チュートリアル6

二次利用ユーザーのためのデータ抽出、変換、登録（ETL）入門

－ETLの理解がデータ活用の近道－

ETL開発の実際

山ノ内祥訓

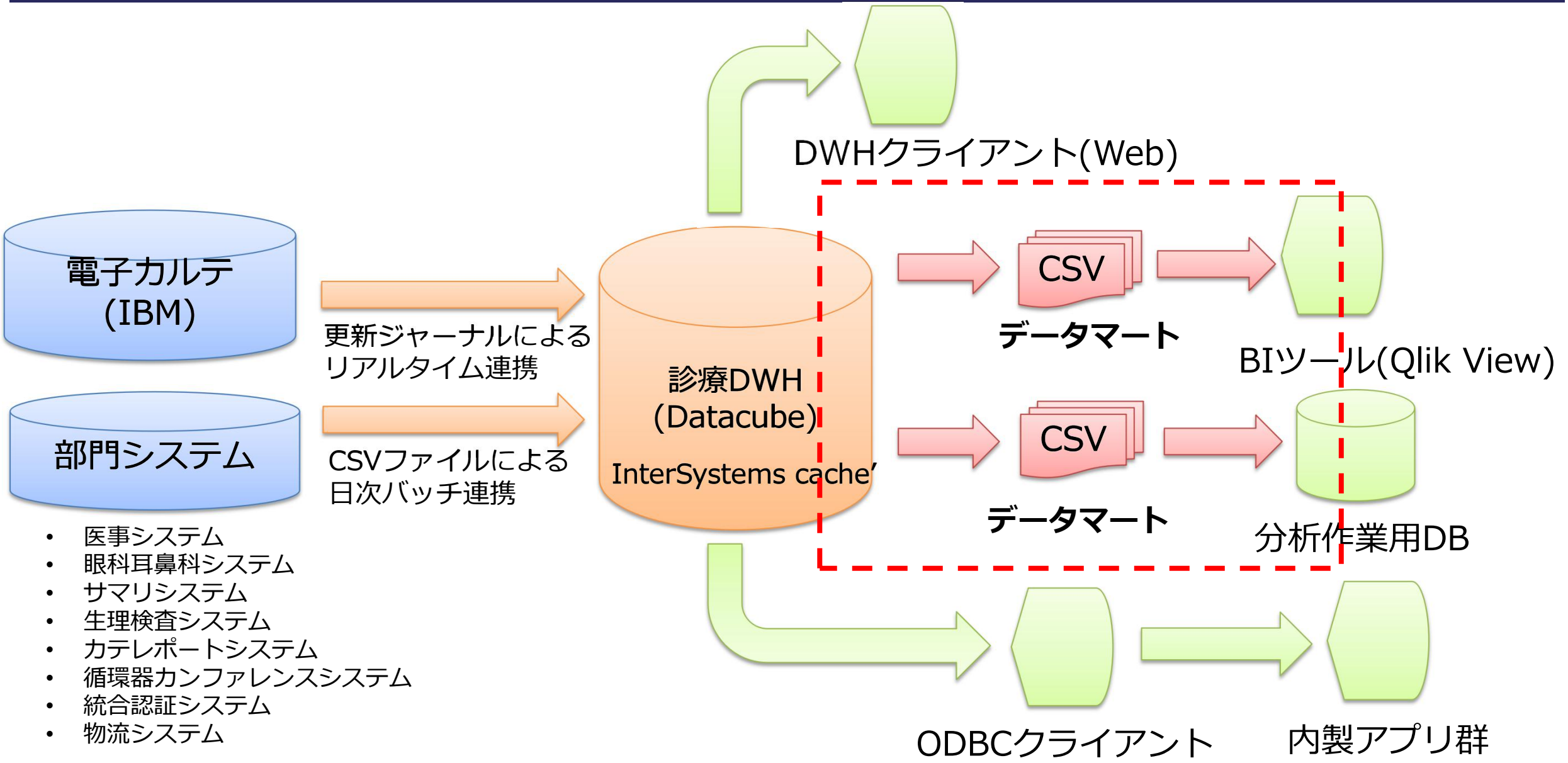
熊本大学病院　総合臨床研究部研究データ管理センター

COI開示　　私が発表する今回の演題について開示すべきＣＯＩはありません。

熊本大学病院について

- 病院紹介
  診療科数/部　　6部門　30科、中央診療部門28
  ベッド数　　845床(一般745床、精神50床)
- 診療実績(令和4年度)
  平均在院日数　　12.7日
  稼働率　　86.28%
  一日平均外来患者数　1453.2人
  一日平均初診患者数　　109.3人
  紹介率　　93.5%
  逆紹介率　　111.3%
- 病院情報システム
  H11 オーダリング導入
  H18 フィルムレス化
  H22 電子カルテ化
  H29 現行システム稼働
  R6/1 次期システム稼働予定

当院のデータ利活用環境



- 医事システム
- 眼科耳鼻科システム
- サマリシステム
- 生理検査システム
- カテレポートシステム
- 循環器カンファレンスシステム
- 統合認証システム
- 物流システム

ETL(Extract,Transform,Load)とは

様々なシステムのデータベースやファイルからデータを抽出し、目的に応じたデータに加工し、加工したデータを別のシステムに読み込ませるためのツールのこと。製品にもよるが基本的には各種データベース、CSV/TSVファイル、XMLファイル、JSONファイル、Excelファイルなどのデータを読み書き可能である。

ただし、専用の製品を使用しなくてもプログラミング言語はデータを加工・変換するのが主要な目的のため、これを用いて実現することも可能である。また、DBMS(Database Management System)やBI(Business Intelligence)ツールにもETL機能が付属していることが多い。

主な商用製品

Talend Data Fabric
https://www.talend.com/jp/
ASTERIA Warp
https://www.asteria.com/jp/warp/
Data Spider
https://www.hulft.com/software/dataspider
AWS Glue
https://aws.amazon.com/jp/glue/
Azure Data Factory
https://azure.microsoft.com/ja-jp/products/data-factory
GCP Dataflow
https://cloud.google.com/dataflow?hl=ja

主なオープンソース製品

Talend Open Studio
https://www.talend.com/jp/products/talend-open-studio/
Apache Nifi
https://nifi.apache.org/

ETL開発の背景

- BIツールを用いて医療安全に関連した臨床モニタリングを運用している。このモニタリング項目は医療の質・安全管理部が要件を取りまとめており、その内容をもとに総合臨床研究部がETLからBIまでの構築作業を行っている。

- 臨床研究の後ろ向き観察研究においては電子カルテや部門システムからのデータ抽出と匿名化、統計解析用データセットへの変換を行う必要があり、総合臨床研究部ではその支援作業を行っている。

※医療情報部門では抽出のみ担っている。

モニタリング指標毎や研究毎にETLを開発することになるため効率的な開発環境が必要

使用するETLの機能などを比較検討した結果、
プログラミング言語を用いて開発することにした。

ETLとしてプログラミング言語を採用した理由

熊本大学 Kumamoto University

- ✓ データの入出力がはっきりしている

  ⇒データの入出力はほぼ固定(ODBC or CSV)なので多様なアダプタは不要。

- ✓ 追加費用が不要である

  ⇒使用する言語はSQL、C#、Pythonだがどれも無償で開発環境を揃えることが可能。

- ✓ 処理の柔軟性が高い

  ⇒プログラムを記述することになるので自由度は一番高い。

- ✓ 部署内の限定的な利用である

  ⇒BIツールはライセンス数が少ないため院内全体への展開ができないので作成者は限定。

- ✓ 学習コストが低い

  ⇒もともとコードを書けるのでわざわざ新しいツールの習得時間は不要。

プログラミング言語を用いてETLを開発するが毎回イチから実装するのは効率が悪い。そのため、ETLの処理を部品化することで効率的に作業ができるETLフレームワークを開発してそれを利用することとした。

ETLにBIツールの機能を使用しなかった理由

最近のBIツールはほとんどの製品でもETL機能が存在する。基本的な機能はそろっており十分実現できることも多い。ただ、今回は下記の理由により採用しなかった。

- BIツールでETLを実装(特に変換の部分)してしまうと、BIツールと密結合された状態となりBIツールの変更が困難になる。
- BIツールは加工処理の共有が困難であり再利用性が低くなる。
- BIツールはデータと一体化されているためバージョン管理が困難である。

ETLにおけるSQL

ほとんどのデータベースはリレーショナルデータベースでありデータ抽出にSQLを使用している。 SQLは構文が簡単なので容易に習得ができること、集計や関数といった処理を組み込むことができることからETLを含めたデータ分析には必須のスキルである。とくにウィンドウ関数、UNION、サブクエリなどを組み合わせるとかなり複雑なことが実現できるようになる

しかし、正しい条件で抽出できたかどうかを検証する手段に乏しく間違いやすい。また、1つの塊として運用するため再利用性が低い。そのため、ある程度分割したSQLを組み合わせてETLを実現する方法を検討した。

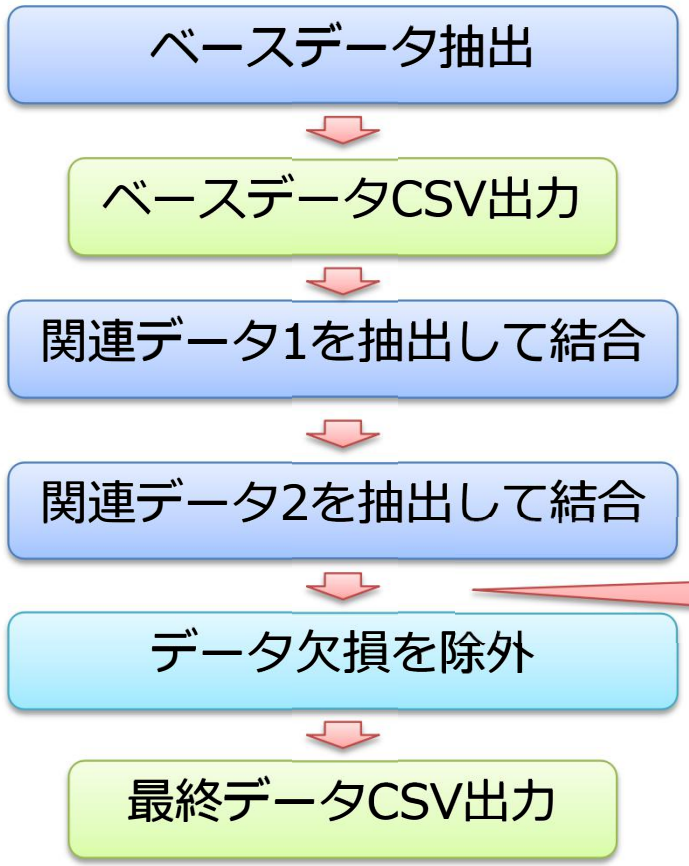
ETLフレームワークの概要

熊本大学
Kumamoto University

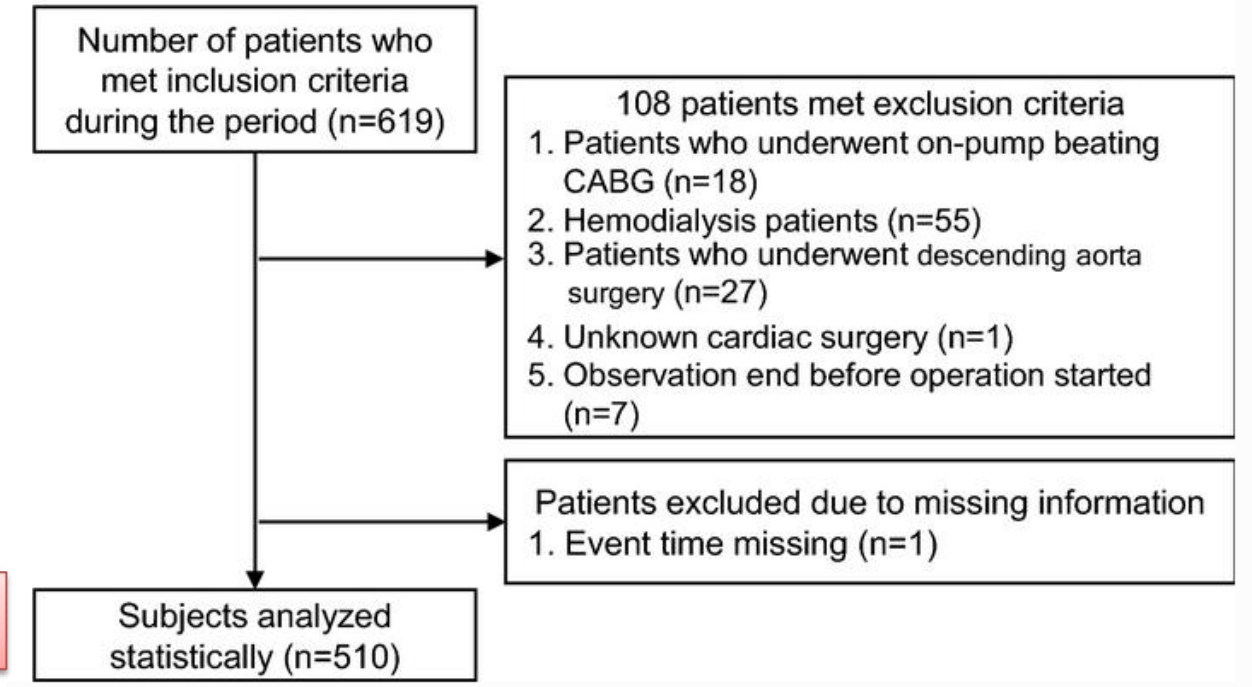
- データベースはODBC接続できるデータベースのみとする。
- ファイルはCSVファイルのみとする。
- SQLの実行結果をもとに次のSQLを実行して結果を結合できるようにする。
- データの加工は原則SQLで実現するがSQLで実現できない場合はフレームワーク側で特別処理を実装する。
- SQL文も含めたETLの定義情報は全てJSONファイルに保存し、起動時にJSONファイルを読み込むことで目的別のETL処理を実現する。
- 全てインメモリ内で行う。

ETLフレームワークの考え方

臨床研究において解析対象集団の選定方法を図示するSelection Flow Chartがある。
これを参考に抽出条件と選択条件の途中経過が判別できるようETLフェーズをコンポーネント化し組み合わせで実行できるように設計した。



途中でワークテーブルをCSVに出力可能



Subjects Selection Flow Chartの例

Okadome Y, Morinaga J, Yamanouchi Y, et al. Increased numbers of pre-operative circulating monocytes predict risk of developing cardiac surgery-associated acute kidney injury in conditions requiring cardio pulmonary bypass. Clin Exp Nephrol. 2023;27(4):329-339.

主なコンポーネント

- SQL読込
- SQL読込&テーブルジョイン
- CSV読込
- CSV読込&テーブルマージ
- CSV読込&テーブルジョイン
- Webサービス読込&テーブルジョイン
- 縦横変換(Flatten)
- 条件フィルタ
- テーブルコピー
- CSV出力

```
[
  {
    "ActionType":"LoadFromODBC",
    "ConnectionName":"KuhDwh",
    "OutputName":"NSBODYRESTRAINT1",
    "SqlTextLines": [
      "SELECT N.PTID,N.NYUINDATE,N.NYUINTIME,N.TAIINDATE,N.TAIINTIME,N.NYUIN
      "FROM m3_data_table_kuh_his.TNYUIN N",
      {"DataType":"Date","Name":"EDDT1","Value":"@ARGS2"},
      {"DataType":"Date","Name":"EDDT2","Value":"@ARGS2"},
    ]
  },
  {
    "ActionType":"LoadJoinFromODBC",
    "ConnectionName":"KuhDwh",
    "JoinType":"First",
    "SourceName":"NSBODYRESTRAINT1",
    "OutputName":"NSBODYRESTRAINT2",
    "SqlTextLines": [
      "select ei.ptid,ei.hasseidate,ei.seqno,ei.wsno,ei.indexkbn,ei.xxkbn,ei.xxsybt,ei.xxseq,et.soapsybt,et.soapseq,",
      " et.recordtype,ei.drid,ei.nsid,ei.dispflg,et.cancelflg,et.linkinfo,ei.startdate,ei.exectime,ei.idxtitle,ei.idxtext,et.emrstatus",
      "from m3_data_table_kuh_his.TEMRIDX ei",
      "join m3_data_table_kuh_his.TEMR et",
      " on ei.ptid=et.ptid and ei.hasseidate=et.hasseidate and ei.seqno=et.seqno and ei.wsno=et.wsno and ei.indexkbn=et.indexkbn and ei.xxkbn=et.
      "where ei.ptid=? and ei.startdate>=? and ei.startdate<=? and ei.indexkbn='1' and ei.xxkbn='20' and ei.dispflg='1' and ei.stopflg<>'1' and et
```

```
Windows PowerShell
PS C:¥病院情報¥医療の質安全管理部> C:¥病院情報¥共通ツール¥KUHDWH¥KuhDwhTransferTool¥KuhDwhExtractProcessTool¥bin¥Debug¥K
uhDwhExtractProcessTool.exe ".¥予期せぬ再手術データ抽出.json" "2022-11-01" "2022-11-30" "予期せぬ再手術データ202211.csv"

2023-06-26 17:01:57,785 [1] INFO  KuhDwhExtractProcessTool.ExtractProcess - Initialize ExtractProcess
2023-06-26 17:01:57,863 [1] INFO  KuhDwhExtractProcessTool.ExtractProcess - - Read Config File:.¥予期せぬ再手術データ抽
出.json
2023-06-26 17:01:58,051 [1] INFO  KuhDwhExtractProcessTool.ExtractProcess - Start ExtractProcess
2023-06-26 17:01:58,051 [1] INFO  KuhDwhExtractProcessTool.ExtractProcess - - Step 1:LoadFromODBC
2023-06-26 17:01:58,067 [1] INFO  KuhDwhExtractProcessTool.OdbcDataLoader - -- Load Data: OPEDATA_0
2023-06-26 17:02:03,895 [1] INFO  KuhDwhExtractProcessTool.ExtractProcess - - Step 2:LoadJoinFromODBC
2023-06-26 17:02:03,898 [1] INFO  KuhDwhExtractProcessTool.OdbcDataLoader - -- Load Join Data: OPEDATA_1 base OPEDATA_0
2023-06-26 17:02:25,395 [1] INFO  KuhDwhExtractProcessTool.ExtractProcess - - Step 3:LoadJoinFromODBC
2023-06-26 17:02:25,395 [1] INFO  KuhDwhExtractProcessTool.OdbcDataLoader - -- Load Join Data: OPEDATA_2 base OPEDATA_1
2023-06-26 17:02:33,770 [1] INFO  KuhDwhExtractProcessTool.ExtractProcess - - Step 4:WriteToCsvFile
2023-06-26 17:02:33,778 [1] INFO  KuhDwhExtractProcessTool.CsvDataWriter - -- Write CSV: Source=OPEDATA_2, File=予期せぬ
再手術データ202211.csv
2023-06-26 17:02:33,817 [1] INFO  KuhDwhExtractProcessTool.ExtractProcess - End ExtractProcess
PS C:¥病院情報¥医療の質安全管理部>
```

ETLフレームワークのソースコード

熊本大学
Kumamoto University

今回開発したETLフレームワークのソースコードをgithubに公開しているので興味のある方はご参照ください。

データベースの接続部分やコンバーター部分など当院専用に開発している部分を除外しているので必要に応じて変更してください。

実際に動作可能な定義ファイルもDBレイアウトの開示になるので非公開です。
サンプルの定義ファイルはプロジェクト内に含まれています。

https://github.com/eolla1013/KuhDwhExtractProcessTool

ETL開発のながれ

熊本大学
Kumamoto University

1. データ表現のイメージをつかむ。
2. 何をもって1レコードとするか決める。
3. データソース側に必要なデータが存在するか調査する。
4. データソース側のテーブルレイアウトと分析用テーブルレイアウトを並べてどう変換するか検討する。
5. 変換方法をSQLの定義ファイルに落とし込む。
6. 小規模データで実行し正しく変換されていること確認する。
7. 過去データをもとにまとめて変換する。
8. 未来データを変換するためにタスクスケジューラなどで定期的にロードできるようにしておく。

事例1　予期せぬ再手術の状況確認

熊本大学 Kumamoto University

■ 背景と目的

DPC様式1に予期せぬ再入院があるが、もう少し診療に直結した質管理の一環として予期せぬ再手術を予定外の手術とみなしその発生状況を確認する。手術後24時間以内と7日以内に予定外の手術があればその手術情報を出力する。最終的にはカルテレビューを行うが全手術患者では数が多く対応しきれないため、対象患者のスクリーニングが必要となった。

■ 入力元

手術実施テーブル

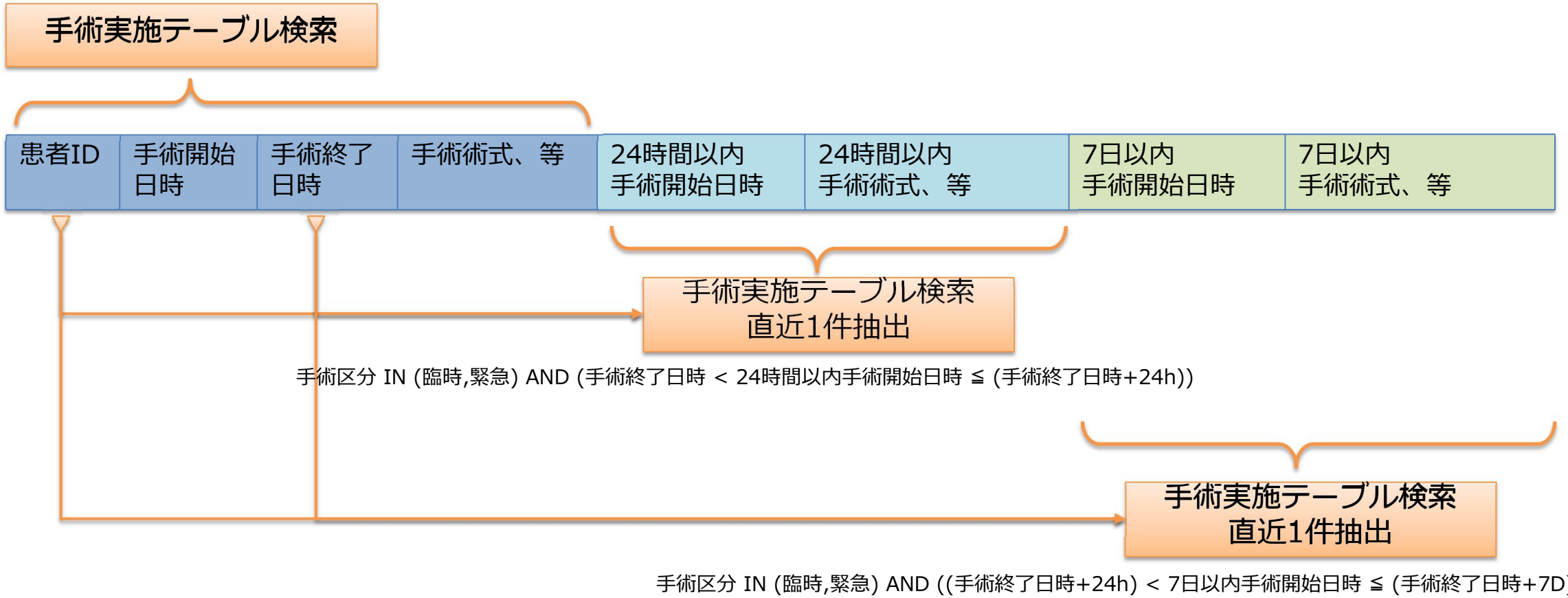
■ 出力先

CSVファイルで出力する。その後QlikView(BIツール)が読み込んで可視化する。

■ データ粒度

手術1件につき1レコードとする。

事例1　予期せぬ再手術の状況確認



事例2　CVC実施記録確認

■背景と目的

CVC(中心静脈カテーテル挿入)は重篤な合併症を引き起こす可能性があるため適切な実施と実施後の観察が求められる。当院ではCVC施行に院内認定資格が必要であり認定資格は1年更新制である。そのため、電子カルテのテンプレート機能でCVC施行記録を登録しておき、認定資格者が適切に施行していること、資格更新に必要な施行回数があることを確認する必要があった。

■入力元

カルテテンプレートテーブル

■出力先

CSVファイルで出力する。その後QlikView(BIツール)が読み込んで可視化する。

■データ粒度

テンプレート1件につき1レコードとする。

事例2　CVC実施記録確認

カルテテンプレート検索(明細結合済)

検索開始日時 ≦ 記載日時 AND 記載日時 ≦ 検索終了日時 AND テンプレートID IN (CVC実施記録テンプレートID)

患者ID	記載日時	記載者	テンプレートID	テンプレート名	テンプレート項目ID	テンプレート項目名	テンプレート項目入力値

テーブル縦横変換

患者ID	記載日時	記載者	テンプレートID	テンプレート名	施行日	術者	インストラクター	補助者	穿刺部位、等

事例3　身体抑制実施記録整合性確認

■背景と目的

患者の身体抑制は人権擁護の観点から原則禁止である。しかしながら転倒転落や点滴自己抜去などの安全面において大きな支障をきたす場合のみ一時的に許容されている。そのため、身体抑制指示や評価、抑制中の観察内容は適切に記録される必要がある。当院では抑制の指示は継続指示オーダ、開始、継続、終了の評価と記録は電子カルテのテンプレートを用いてそれぞれ記録しているがその整合性が取れているかどうかを調査する必要があった。

■入力元

入院テーブル、継続指示オーダテーブル、カルテテンプレートテーブル、患者在院検索Webサービス

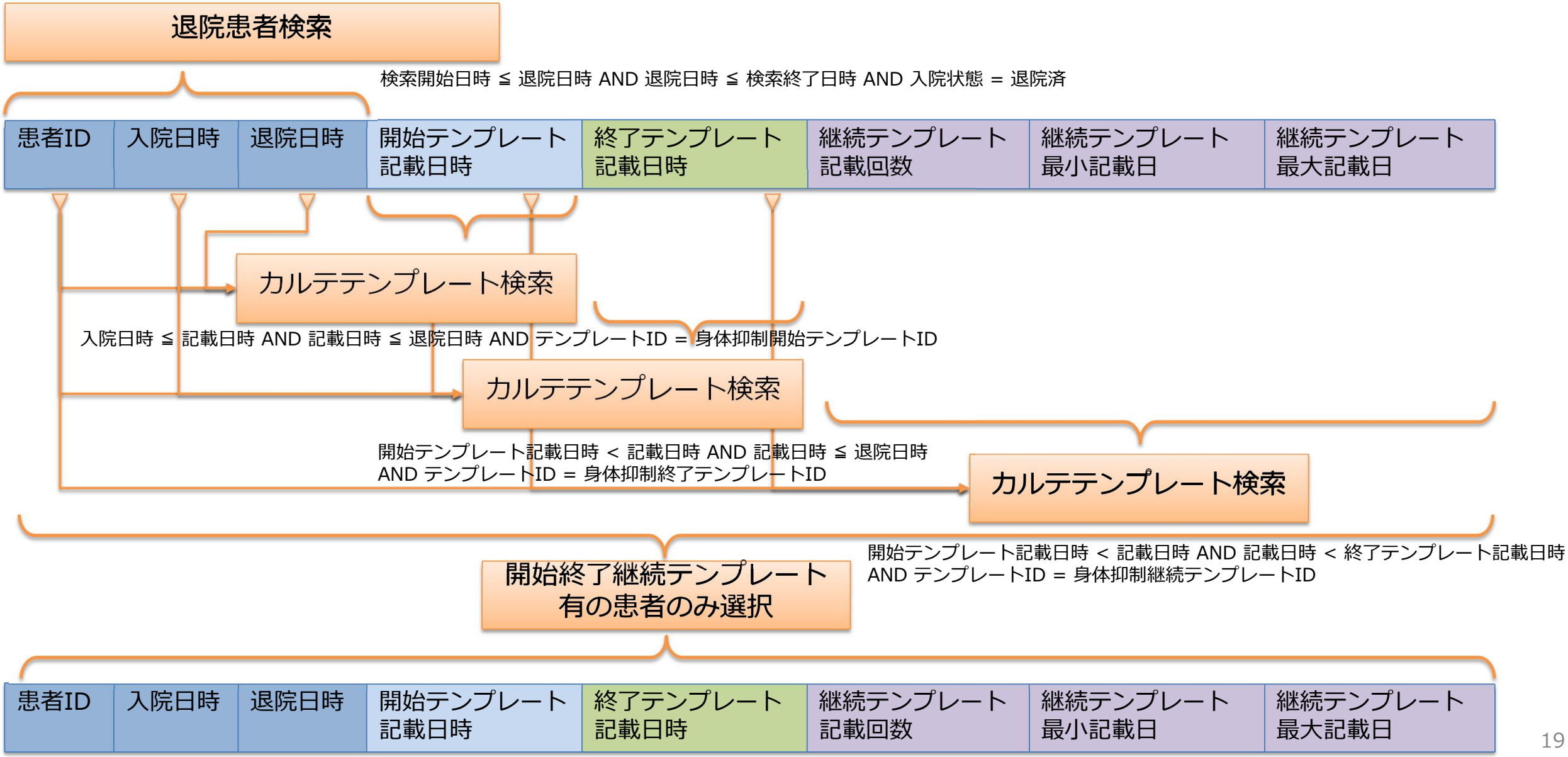
■出力先

CSVファイルで出力する。その後QlikView(BIツール)が読み込んで可視化する。

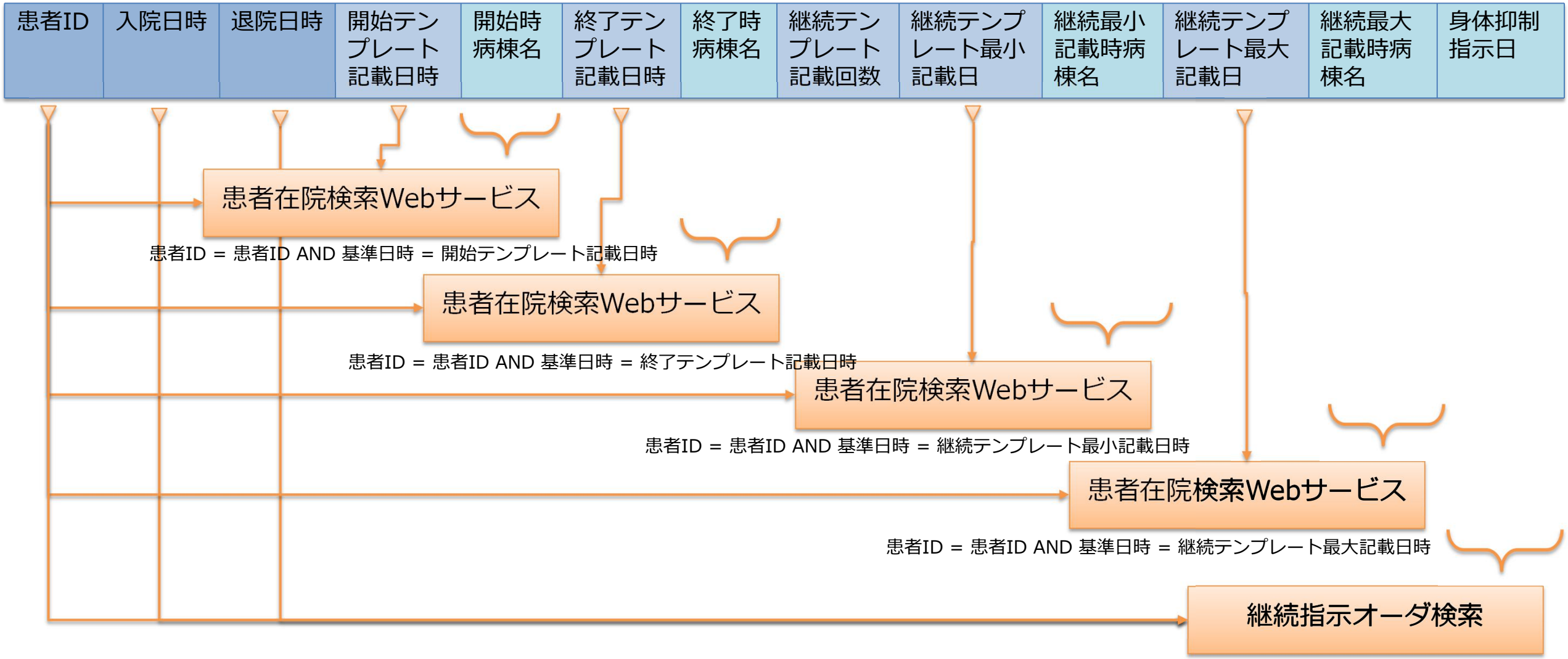
■データ粒度

入院1件につき1レコードとする。

事例3　身体抑制実施記録整合性確認



事例3　身体抑制実施記録整合性確認



まとめ

現行システムにおいてBIツール向けのETLは自作のETLフレームワークを用いている。このフレームワークの特徴である定義ファイル内にSQLを記載していく方式は作業の効率化に寄与している。

現在このツールを用いたモニタリング項目は14件運用している。

ただ、使いこなせる人間が限定されている現状がある。特に臨床評価指標の可視化は院内でも重要視され始めており環境の更新と担い手の確保が課題である。

次期システムに向けて

2024年1月に稼働予定の次期病院情報システムにおいて次のような機能強化を予定している。

- ✓ BI機能を有するDWHに更新
- ✓ SDMの新規導入
- ✓ ESB(Enterprise Service Bus)の新規導入

ESBはノーコードでシステム間のデータ連携を実装できるため、高度な専門的知識を持っていなくてもETLの開発が可能である。

また、院内人材の教育もこのシステムの導入に合わせて実施する予定となっており、更なるデータ利活用が進む予定である。

次期システムのデータ利活用環境(予定)