネットワーク 8 1 0 は、TCP / IP、SNA、IPX、AppleTalk などを含みこれらに限定されない様々な商業的に利用可能なプロトコルのいずれかを使用してデータ通信をサポートすることができる、当業者によく知られた任意のタイプのネットワークであり得る。単に例示であるが、ネットワーク 1 5 1 0 は、イーサネット（登録商標）ネットワークおよび/またはトー

50

クンリングネットワークなどのローカルエリアネットワーク（LAN）、広域ネットワーク、仮想プライベートネットワーク（VPN）を含むがこれに限定されない仮想ネットワーク、インターネット、イントラネット、エクストラネット、公衆交換電話網（PSTN）、赤外線ネットワーク、無線ネットワーク（たとえば、IEEE 802.11プロトコルスイート、当該技術において知られるBluetooth（登録商標）プロトコル、および/または任意の他の無線プロトコルにおいて運用されるネットワーク）、ならびに/またはこれらおよび/もしくは他のネットワークの組み合わせであり得る。

【0122】

また、システム環境800は、汎用コンピューター、専用サーバーコンピューター（たとえば、PCサーバー、UNIXサーバー、ミッドレンジサーバー、メインフレームコンピューター、ラックマウントサーバーなどを含む）、サーバーファーム、サーバークラスター、または任意の他の適切な構成および/または組み合わせであり得る1つ以上のサーバーコンピューター812を含む。様々な実施形態において、サーバー812は、上記の開示において記載される1つ以上のサービスもしくはソフトウェアアプリケーションを実行するように適合され得る。たとえば、サーバー812は、本開示の実施形態に係る、上記の処理を行なうためのサーバーに対応し得る。

【0123】

サーバー812は、上記のいずれかを含むオペレーティングシステム、ならびに商業的に利用可能なサーバーオペレーティングシステムのいずれかを実行し得る。また、サーバー812は、HTTPサーバー、FTPサーバー、CGIサーバー、Java（登録商標）サーバー、データベースサーバーなどを含む、様々な付加的なサーバーアプリケーションおよび/またはミッドティアアプリケーションのいずれかを実行し得る。例示的なデータベースサーバーは、Oracle、Microsoft、Sybase、IBMなどから商業的に入手可能なものも含むが、これらに限定されない。

【0124】

また、システム環境800は、1つ以上のデータベース814、816を含み得る。データベース814、816は、様々な場所に置かれ得る。たとえば、データベース814、816のうちの1つ以上は、サーバー812に対してローカルの（および/またはそこに置かれる）非一時的記憶媒体上に置かれ得る。代替的に、データベース814、816は、サーバー812から遠隔に設けられ得て、ネットワークベースもしくは専属接続を介してサーバー812と通信し得る。実施形態の1つのセットにおいて、データベース814、816は、当業者によく知られるストレージエリアネットワーク（SAN）内に置かれ得る。同様に、サーバー812に帰する機能を行なうために必要な任意のファイルは、適宜、サーバー812にローカルに記憶され得る、および/または遠隔的に記憶され得る。実施形態の1つの集合において、データベース814、816は、SQLフォーマットの命令に応答してデータを記憶、更新、および検索するように適合される、Oracleによって提供されるデータベースなどのリレーショナルデータベースを含み得る。

【0125】

図9は、本開示の実施形態に従って使用され得るコンピューターシステム900を示す簡易化されたブロック図である。たとえば、サービスプロバイダーコンピューター106は、システム900などのシステムを使用して実施され得る。コンピューターシステム900は、バス901を介して電気的および/または通信可能に結合され得るハードウェア要素を含んで示される。ハードウェア要素は、1つ以上の中央処理ユニット（CPU）902、1つ以上の入力デバイス904（たとえば、マウス、キーボードなど）、および1つ以上の出力デバイス906（たとえば、ディスプレイデバイス、プリンターなど）を含み得る。コンピューターシステム900は、1つ以上の記憶デバイス908も含み得る。たとえば、記憶デバイス908は、ディスクドライブ、光学記憶デバイス、ならびにプログラム可能および/またはフラッシュ更新可能であるランダムアクセスメモリー（RAM）および/もしくは読み取り専用メモリー（ROM）などのソリッドステート記憶デバイスなどのデバイスを含み得る。

【 0 1 2 6 】

コンピュータシステム 9 0 0 は、コンピュータ読取可能記憶媒体リーダー 9 1 2、通信サブシステム 9 1 4（たとえば、モデム、ネットワークカード（無線もしくは有線）、赤外線通信デバイスなど）、ならびに上記のような R A M および R O M デバイスを含み得るワーキングメモリー 9 1 8 を付加的に含み得る。一部の実施形態において、コンピュータシステム 9 0 0 は、デジタル信号プロセッサ（ D S P ）、専用プロセッサなどを含み得る、処理加速ユニット 9 1 6 も含み得る。

【 0 1 2 7 】

コンピュータ読取可能記憶媒体リーダー 9 1 2 は、コンピュータ読取可能記憶媒体 9 1 0 にさらに接続され得て、これらは合わせて（および、選択的には、記憶デバイス 9 0 8 と組み合わせて）包括的に、遠隔、ローカル、固定、および / またはリムーバブル記憶デバイスと、コンピュータ読取可能情報を一時的および / またはより永続的に記憶するための記憶媒体とを加えたものを表わす。通信システム 9 1 4 は、ネットワーク 9 5 2、および / またはシステム環境 9 0 0 に関して上で記載される任意の他のコンピュータとデータが交換されることを許容し得る。

【 0 1 2 8 】

コンピュータシステム 9 0 0 は、ワーキングメモリー 9 1 8 内に位置しているものとして示され、オペレーティングシステム 9 2 0 および / またはアプリケーションプログラム（クライアントアプリケーション、ウェブブラウザ、ミッドティアアプリケーション、 R D B M S などであり得る）および / または他のコードソフトウェア要素も含み得る。例示的な実施形態において、ワーキングメモリー 9 1 8 は、上記のような信頼当事者およびオープン許可に関連する処理のための実行可能コードおよび関連付けられるデータ構造を含み得る。なお、コンピュータシステム 9 0 0 の代替的な実施形態は、上記のものからの様々な変更を有し得ることを理解すべきである。たとえば、カスタマイズされるハードウェアも使用され得る、および / または特定の要素がハードウェア、ソフトウェア（タブレットなどのポータブルソフトウェアを含む）、またはその両方で実施され得る。さらに、ネットワーク入力 / 出力デバイスなどの他のコンピューティングデバイスへの接続も採用され得る。

【 0 1 2 9 】

図 1 0 は、本発明の他の実施形態に従う、本願明細書に記載されるリレーションに対する値ベースのウィンドウの管理を実施するための処理 1 0 0 0 を示す例示的なフロー図を示す。少なくとも図 1（およびその他）に示される 1 つ以上のサービスプロバイダーコンピュータ 1 0 6（たとえば、少なくとも入力リレーションモジュール 1 4 8、コンフィギュラブルウィンドウ演算子モジュール 1 5 0、または出力リレーションモジュール 1 5 2 を利用する）は、図 1 0 の処理 1 0 0 0 を行ない得る。処理 1 0 0 0 は、入力リレーションを処理するように構成されるクエリ（たとえば、連続クエリ）を識別することを含むことによってブロック 1 0 0 2 において開始され得る。ブロック 1 0 0 4 において、処理 1 0 0 0 は、コンフィギュラブルウィンドウ演算子に基づいてクエリを開始して出力リレーションを生成することを含み得る。ブロック 1 0 0 6 において、処理 1 0 0 0 は、入力イベントを生成するとともにそれを C Q L エンジンおよび / または C Q サービス 2 0 0 に送信することを含み得る。一部の例において、入力イベントは、新しいイベントが C Q L エンジンおよび / または C Q サービス 2 0 0 において受信される時に生成され得る。たとえば、入力イベントは、上記の「CallCentreRecords」入力リレーション 3 0 2 における呼出元によって行なわれる新しいコールを含み得る。ブロック 1 0 0 8 において、処理 1 0 0 0 は、入力イベントおよび校正可能ウィンドウ演算子を受信し、出力イベントを作ることを含み得る。出力イベントは、たとえば、「CallCenterRecords」入力リレーション 3 0 2 における「コール時間」属性に対してコンフィギュラブルウィンドウ演算子が適用される時に「CallCenterRecords」出力リレーション 3 0 4 に挿入されるデータレコードを含み得る。

【 0 1 3 0 】

図 1 1 は、本発明の特定の実施形態に従って使用され得るサービスプロバイダーデバイス 1 1 0 1 を示す簡易化されたブロック図である。サービスプロバイダーデバイス 1 1 0 1 のブロックは、本発明の原則を実行するために、ハードウェア、ソフトウェア、またはハードウェアとソフトウェアの組み合わせによって実施され得る。当業者は、図 1 1 に記載されるブロックが組み合わせられ得る、またはサブブロックに分離され得て、上記の本発明の原則が実施されることを理解する。このため、本願明細書における記載は、可能性のある組み合わせもしくは分離、または本願明細書に記載される機能ブロックのさらなる定義を支持し得る。

【 0 1 3 1 】

図 1 1 に示されるように、サービスプロバイダーデバイス 1 1 0 1 は、入力リレーションユニット 1 1 0 2 と、コンフィギュラブルウィンドウ演算子ユニット 1 1 0 3 と、出力リレーションユニット 1 1 0 5 と、提供ユニット 1 1 0 7 とを含み得る。入力リレーションユニット 1 1 0 2 は、入力リレーションを識別する連続クエリを受信するように構成され得て、ここで入力リレーションは、アプリケーションに関連するデータレコードの有界集合である。コンフィギュラブルウィンドウ演算子ユニット 1 1 0 3 は、入力リレーションの処理に関連付けられるコンフィギュラブルウィンドウ演算子を識別するように構成され得る。出力リレーションユニット 1 1 0 5 は、コンフィギュラブルウィンドウ演算子に少なくとも部分的に基づいて連続クエリを実行して出力リレーションを生成するように構成され得る。提供ユニット 1 1 0 7 は、連続クエリの実行に少なくとも部分的に基づいて出力リレーションのデータレコードを提供するように構成され得る。一実施形態において、各ユニットは、コンピュータプログラム命令を読み取ることによって対応する処理を行なうプロセッサとして実施され得る。

【 0 1 3 2 】

一実施形態において、入力リレーションは、アプリケーションに関連する受信中の連続入力データストリームに少なくとも部分的に基づいて生成される外部データソースであり得る。

【 0 1 3 3 】

一実施形態において、入力リレーションは、過去データのデータベースに記憶されるアプリケーションに関連する情報に少なくとも部分的に基づいて生成される外部データソースであり得る。

【 0 1 3 4 】

一実施形態において、入力リレーションは、アプリケーションに関連する 1 つ以上のアーカイブされたりレーションに少なくとも部分的に基づいて生成される外部データソースであり得る。

【 0 1 3 5 】

一実施形態において、コンフィギュラブルウィンドウ演算子は、入力リレーションに対して定義されるジェネリック値ウィンドウ演算子であり得て、ここで出力リレーションユニット 1 1 0 5 は、出力リレーションを生成するために入力リレーションにおける属性に対してジェネリック値ウィンドウ演算子を適用するようにさらに構成され得る。

【 0 1 3 6 】

一実施形態において、コンフィギュラブルウィンドウ演算子は、入力リレーションに対して定義される現在時間値ウィンドウ演算子であり得て、ここで出力リレーションユニット 1 1 0 5 は、出力リレーションを生成するために入力リレーションにおける属性に対して現在時間値ウィンドウ演算子を適用するようにさらに構成され得る。

【 0 1 3 7 】

一実施形態において、コンフィギュラブルウィンドウ演算子は、入力リレーションに対して定義される現在期間値ウィンドウ演算子であり得て、ここで出力リレーションユニット 1 1 0 5 は、出力リレーションを生成するために入力リレーションにおける属性に対して現在期間値ウィンドウ演算子を適用するようにさらに構成され得る。

【 0 1 3 8 】

一実施形態において、提供ユニット１１０７は、出力リレーションを表示するようにさらに構成され得て、ここで出力リレーションは、コンフィギュラブルウィンドウ演算子によって定義される特定の範囲内に属性値がある入力リレーションからのデータレコードの部分集合を含む。

【０１３９】

図１２は、本発明の特定の実施形態に従って使用され得るサービスプロバイダーデバイス１２０１を示す簡易化されたブロック図である。サービスプロバイダーデバイス１２０１のブロックは、本発明の原則を実行するために、ハードウェア、ソフトウェア、またはハードウェアとソフトウェアの組み合わせによって実施され得る。当業者は、上記の本発明の原則を実行するために図１２のブロックが結合され得る、またはサブブロックに分離され得ることを理解する。このため、本願明細書における記載は、可能性のある組み合わせもしくは分離、または本願明細書に記載の機能ブロックのさらなる定義を支持し得る。

【０１４０】

図１２に示されるように、サービスプロバイダーサービス１２０１は、入力リレーションユニット１２０２、コンフィギュラブルウィンドウ演算子ユニット１２０３、および出力リレーションユニット１２０５を含み得る。入力リレーションユニット１２０２は、入力リレーションを処理するように構成されるクエリを受信および識別するように構成され得る。コンフィギュラブルウィンドウ演算子ユニット１２０３は、入力リレーションの処理に関連付けられるコンフィギュラブルウィンドウ演算子を識別するように構成され得る。出力リレーションユニット１２０５は、出力リレーションを生成するためにコンフィギュラブルウィンドウ演算子に少なくとも部分的に基づいてクエリを実行するように構成され得る。一実施形態において、各ユニットは、コンピュータープログラム命令を読み取ることによって対応する処理を行なうプロセッサとして実施され得る。

【０１４１】

一実施形態において、入力リレーションは、アプリケーションに関連する受信中の連続入力データストリームに少なくとも部分的に基づいて背市営される外部データソースであり得る。

【０１４２】

一実施形態において、入力リレーションは、過去データのデータベースに記憶されるアプリケーションに関連する情報に少なくとも部分的に基づいて生成される外部データソースであり得る。

【０１４３】

一実施形態において、入力リレーションは、アプリケーションに関連する１つ以上のアーカイブされたりリレーションに少なくとも部分的に基づいて生成される外部データソースであり得る。

【０１４４】

一実施形態において、コンフィギュラブルウィンドウ演算子は、入力リレーションに対して定義されるジェネリック値ウィンドウ演算子であり得て、出力リレーションユニット１２０５は、出力リレーションを生成するために入力リレーションにおける属性に対してジェネリック値ウィンドウ演算子を適用するようにさらに構成され得る。

【０１４５】

一実施形態において、サービスプロバイダーデバイス１２０１は、連続クエリの実行に少なくとも部分的に基づいて出力リレーションのデータレコードを提供するように構成される提供ユニット１２０７をさらに含み得る。

【０１４６】

一実施形態において、提供ユニット１２０７は、出力リレーションを表示するようにさらに構成され得て、ここで出力リレーションは、コンフィギュラブルウィンドウ演算子によって定義される特定の範囲内に属性値がある入力リレーションからのデータレコードの部分集合を含む。

【０１４７】

10

20

30

40

50

コードもしくはコードの一部を記憶する記憶媒体およびコンピューター読み取り可能媒体は、当該技術において知られる、もしくは使用される任意の適切な媒体を含み、この媒体には、コンピューター読み取り可能命令、データ構造、プログラムモジュール、もしくは他のデータなどの情報の記憶および/または伝達のための任意の方法および技術において実施される揮発性および不揮発性（非一時的）のリムーバブルおよび非リムーバブル媒体などの記憶媒体および通信媒体が含まれるが、これらに限定されず、RAM、ROM、EEPROM、フラッシュメモリー、もしくは他のメモリー技術、CD-ROM、デジタルバーサタイルディスク（DVD）、もしくは他の光学記憶部、磁気カセット、磁気テープ、磁気ディスク記憶部、もしくは他の磁気記憶デバイス、データ信号、データ伝達、もしくは所望の情報を記憶もしくは伝達することができるとともにコンピューターによってアクセスすることができる任意の他の媒体が含まれ得る。

10

【0148】

本開示の具体的な実施形態が記載されるが、様々な変形、変更、代替的構造、および均等物もまた本開示の範囲内に含まれる。本開示の実施形態は、特定の具体的なデータ処理環境内における動作に限定されず、複数のデータ処理環境内において自由に動作する。加えて、特定の連続したトランザクションおよびステップを使用して本開示の実施形態が記載されるが、本開示の範囲が記載される連続したトランザクションおよびステップに限定されないことは当業者にとって明らかである。

【0149】

さらに、本開示の実施形態は、ハードウェアおよびソフトウェアの特定の組み合わせを使用して記載されるが、ハードウェアおよびソフトウェアの他の組み合わせも本開示の範囲内にあることを認識すべきである。本開示の実施形態は、ハードウェアのみ、ソフトウェアのみ、またはこれらの組み合わせで実施され得る。

20

【0150】

このため、明細書および図面は、限定的ではなく例示的であるものとして考慮される。しかしながら、より広い主旨および範囲から逸脱することなく追加、削減、削除、および他の変更および変化が行なわれ得ることは明らかである。本開示の特徴を提供するための例示的な方法およびシステムは、上に記載されている。これらのシステムおよび方法の一部もしくはすべては、上記の図1から図12に示されるものなどのアーキテクチャによって少なくとも部分的に実施され得るが、必要ではない。

30

【0151】

構造的特徴および/または方法論的な動作に対して具体的な言語で実施形態が記載されるが、開示は必ずしも記載される具体的な特徴もしくは動作に限定されないことが理解される。むしろ、具体的な特徴および動作は実施形態を実施する例示的な形態として開示されている。とりわけ、「can」、「could」、「might」、または「may」などの条件的な言語は、具体的に述べられない限り、または使用される文脈において理解されない限り、特定の実施形態が特定の特徴、要素、および/またはステップを含み、他の実施形態が含み得ないものとして伝えることを概して意図している。したがって、このような条件的な言語は、1つ以上の実施形態に対して特徴、要素、および/またはステップが必要であること、またはユーザー入力もしくは命令の有無に関わらずこれらの特徴、要素、および/またはステップが含まれるかもしくは任意の特定の実施形態において行なわれるかを決定するためのロジックを1つ以上の実施形態が必ず含む必要があることを暗に示すことを概して意図していない。

40

【図 1】

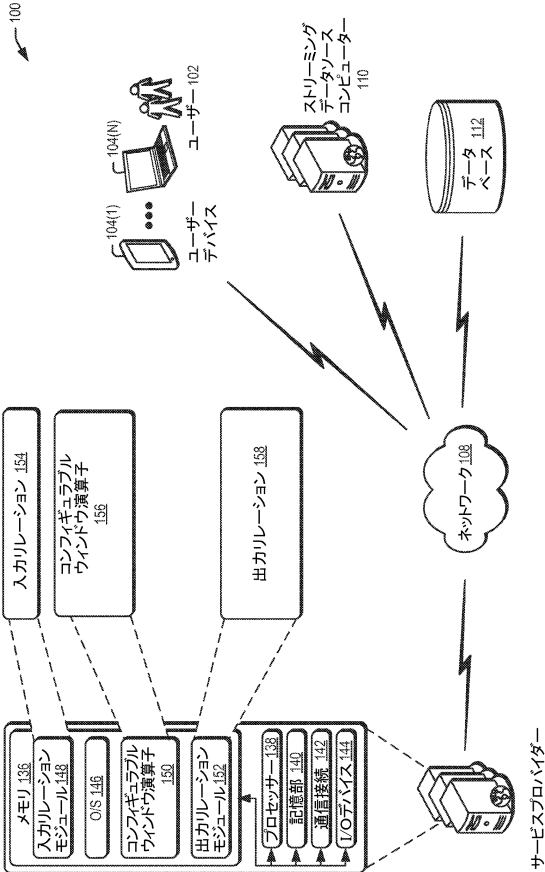


FIG. 1

【図 2】

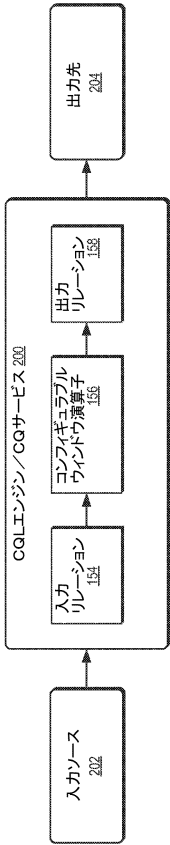


FIG. 2

【図 3】

Timestamp	callerName	callTime
21/01/2012 9:00:00 AM	Robin	21/01/2012 8:30:00 AM
21/01/2012 9:00:00 AM	Michael	21/01/2012 8:45:00 AM
21/01/2012 10:00:00 AM	Sandeep	21/01/2012 9:30:00 AM
21/01/2012 10:00:00 AM	Anand	21/01/2012 11:00:00 AM
21/01/2012 3:00:00 PM	Umesh	21/01/2012 1:15:00 PM
21/01/2012 3:00:00 PM	Alex	21/01/2012 2:30:00 PM
21/01/2012 3:00:00 PM	Sundar	

コンフィギュラブルウィンドウ演算子 (シネリック) 303

Timestamp	SIGN	callerName	callTime
21/01/2012 9:00:00 AM	+	Robin	21/01/2012 8:30:00 AM
21/01/2012 9:00:00 AM	+	Michael	21/01/2012 8:45:00 AM
21/01/2012 10:00:00 AM	-	Robin	21/01/2012 8:30:00 AM
21/01/2012 10:00:00 AM	-	Michael	21/01/2012 8:45:00 AM
21/01/2012 10:00:00 AM	+	Sandeep	21/01/2012 9:00:00 AM
21/01/2012 10:00:00 AM	+	Anand	21/01/2012 9:30:00 AM
21/01/2012 3:00:00 PM	-	Sandeep	21/01/2012 9:00:00 AM
21/01/2012 3:00:00 PM	-	Anand	21/01/2012 9:30:00 AM
21/01/2012 3:00:00 PM	+	Sundar	21/01/2012 2:30:00 PM

FIG. 3

【図 4】

Timestamp	callerName	callTime
21/01/2012 9:59:00 AM	Robin	21/01/2012 9:30:00 AM
21/01/2012 9:59:00 AM	Michael	21/01/2012 9:45:00 AM
21/01/2012 10:59:00 AM	Sandeep	21/01/2012 10:00:00 AM
21/01/2012 10:59:00 AM	Anand	21/01/2012 10:30:00 AM
21/01/2012 3:59:00 PM	Umesh	21/01/2012 11:00:00 AM
21/01/2012 3:59:00 PM	Alex	21/01/2012 1:15:00 PM
21/01/2012 3:59:00 PM	Sundar	21/01/2012 3:30:00 PM

コンフィギュラブルウィンドウ演算子 (現在時間) 403

Timestamp	SIGN	callerName	callTime
21/01/2012 9:59:00 AM	+	Robin	21/01/2012 9:30:00 AM
21/01/2012 9:59:00 AM	+	Michael	21/01/2012 9:45:00 AM
21/01/2012 10:59:00 AM	-	Robin	21/01/2012 9:30:00 AM
21/01/2012 10:59:00 AM	-	Michael	21/01/2012 9:45:00 AM
21/01/2012 10:59:00 AM	+	Sandeep	21/01/2012 10:00:00 AM
21/01/2012 10:59:00 AM	+	Anand	21/01/2012 10:30:00 AM
21/01/2012 3:59:00 PM	-	Sandeep	21/01/2012 10:00:00 AM
21/01/2012 3:59:00 PM	-	Anand	21/01/2012 10:30:00 AM
21/01/2012 3:59:00 PM	+	Sundar	21/01/2012 3:30:00 PM

FIG. 4

【図 5】

Input Timestamp	callerName	callTime
21/01/2012 9:59:00 AM	Robin	21/01/2012 8:30:00 AM
21/01/2012 9:59:00 AM	Michael	21/01/2012 9:45:00 AM
21/01/2012 10:59:00 AM	Sandeep	21/01/2012 10:06:00 AM
21/01/2012 10:59:00 AM	Anand	21/01/2012 10:30:00 AM
21/01/2012 3:59:00 PM	Umesh	21/01/2012 11:00:00 AM
21/01/2012 3:59:00 PM	Alex	21/01/2012 11:15:00 PM
21/01/2012 3:59:00 PM	Sunder	21/01/2012 3:30:00 PM
22/01/2012 9:15:00 AM	Eric	22/01/2012 9:00:00 AM

Output Timestamp	callerName	callTime
21/01/2012 9:59:00 AM	Michael	21/01/2012 9:45:00 AM
21/01/2012 10:59:00 AM	Sandeep	21/01/2012 10:06:00 AM
21/01/2012 10:59:00 AM	Anand	21/01/2012 10:30:00 AM
21/01/2012 3:59:00 PM	Umesh	21/01/2012 11:00:00 AM
21/01/2012 3:59:00 PM	Alex	21/01/2012 11:15:00 PM
22/01/2012 09:00:00 AM	Michael	21/01/2012 9:45:00 AM
22/01/2012 09:00:00 AM	Sandeep	21/01/2012 10:06:00 AM
22/01/2012 09:00:00 AM	Anand	21/01/2012 10:30:00 AM
22/01/2012 09:00:00 AM	Umesh	21/01/2012 11:00:00 AM
22/01/2012 09:15:00 AM	Alex	21/01/2012 11:15:00 PM
22/01/2012 09:15:00 AM	Eric	22/01/2012 9:00:00 AM

FIG. 5

【図 6】

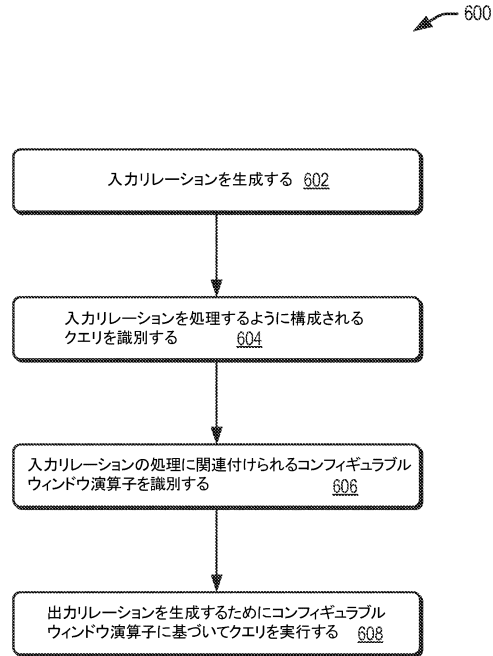


FIG. 6

【図 7】

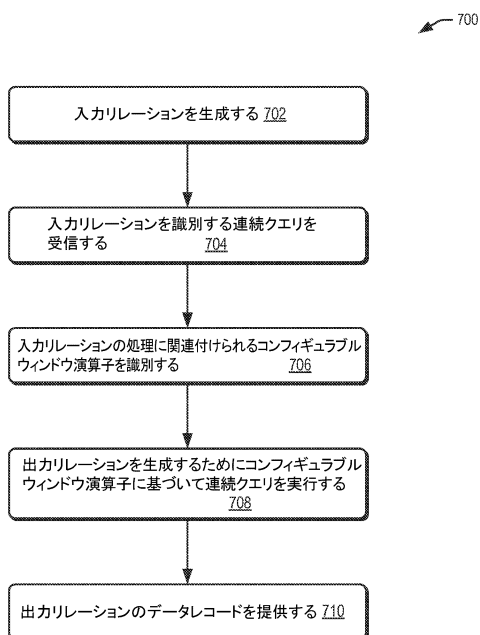


FIG. 7

【図 8】

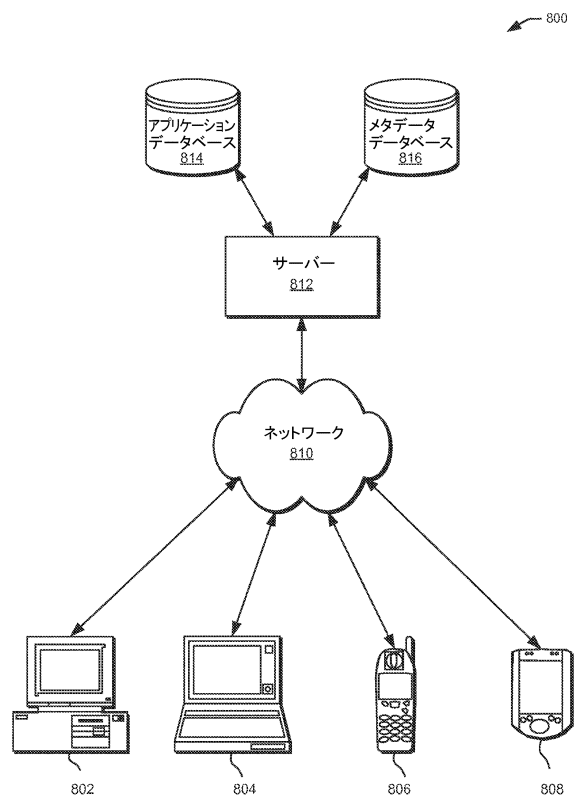


FIG. 8

【図 9】

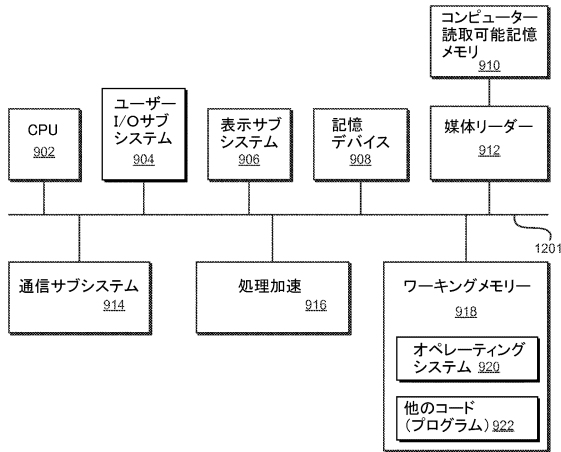


FIG. 9

【図 10】

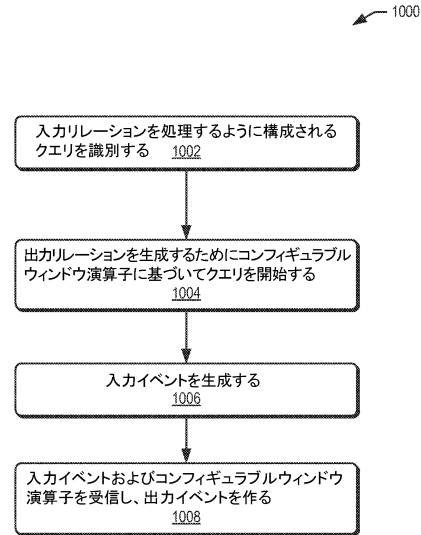


FIG. 10

【図 11】

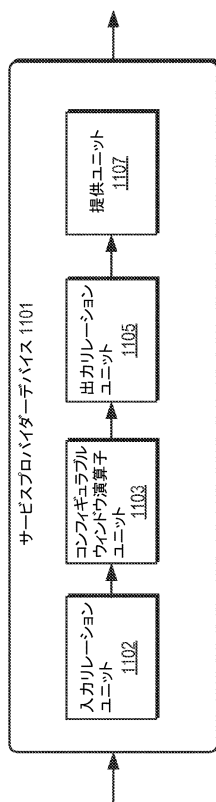


FIG. 11

【図 12】

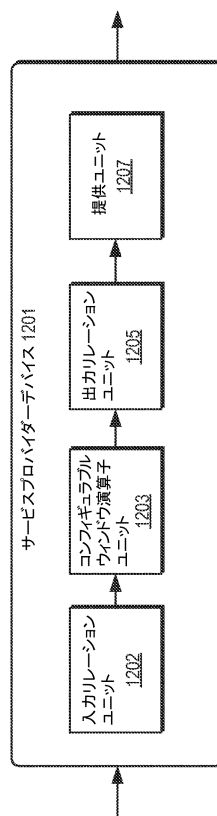


FIG. 12

フロントページの続き

- (72)発明者 スリニバサン, アナンド
インド、560076 カルナータカ、バンガロール、パナーガッタ・ロード、エヌ・エス・パ
リヤ、マントリ・エレガンス、イー・303
- (72)発明者 デシュムク, ウンメシュ・アニル
インド、440025 マハーラーシュトラ、ナーグプル、ソネガアン、エイチ・ビィ・エステー
ト・ニアー・シブ・マンディール、プラット・ナンバー・222 - エイ

審査官 早川 学

- (56)参考文献 特開2010-108073(JP, A)
国際公開第2011/142026(WO, A1)
米国特許出願公開第2011/0032260(US, A1)
国際公開第2012/172600(WO, A1)
ARASU, Arvind et al., The CQL continuous query language: semantic foundations and quer
y execution, The VLDB Journal(2006) [online], Springer-Verlag, 2006年 6月 1日
, Vol.15, No.2, pp.121-142, [検索日:2018.05.14], Internet<URL: [https://www.microsoft.com
/en-us/research/wp-content/uploads/2016/02/cql.pdf](https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/cql.pdf) >
- (58)調査した分野(Int.Cl., DB名)
G06F 16/00 - 16/958