【セッションNo．2】

ビッグデータ活用！
Delphi/400からのNoSQL連携テクニック

株式会社ミガロ.
RAD事業部　技術支援課
吉原　泰介

■ はじめに ～ビッグデータとは～

- 近年、情報量の急速な増加により、さまざまな事業で「ビッグデータ」の活用が注目をされています。
「ビッグデータ」とは、いわゆる巨大なデータ群を指します。
具体的には、モバイル端末、IoT機器、インターネット等を通した位置情報・行動履歴や、ホームページやコンテンツの閲覧・視聴に関する情報から得られる膨大なデータなどが該当します。

こうしたビッグデータは単に大容量であるだけでなく、
非定型でかつリアルタイム性が高いことが特徴です。

本セッションではビッグデータを扱う基礎知識から、Delphi/400を使った具体的なプログラミングテクニックまでをご紹介致します。

【アジェンダ】

1.ビッグデータとNoSQL

- 1-1.NoSQLとは？
- 1-2.NoSQLの種類
- 1-3.RDBMSとNoSQLの使い分け

2. Delphi/400からのNoSQL操作

- 2-1.FireDACのNoSQLコンポーネント
- 2-2.MongoDBとは？
- 2-3.データメンテナンスプログラムの作成
- 2-4.コーディングによるデータ操作機能の追加
- 2-5.EnterPrise Connectorsによる対応DB拡張

3.まとめ

1.ビッグデータとNoSQL

1.ビッグデータとNoSQL

1-1.NoSQLとは？

ビッグデータと呼ばれる膨大なデータはNoSQLで使用されることが多い。

NoSQLとは「Not only SQL」の略で、名前の通り

SQL言語を使わず（SQL言語に限定されず）にデータ操作ができるデータベースのこと

NoSQLはRDBMS（リレーショナルデータベース）と性質が異なり、
データをテーブルで管理するのではなく、構造（スキーマ）に
縛られない、スケーラブルな管理が可能。

※広義ではRDBMS以外のDB全てをNoSQLと呼ぶ場合もある。

1.ビッグデータとNoSQL

1-2.NoSQLの種類

NoSQLは大きく分けて4つの種類に分類することができる。

キーバリュー指向型

キーを基準にデータ管理を行う。拡張性に優れ、データ読込が高速。
DB例：Redis、Amazon DynamoDBなど

ドキュメント指向型

ドキュメント形式のデータ管理を行う。JSON等の スキーマレスな構造にも対応。
DB例：MongoDB、CouchDBなど

ワイドカラムストア指向型

データは行で管理するが、アクセスはキーを使用して行う。データ書込が高速。
DB例：Hadoop、Cassandraなど

グラフ指向型

ノード、リレーション、プロパティの3要素でデータ管理を行う。データのグラフ化が容易。
DB例：Neo4j、Amazon Neptuneなど

1.ビッグデータとNoSQL

1-3.RDBMSとNoSQLの使い分け

主要なデータベースと言えばDB2やSQLServer、Oracleなどが代表的。
では、なぜビッグデータはRDBMSではなく、NoSQLで扱うか？

＜RDBMSとNoSQLの優位性比較＞

	RDBMS	NoSQL
定型データ	○	
非定型データ		○
データ間結合	○	
一貫性・整合性	○	
大量データ処理速度		○
DBの拡張		○

ビッグデータの運用に向いている

ただしNoSQLがRDBMSの代わりになるわけではない！

1.ビッグデータとNoSQL

1-3.RDBMSとNoSQLの使い分け

【RDBMSの特徴】→　整合性・一貫性が必要な基幹データはRDBMSが得意

- データを表構造で登録管理して、複数の表を関連定義で制御可能
- 整合性・一貫性のあるデータが保証
- データ量が増えると定義や一貫性の維持の為、パフォーマンスに影響

→　重要な基幹データなど、整合性が必要な定型データに向いている

RDBMS構造例

複数表で関連付
（リレーション）

社員マスタ

社員No	氏名	氏名カナ	年齢	部署コード
1	佐久間大輔	サクマダイスケ	32	2
2	大隈良子	オオクマリョウコ	26	5
3	長井浩二	ナガイコウジ	36	4
4	吉田雄一	ヨシダユウイチ	33	4
5	斎藤恵美	サイトウエミ	28	3
6	山口敏郎	ヤマグチトシロウ	33	
7	佐[illegible]介	サトウケイスケ	43	
8	遠藤[illegible]	エンドウミツル	44	3
9	加藤美紀	[illegible]	47	1
10	江川[illegible]		29	3
……		……	……	……

部署マスタ

部署コード	部署名
1	人事部
2	経理部
3	第一営業部
4	第二営業部
[illegible]	[illegible]三営業部
[illegible]	[illegible]製品開発部
……	……

複数表で関連付
（リレーション）

売上ファイル etc

……	売上金額	担当者No
……	300000	8
……	120000	5
……	50000	5
……	228000	2

キー重複や
整合性を管理

1.ビッグデータとNoSQL

1-3.RDBMSとNoSQLの使い分け

【NoSQLの特徴】→　リアルタイムな大量登録や非定型データ処理が得意

- 構造に縛らず、自由なデータのサイズ、項目で拡張でき、変化に強い
- データ量が増えても関連性はない為、パフォーマンスやDB拡張に強い
- 整合性・一貫性は保証されない

→　非定型なデータやリアルタイムな大量データ処理に向いている

NoSQL構造例

新しい情報を自由に追加できるので整合性はないが、変化に強い！

社員No	氏名
1	佐久間大輔
2	大隈良子
3	長井浩二
4	吉田雄一
5	斎藤恵美

とりあえず
社員名を登録しておこう

社員No	氏名	氏名カナ
1	佐久間大輔	
2	大隈良子	
3	長井浩二	
4	吉田雄一	
5	斎藤恵美	
6	山口敏郎	ヤマグチトシロウ
7	佐藤啓介	サトウケイスケ

氏名のカナ読みが
あったほうがいいな

社員No	氏名	氏名カナ	年齢
1	佐久間大輔		
2	大隈良子		
3	長井浩二		
4	吉田雄一		
5	斎藤恵美		
6	山口敏郎	ヤマグチトシロウ	
7	佐藤啓介	サトウケイスケ	
8	遠藤充	エンドウミツル	44
9	加藤美紀	カトウミキ	47
10	江川登	エガワノボル	29
….	….	….	….

年齢情報も
追加しよう

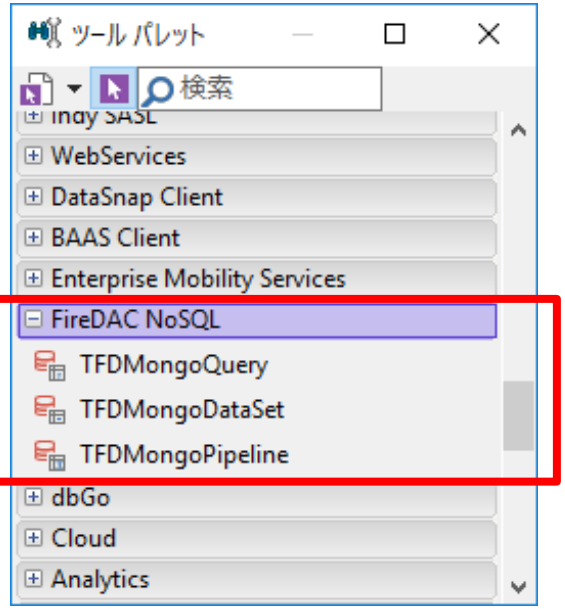
2. Delphi/400からのNoSQL操作

2. Delphi/400からのNoSQL操作

2-1.FireDACのNoSQLコンポーネント

Delphi/400 10 Seattle以降では、FireDACが実装されており、ツールパレットで「**FireDAC NoSQL**」というカテゴリにMongoDB専用のコンポーネントが用意されている。

ただしMongoDBはRDBMSと異なり、SQLを使用しないので、データ操作にはNoSQLの実装知識も必要！



2. Delphi/400からのNoSQL操作

2-2.MongoDBとは？

MongoDBは、クロスプラットフォーム対応のオープンソースの
NoSQLデータベース（ドキュメント指向型）。
NoSQLでは最もシェアを獲得しており、世界のDBランキングでも5位。
JSONをドキュメントとして扱うことができ、大量データ処理に最適。
ただしRDBMSと違い、トランザクションなどの整合性は持たない。

Rank			DBMS	Database Model	Score		
Oct 2018	Sep 2018	Oct 2017			Oct 2018	Sep 2018	Oct 2017
1.	1.	1.	Oracle	Relational DBMS	1319.27	+10.15	-29.54
2.	2.	2.	MySQL	Relational DBMS	1178.12	-2.36	-120.71
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1058.33	+7.05	-151.99
4.	4.	4.	PostgreSQL	Relational DBMS	419.39	+12.97	+46.12
5.	5.	5.	MongoDB	Document store	363.19	+4.39	+33.79
6.	6.	6.	DB2	Relational DBMS	179.69	-1.38	-14.90
7.	↑ 8.	↑ 9.	Redis	Key-value store	145.29	+4.35	+23.24
8.	↓ 7.	↑ 10.	Elasticsearch	Search engine	142.33	-0.28	+22.09
9.	9.	↓ 7.	Microsoft Access	Relational DBMS	136.80	+3.41	+7.35
10.	10.	↓ 8.	Cassandra	Wide column store	123.39	+3.83	-1.40

出典元：DB-Engines　URL:https://db-engines.com/en/ranking

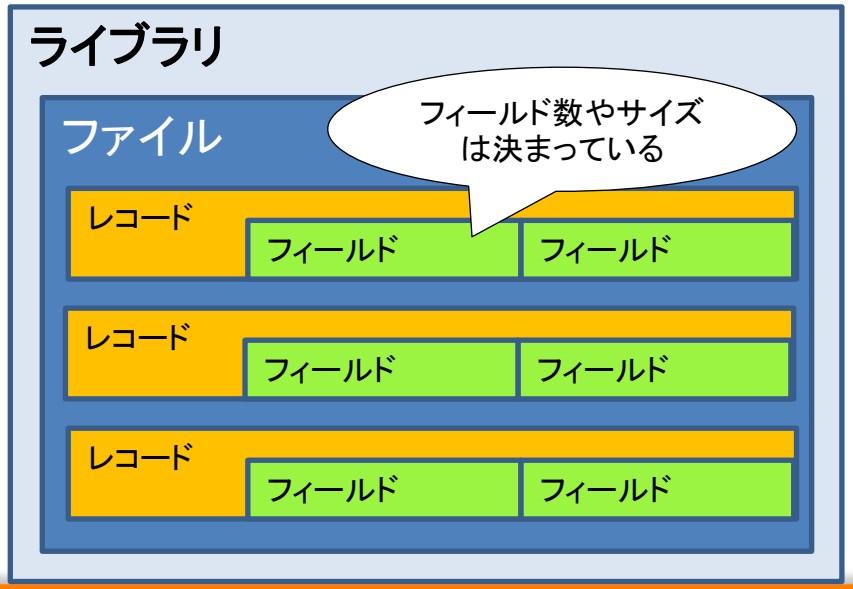
2. Delphi/400からのNoSQL操作

2-2.MongoDBとは？

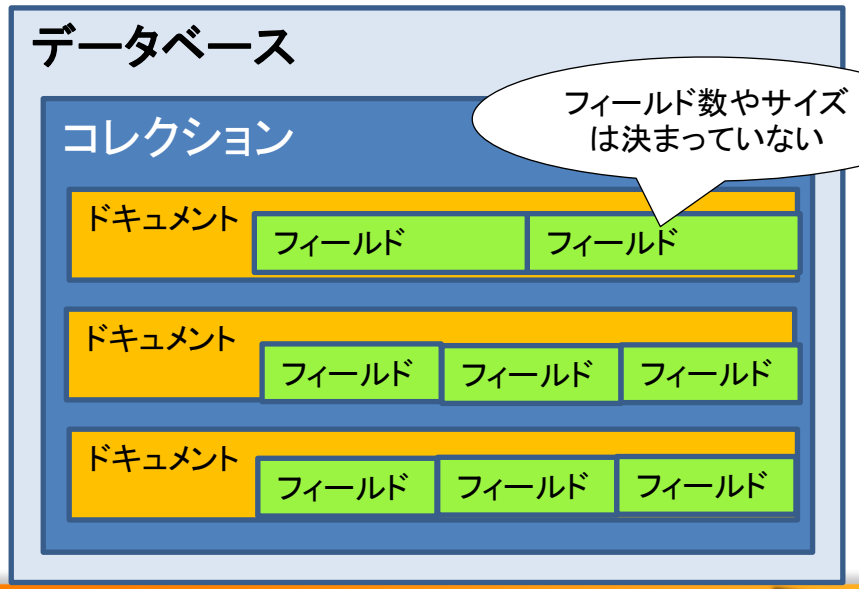
IBM i とMongoDBの構造的な違い

IBM i 用語	MongoDB 用語
ライブラリ	データベース
ファイル	コレクション
レコード（行）	ドキュメント
フィールド（列）	フィールド

IBM i 例



MongoDB 例



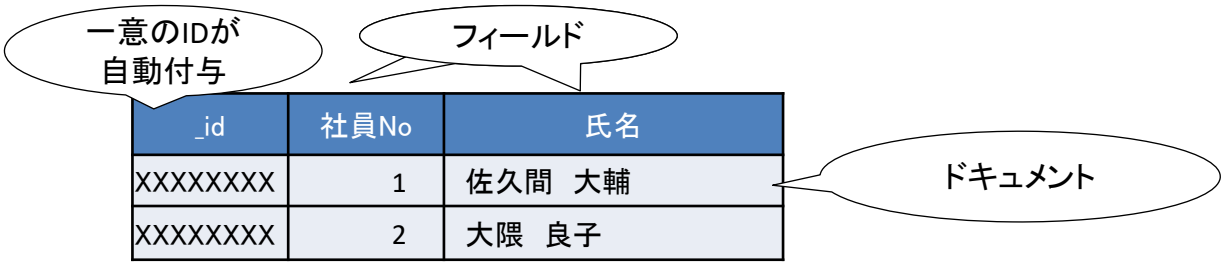
2. Delphi/400からのNoSQL操作

2-2.MongoDBとは？

【本セッションで使用するMongoDB情報】

バージョン：MonogoDB 4.0.2
ポート：27017
データベース：D400Mongo
コレクション：Tec2018

```
MongoDB shell version v4.0.2
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 4.0.2
```



Tec2018の実際のデータ

```
> db.Tec2018.find()
{ "_id" : ObjectId("5bc6e59b061ae6324c006dd1"), "社員No" : " 1", "氏名" : "佐久間　大輔" }
{ "_id" : ObjectId("5bc6e5b4061ae6324c006dd2"), "社員No" : " 2", "氏名" : "大隈　良子" }
{ "_id" : ObjectId("5bc6e5c5061ae6324c006dd3"), "社員No" : " 3", "氏名" : "長井　浩二" }
{ "_id" : ObjectId("5bc6e5ef061ae6324c006dd5"), "社員No" : " 4", "氏名" : "吉田　雄一" }
{ "_id" : ObjectId("5bc6e5fe061ae6324c006dd6"), "社員No" : " 5", "氏名" : "斎藤　恵美" }
{ "_id" : ObjectId("5bc6e611061ae6324c006dd7"), "社員No" : " 6", "氏名" : "山口　敏郎" }
{ "_id" : ObjectId("5bc6e620061ae6324c006dd8"), "社員No" : " 7", "氏名" : "佐藤　啓介" }
{ "_id" : ObjectId("5bc6e62a061ae6324c006dd9"), "社員No" : " 8", "氏名" : "加藤　美紀" }
{ "_id" : ObjectId("5bc6e636061ae6324c006dda"), "社員No" : " 9", "氏名" : "遠藤　充" }
```

2. Delphi/400からのNoSQL操作

2-3.データメンテナンスプログラムの作成

作成するプログラム例

_id	社員No	氏名
5[illegible]DD1	1	佐久間　大輔
5BC6E5[illegible]C006DD2	2	大隈　良子
5BC6E5C5061AE6324C006DD3	3	長井　浩二
5BC6E5EF061AE6324C006DD5	4	吉田 雄一
5BC6E5FE061AE6324C006DD6	5	斎藤　恵美
5BC6E611061AE6324C006DD7	6	山口　敏郎
5BC6E620061AE6324C006DD8	7	佐藤　啓介
5BC6E62A061AE6324C006DD9	8	加藤　美紀
5BC6E636061AE6324C006DDA	9	遠藤　充

2. Delphi/400からのNoSQL操作

2-3.データメンテナンスプログラムの作成

MongoDBの操作に使用するコンポーネント

TFDConnection

FireDACのデータベース接続用コンポーネント。
TFDphysMongoDriverLinkでドライバ情報を読み込むことでMongoDB専用の設定が可能。

TFDphysMongoDriverLink

FireDACでMongoDBを操作する為の専用ドライバ情報をプログラムに提供するコンポーネント。
配置するだけで設定は不要。

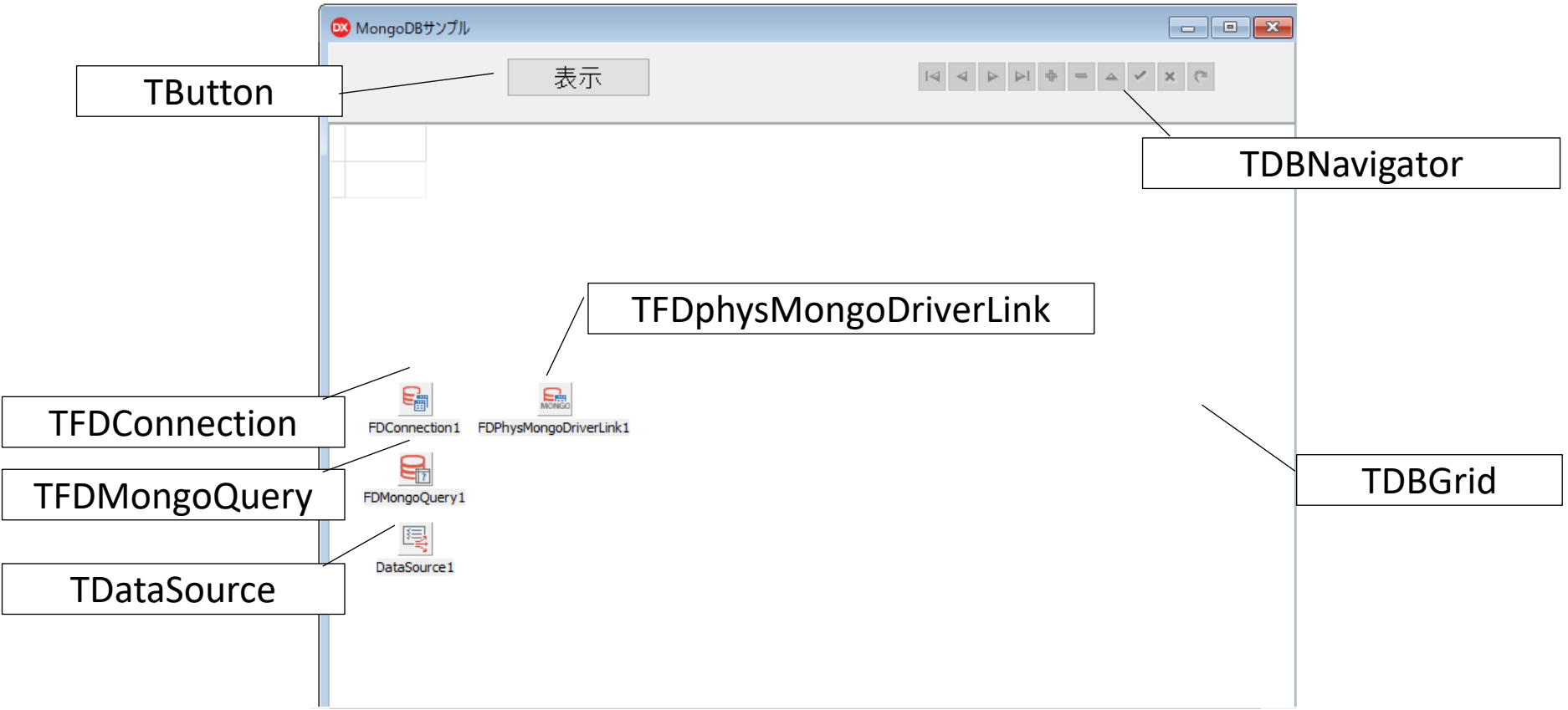
TFDMongoQuery

FireDACでMongoDBのデータを操作するための専用コンポーネント。
通常のTFDQueryはSQL用なので使用しない。

2. Delphi/400からのNoSQL操作

2-3.データメンテナンスプログラムの作成

コンポーネントの配置



2. Delphi/400からのNoSQL操作

2-3.データメンテナンスプログラムの作成

コンポーネントの設定①

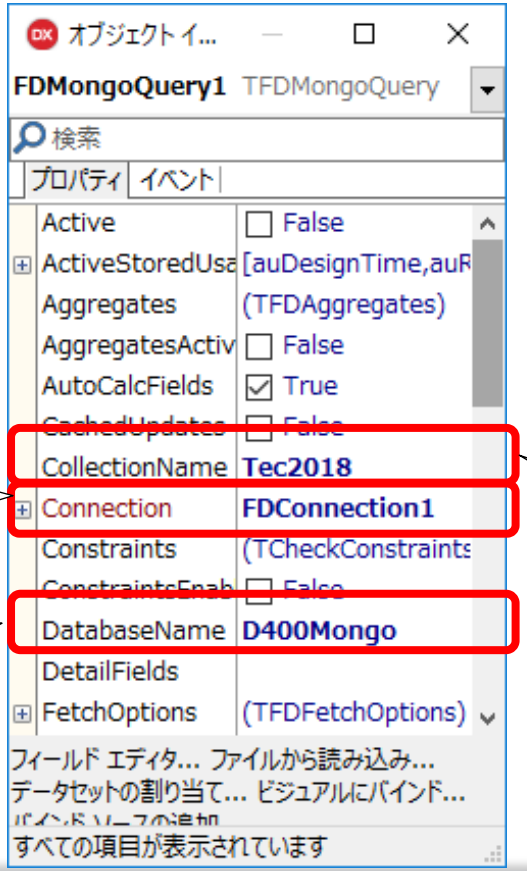
TFDConnectionの設定

2. Delphi/400からのNoSQL操作

2-3.データメンテナンスプログラムの作成

コンポーネントの設定②

TFDMongoQueryの設定



①Connectionは
「FDConnection1」

②DataBaseNameは
「D400Mongo」

③CollectionNameは
「Tec2018」

2. Delphi/400からのNoSQL操作

2-3.データメンテナンスプログラムの作成

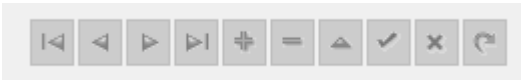
コンポーネントの設定③

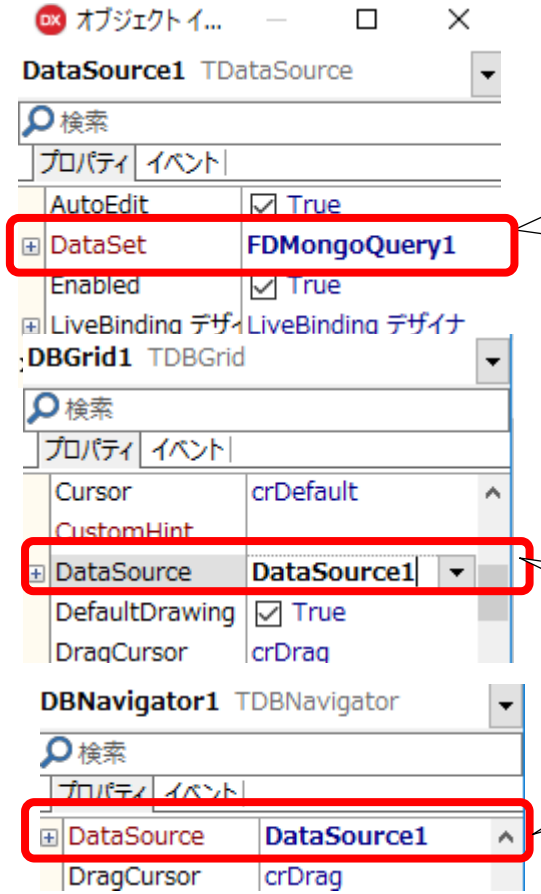
TDataSourceの設定

DataSource1

TDBGridの設定

TDBNavigatorの設定





DataSetに
「FDMongoQuery1」を設定

Datasourceに
「DataSource1」を設定

2. Delphi/400からのNoSQL操作

2-3.データメンテナンスプログラムの作成

データベース接続の実装

OnCreateイベント(起動時に接続)

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  FDConnection1.Connected := True;  //MongoDB接続
end;
```

データ表示の実装

OnClickイベント(表示処理)

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  FDMongoQuery1.Close;
  FDMongoQuery1.Open;
end;
```

通常のRDBMSと同じ操作

2. Delphi/400からのNoSQL操作

2-3.データメンテナンスプログラムの作成

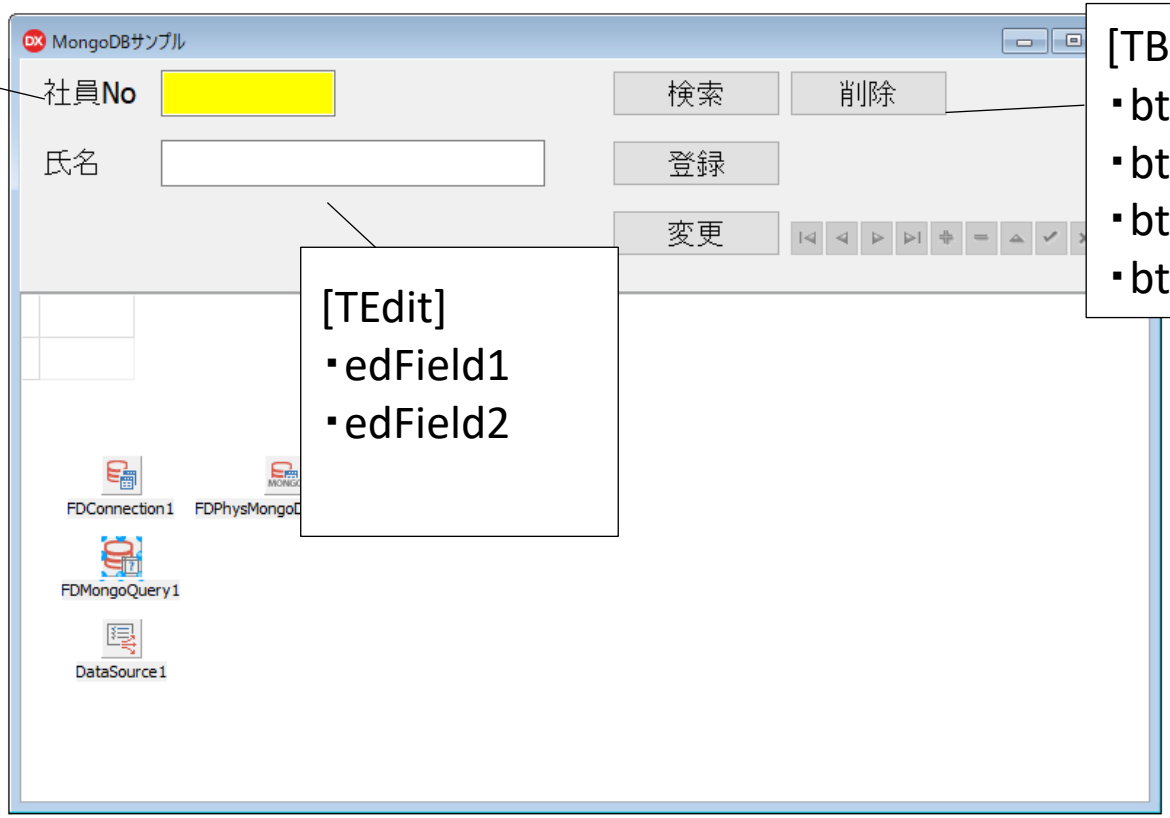
_id	社員No	氏名
5BC6E59B061AE6324C006DD1	1	佐久間　大輔
5BC6E5B4061AE6324C006DD2	2	大隈　良子
5BC6E5C5061AE6324C006DD3	3	長井　浩二
5BC6E5EF061AE6324C006DD5	4	吉田 雄一
5BC6E5FE061AE6324C006DD6	5	斎藤　恵美
5BC6E611061AE6324C006DD7	6	山口　敏郎
5BC6E620061AE6324C006DD8	7	佐藤　啓介
5BC6E62A061AE6324C006DD9	8	加藤　美紀
5BC6E636061AE6324C006DDA	9	遠藤　充

2. Delphi/400からのNoSQL操作

2-4.コーディングによるデータ操作機能の追加

コンポーネントの追加

[TLable]
・lbField1
・lbField2



[TEdit]
・edField1
・edField2

[TButton]
・btnSearch
・btnAdd
・btnChange
・btnDelete

2. Delphi/400からのNoSQL操作

2-4.コーディングによるデータ操作機能の追加

データベース接続の追加実装

グローバル変数宣言

```
private
   { Private 宣言 }
   FCon: TMongoConnection; //MongoDB接続情報
   FEnv: TMongoEnv;        //MongoDB環境情報
```

OnCreateイベント(起動時に接続)

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  FDConnection1.Connected := True;  //MongoDB接続
  FCon := TMongoConnection(FDConnection1.CliObj); //接続情報取得
  FEnv := FCon.Env;                               //環境情報取得
end;
```

データ操作の処理で
必要になる！

2. Delphi/400からのNoSQL操作

2-4.コーディングによるデータ操作機能の追加

データ検索機能の実装

OnClickイベント(検索処理)

```
procedure TForm1.btnSearchClick(Sender: TObject);
begin
  FDMongoQuery1.Close;
  //No条件があれば設定
  if (edField1.Text <> '') then
  begin
    FDMongoQuery1.QMatch :=
    '{'+ QuotedStr(lbField1.Caption) + ':' + QuotedStr(edField1.Text) + '}';
  end
  //No条件がなければ全検索
  else
  begin
    FDMongoQuery1.QMatch := '';
  end;
  FDMongoQuery1.Open;
end;
```

【検索条件】
SQLのWhere句相当
{フィールド名:絞り込み値}

2. Delphi/400からのNoSQL操作

2-4.コーディングによるデータ操作機能の追加

データ登録機能の実装

OnClickイベント(登録処理)

```
procedure TForm1.btnAddClick(Sender: TObject);
var
  MongoDoc: TMongoDocument;
begin
  MongoDoc := FEnv.NewDoc;  //ドキュメント生成
  try
    MongoDoc.Add(lbField1.Caption, edField1.Text)
            .Add(lbField2.Caption, edField2.Text);
    FCon['D400Mongo']['Tec2018'].Insert(MongoDoc);
  finally
    MongoDoc.Free;          //ドキュメント破棄
  end;

  FDMongoQuery1.Close;
  FDMongoQuery1.Open;

end;
```

【登録内容】
SQLのInsert句相当
Add(フィールド名、フィールド値)

データを開き直す
(Refreshではダメ)

2. Delphi/400からのNoSQL操作

2-4.コーディングによるデータ操作機能の追加

データ更新機能の実装

OnClickイベント(更新処理)

```
procedure TForm1.btnChangeClick(Sender: TObject);
begin
  FCon['D400Mongo']['Tec2018'].Update()
    .Match()
      .Add(lbField1.Caption, edField1.Text)
    .&End
    .Modify()
      .&Set()
        .Field(lbField2.Caption, edField2.Text)
      .&End
    .&End
  .Exec;

  FDMongoQuery1.Close;
  FDMongoQuery1.Open;
end;
```

【更新条件】
SQLのWhere句相当
Add(フィールド名、フィールド値)

【更新内容】
SQLのSet句相当
Field(フィールド名、フィールド値)

データを開き直す
(Refreshではダメ)

2. Delphi/400からのNoSQL操作

2-4.コーディングによるデータ操作機能の追加

データ削除機能の実装

OnClickイベント(削除処理)

```
procedure TForm1.btnDeleteClick(Sender: TObject);
begin
  FCon['D400Mongo']['Tec2018'].Remove()
    .Match()
      .Add(lbField1.Caption, edField1.Text)
    .&End
  .Exec;

  FDMongoQuery1.Close;
  FDMongoQuery1.Open;
end;
```

【削除条件】
SQLのWhere句相当
Add(フィールド名、フィールド値)

データを開き直す
(Refreshではダメ)

2. Delphi/400からのNoSQL操作

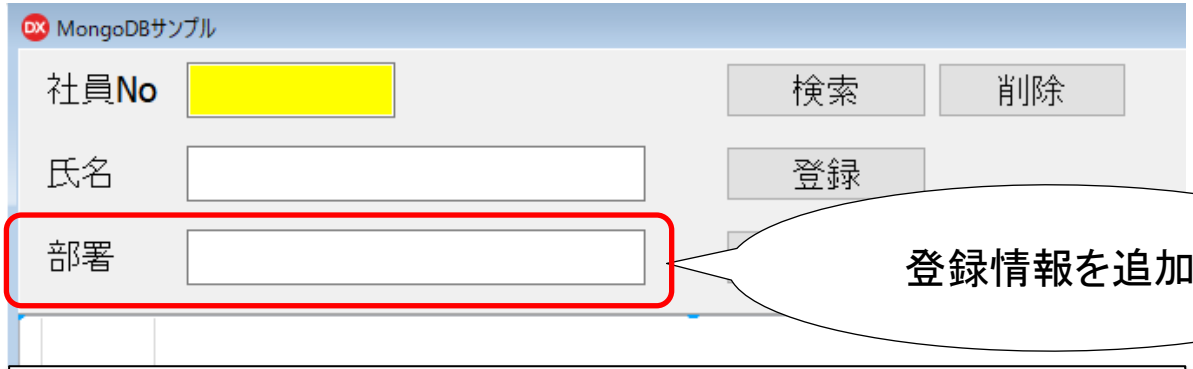
2-4.コーディングによるデータ操作機能の追加

完成！

2. Delphi/400からのNoSQL操作

2-4.コーディングによるデータ操作機能の追加（補足）

登録フィールドを追加する



OnClickイベント（登録処理）

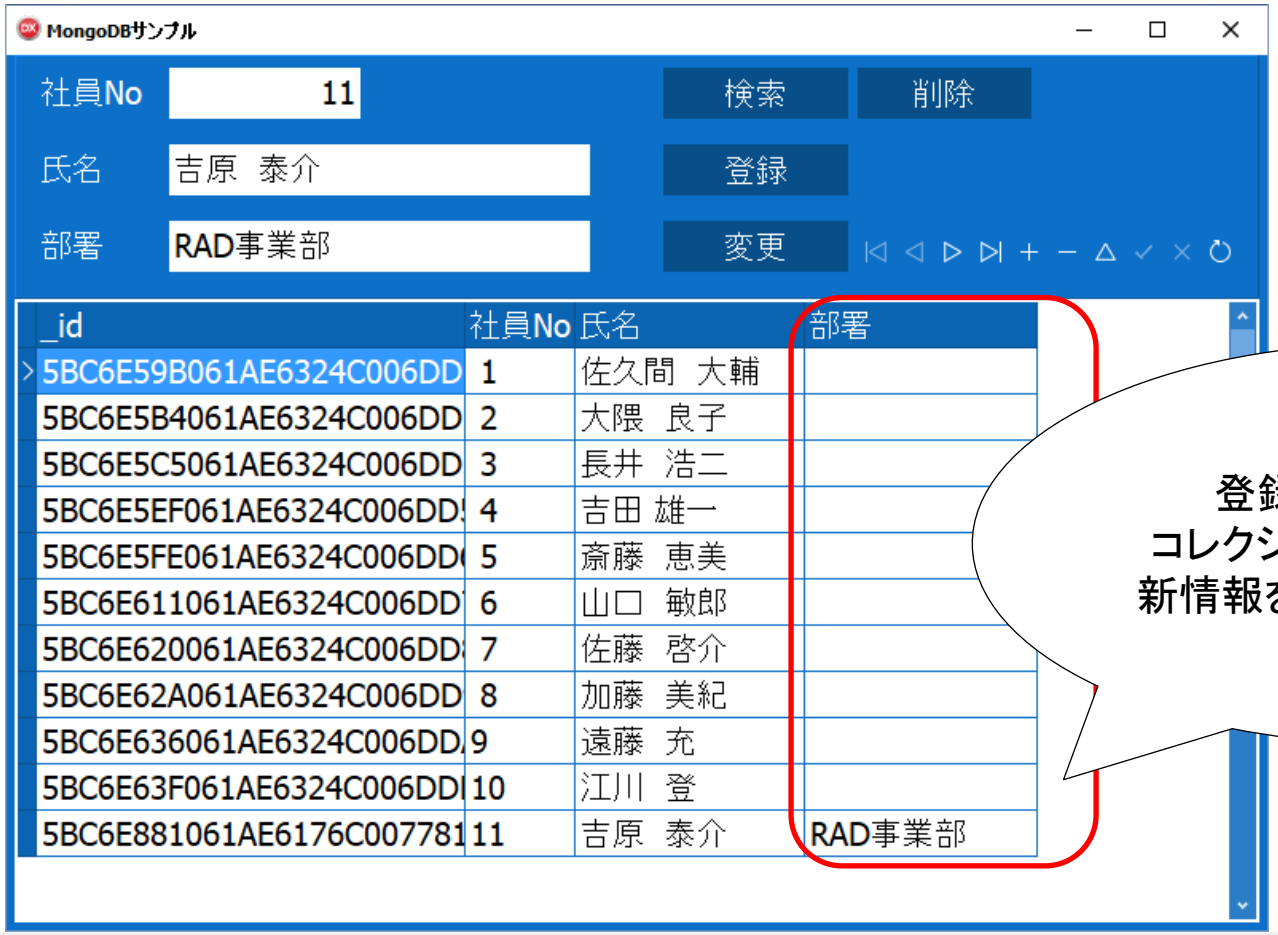
```
procedure TForm1.btnAddClick(Sender: TObject);
var
  MongoDoc: TMongoDocument;
begin
  MongoDoc := FEnv.NewDoc;  //ドキュメント生成
  try
    MongoDoc.Add(lbField1.Caption, edField1.Text)
            .Add(lbField2.Caption, edField2.Text)
            .Add(lbField3.Caption, edField3.Text);
    FCon['D400Mongo']['Tec2018'].Insert(MongoDoc);
```

ドキュメントに登録する
フィールドを追加

2. Delphi/400からのNoSQL操作

2-4.コーディングによるデータ操作機能の追加（補足）

NoSQLではDDSを再定義することなく拡張される



登録情報が増えると
コレクション側が自動拡張。
新情報をすぐに追加できる！

2. Delphi/400からのNoSQL操作

2-4.コーディングによるデータ操作機能の追加（補足）

パフォーマンス：1万件のデータ登録

OnClickイベント(一括登録処理)

```
procedure TForm1.btnSpeedClick(Sender: TObject);
var
  MongoDoc: TMongoDocument;
  i: integer;
  No: String;
  ms: cardinal;
begin
  ms := GetTickCount;
  //10000件のループでデータ登録を行う
  for i := 1 to 10000 do
  begin
    No := FormatFloat('00000', i);
    MongoDoc := FEnv.NewDoc;
    try
      MongoDoc.Add(lbField1.Caption, No)
              .Add(lbField2.Caption, edField2.Text + No)
              .Add(lbField3.Caption, edField3.Text + No);
      FCon['D400Mongo']['Tec2018'].Insert(MongoDoc);
```

データはループ値を
元にダミーデータを登録

2. Delphi/400からのNoSQL操作

2-4.コーディングによるデータ操作機能の追加（補足）

パフォーマンス：1万件のデータ登録

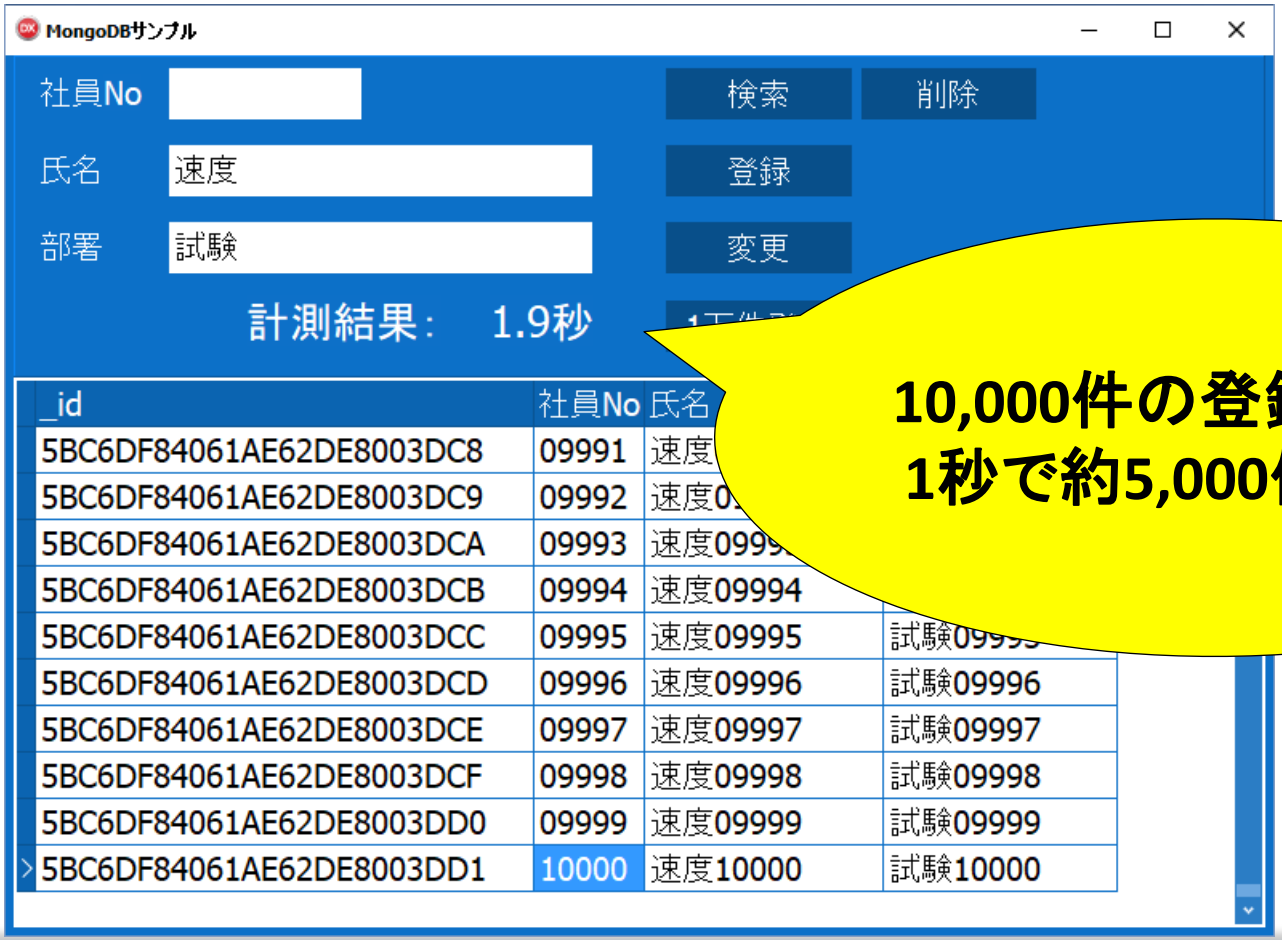
OnClickイベント(一括登録処理　続き)

```
  finally
      MongoDoc.Free;
    end;
  end;

  FDMongoQuery1.Close;
  FDMongoQuery1.Open;

  //結果をラベルに表示
  lbSpeed.Caption := FormatFloat('0.0', (GetTickCount - ms) / 1000) + '秒';

end;
```

2. Delphi/400からのNoSQL操作

2-4.コーディングによるデータ操作機能の追加（補足）

パフォーマンス：1万件のデータ登録



10,000件の登録が1.9秒
1秒で約5,000件登録！

2. Delphi/400からのNoSQL操作

2-4.コーディングによるデータ操作機能の追加（補足）

【MongoDBを使う上での注意点】

RDBMSと違い、下記点はアプリケーション側で考慮が必要

- **トランザクションでの排他は使えない**
- **データの整合性はDB側で保証されない**

パフォーマンスや自由度が高くなる反面、
RDBMSのようなデータ精度は犠牲にしているので
用途によって使い分けることがNoSQL活用の最重要ポイント

2. Delphi/400からのNoSQL操作

2-5.EnterPrise Connectorsによる対応DB拡張（有償）

「EnterPrise Connectors」はFireDACを使ってsalesforceやAWS、SAP、ERP、**ビッグデータDB**、決済、ソーシャルサービスなど、80種類を超えるクラウドやエンタープライズサービスへの機能を追加できるソリューション。これにより下記のようなNoSQLがDelphi/400で扱えるように拡張できる。

NoSQL & ビッグデータ対応

- Amazon Athena
- Amazon DynamoDB
- Amazon Redshift
- Apache Cassandra
- Apache HBase
- Apache Hadoop Hive
- Apache Spark SQL
- Azure Cosmos DB
- Azure Table
- Elasticsearch
- Couchbase Server
- Google BigQuery
- HPCC Systems
- IBM Cloudant NoSQL DB
- Microsoft Access
- Microsoft Active Directory
- MongoDB
- Redis
- xBase

3.まとめ

3.まとめ

- ビッグデータではNoSQLの技術が活用されている。
- FireDACでは標準でMongoDBが利用できる。
- NoSQLはRDBMSの代用ではないので、
  用途によってDBを使い分けて連携することが重要。
  （Delphi/400ではそうした連携が得意）
- EnterpriseConnectorsを使って拡張すれば
  様々なNoSQLDBがFireDACで利用できる。