

# DynamoDB設計 ベストプラクティス事例紹介

---

【RDB開発者向け】

2023年10月26日

Ragate

- **自己紹介**
- **モダンでスケーラブルな DB 設計の考え方**
- **DynamoDB実践テクニック & 事例紹介**
  - 事例 - J Sync株式会社
- **まとめ**
- **自社製品 “DynamoDB 専門設計ツール”のご紹介**

# 自己紹介

(はじめまして🙏)



**久保 翔太**

**プロジェクトマネージャー**  
**( AWS Solutions Architect Professional 認定 )**

**休日の過ごし方**

**1歳7ヶ月の娘👧と公園で遊ぶ**  
**草野球 (外野手)**

## [ 会社概要 ]

社名 Ragate株式会社（ラーゲイト）

---

ミッション 最新技術を大衆化する

---

所在地 東京都中央区日本橋富沢町6-4

---

設立年月日 2017年5月25日

---

資本金 20,000,000円

---

認定 / 受賞

- Amazon DynamoDB SDP 認定
- 内製化支援推進 AWS パートナー企業
- Rising Star Partners of the Year – Japan 受賞 (2022年度)
- Amazon Lambda SDP 認定

---

## [わたしたちの強み]

- ✓ 独自のDynamoDB設計書で手戻りの無いDynamoDB開発
- ✓ AWSサーバーレスを主軸とした、低コスト/高速なAWS構築
- ✓ 豊富なIaC開発実績とノウハウによる低工数なインフラ構築
- ✓ AWSのNoSQLの高度な知見

# モダンで スケーラブルな DB 設計の考え方 (一歩進んだデータベース設計)

## [ データベースの選択 ]

**一つのデータベースで完結を目指さない**  
**ユースケースに応じてデータベースを使い分ける**

- ワークロードに応じて適切なデータベースを選択
- AWSでは数多くのデータベースの選択肢がある

**適切に使い分ける**



# モダンでスケーラブルな DB 設計の考え方

[ 愛用してます ]

キーバリュー

全文検索

リレーショナル

ドキュメント

グラフ



**Dynamo  
DB**



**OpenSearch**



**RDS/Aurora**



**Document  
DB**



**Redshift**

**データベースごとにユースケースを理解し  
適切に使い分けることが大事**

## [ DynamoDBのユースケース ]

# 大規模でも**低レイテンシー** **高速**なデータアクセス

- 大量のデータとイベントを扱うゲーム
- toC向けの大規模なウェブアプリケーション
- 大量データを扱う広告配信 etc.

## [ DynamoDBが解決する課題 ]

### スケール予測の難しいアプリケーション開発

- スタートアップでスケール規模が予測できない
- 低レイテンシー/高速が求められる

### RDBでのスケールアウトの限界

- スケールアップがDBの物理性能や容量に依存
- スケールアウトにはホストの分散などが検討必要

## [ DynamoDBによる課題解決 ]

### ▶ Amazon DynamoDB

どんな規模にも対応する

高速で柔軟なキーバリューストアデータベース

- 規模に関わらず数ミリ秒のレスポンス
- 大量のリクエストの処理可能
- 毎秒2000万を超えるリクエストに対応
- スタートアップビジネスのお財布に優しい

# DynamoDB テクニック & 事例紹介

(インデックスによる、効率良い探索を目指す)

[はじめに]

## DynamoDBの設計=インデックス設計

- インデックスを用いた効率の良い探索を行う
- Scan はなるべく避ける
- インデックスやファセットを活用する
- 最小のテーブル数で設計する
- 複雑な複合検索や全文検索はOpenSearchを併用

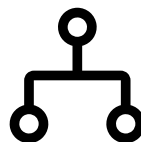
## [ 新規開発における主な設計プロセス ]

**STEP 1**



画面構成案の作成

**STEP 2**



シーケンス図作成

**STEP 3**



モデリング

API 開発の場合は SKIP

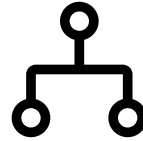
## [ 新規開発における主な設計プロセス ]

STEP 1



画面構成案の作成

STEP 2



シーケンス図作成

STEP 3



モデリング

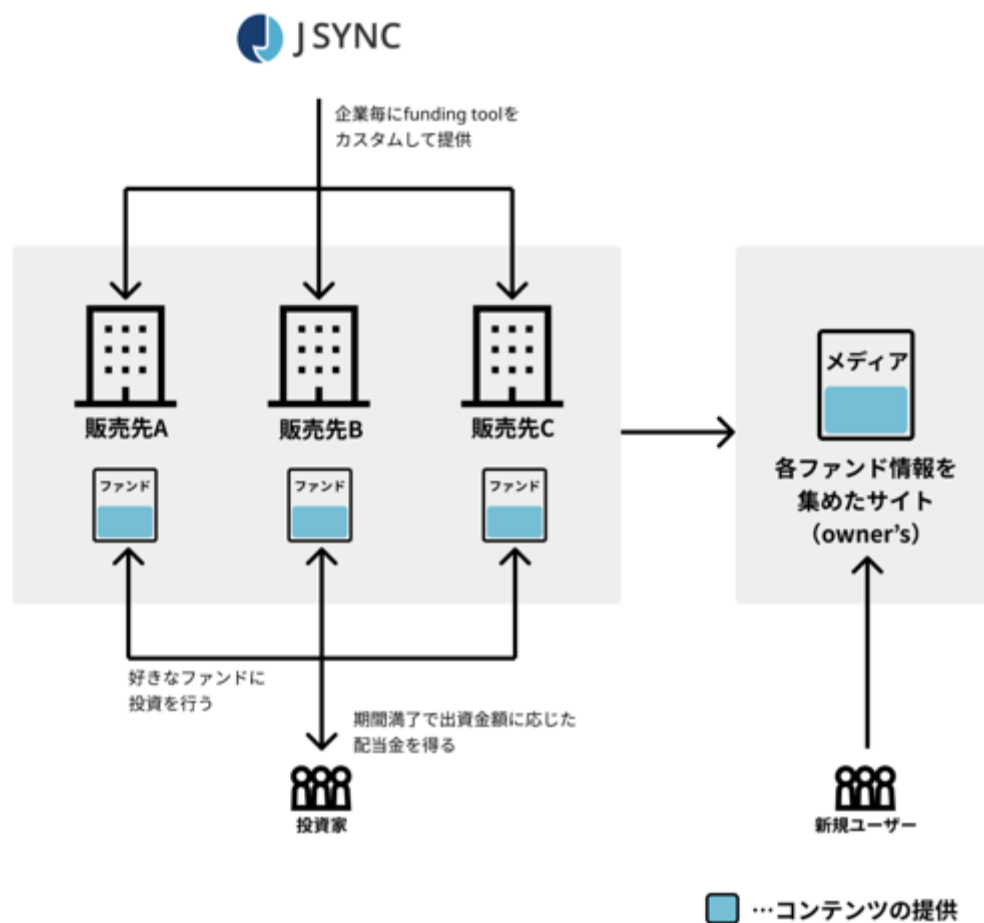
API 開発の場合は SKIP

**必要なデータと  
アクセスパターンを明確化**



# 事例：DynamoDB実践テクニック & 事例紹介

## [ サービス概要 ]



会社名	JSync株式会社
サービス名	ファンディングツール Owner's
サービス内容	不動産クラウド ファンディングシステム
開発期間	6ヶ月
対応範囲	AWS 開発 API 開発 WEBフロントエンド開発

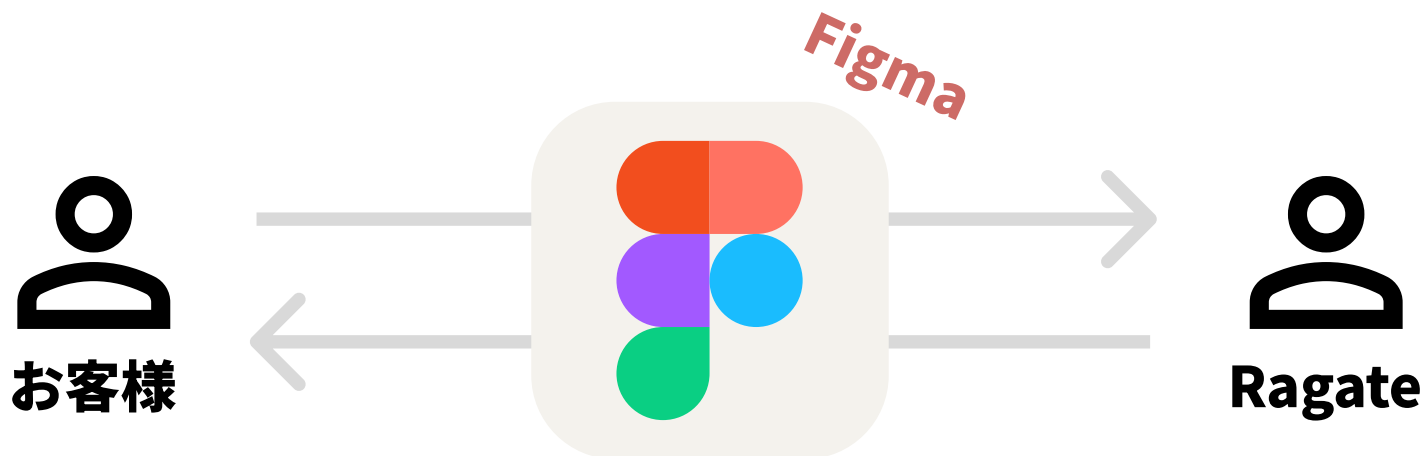
実績URL： <https://aws.amazon.com/jp/solutions/case-studies/jsync-case-study/>

# 事例：DynamoDB実践テクニック & 事例紹介

画面構成案の作成

シーケンス図作成

モデリング

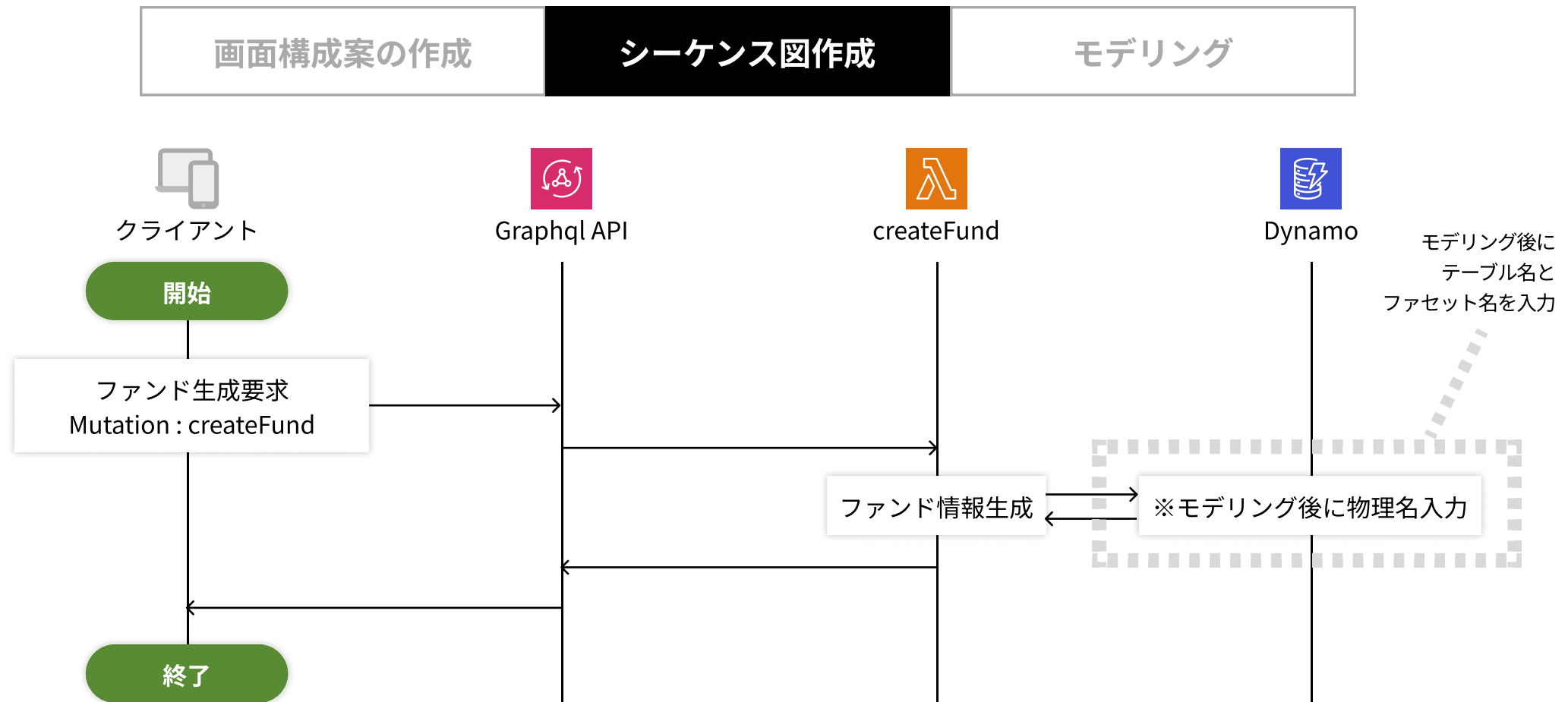


Figmaを活用しお客様の仕様共有コストを削減

構成案からデータアクセス要件を把握

複雑な検索要件にはOpenSearchを検討

# 事例：DynamoDB実践テクニック & 事例紹介



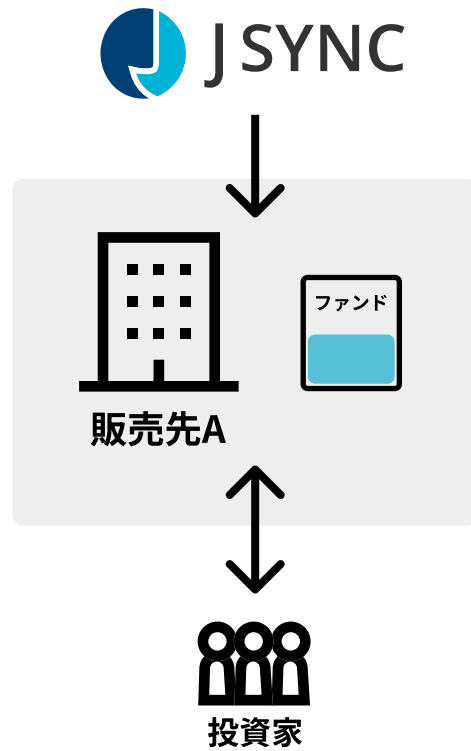
LambdaからDynamoDBへのアクセスパターンを把握  
バッチ処理の必要有無を把握

# 事例：DynamoDB実践テクニック & 事例紹介

画面構成案の作成

シーケンス図作成

モデリング



ファンド情報を  
DynamoDB設計に起こすとき、  
考えることは何か？



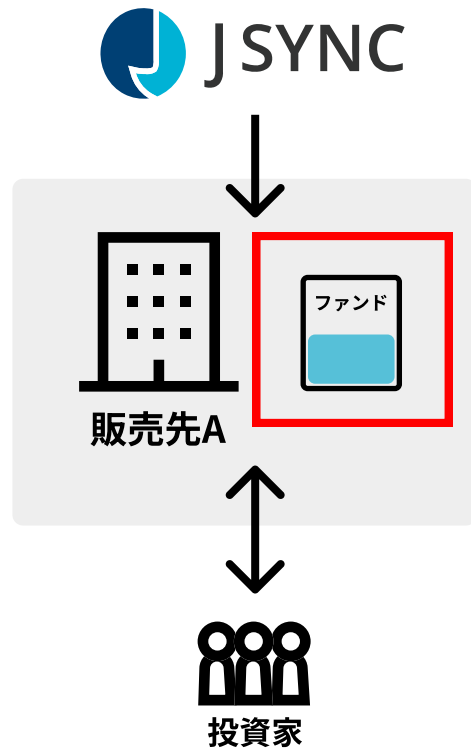
- そもそもファンド情報も複数分かれる..
- 基本的なファンド情報
  - ユーザーが投資したファンド
  - ユーザーがお気に入りしたファンド
  - etc

# 事例：DynamoDB実践テクニック & 事例紹介

画面構成案の作成

シーケンス図作成

モデリング



ファンド情報を  
DynamoDB設計に起こすとき、  
考えることは何か？



- そもそもファンド情報も複数分かれる..
- 基本的なファンド情報
  - ユーザーが投資したファンド
  - ユーザーがお気に入りしたファンド
  - etc

どこに所属する情報なのかを把握すること

画面構成案の作成

シーケンス図作成

モデリング

今回のケースでは  
2つのテーブルに分かれる

販売先企業

企業情報

ファンド情報

ブログ情報

etc..

ユーザー

顧客情報

投資したファンド情報

通知情報

etc..

画面構成案の作成

シーケンス図作成

モデリング

## ファンドテーブルを 1つ作れば良いじゃん

販売先企業

企業情報

ブログ情報

etc..

ファンド

ファンド情報

etc..

ユーザー

顧客情報

通知情報

etc..

画面構成案の作成

シーケンス図作成

モデリング

## テーブルを増やしていくと 管理しなければいけないことが増えていく

- キャパシティユニット
- CloudWatchとの連携
- バックアップ
- インデックス数（上限緩和可能な為、そこまで重要視しない）

ただし、テーブルを  
1つにすることが正義では無い



画面構成案の作成

シーケンス図作成

モデリング

## テーブル数が少ないことのメリット

### インフラ費用の削減

インデックスを使用して効率よくデータ取得を行うことにより、RDSより確実に費用は安くなります。

### 複雑な階層データをシンプルなクエリで取得可能

Jsonを使用して、複数テーブルを取得する等の手間が無くなるなり、実装コストが軽減される。

### テーブル管理が楽になる

運用コストが軽減されます。目に見えにくいコストですがバカには出来ません。

# 事例：DynamoDB実践テクニック & 事例紹介

画面構成案の作成

シーケンス図作成

モデリング

## 販売先企業テーブル

### 企業情報ファセット

Attribute name	Attributes type	Key type	Composite	Sample Value	Comment
Id	String	PartitionKey		7b10d450-b0c3-4aaa-a65d-2c6bc	企業ID
Sk	String	SortKey		About	
Name	String			Ragate	企業名

### ファンド詳細情報ファセット

Attribute name	Attributes type	Key type	Composite	Sample Value	Comment
Id	String	PartitionKey		7b10d450-b0c3-4aaa-a65d-2c6bc	企業ID
Sk	String	SortKey	Fund#{FundId}	Fund#66798edc-8568-4e8b-9c2d-	
FundId	String			4df0d450-b0c3-4dcf-a65d-2c6bd-	ファンドID
Status	String			public	ファンドの状態
FundPublishedAt	String			2023-10-26T16:01:56+09:00	

### GSI ファンド一覧

Attribute name	Key type	Comment
Id	PartitionKey	企業ID
FundPublishedAt	SortKey	ファンド情報を公開した日

※実際の設計とは異なります

# 事例：DynamoDB実践テクニック & 事例紹介

画面構成案の作成

シーケンス図作成

モデリング

## ユーザーテーブル

### ユーザー情報ファセット

Attribute name	Attributes type	Key type	Composite	Sample Value	Comment
Id	String	PartitionKey		7b10d450-b0c3-4aaa-a65d-2c6bc	ユーザーID
Sk	String	SortKey		About	
Name	String			久保翔太	氏名
Status	String			InActive	ユーザーの状態

### 投資ファンド情報ファセット

Attribute name	Attributes type	Key type	Composite	Sample Value	Comment
Id	String	PartitionKey		7b10d450-b0c3-4aaa-a65d-2c6bc	ユーザーID
Sk	String	SortKey	Investment#{FundId}	Investment#66798edc-8568-4e8b	
FundId	String			4df0d450-b0c3-4dcf-a65d-2c6bd	ファンドID
InvestmentAmount	String			10000	投資額
DepositStatus	String			deposited	入金状況

※実際の設計とは異なります

# よくあるご質問

Q：テーブルは絶対に一つにするべきですか？

A：そんなことはありません。テーブルを分ける場合もあります。

テーブルのアクセス数が多い等、テーブル単位で管理しなければいけない要件が発生する場合はテーブルを分けます。

Q：特定の日付のデータを取得したい場合はどのようにしたら良いか？

A：データ抽出アプローチのテクニックの1つである、  
スパースインデックスを使用しましょう。

例えば... GSI 投資額が到達して一週間が経ったファンド一覧

Attribute name	Key type	sample value	Comment
Id	PartitionKey	7b10d450-b0c3-4aaa-a65d-2c6bda81	企業ID
AchievedAt	SortKey	2023-10-26T16:01:56+09:00	投資が達成した日付を登録


- ✓ プロジェクトに応じてデータベースを選択
- ✓ DynamoDBを設計する前に  
必要なデータとアクセスパターンを可視化
- ✓ DynamoDBの設計＝インデックス設計

# 良いDynamo設計ライフを

自社製品 /

# DynamoDB 専門 設計ツールのご紹介




( 最新技術を大衆化する )

 OpenAI 搭載

The Ultimate DynamoDB Design Tool

# 国内初の DynamoDB専用設計ツール

DynamoDB設計を効率化する洗練されたGUIと、  
設計テンプレートでプロジェクトをクイックスタート。

製品ダウンロード   

ブラウザで起動



お問い合わせ：久保 翔太

 [s.kubo@ragate.co.jp](mailto:s.kubo@ragate.co.jp)

# Ragate