情報処理学会論文誌　デジタルプラクティス　Vol.5　No.2　20–34 (Apr. 2024)

デジタルバンクにおけるデータマート構築に関する実践：みんなの銀行を事例として

神辺 圭一[1,a)]　江里口 剛喜[2,b)]

受付日 2023年9月15日, 採録日 2024年1月19日

概要：本稿は，日本初のデジタルバンク「みんなの銀行」におけるデータ分析基盤について，主にデータウェアハウス（DWH）の観点から記したものである．DWH に連携された様々なデータは，前処理を経て「データマート」と呼ばれる用途別のテーブルに格納される．筆者らは，データ分析に携わるデータサイエンティストの立場から用途の異なるデータマートを効率的に構築するための手法を考案し，実践した．また，構築したデータマートを用いて社内のデータ利活用業務を推進し，「データの民主化」にどのような形で貢献したかについて述べる．

キーワード：データウェアハウス，DWH，データマート，データの民主化，デジタルバンク

Practice on Data Mart Design in a Digital Bank: Case Study of Minna Bank

Keiichi Shinbe[1,a)]　Goki Eriguchi[2,b)]

Received: September 15, 2023, Accepted: January 19, 2024

Abstract: This paper describes the data analysis infrastructure of the first digital bank in Japan "Minna Bank", mainly from the perspective of data warehouse (DWH). Various data added to the DWH are stored in tables for specific purposes called "Data Marts" after preprocessing. We devised and implemented methods for efficiently creating Data Marts from the perspective of data scientists who engage in data analysis. In addition, we will discuss how we promoted data utilization with Data Marts and contributed to "Data Democratization" in the company.

Keywords: data warehouse, DWH, data mart, data democratization, digital bank

1. はじめに

近年，企業が保有する様々なビッグデータを一元管理するための基盤であるデータウェアハウス（以下 DWH）の重要性が高まっている．DWH とは，意志決定を支援するための「サブジェクト指向で構成され，統合化され，時系列で，恒常性を持つデータの集合体」[1] とされ，基幹系システムから分析用のデータを抽出し，逐次的に蓄積する「データの倉庫（ウェアハウス）」に相当する．DWH では，基幹系システムで発生したレコードの追加や更新，削除をテーブルに「追記」する．DWH のデータベースは時系列で保存され，原則として変更や削除をしないため，基幹系システムの現在のデータの状態によらず，任意の期間を対象とした分析が可能となる．

DWH の動作基盤にはオンプレミスのアプライアンスが利用されるほか，近年は PaaS（Platform as a Service）と呼ばれるクラウドサービスによる利用形態も増えている．クラウドを用いた DWH は，①ストレージの容量拡張が容易であり，②分析処理に必要な計算リソースを即時的に確保できるといった利点がある．

DWH に格納されたデータは構造化されており[*1]，多くのシステムでは標準 SQL を用いて操作することができる．DWH には全社的な情報が集約・一元管理されているた

1　（株）みんなの銀行（現所属：福岡工業大学）
Minna Bank, Ltd., Fukuoka city, Fukuoka 810–0002, Japan
(Present Affiliation: Fukuoka Institute of Technology)
2　（株）みんなの銀行
Minna Bank, Ltd., Fukuoka city, Fukuoka 810–0002, Japan
a)　shinbe@fit.ac.jp
b)　g.eriguchi@minna-no-ginko.com

[*1] DWH の前段に動画や音声，テキストといった非構造データも包括して記録するための「データレイク」を設置するケースもある．

め，社内の膨大な情報にワンストップでアクセスすることが可能である．一方，DWH には様々な情報が格納されているため，エンドユーザが目的のテーブルを探し出し，必要なデータを取り出すには手間を要する場合がある．そこで，作業の効率化のためデータベースから特定の用途（サブジェクト）に必要なデータを抽出し，整形・加工したサブセットをあらかじめ作成することがある．これは「データマート」と呼ばれ，企業であれば組織全体で KPI[*2] などの数値を共有するために作成するケースや，部門などの業務に合わせて設置することが多い．データサイエンティストやデータエンジニア，データアナリストといったデータの専門家が分析の効率化ために作成したビューやテーブルも再利用性がある場合はこの中に含まれる．

DWH にデータマートが存在しない場合，エンドユーザは用途ごとに整形・加工される前のテーブルから手動でデータを取得し，不要なカラムやレコードを除外する処理を自前で行う必要がある．そのため，作業者の「解釈の相違」により，データの取得結果や集計値に差異が生じる恐れがある．社内の専門家がデータマートを作成し，統一的なデータ参照先としてエンドユーザに提供することは，データの品質管理の観点からも有用である．

組織のメンバ全員が必要なときに必要なデータへアクセスし，利活用できる状態を「データの民主化」という [2]．データの民主化を推進するには，社内の誰もがデータに“簡便に”アクセスできる環境の整備が必須である．用途別に作成されたデータマートはデータの民主化を推進するための強力なツールであるが，DWH 上のデータマートを参照するには通常 SQL 言語による操作が必要であり，専門家以外には敷居が高い．そのため，データマートにアクセスする手段として，BI（Business Intelligence）ツールをフロントエンドアプリケーションとして導入するケースが多い．この場合，データマートは BI ツールのバックエンドデータベースとして動作する．

BI ツールの可視化機能（ダッシュボード）を利用することで，エンドユーザは GUI 操作でデータを探索し，簡易的な集計等を自分自身で行うことが可能となる．一方，BI ツール向けに最適化されたデータマートを個別に開発・運用するには相応の工数を要するため，データ分析や社内業務向けに作成したデータマートを転用したり，データマート化する前のテーブルを BI ツールから直接参照することがある．この場合，BI ツールの機能を用いてデータ整形や加工を行うことになるが，以下の問題が生じやすい．

- 複数のテーブルやデータマートを BI ツール上で動的に結合（JOIN）すると，単独のデータマートで処理するよりも性能面のオーバヘッドが生じやすい
- BI ツールが読み込むデータ量（レコード数）が多い場合メモリがオーバフローし，処理が中断することがある
- 大量のレコードに対する集計処理を BI ツール上で動的に行うと，処理完了までに長時間要する場合がある
- テーブルやデータマートに利用頻度の低いカラムが含まれていることがあり，処理速度の低下やデータ書き出し項目の選択時に混乱を招きやすい

そこで，筆者らは DWH 上のデータマートを「階層化」および「部品化」することで再利用性を高め，業務特化型と BI ツール用のデータマートの構成要素を可能な限り共通化した．これにより，比較的少ない工数で BI ツール用のデータマートを作成することが可能となり，上記の問題について一定の解決策を得た．本稿では，その具体的な手法について述べる．さらに筆者らが作成したデータマートが社内のデータ民主化にどのように貢献したかについて言及し，効果と今後の課題について考察する．

2. DWH のデータモデリング手法

ビル・インモンは，データの正規化を重視し，冗長性を可能な限り回避するため，企業全体のデータを一か所に集約（1 Fact 1 Place の原則）し，トップダウンでデータマートを構築するアプローチ [3] を 1990 年代に提唱した．全社的なデータが統合かつ正規化されることで冗長性が排除され，業務プロセスが明確になるが，DWH の設計に要する初期コスト（時間・開発工数）が大きくなりやすい．また，テーブル間の参照関係が複雑になり，エンドユーザがデータ利活用するための習熟コストも高くなる傾向にある．

ラルフ・キンボールは，データの前処理を行う「ステージング領域」とデータマートを配備する「プレゼンテーション領域」の 2 層構造からなる DWH アーキテクチャ [4] と，ディメンショナルモデルと呼ばれるデータモデル [5] を提唱した．ディメンショナルモデルとは，分析対象の値を含むファクトテーブルと分析軸となる値を持つディメンションテーブルから構成される．本モデルはビジネス要件や分析観点から必要な要素を結合してデータマートを構築するボトムアップのアプローチであり，データの正規化を追求しない点がインモンの手法とは異なる．個々のデータマートの構造が単純なため構築コストを低く抑えられる一方，局所最適なデータマートが乱立する恐れがある．

ディメンションテーブルのレコード更新を行う手法に，スローリーチェンジングディメンション（以下 SCD）がある．SCD には Type 0 から Type 7 までの手法が定義されている [4]．その中で，データの変更が発生するたびに新規レコードを作成し，新旧データを行として保持する Type 2 が多用される．Type 2 には複数の実装方法がある

[*2] Key Performance Indicator（重要業績評価指標）の略．

が，ディメンションテーブルのレコードに開始日・終了日のカラムを定義し，この日付の差異で状態の変化を検知する方式が一般的である．開始日カラムには，レコードの変更が生じた日付を登録し，終了日カラムには null または 2099-12-31 といったシステムの寿命を超えた未来の日付を一旦登録する．基幹系システムのデータに変更が生じると，DWH のディメンションテーブルにレコードが 1 件追加され，先と同様の手順で開始日・終了日を設定する．さらに，変更前のレコードの終了日を再設定することで，基幹系システムのデータがいつからいつまで有効であったかを DWH 上で把握することが可能になる（**図 1**）．

ダン・リンステットは DWH のアーキテクチャを 3 層に分けた DataVault 2.0（以下 DV2.0）と呼ばれるデータモデリング手法 [6] を提唱した．DV2.0 では，「ステージング領域」と「プレゼンテーション領域」の中間に「Data Vault 領域」（以下 EDW[*3]）を設ける．EDW には Hub / Link / Satellite の 3 種類のテーブル群から構成されたデータモデルを構築する．DV2.0 は複数のデータソースへの対応が容易であり，監査性やスケーラビリティに優れ，クラウド DWH との親和性が高い．一方，ステージング領域のテーブルを EDW で効果的に分割（疎結合）にするには，業務プロセスの理解とドメイン知識を有したデータエンジニアの関与が不可欠であり，データマート構築の難易度や保守コストが相対的に高くなるといった課題もある．

DWH では，「大福帳型」と呼ばれるモデルも利用される [7]．大福帳モデルは，トランザクションデータとマスタデータが 1 つのテーブルに統合された状態でデータを保持する．表計算ソフトのようにフラットなデータ構造であることから専門家以外にも分かりやすく，簡易的なデータ分析や抽出に適しているといった利点がある．テーブルが正規化されていないため，データが冗長化するといったデメリットはあるものの，DWH で利用されることが多い列指向データベースとの親和性が高い[*4]．また，ストレージ単価の安いクラウド DWH ではデータの重複についてもコスト観点から許容されるケースが多いと考えられる．

筆者らのデータマートでは，主にキンボールの DWH アーキテクチャを参考にしたが，データ構造はエンドユーザの利便性に考慮して大福帳型に近い形態を取る．

顧客ID	ステータス	レコード更新日	開始日	終了日
AAA	0	2023-07-01	2023-07-01	2099-12-31

↓

顧客ID	ステータス	レコード更新日	開始日	終了日
AAA	0	2023-07-01	2023-07-01	2023-07-02
AAA	1	2023-07-03	2023-07-03	2099-12-31

本例では、**2023年7月3日**に 顧客ID＝AAA のステータスが 0 から 1 に変化したことでDWHにレコードが追加された。同時に一つ前のレコードの終了日が書き換わった

図 1　SCD Type 2 方式による DWH への変更内容の反映例
Fig. 1　Appending data to DWH by SCD type 2.

[*3] Enterprise Data Warehouse の略．

3.　みんなの銀行のデータ分析基盤

みんなの銀行[*5] は，2021 年 5 月に一般開業した日本初の「デジタルバンク」[8] である．デジタルバンクとは，「デジタル起点で発想し，ゼロベースで設計された次世代の銀行」[9] を指す．みんなの銀行は，1981 年以降生まれのデジタルネイティブ世代（ミレニアム世代および Z 世代）を主なターゲットとしてスマートフォン完結型の銀行サービスを提供しており，スマートフォンアプリケーション（以下スマホアプリ）は 2023 年 8 月現在 220 万ダウンロードを記録し，口座開設数は 75 万件を突破した．

みんなの銀行には有人店舗や通帳，キャッシュカードは存在しない．署名捺印を伴う書類手続きや郵送といった非デジタルのチャネルを経由せず，スマホアプリからいつでも直接口座開設することができる．残高照会，送金，貯蓄預金，ローン等の銀行機能はすべてスマホアプリからのみ利用可能であり，Web ブラウザ経由の「インターネットバンキング」機能は提供していない．現金の入出金は，全国のセブンイレブンに設置された（株）セブン銀行の ATM が利用可能であり，キャッシュカードの代わりにスマホアプリを介して認証する．

みんなの銀行の勘定系システムは，米 Google 社の IaaS（Infrastructure as a Service）である Google Cloud 上に構築され，24 時間 365 日の連続運用を行っている [10]．勘定系のデータベースにはフルマネージドリレーショナルデータベースサービスである Cloud Spanner [11] を採用し，東京と大阪のデータセンターに常時同じデータを同じタイミングで書き込む「東阪両現用」の仕組みを構築することで，大規模災害などへの耐性を高めている．

データ分析基盤である DWH には，同じく Google Cloud のサービスである BigQuery [12] を利用している．BigQuery は，データがカラム単位で保存される「列指向データベース」であり，TB〜PB クラスのビッグデータへ効率的にアクセスすることができる [13]．BigQuery では，一般的なリレーショナルデータベースと同様にテーブルやビューを定義することができ，標準的な SQL 構文が使用可能である．一方，①キー制約が存在しない，②データ型の種類が勘定系のリレーショナルデータベースと比べ

[*4] 列指向型のデータベースでは，列方向（カラム）の呼び出しは行方向よりもリソースを必要とせず，都度結合を必要としない大福帳型のテーブルはパフォーマンスの観点からも有利である．
[*5] https://www.minna-no-ginko.com/

て少ないといった相違がある．利用料は，データの保存量やクエリ実行時の計算リソースに対して発生する．

本章では，筆者が属するみんなの銀行データサイエンティストチーム（Data Creation Group，以下 DCG）が DWH 上に構築したデータマートの特長について述べる．二重かぎ括弧の付いたワードは，本 DWH 固有の用語である点に留意されたい．

3.1　データマートの設計方針

DCG が DWH に構築するデータマートは，以下の要件を満たすものとして設計した．

■データモデリング

A. エンドユーザが必要とする項目が網羅され，不要な項目が除かれること
B. データの齟齬が発生しない設計が行われること
C. 作成者（DCG）が同一の処理を繰り返し行わずに済むように，再利用可能な構造を有すること．
D. 用途の異なるデータマートを効率的に作成できること

■データアーキテクチャ

E. アクセス手段によらず，選択・集計処理が可能な限り短時間で終わること
F. DWH や BI ツールの計算リソースの範囲で動作すること
G. PaaS の利用料を可能な限り節約すること
H. 将来にわたって計算リソースやストレージが不足せずに利用できること

■データクオリティ

I. 明確な命名ルールが定められていること
J. 定義内容を適切に管理するための仕組みを有すること

■データセキュリティ・ガバナンス

K. 適切なアクセス制御が行われること
L. 適切な個人情報保護や情報漏洩対策等の安全管理措置が取られること

DCG が取り組むデータマートでは各項目を必須要件と考えているが，特に C・D・E の実現を重視した．そのため，エンドユーザの利用可能な分析軸の組み合わせを固定化し，分析粒度に「日次」単位といった制約を加えるといった，データマートのモデリングに関する工夫を行い，開発の効率化や処理の高速化を図っている．

3.2　スタースキーマと逆スタースキーマ

第 2 章で触れたディメンショナルモデルでは，ファクトテーブル（以下 FT）と呼ばれる分析対象データ（例：取引明細，ログイン履歴）とディメンションテーブル（以下 DT）に当たる分析軸（例：顧客マスタ，カレンダマスタ）が，外部キーと主キー[*6] で結合（JOIN）する構造を有する．本モデルは，FT を中心に複数の DT が結びつく構造から「スタースキーマ」とも呼ばれ，FT と DT のレコードの対応関係は通常 N:1 となる[*7]．スタースキーマに基づいて設計されたデータマートでは，DT 由来の分析軸を組み合わせることで柔軟な多次元データ分析（OLAP; Online Analytics Processing）[*8] が可能となる．一方，BI ツール上で分析軸を選択するたびに集計処理が始まるため，FT のレコード数によっては動作が緩慢になることも考えられる．

そこで筆者らはディメンショナルモデルの主従関係を逆に捉え，DT ごとに FT を結合した『逆スタースキーマ』型のデータマートを DWH へ構築することにした．逆スタースキーマとは，以下の手順で作成したデータマートを指す．

1. DT のレコードを主キー（例：顧客 ID）ごとに 1 日単位に分割し，日付カラムを追加．SCD Type 2 の開始日・終了日の情報を元に，各日の最終状態を補完（3.5 節参照）．すなわち，主キー 1 件あたり，1 日 1 レコードが生成される[*9]
2. FT のレコードを DT の外部キーで日別に集約（GROUP BY）し，日付カラムを追加．「合計」や「イベント発生回数」[*10] といった集計値に変換したうえで，両テーブルが 1:1 で結合するように前処理を行う
3. DT と FT を主キー・外部キーで結合．キーには，DT 由来のもの（例：顧客 ID）に加え，1. で生成した日付カラムも含まれる

本方式は分析軸や分析粒度がある程度定型化していることが前提になるが，DT と FT の対応関係が原則として 1:1 であり FT 側で事前集計が可能となることから，処理時間や計算リソースの節約につながることが期待される【要件 E・F】．

スタースキーマと逆スタースキーマの概念図を**図 2** に示す．次節からは，逆スタースキーマに基づくデータマートの設計手順の詳細について述べる．

3.3　DWH の論理構成

みんなの銀行の DWH は，大きく『データ格納層』『データ参照層』『データマート層』の 3 層から構成される

[*6] 代替キーを用いることもあるが，本稿では「主キー」で表記を統一する．

[*7] たとえば，顧客の取引明細テーブルを FT とした場合，同一顧客の取引レコードは FT に複数件存在しうるが，DT の顧客マスタには，取引発生時点で有効な顧客レコードは通常 1 件しか存在しない．

[*8] 多次元データ分析では，「スライシング」「ダイジング」「ドリル」といった解析手法を用いる．

[*9] 分析粒度をより細かくするには，たとえば 1 時間単位の最終状態を取得することも考えられる．この場合，主キー 1 件あたり 1 日 24 レコードが生成される．

[*10] たとえば，特定のサービスを利用した回数を顧客 ID（外部キー）ごとに日別で集計する．

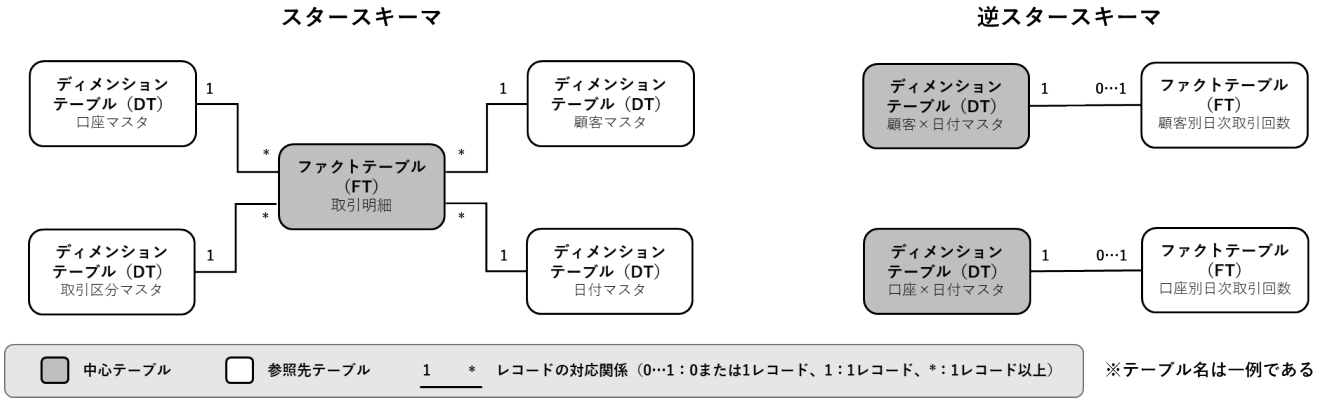

図 2　スタースキーマと逆スタースキーマの概念図

Fig. 2　Conceptual diagram of star schema and reverse star schema.

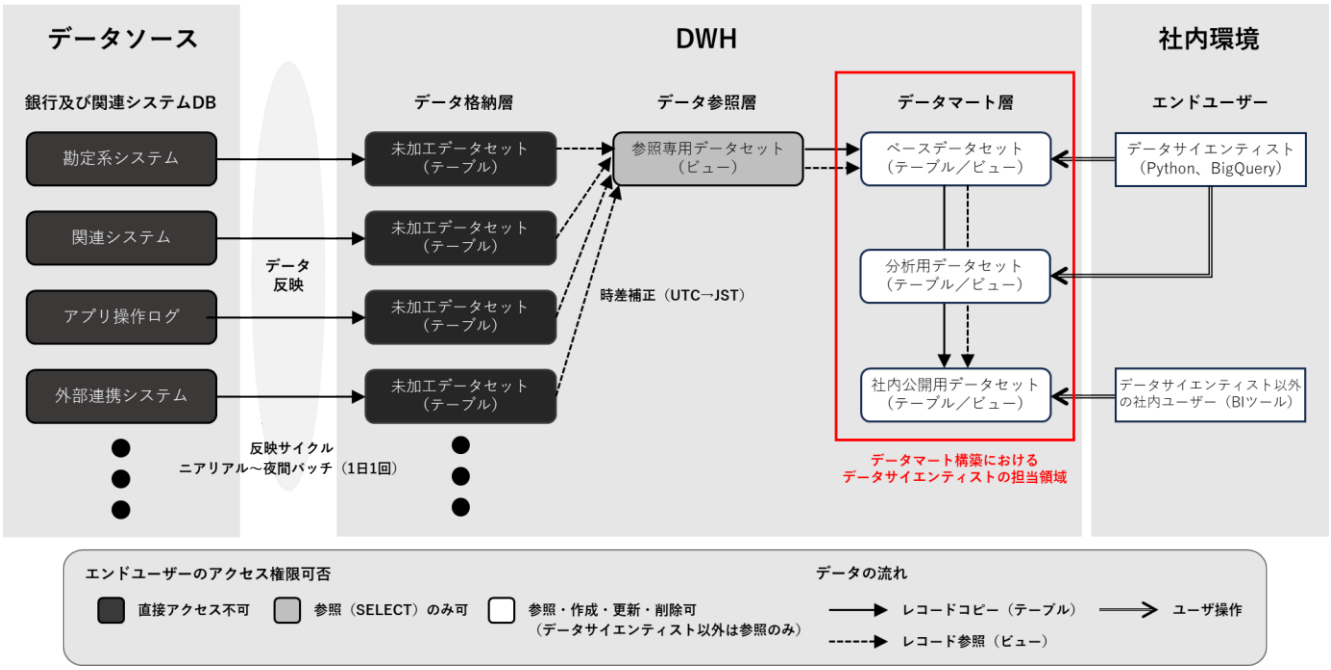

図 3　みんなの銀行 DWH の概略図とデータの流れ

Fig. 3　Overview of data flow in Minna Bank.

（**図 3**）.

DWH の動作基盤である BigQuery では，テーブルやビューの集合体を「データセット」と呼ばれる単位で管理する[*11]．BigQuery では，データセットまたは内包するテーブル・ビュー単位でアクセス制御を行うことが可能である．

DWH の最初の階層に当たる『データ格納層』には物理テーブルを配置し，「データソース」に相当する勘定系システムや関連システム，スマホアプリ操作ログ，連携する外部サービス（SaaS）で発生した各種データを可能な限りオリジナルのデータ形式を保ったまま取り込む[*12]．

データ格納層のデータセットはおおむねデータソース側のデータベース単位で作成され，発生源（データソース）と書込先（DWH）のテーブルは原則一対一で対応する．データの反映サイクルは，数分程度の時間差（ニアリアル）から最長で 1 日 1 回（夜間バッチ処理）であり，業務上の重要度やデータソース側の仕様により頻度が決定される．

『データ参照層』と呼ばれる階層のデータセットには，データ格納層のテーブル群を網羅的に SQL のビュー形式へ変換したものが配置される．本階層のデータセットは 1

[*11] BigQuery には，「プロジェクト」と呼ばれる最上位層が存在し，各データセットはいずれかのプロジェクトに属する．利用料はプロジェクト単位で集計される．

[*12] DWH のデータ連携日時を記録したカラムの付与や BigQuery で利用可能なデータ型への変換といった処理は付随的に行われる．

つであり，データ格納層では異なるデータセットに保存されていたテーブルが同一場所にビューとして定義される．ビューを用いると 2 つ以上のテーブルを結合して仮想的な表を作成することが可能であるが，データ参照層のそれはデータ格納層の物理テーブルと一対一で対応しており，テーブルと同数のビューが作成される．これらのビューは参照専用であり，データ格納層の実体（物理テーブル）に対して，データ参照層側からレコードの追加や更新，削除を行うことはできない．データ参照層を中間に設けることで，DWH に連携された生データを直接操作することなく，読み取り専用の状態で安全に取り扱うことが可能となる【要件 K】．なお，データ参照層に定義したビューではカラムの追加や削除は原則行わないが，協定世界時（UTC）で保存されたタイムスタンプを日本標準時（JST）に変換する処理を行っている．

データ格納層とデータ参照層のテーブルおよびビューは，DCG は直接編集することができない．本階層の実装作業や保守は，データエンジニアリングチーム（DWH Group，以下 DWHG）の所管となる．DWHG は，データソースとデータ格納層とをつなぐデータパイプラインの設計やデータ参照層のビューの定義といった一連のデータフローに関わる業務を担当しており，データ参照層と次層の境界が DCG との責任分界点となる[*13]．

『データマート層』は，データ分析や可視化，特定の社内業務[*14] に用いるデータを出力するための階層である．本層の一部はビジネス部門[*15] のメンバによるアクセスも想定した領域となる．データマート層に配備するテーブルやビューは，DCG が設計と実装を担当する．本層はさらに複数のデータセットから構成され，それぞれ役割を有している．詳細は次節で説明する．

3.4　データマート層の内部構造①〜データ基底層

データマート層は，内部でさらに 3 層に細分化される（図 4）．各層は原則としてビューで定義するが，用途やデータの特性に応じてバッチ処理でレコードを生成し，物理テーブルに投入する場合もある．

まず，『データ基底層』では，以下の前処理を行う【要件 A】．

1. 不要カラムの削除
2. 不要レコードの削除
3. データ型の変換
4. レコードの有効期間の付与

1. では，データソース由来のカラムのうち，暗号化されているもの（例：氏名や電話番号等の個人情報）やデータ分析の用途では明らかに利用しないもの（例：システムが自動生成するメッセージや ID 等）を除外する．これは，後続の階層からの参照を防ぐ意味合いもある．

2. では，サービス開始日以前のテストデータ[*16] や不整合を除外する処理を行う．DWH 上で不整合が見つかった場合，原則として「データパッチ」と呼ばれるデータ格納層への補正処理が DWHG によって実施される．しかし，修正に時間を要する場合や厳密には不整合ではないものの分析の観点からは対象外とすべきレコードが存在する場合は，本層でそれを取り除く．

3. では，カラムのデータ型を利用に適した型に変換（いわゆるキャスト）する．具体的には，以下のような処理を行う．

- データ参照層では BigQuery のデータ型が文字列型（STRING）で定義されているが実際のデータは数字しか含んでいないものについて，カラムの用途を精査のうえ，整数型（INT64）や浮動小数点型

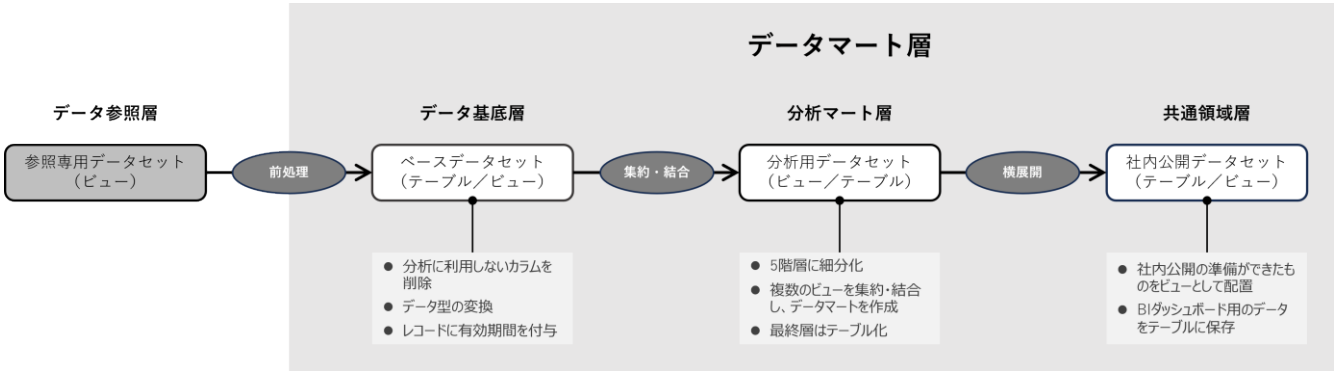

図 4　データマート層のカスケード構造
Fig. 4　Cascading structure of data mart layer.

[*13] 「データマート層」の権限設定や本層を含む DWH 全体の管理業務は DWHG が担当している．
[*14] 顧客コミュニケーションのための対象者抽出や定期的に発生する各種報告のための集計作業等が該当する．
[*15] マーケティングやユーザ調査を行う部署等が該当する．
[*16] 本番環境上で実施したフィールドテストのデータ等が該当する．取引上は有効なデータであるが，顧客向けにサービスを開放する前のレコードであるため，データ分析上は除外するケースが多い．

（FLOAT64）に変換
- 日付を意図するカラムが文字列型で定義されている場合は日付型（DATE）に変換

4. では，DWH に追加されたレコードに対し，データソース側ではいつからいつまでがこの値であったかを示す「開始日」「終了日」カラムを付与する[*17]．第 2 章で述べた SCD Type 2 と同じ考え方であるが，ここでは終了日を次のレコードの開始日の 1 日前となるように設定する[*18]．そのため，データソース側で同日中に複数回のデータの変更が生じた場合，当日の最終更新レコード“以外”は開始日＞終了日と表示され，一見矛盾した前後関係にみえる．これは同日中の最終状態とそれ以外のレコードを DWH 上で簡便に識別できるようにした結果である[*19]（**図 5**）．

「開始日」「終了日」カラムが付与されるのは顧客や口座情報といったマスタテーブルに限られ，取引明細や振込履歴等，都度レコードが発生するイベントテーブル（トランザクション）や日次残高のように 1 日 1 レコードしか発生しないテーブルに対しては適用されない．

3.5　データマート層の構造②〜分析マート層

『分析マート層』では，以下の処理を行う．本層以降で定義するテーブルおよびビューを狭義の「データマート」と呼称する．各処理には前後関係が存在し，それぞれの段階に Tier と呼ばれるレベルを設定している．Tier は現在

開始日・終了日付与前

顧客ID	ステータス	レコード更新日時
AAA	0	2023-07-01 06:11:32
AAA	1	2023-07-01 23:50:17
AAA	2	2023-07-02 10:32:26

開始日・終了日付与後

顧客ID	ステータス	レコード更新日時	開始日	終了日
AAA	0	2023-07-01 06:11:32	2023-07-01	2023-06-30
AAA	1	2023-07-01 23:50:17	2023-07-01	2023-07-01
AAA	2	2023-07-02 10:32:26	2023-07-02	2099-12-31

同日中に複数回のレコード更新が生じた場合は、最終レコード以外の「終了日」が「開始日」よりも前の日付となる

同日中の最終状態のみを取得した場合

顧客ID	ステータス	レコード更新日時	開始日	終了日
AAA	1	2023-07-01 23:50:17	2023-07-01	2023-07-01
AAA	2	2023-07-02 10:32:26	2023-07-02	2099-12-31

SQLのWHERE句のクエリに 開始日 <= 終了日 を指定することで、当日中の最終レコードを取得できる

図 5　開始日・終了日カラムの付与例
Fig. 5　Adding columns of start dates and end dates.

1 から 5 まで存在し，分析マート層のデータマートは必ず Tier 1 を経て 2〜5 の段階へ進む．ただし，Tier 2・3 は省略することがある．すなわち Tier の高いデータマートは低い Tier を SQL の FROM 句に指定できるが，逆は不可である．各 Tier の役割を以下に示す．

1. データ基底層で付与した開始日・終了日を元に，日次別の最終状態のレコードを生成《Tier 1》
2. Tier 1 のテーブルやビューには，単独では使いづらいものが存在．その場合，複数の Tier 1 を結合して最小の「意味のある」データマートを作成．または，Tier 1 に含まれるデータの一部を切り出し，独立したデータマートを定義《Tier 2》
3. Tier 1・2 のデータマートを結合（JOIN）し，特定のキーでレコードを集約（GROUP BY）《Tier 3》
4. Tier 1〜3 を組み合わせて，特定の業務に最適化したデータマートを作成《Tier 4》
5. Tier 4 のデータマートをテーブルに変換．バッチ処理でレコードを追加し，静的に保存《Tier 5》

Tier 1 では，開業日から現在までの連続日を記録したカレンダーテーブルとマスタテーブルの主キーを交差結合（CROSS JOIN）し，開始日・終了日の期間に含まれるデータに変化がない日のレコードを「補完」する．さらに，開始日と終了日のカラムを削除し，代わりに該当日を示す「TBL」（テーブル基準日）カラムを追加する．TBL は日付型のカラムであり，抽出条件に TBL＝特定の日付 を指定することでその日に存在した主キーに紐づくマスタテーブルの最終状態を取得することができる[*20]（**図 6**）．

本方式はカレンダーとの掛け合わせになるためレコード数が時間経過とともに膨大になるが，ビューとして実装することで物理的なストレージを消費せずに日次断面のスナップショットを取得することが可能となる【要件 H】．動的にレコードを生成することから，静的保存されたテーブルにアクセスするよりも計算リソース（BigQuery では「スロット」と呼ぶ）を消費するものの，クラウド DWH の分散処理により，交差結合の結果が仮に数億レコードに達した場合でも，数分以内に処理を終えるケースが多い．なお，データソース側で「物理削除」を行うと Tier 1 データマートに削除時点の最終状態が正しく反映されないため，データソース側のレコード削除は原則として「論理削除」で処理する必要がある．

イベントテーブルについても同様に TBL カラムを追加する．ただし，トランザクションが発生したタイムスタンプの日付部分を TBL の値に用いるため，マスタテーブルのようなレコードの補完処理（カレンダーとの交差結合）

[*17] 本 DWH では，開始日・終了日はタイムスタンプ型ではなく，日付型で管理している．
[*18] 最新のレコードの終了日は，2099-12-31 とする．
[*19] 本仕様は，開発パートナーであるアクセンチュア（株）の黒田亮氏のアイデアに基づく．
[*20] 同日中の複数回の値の変化はこの時点で最終状態に一本化される．

は行わない（図 7）．

Tier 2 では，必要に応じて 2 種類の処理を行う．まず，以下に該当する場合に複数の Tier 1 データマートを結合する．

1. 主キーと他のテーブルの外部キーが直接結合できない場合に，キーを結びつける中間テーブルをあらかじめ結合しておき，候補キーを追加．Tier 2 として定義するかは，利用頻度によって判断
2. 2 つ以上のテーブルの和集合（UNION）を取ることでデータの利便性が向上する場合に，テーブルを統合（例：Tier 1 では別テーブルになっていた口座開設と口座解約の履歴を一本化）

本処理を行ったデータマートは，『マルチテーブル』と呼ぶ．

逆に Tier 1 データマートの集合の一部を切り出して Tier 2 に定義する場合がある．たとえば，Tier 1 の「口座」テーブルのデータマートには，「普通預金口座」と「貯蓄預金口座」の 2 種類の口座種別の異なるレコードが含まれている．データ分析は双方の口座種別を分離して行うことが多いため，Tier 2 では，2 つのサブセットに集合を分解する．本処理を行ったデータマートは，『サブテーブル』と呼ぶ．

Tier 3 では，Tier 1・2 データマート単独または複数結合したものを特定のキーに基づいて集約（GROUP BY）する．たとえば，「取引明細」テーブルのデータマートには，複数の入出金の履歴（トランザクション）が含まれおり，そのままでは顧客ごとの利用頻度を集計することができない．そこで，顧客 ID かつ TBL 別にレコードを集約することで，取引回数や合計取引額の日ごとの集計が可能になる（図 8）．

Tier 4 では，Tier 1〜3 で作成したデータマートを組み合わせて，データ分析や特定の業務に最適化された統合データマートを作成する．たとえば，顧客アクティビティデータマートでは，顧客 ID をキーにして属性情報（性別，年代，職業等）や口座残高，各サービスの利用状況（集計値）を横方向に結合する．

BigQuery は一度に実行できる計算リソースに上限あり，テーブル（ビュー）の結合対象が数十に達する場合やクエリが複雑化すると処理が中断することがある．本制約を回避するには結合回数を制限する必要があり，Tier4 ではあえて複数のデータマートに分割した状態で定義する場合がある【要件 F】．

Tier 5 では，Tier 4 で実装したデータマートをテーブル形式に変換する．Tier 4 までのデータマートは一部を除いてビューで定義されており，レコードは SQL を実行するたびに動的に生成される．先述のとおり，ビュー形式のデータマートはストレージを消費せずに常に最新データを取得できるメリットがある一方，Tier の階層を進むごとにデータマートの結合数が急増し，以下の問題が生ずるこ

データ基底層

顧客ID	ステータス	開始日	終了日
AAA	0	2023-07-01	2023-07-01
AAA	1	2023-07-02	2023-07-03
AAA	2	2023-07-04	2099-12-31
BBB	0	2023-07-04	2023-07-03
BBB	1	2023-07-04	2023-07-04
BBB	2	2023-07-05	2099-12-31

分析マート層（Tier 1）

顧客ID	ステータス	TBL
AAA	0	2023-07-01
AAA	1	2023-07-02
AAA	1	2023-07-03
AAA	2	2023-07-04
AAA	2	2023-07-05
AAA	2	2023-07-06
BBB	1	2023-07-04
BBB	2	2023-07-05
BBB	2	2023-07-06

2023年7月6日を現在日とした場合の例。同じ日のステータス遷移は最終状態のみを保持。ステータスに変化がない場合は、前日の値を繰り返す

図 6　Tier 1 のレコード補完処理例（マスタテーブル）

Fig. 6　Process of record completion in master table of tier 1.

データ基底層

顧客ID	取引ID	取引額	取引日時
AAA	xyz	2564	2023-07-01 15:46:23
BBB	uvw	330	2023-07-02 12:05:18
BBB	rst	110	2023-07-02 20:33:49
AAA	opq	45987	2023-07-03 07:15:20

分析マート層（Tier 1）

顧客ID	取引ID	取引額	取引日時	TBL
AAA	xyz	2564	2023-07-01 15:46:23	2023-07-01
BBB	uvw	330	2023-07-02 12:05:18	2023-07-02
BBB	rst	110	2023-07-02 20:33:49	2023-07-02
AAA	opq	45987	2023-07-03 07:15:20	2023-07-03

図 7　Tier 1 のレコード補完処理例（イベントテーブル）

Fig. 7　Process of record completion in event table of tier 1.

Tier 1・2

顧客ID	取引ID	取引額	取引日時	TBL
AAA	xyz	2564	2023-07-01 15:46:23	2023-07-01
BBB	uvw	330	2023-07-02 12:05:18	2023-07-02
BBB	rst	110	2023-07-02 20:33:49	2023-07-02
AAA	opq	45987	2023-07-03 07:15:20	2023-07-03

Tier 3

顧客ID	取引回数	合計取引額	TBL
AAA	1	2564	2023-07-01
BBB	2	440	2023-07-02
AAA	1	45987	2023-07-03

例では、顧客ID＝AAA・BBB の取引履歴から、TBLごとの取引回数と合計取引額を集計した

図 8　Tier3 のレコード集約例

Fig. 8　Process of record aggregation of tier 3.

とがある．

- クエリの結果取得に時間を要する
- 消費スロットに応じた利用料が増加する

そのため，Tier 5 はあえてテーブルとして定義し，BigQuery の「クエリのスケジューリング」機能を用いて，Tier 4 のデータを 1 日 1 回バッチ処理で追加する【要件 E・F・G】．反映対象は前回バッチ実行以降の差分のみとし，既存データの洗い替えは原則行わない[*21]．

Tier 5 は，Tier 4 の単一データマート（ビュー）をテーブル化するパターンと，複数の Tier 4 のデータを一時テーブルに取り込んでから最終テーブルへ反映するパターンの 2 種類が存在する．Tier 5 では，「パーティション分割テーブル」の設定も合わせて行う．BigQuery は通常，SQL の WHERE 句で条件を指定してレコードの絞り込みを行った場合もテーブルに対するフルスキャンが実行されるが，「パーティショニング列」と呼ばれるカラムを設定しておくことで，クエリ実行時のスキャン量を減らすことができる．たとえば，パーティショニング列に TBL カラムを指定し，WHERE 句で当該カラムに対する日付指定や期間指定を行うことで，スキャン量が十〜数百分の一に削減可能なケースもある．本機能の活用により，応答時間の改善や利用料の軽減が期待される【要件 E・G】．

Tier に基づいたデータマートの「階層化」を行うことで，構成要素の「部品化」が推し進められ，集計等の繰り返し発生しうる処理を一元化することが可能となる．構成要素の再利用性を高めることは，データマート構築の効率化のみならず，設計者の「解釈の相違」による出力結果の不一致を防止することにもつながる【要件 B・C】．

分析マート層をディメンショナルモデルの観点からみた場合，Tier 1 および 2 は DT および FT に該当し，Tier 3 は集約された FT に相当する．逆スタースキーマの適用事例である本 DWH では，Tier 1・2 のマスタテーブルに Tier 3 のイベントテーブルの集約結果を結合して Tier 4・5 を作成する．Tier 5 は Tier 4 を静的なテーブルに変換したものであるため，もはや動的な結合（JOIN）は存在しない．これは，リレーショナルデータベースにおける「第一正規形」に相当する．つまり，逆スタースキーマの最終形（Tier 5）は，単一のテーブルのみで業務に必要な情報がすべて網羅された大福帳型モデルに近い形態となる．

分析マート層には，DCG が主に利用するデータ分析用データマートに加えて，社内の各種業務に利用できるものが多数含まれている．しかし，本層のデータマートは DCG のみがアクセス可能な領域に格納されているため，他のチームメンバは直接参照することはできない．そこでビジネス部門向けのデータマートを次節で述べる領域へ展開し，アクセス権限の問題に対応した．

3.6　データマート層の構造③〜共通領域層

『共通領域層』は，DCG 以外のメンバも参照可能なデータセットである．特定の社内業務向けに作成したデータマートおよび「データの民主化」の観点からデータ分析以外の業務でも利活用可能と DCG が判断したものについて分析マート層から共通領域層へ展開する【要件 K】．2 層間の参照関係を**図 9** に示す．

共通領域層に配置するデータマートは，大きく 2 種類に分類される．

1. Tier 1〜3 および Tier 5 データマートをビューとして定義したもの[*22]
2. BI ツールから参照するためのデータマートをテーブルとして定義したもの

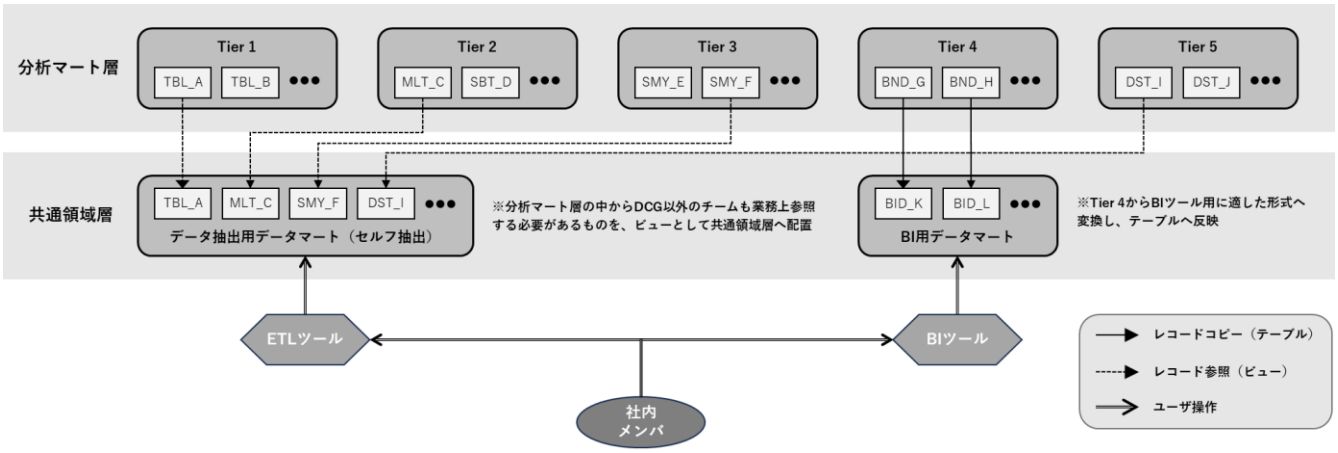

図 9　分析マート層と共通領域層の関係

Fig. 9　Relationship between analysis mart layer and common area layer.

[*21] カラム追加時や集計ロジックに変更が生じた場合は，Tier 5 の全レコードを差し替えることがある．

[*22] Tier 4 は，Tier 5 と同一のスキーマであることが多いため，共通領域層からは参照しない．

1. は，業務上必要なデータの抽出作業を DCG 以外のメンバが自ら行えるようにするために，関連するデータマートを整備し，ビューとして配置したものである．各チームメンバは，GUI ベースの ETL ツール[*23] や SQL を Web ブラウザから直接実行可能な BigQuery Web UI 経由でこれらのデータマートにアクセスすることができる．利用方法や事例は社内 Wiki に順次公開し，ドキュメントの整備と共有を進めている【要件 J】.

2. は，BI ツール向けのデータマートに該当する．詳細は次節で述べる．

3.7　BI ダッシュボード用データマートの構築

本 DWH では，データ分析および業務特化型のデータマートと，BI ツールから参照するダッシュボード用データマートを可能な限り共通の仕組みで作成できるようした【要件 D】.

BI ツールでは，一般的に「ディメンション」と呼ばれる分析軸を用いて集計対象となるレコード（「メジャー」に相当）を絞り込む．ディメンショナルモデルのスタースキーマは，BI ツールの多次元データ分析機能との親和性は高いが，データ量によっては集計処理に時間を要することがある．一方，ダッシュボードによる可視化や KPI 等のレポーティングに BI ツールを用いる場合は分析軸が固定的であるため，ディメンションを組み合わせた探索的な分析ニーズは相対的に低い．

一例として，「顧客ごとのサービス利用状況ダッシュボード」を作成するケースを考える．本例では，DT に当たる「顧客」テーブルに FT から「サービス利用状況」に該当する値を事前に集計し，結合することで BI ツール用のデータマートを作成する．これは，逆スタースキーマによる実装モデルにほかならない．スタースキーマと比較した場合，ディメンションの種類や分析粒度は固定されるが，集計ロジックを分離することが可能となるため，BI ツールの処理速度の改善やデータマートを横展開する[*24] 際の工数削減に有効である．

データ分析ならびに業務特化用のデータマートは Tier 5 を構築することで完成する．一方，データマートを BI ツールのダッシュボード用途に使用する場合は，集計値（発生頻度，合計額等）に加えて，TBL 当日または口座開設以降 TBL までのイベント発生有無を 0・1 の 2 値に変換（フラグ化）したカラムを追加することが多い．これは，BI ツール上でチェックボックスやラジオボタンをレコード抽出（分析軸による条件指定）のインタフェースに用いた場合に，2 値化されたカラムを検索対象とするからである．BI ツール側で 2 値化したカラムを動的に追加することも可能であるが，処理の高速化のため，本 DWH ではあらかじめデータマート側でフラグに変換しておく．

BI ツール用のデータマートは，Tier 4 から Tier 5 へ進むフローから分岐する形で共通領域層に定義する（図 9 参照）．データマートはテーブルであり，Tier 4 からフラグ化した値をバッチ処理で挿入する．本手順に基づくことで比較的少ない工数でデータ分析または社内業務用のデータマートをダッシュボード用に転用することが可能となる．

3.8　命名規則

本 DWH の各テーブル・ビュー名は，原則として以下の命名規則に沿って付与する【要件 I】.

- 名前の先頭にカテゴリごとに設定された 3 文字の識別子とアンダーバーを追加（例：CST_，DEM_）
- 原則としてテーブル・ビュー名は複数形で表記（例：CST_Customers，DEM_Accounts）
- 識別子を除く名称部分はアッパーキャメルケースで表記（例：DST_CustomerActivities，TBL_Account Transactions）

ここでいう「カテゴリ」とは，テーブル・ビューを用途ごとに分類したものであり，データ基底層までは，データソース由来の識別子を使用する．分析マート層以降は，Tier のレベルに応じた識別子を DCG が独自に設定する．分析マート層および共通領域層に対する識別子の付与ルールと配備形式を**表 1** に示す．

3.9　テーブル定義書

データ基底層以降のテーブルやビューのスキーマについては社内 Wiki に定義情報を登録している【要件 J】．主な記載内容は以下のとおりである．

- 概要・目的
- 種別（テーブルまたはビュー）
- データ連携頻度（ニアリアル，バッチ処理等）
- カラム定義（論理名，物理名，主キー，データ型，欠

表 1　分析マート層・共通領域層の識別子付与ルール
Table 1　Prefix assignment rule for data marts.

Tier	識別子	識別子の意味	配置形式	
			分析マート層	共通領域層
1	TBL	Table基準日変換	原則ビュー	ビュー
2	MLT	Multi Table	原則ビュー	ビュー
2	SBT	Sub Table	原則ビュー	ビュー
3	SMY	Summary	原則ビュー	ビュー
4	BND	Binder	原則ビュー	-
5	DST	Destination	テーブル	ビュー
-	BID	BI Dashboard	-	テーブル

[*23] Extract（抽出），Transform（変換・加工），Load（ロード）といった一連の処理を行うアプリケーション．
[*24] たとえば，顧客を起点としたデータマートを複数作成する場合に，Tier 3 の集計済データマートを再利用することが考えられる．

損・空白有無，型変換有無，備考）
- データの確認状況チェックリスト
- テーブルの関係とデータの流れを示したフロー図
- 定義用の SQL
- 更新履歴

定義内容に大きな変更が生じた場合は，ページを複製のうえ編集を行い，定義書を新旧バージョンに分割する．Wiki のタグ機能を用いて新ページにバージョン番号を登録し，テーブル定義書の一覧には最新版のみを初期表示する．

3.10　個人情報保護および情報セキュリティ対策

本 DWH では，氏名・住所（市区町村より細かい情報）・電話番号・メールアドレス等の本人到達性のある情報はすべて暗号化されている．DCG およびデータマート利用者は暗号化されたカラムを復号化する（＝平文に戻す）ことはできない．また，「個人識別符号」やパスワード等の「不正に利用されることにより財産的被害が生ずる恐れがある個人データ」は勘定系システムから DWH に連携されないため，参照不可である[*25]．【要件 L】．

個人情報の取り扱いについては，スマホアプリおよび Web サイトの利用規約・ポリシーの「個人情報等の利用目的」[14] に記されている．DWH のデータ利用については，主に「データ分析やアンケートならびに市場調査の実施等による各種金融商品やサービスの研究・開発等，お客さまへのサービス品質の向上を図るため」の項目が該当し，目的の範囲内で実施している．

DCG を含む社内のメンバが DWH にアクセスする手段は制限されており，VDI（Virtual Desktop Infrastructure）と呼ばれる仮想デスクトップ環境内でのみ利用可能である．VDI の操作はすべて録画されており，監査証跡として操作ログは一定期間保持される．これは業務外利用を防ぐ牽制の意味合いもある．また，VDI と PC 間のファイルのやり取りは，すべて専用のファイル転送サービスを経由する必要がある．本サービスへファイルをアップロードするには上長の承認が必要であり，ここでも相互牽制の仕組みが取り入れられている．なお，USB メモリ等の外部記憶媒体は PC 上では使用不可となっている．

メールによる外部とのやり取りが発生する場合も送信のたびに上長による承認を必須としている．さらに IP アドレスによるアクセス制限やシングルサインオン等による利用者別のアクセスコントロールも行われており，複数の情報漏洩対策が取られている【要件 K・L】．

[*25] データ分析の操作過程は後述の VDI により記録されており，DWH 内外のデータと突合することで個人を特定することはない．

4.　運用状況と成果

2023 年 8 月現在，DWH 上には合計 682 個のテーブルおよびビューが定義されている．うち，445 個が DCG によるデータ確認・補正を実施したものであり，Tier 1 以降のデータマートは 284 個に達する．

新サービスや既存サービスの拡充に伴い，データソースにテーブルやカラムが追加されることがある．その場合，DWHG による DWH へのデータパイプラインの実装が完了してから，データ基底層以降の整備に着手する．データ参照層へのビューの追加は，年平均 20 件前後発生している．

公開領域層にデータマートが整備される以前は，DCG にデータ抽出依頼が来ることも多かったが，各自が直接データに「触れられる」環境を整備することで，定型的な作業については所属チーム内で完結できるようになった．これにより，社内に次の好循環をもたらした．

1. ダッシュボードの簡易抽出・集計機能を用いて，公式な統計データを自分自身で取得し，「正式」な数字として利活用
2. DCG が事前にデータをチェックし不整合等を取り除いたデータマートを構築することで，各チームのメンバは業務に必要なデータを正確に取得可能
3. 業務ごとに最適化されたされたデータマートを設置することで，作業者のスキルに依存せずにデータ抽出を行えるようになり，工数削減や属人性の排除に寄与
4. データの専門家である DCG は，データ抽出依頼に追われることなくより高度な分析業務へ注力

これは社内における「データの民主化」の 1 つの形であるといえよう．成果の詳細について，次節以降に述べる．

4.1　社内におけるデータマートの活用事例①：公式な数値の提供元としてのダッシュボード

DCG が BI ツール用のデータマートを整備したことにより，KPI 等のビジネス観点の計数をダッシュボードから容易に確認できるようになった．ダッシュボード上では指標ごとの時間推移を追うこともでき，多くの社内のメンバが参照している．また，ダッシュボードからは CSV 形式での書き出しも可能であるため，メンバ自身が「セルフ抽出」を行い，公式な数字として利活用している．

4.2　社内におけるデータマートの活用事例②：データの不整合の防止

データマートが整備される以前は，ビジネス部門のメンバは ETL ツールや BI ツール等を用いてデータ参照層のビューを直接参照し，データ集計や書き出し等の作業を行っていた．3.4 節で述べたように，データ参照層のビューには，通常の業務では使用しないテスト用のデータ

や重複データが含まれていることがあり，手動で取り除く必要があった．ところが，メンバによって除外すべきレコードの解釈やデータの理解度に「振れ幅」があり，同一目的で出力されたデータであっても，件数の相違が発生することもあった．特に「データの理解」については，DCG が調査しなければ除外すべきかの判断が難しいケースもあり，メンバの「努力」に任せるには限界があったものの，データマート化されたことによりこれらの問題は解消された．さらに，繰り返し発生する業務用の集計についてもデータマート側で処理を行うことにより，作業の効率化が図られた．

4.3　社内におけるデータマートの活用事例③：属人性の排除と作業効率化

本項目は，主に特定の顧客に対するメッセージ配信[*26]やキャンペーン対象者の抽出といったマーケティング施策の実施に関する作業が該当する．データマートが整備される前は，ビジネス部門のメンバが ETL ツールを用いて，GUI アプリケーションに複数のデータ参照層やデータ基底層のテーブル（ビュー）を読み込み，主キーでテーブル間を結びつけ，対象者を抽出する作業を行ってきた．本作業は，様々な顧客属性やサービス利用状況を組み合わせて対象者を絞り込むことから，10 以上のテーブルを参照するケースもあった．そのため，ETL ツールの操作にある程度習熟したメンバであっても，準備に 1 施策あたり 1 時間程度を要していた．さらに「ワークフロー」と呼ばれる設計画面上でどのような操作が行われているかを他のチームメンバが理解するには相応の時間とスキルが必要なため，作業が属人化する傾向にあった[*27]．このような状況を解消すべくデータマートの整備を進めた結果，データ抽出に必要な属性情報やサービス利用状況等の情報をワンストップで取得できるようになり，ETL ツールでワークフローを都度設計する必要がなくなった．つまり当該作業を BI ツールで完結できるようになったため，大幅な省力化を実現し，属人性を排除することにもつながった．

データマートの整備による工数削減効果が大きかった事例として，「CheerBox」[15] の対象者抽出作業が挙げられる．CheerBox とは，みんなの銀行のサービスである「ボックス」[16] 機能を用いたキャンペーンを指す．ボックスは目的別に貯められる貯蓄預金を指し，普通預金（みんなの銀行では「ウォレット」と呼ぶ）とは別口座として扱われる．顧客は口座開設と同時に普通預金（ウォレット）と貯蓄預金（ボックス）の 2 口座を保有することになる．ウォレットとボックス間のお金の移動はスマホアプリ上で簡単に行え，普段使いのお金はウォレットに，一時的に取り分けたり中長期的に貯めていく場合はボックスに移しておくといった使い方が可能である．ボックスは目的に応じてさらに最大 20 個まで仕分けることができ，それぞれの「箱」に自由に名前を付けることができる．たとえば，「家賃」というボックスを作成して毎月の支払いに備えたり，「旅行」といった名前で資金をためていくといった使い方もできる．また，ウォレットのお金を定期的にボックスに移動し貯金するスケジュール機能も備わっている．CheerBox キャンペーンはボックスの「自由に名前を付けて任意の額を貯金できる」機能を利用して，みんなの銀行が指定するキーワードをボックス名に含む場合に，残高に応じた一定額を支援先に「寄付」する取り組みである．たとえば，あるスポーツチームを応援する CheerBox キャンペーンがあった場合，顧客がボックス名にチーム名等のキーワード[*28] をつけておくと，キャンペーン開始から終了時までのボックスの平均残高の一部（多くの場合 1%）をみんなの銀行が支援先へ送る（贈る）スキームとなる．寄付によって顧客のボックス残高が減ることはない．また，寄付を受けた支援先は，CheerBox で応援してくれた顧客に対して抽選でグッズを進呈することがあり，顧客と支援先とのコミュニケーション活性化にも寄与している．

支援先に寄付を行う際やグッズ進呈のための抽選を行うには，CheerBox キャンペーンに参加した顧客の抽出作業が必要である．本作業は，以下の要素を考慮しながら実施する．

1. CheerBox に含まれるキーワードの表記ブレの補正
2. ボックスの正確な日次最終残高の抽出と集計

1. は，キーワードの入力は顧客が行うため，様々な入力パターンが発生しうる．たとえば，「ABC」という文字列を CheerBox のキーワードに指定した場合，全角・半角・大文字・小文字のブレやスペースの有無，「ABC 応援」「がんばれ ABC！」といった部分一致についても考慮する必要がある．これらのパターンを ETL ツール上で完全に網羅するのは難しい場合があり，データマート側のロジックとして表記ブレを補正することで抽出作業の効率化と抜け漏れの防止を図った．

2. はキャンペーン実施期間中に設定された CheerBox の最終残高を日別に計算し，平均残高を計算する．ボックスはいつでも名前を変えることができるため，昨日まで存在した CheerBox の対象ボックスが翌日には別名に変更され対象外となることもあり得る．さらに，キャンペーン終

[*26] 配信手段としては，「プッシュ通知」「アプリ内ポップアップ」「メール」等がある．メールアドレスは暗号化された状態で対象者抽出を行うため，作業者が平文のアドレスを視認することはない．

[*27] GUI の ETL ツールは，SQL のように集計ロジックが言語化されないため，画面上のフローを 1 つずつ追ってデータの流れを確認する必要がある．

[*28] 具体的な対象キーワードについては，みんなの銀行の CheerBox キャンペーンサイトに掲載される．

了直後に別のボックスやウォレットに資金が移動することもあるため，各日の深夜 0 時ちょうどの残高を取得する必要がある．

ビジネス部門のメンバはこれらの作業を ETL ツールを用いて行っていたが，様々なテーブルの結合とレコードの除外処理が必要であり，ワークフローが複雑化していた．そのため，抽出処理が長時間化したり，ロジックの正確性の検証を GUI 画面上で行うのが難しいという課題もあった．また，CheerBox のキーワード抽出ルールも徐々に複雑化しており，1 つのキャンペーン用の抽出ワークフローを ETL ツールで設計するのに，熟練した作業者であっても数時間要していた．そこで，CheerBox のキーワード正規化処理と正確な日次最終残高を結合したデータマートを作成することで，抽出作業が 1 キャンペーンあたり約 15 分にまで短縮された．

4.4 社内におけるデータマートの活用事例④：DCG の役割の高度化

DCG がデータマートを整備し，各チームメンバ向けに公開領域層で公開したことにより，他部署からの「単純」なデータ抽出依頼は，データマート公開前と比べ減少した．DCG は，既存のデータマートでは抽出が困難な難易度の高い依頼を引き続き担当しているが，削減された工数はより高度なデータ分析や機械学習モデルの構築，マーケティングオートメーション（MA）の整備等に振り向けている．

MA は特定の条件や時間経過等をトリガにして，自動的にメッセージ配信を行う機能である．対象者抽出も全自動であるため，DCG やビジネス部門の工数削減効果は大きい．MA 用の対象者抽出は SQL を使用する場合や，Python スクリプトを経由するケースがある．後者は，顧客ごとの（施策に対する）反応率を機械学習モデルで推定し，確率の高いセグメントに含まれる対象者を抽出のうえ，自動的にメッセージ配信するといった運用に用いている．

5. まとめと今後の課題

みんなの銀行の開業から 2 年が経過し，データ分析や社内業務に必要なデータマートや BI ダッシュボードについては，完全ではないものの必要十分な環境を整備することができた．その中で，クラウド DWH 時代に適したデータマートの形態を模索し，逆スタースキーマの考え方にたどり着いた．データマートを階層化し，Tier のレベルに応じてビューやテーブルを細かく作り分けることは，データマートの再利用性を高め，似たような処理を繰り返し作成することを防ぐ意味合いがある．同一用途の集計カラムをデータマートごとに作成すると，作成者の「解釈の違い」により集計値が一致しない状況も発生しうるが，本稿の考えに基づいてデータマートを「部品化」することでこのような事態を回避できた．また，「部品」の組み合わせでデータマートを構築できるということは，「データ分析用」「特定の社内業務用」「BI ツール用」データマートのロジックを共通化できるということであり，DCG の開発工数の削減にもつながった．

本稿で述べたこれらの考え方は，業態や保有するデータの構成・構造の違いを超えて，様々なデータマートへの幅広い応用が可能だと考える．

今後の DCG の取り組みとしては，以下が挙げられる．

1. データカタログの整備とテーブル定義書の半自動生成
2. データマート定義のバージョン管理のシステム化
3. データ異常検知の仕組みの整備
4. DWH 連携データ障害発生時のデータマートのリカバリ半自動化
5. マテリアライズドビューを用いた処理の高速化，テーブルの代替
6. 「スーパーデータマート」の作成

1. は，「データカタログ」と呼ばれるメタデータ管理ツールの本格的な導入を進める．データカタログを用いることで，テーブル定義情報に加えて，作成者や作成日，データの機密度，アクセス権限といったデータセキュリティやデータマネジメントに関わるメタデータを包括的に管理することが可能となる．テーブル定義書の半自動生成により DCG の管理工数の削減にもつながることが期待されるほか，定義書にコメントやカラムの別名等の付加情報を追記することで，データの所在を確認しやすくなるといったメリットもある．これは，各チームメンバが業務上必要な情報を探索するうえでも有効な仕組みである．

2. は，Git 等のバージョン管理システムを用いてデータマートの定義用 SQL を体系的に管理することを想定している．データマートの構築は少人数・小規模で始めたこともあり，これまで SQL のバージョン管理は手動であったが，今後のデータマートのボリューム増加を考慮して，システム化に取り組んでいく．

3. は，DWH に連携されたデータに対する，主キーの重複やデータ欠損，異常値検知をモニタリングする仕組みを検討中である．BigQuery は，データを時系列で蓄積する DWH の仕様上，主キーや外部キーによるキー制約機能が備わっていないため，値の一意性は別の仕組みで担保する必要がある．キーの重複を検知する取り組みについては DWHG でも実施しており，今後は連携して精緻化を進める．

4. は，データソースからデータ格納層へのデータ反映に失敗した場合，後続のフローに影響することがあり，そのリカバリ工数の削減を課題としている．不具合発生時の復旧作業は，データ参照層までは DWHG の管轄となるが，後続の階層は DCG の担当となる．DCG が作成した

データマートのうちビューとして実装されているものについては，データ格納層に正しいデータが連携され次第，自動的に「復旧」する．一方，Tier 5 や一部のデータマートはバッチ処理でテーブルに差分データが反映されるため，スクリプトの再実行や手動でのレコード削除が必要になることもある．当該作業はイレギュラーであるため，ミスや属人化の防止の観点からもリカバリ処理の半自動化について検討中である．

5. は，データマートの一部をマテリアライズドビューに変更することを検討している．マテリアライズドビューとは，ビューとテーブルとの中間的な要素を持ったビューであり，レコードを一定期間キャッシュした状態で動作する．クエリが複雑なビューでは劇的な高速化が期待できる一方，最新データを反映するには定期的なリフレッシュ操作が必要となる．そこで，マテリアライズドビューの特性を踏まえてテーブルとして実装している Tier 5 データマートの中で置き換え可能なものがないかを検証中である．

6. は，Tier 5 をさらに集約し，特定のデータマート（仮に『スーパーデータマート』と呼ぶ）を参照すれば，DCG を除く社内のデータ利活用業務の大半を網羅できるようなスーパーセットの構築検討を開始している．また，歴史的な経緯により，現在はエンドユーザがデータにアクセスするための手段（アプリケーション）が複数に分かれているため，PaaS ベースの BI ツールに統一し，将来的にはデータの参照先をスーパーデータマートに一本化することも視野に入れている．

今後は構築したデータマートをより一層活用して，顧客ごとに最適なサービス提案を行う「ハイパーパーソナライズ」の展開や，マーケティング施策や消費性ローンサービス等で用いる機械学習モデルの高度化・深化といった，データサイエンスとエンジニアリング，ビジネスを融合した取り組みに注力していきたい．

謝辞　本稿で述べた DWH 環境およびデータマートの構築に関して，（株）みんなの銀行データクリエイショングループ，ゼロバンク・デザインファクトリー（株）DWH グループならびにアクセンチュア（株）の関係各位の協力をいただいた．ここに謹んで謝意を表する．

特記事項　文中の社名，商品名，サービス名等は，一般に各社の商標または登録商標である．本稿の図表はイメージであり，実際のテーブルスキーマを示したものではない．本稿の内容は筆者の見解に基づいてまとめられたものであり，筆者の属する組織の公式見解を示すものではないことを付記する．

参考文献

[1] Inmon, W. H. and Hackathorn, R. D.: 藤本康秀（監訳）：よくわかるデータウェアハウス活用法，インターナショナル・トムソン・パブリッシング・ジャパン（1996）．
[2] Zeng, J., and Glaister, K. W.: Value Creation from Big Data: Looking inside the Black Box, Strategic Organization (16: 2), SAGE Publications, pp.105–140 (2018).
[3] データ分析を支える技術 データモデリング再入門，〈https://dev.classmethod.jp/articles/devio2022-primer-of-data-modeling/〉（参照 2023-11-30）．
[4] DAMA International: DAMA-DMBOK Data Management Body of Knowledge 2nd Edition, Technics Publications (2017).
[5] Kimball, R. and Ross, M.: The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, Wiley (2013).
[6] Linstedt, D. and Olschimke M.: Building a Scalable Data Warehouse with Data Vault 2.0, Morgan Kaufmann (2015).
[7] 八木綾子，金田典久，佐藤重雄，早川孝之：データウエアハウスシステムにおける大規模マスタデータアクセス方法，第 67 回全国大会講演論文集，pp.43–44 (2005)．
[8] 「デジタルバンク」はネット銀行と何が違うのか．「みんなの銀行」の狙い，〈https://www.watch.impress.co.jp/docs/topic/1327280.html〉（参照 2023-11-30）．
[9] 新常識に挑む「みんなの銀行」の戦略．ネット銀行と異なるデジタルバンクとは，〈https://hiptokyo.jp/hiptalk/minnanoginko/〉（参照 2023-11-30）．
[10] みんなの銀行：日本初の「デジタルバンク」として Google Cloud に勘定系を構築．Cloud Spanner で銀行基幹システムで求められる可用性を実現〈https://cloud.google.com/blog/ja/topics/customers/minna-no-ginko-spanner〉（参照 2023-11-30）．
[11] Cloud Spanner | Google Cloud，〈https://cloud.google.com/spanner?hl=ja〉（参照 2023-11-30）．
[12] BigQuery エンタープライズ向けデータ ウェアハウス | Google Cloud〈https://cloud.google.com/bigquery?hl=-ja〉（参照 2023-11-30）．
[13] BigQuery ストレージの概要 | Google Cloud，〈https://cloud.google.com/bigquery/docs/storage_overview?hl=ja〉（参照 2023-11-30）．
[14] 個人情報等の利用目的 | みんなの銀行〈https://corporate.minna-no-ginko.com/privacy/purpose/〉（参照 2023-11-30）．
[15] みんなの Cheer Box | みんなの銀行〈https://www.minna-no-ginko.com/cheerbox/〉（参照 2023-11-30）．
[16] 目的別に貯められる貯蓄預金「ボックス」| みんなの銀行〈https://www.minna-no-ginko.com/service/box/〉（参照 2023-11-30）．

神辺 圭一（正会員）

電気通信大学大学院修了．博士（学術）．（公財）高輝度光科学研究センター，（株）ふくおかフィナンシャルグループ，（株）みんなの銀行を経て，2024 年から福岡工業大学情報工学部助教，現在に至る．

江里口 剛喜（非会員）

久留米大学大学院修了．修士．聖路加国際病院，九州大学病院を経て，2016 年福岡銀行入行．同年 iBank マーケティング（株）出向．2020 年（株）みんなの銀行兼務．iBank マーケティングデータ共創部部長兼みんな銀行データクリエイショングループリーダー，現在に至る．