
Micro Focus Visual COBOL チュートリアル

OpenESQL アシスタントを利用したデータベースアクセス

1. 目的

レガシーな COBOL 言語で開発された企業システムからデータベースのアクセスを行うには複雑な技術を要するというイメージをお持ちの方も多いかもしれませんが、Visual COBOL を使えば驚くほど簡単に開発が可能です。一般的にリレーショナル・データベースのデータ操作には SQL を利用します。SQL は、COBOL と異なるデータベース言語となるため、COBOL コンパイラは SQL 文を解釈できません。そこで、COBOL 上で SQL 文を利用する場合、EXEC SQL と END-EXEC で囲んだ部分のみに SQL 文を記述し、COBOL と区別させます。この SQL 文が埋め込まれたソースを Oracle などデータベースベンダが提供するプリコンパイラに渡すことで、SQL 文の部分をデータベースベンダが提供する API コールに展開した COBOL プログラムを生成します。プログラマーが当初作成するソースはこのソースではなくプリコンパイル展開する前のソースになるためデバッグ時のコードと作成時のコードに相違が生まれてしまいます。

Visual COBOL はプログラマーが実際にメンテナンスする埋め込み SQL 文が入ったソースを IDE に認識させるべく、プリコンパイルとコンパイルをシングルステップで処理できるようにしました。プリコンパイル、もしくはそれに相当する処理を Visual COBOL が内部的に処理します。これにより、プリコンパイル後のソースは扱うことがなくなるため、このソースに関する考慮は不要となりました。

Visual COBOL はこのプリコンパイルを意識させない技術に関して技術や利用する DBMS に応じて柔軟に技術選択いただけるよういくつかのオプションを用意しています。その中でも今回紹介する OpenESQL は、マイクロフォーカス純正のプリプロセッサです。また、OpenESQL によるデータベースアクセスを行う場合、「OpenESQL アシスタント」という開発補助ユーティリティを利用できます。

このチュートリアルでは、OpenESQL アシスタントを使用して OpenESQL によるデータベースアクセスの方法を学びます。

2. 前提条件

本チュートリアルは、下記の環境を前提に作成されています。

- データベースサーバー

OS	Windows Server 2019 Standard Edition (64bit)
DBMS 製品	SQL Server 2016 Enterprise Edition

- 開発クライアント ソフトウェア

OS	Windows 10 Enterprise Edition
COBOL 製品	Visual COBOL 7.0

- チュートリアル用サンプルデータベーススクリプト

下記のリンクから事前にチュートリアル用のサンプルデータベースの SQL ファイルをダウンロードして、任意のフォルダに解凍しておいてください。

[サンプルデータベースの SQL スクリプトダウンロード](#)

内容

1. 目的
2. 前提条件
3. チュートリアル手順の概要
 - 3.1. データベースの準備
 - 3.2. Eclipse プロジェクトの作成と設定
 - 3.3. COBOL プログラムの作成とオプション設定
 - 3.4. ODBC データソースの設定
 - 3.5. クエリーの組み立てとテスト
 - 3.6. SQL プログラムの生成と埋め込み
 - 3.7. 追加のコードを記述
 - 3.8. プログラムを実行し動作を確認

3. チュートリアル手順の概要

3.1. データベースの準備

1) チュートリアル用データベースの作成

チュートリアルで使用するデータベースとサンプルテーブル、データのインポートを行う SQL を実行します。

① SQL Server Management Studio の実行

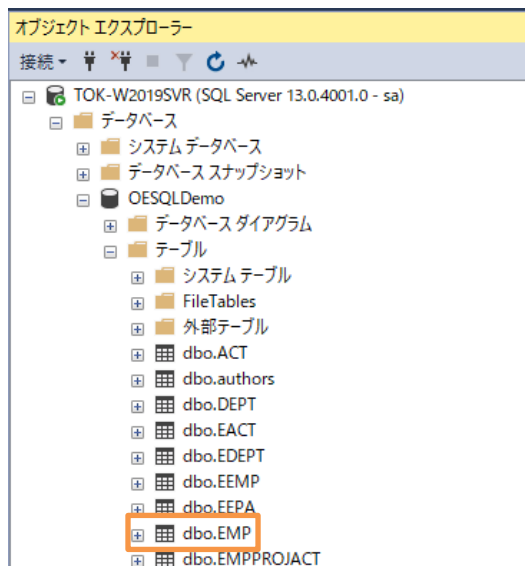
Microsoft SQL Server Management Studio を起動し、管理者でログインを行います。

② SQL スクリプトの実行

ダウンロードした SQL スクリプトの内容をコピー/ペーストしてクエリーを実行します。

③ 実行結果の確認

初めて実行する場合はデータベースが存在しないので drop database 文だけが失敗します。[オブジェクトエクスプローラー] ツリーにてデータベースとして「OESQLDemo」が作成されており、EMP テーブルが存在することを確認します。



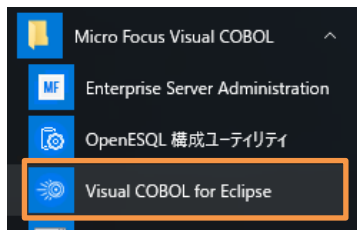
④ SQL Server Management Studio の終了

閉じるボタンにて「Microsoft SQL Server Management Studio」を終了します。

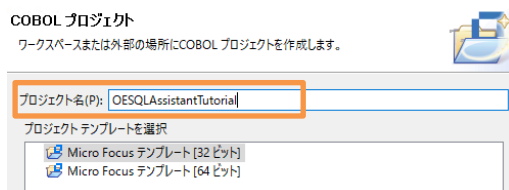
3.2. Eclipse プロジェクトの作成と設定

1) Visual COBOL for Eclipse の起動とプロジェクトの作成

- ① スタートメニューより [Micro Focus Visual COBOL] > [Visual COBOL for Eclipse] を選択します。



- ② ワークスペースの選択画面は任意のワークスペースを指定して、[起動(L)] ボタンをクリックします。
- ③ Eclipse メニューより、[ファイル(F)] > [新規(N)] > [COBOL プロジェクト] を選択し、プロジェクト名に "OESQLAssistantTutorial" を指定して、[終了(F)] ボタンをクリックします。



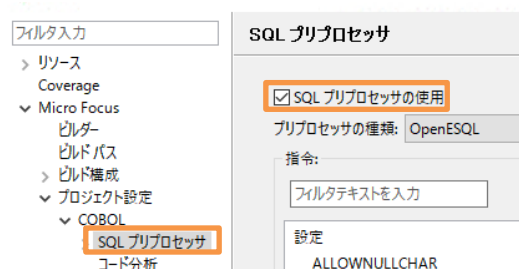
3.3. COBOL プログラムの作成とオプション設定

1) COBOL プログラムを作成

- ① COBOL エクスプローラーにて作成した「OESQLAssistantTutorial」プロジェクトを右クリックし、コンテキストメニューから [新規作成(N)] > [COBOL プログラム] を選択します。
- ② [含まれるプロジェクト] に「OESQLAssistantTutorial」が指定されていること、新規ファイル名が「Program1.cbl」になっていることを確認し、[終了(F)] ボタンをクリックします。

2) プロジェクトプロパティの指定

- ① プロジェクトの構成を変更します。COBOL エクスプローラーにて作成した「OESQLAssistantTutorial」プロジェクトを右クリックし、コンテキストメニューから [プロパティ(R)] を選択します。
- ② プロパティ設定ダイアログが表示されます。[Micro Focus] > [プロジェクト設定] > [COBOL] > [SQL プリプロセッサ] をクリックし、[SQL プリプロセッサの使用] にチェックします。

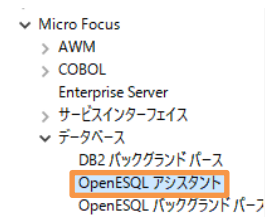


- ③ [プリプロセッサの種類] では「OpenESQL」を選択します。
- ④ [指令] リストから [BEHAVIOR] にて「ANSI」を選択し、[Apply and Close] ボタンをクリックします。

設定	値
AUTO COMMIT	
AUTO FETCH	
BEHAVIOR	ANSI
BGP	
CHECK	
CHECKDUPCURSOR	
CHECKSINGLETON	
CONCAT	

3) OpenESQL アシスタントの設定変更

- ① Eclipse メニューより [ウィンドウ(W)] > [設定(P)] を選択し、設定ダイアログを開き、[Micro Focus] > [データベース] > [OpenESQL アシスタント] と遷移します。



- ② [DCLGEN] の [COBOL 変数の型] を「標準」から「SQL TYPE」へ変更します。

▼ DCLGEN	
ホスト変数の接頭語	表名
インジケータ変数	コピーブック末尾に生成する
構文名	DCL + 変名
COBOL 変数の型	SQL TYPE
レベル 49を使う	false

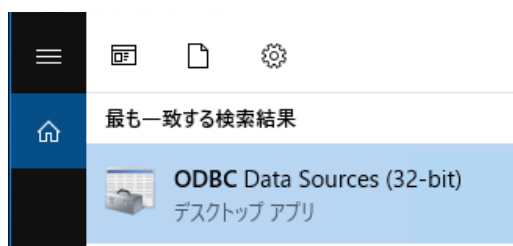
- ③ [一般] の [モード] に「ODBC」を設定し、[Apply and Close] ボタンをクリックします。

▼ 一般	
自動的にカタログ	False
コードの生成	そのまま
項目を表名順に表示する	True
モード	ODBC

3.4. ODBC データソースの設定

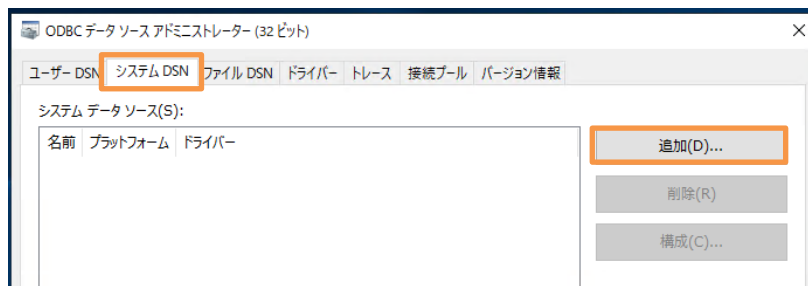
1) ODBC データソースアドミニストレーター (32 ビット) の起動

- ① スタートメニュー横の検索にて "ODBC" とタイプし、検索候補から「ODBC Data Sources (32 ビット)」を起動します。

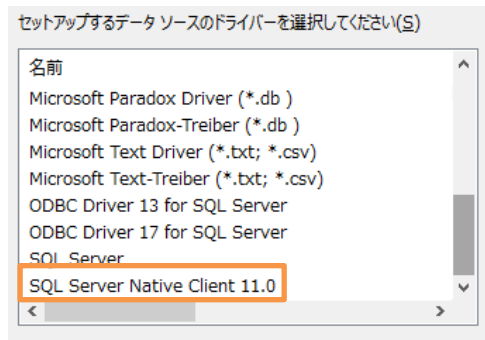


2) データソースの設定と接続確認

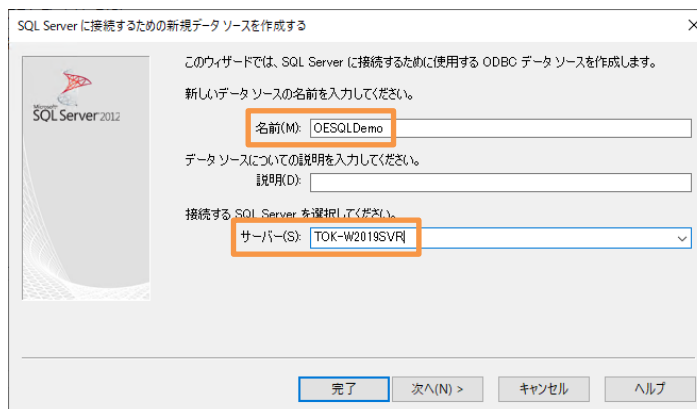
- ① ODBC データソースアドミニストレーター (32 ビット) 画面より [システム DSN] タブをクリックし、[追加(D)...] ボタンをクリックします。



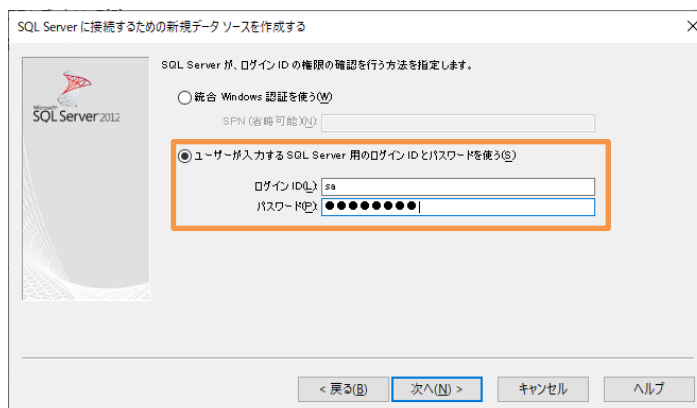
- ② [データソースの新規作成] ダイアログが表示されるので「SQL Server Native Client 11.0」を選択し、[完了] ボタンをクリックします。



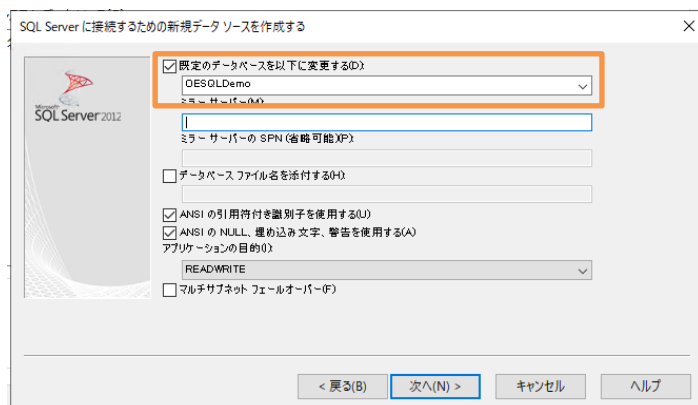
- ③ [Name] フィールドは、「OESQLEDemo」と入力し、[Server] フィールドは接続する SQL Server 名を指定し、[次へ(N)] ボタンをクリックします。もし、不明な場合は、自社の DBMS 管理者に確認してください。



- ④ 設定されている認証方式を選択し [次へ(N)] ボタンをクリックします。不明な場合は、自社の DBMS 管理者に確認してください。（この例では混合モード認証で設定しています）



- ⑤ [既定のデータベースを以下に変更する(D):] にチェックを入れて、作成したデータベース「OESQLDemo」を指定し、[次へ(N)] ボタンをクリックします。



- ⑥ [完了] ボタンをクリックして設定を保存します。[データソースのテスト(T)] をクリックして正常に接続できるか確認してください。

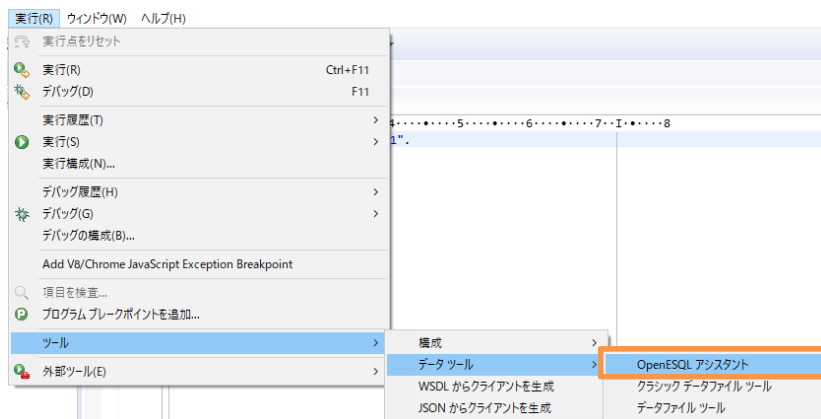


- ⑦ [OK] ボタンをクリックすると ODBC データソースアドミニストレーター (32 ビット) 画面に戻ります。

3.5. クエリーの組み立てとテスト

1) OpenESQL アシスタントの起動

- ① Eclipse メニューより、[実行(R)] > [ツール] > [データツール] > [OpenESQL アシスタント] を選択し、「OpenESQL アシスタント」を起動します。

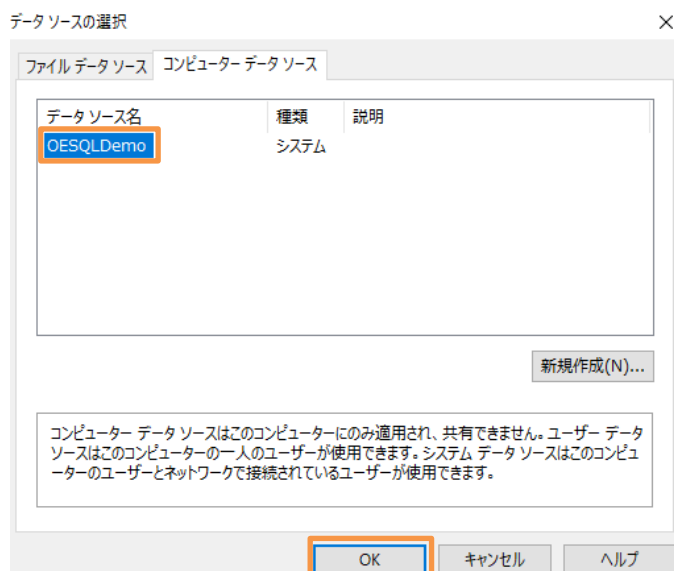


2) テーブルの選択

- ① OpenESQL アシスタントの画面より、左側のツリービュー内に見えている「OESQLDemo」にチェックを入れます。



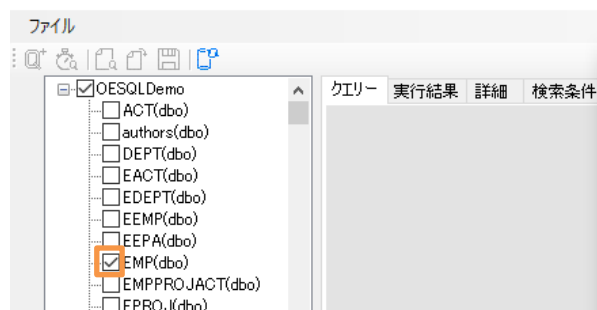
- ② データソースの選択画面が表示されるので 3.4 で設定した ODBC データソースを指定します。



下記ダイアログが表示された場合は、データベースのログイン情報を入力した上で、[OK] をクリックします。

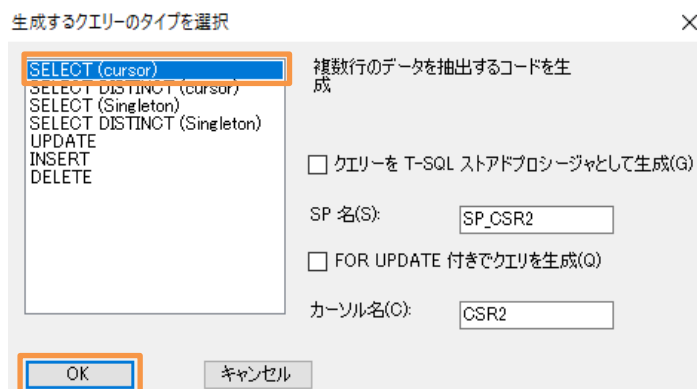


- ③ テーブルの一覧が表示されるので「EMP」にチェックを入れます。



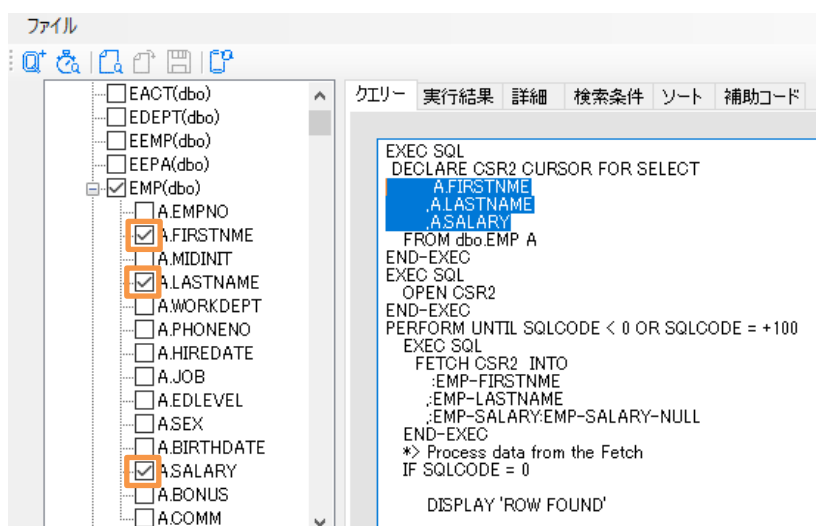
- ④ [生成するクエリーのタイプを選択] ダイアログが表示されます。

- ⑤ 今回は、「SELECT (cursor)」を選択し、[OK] ボタンをクリックします。[クエリー] タブ内にサンプルとなるコードが生成されます。



3) カラムの選択

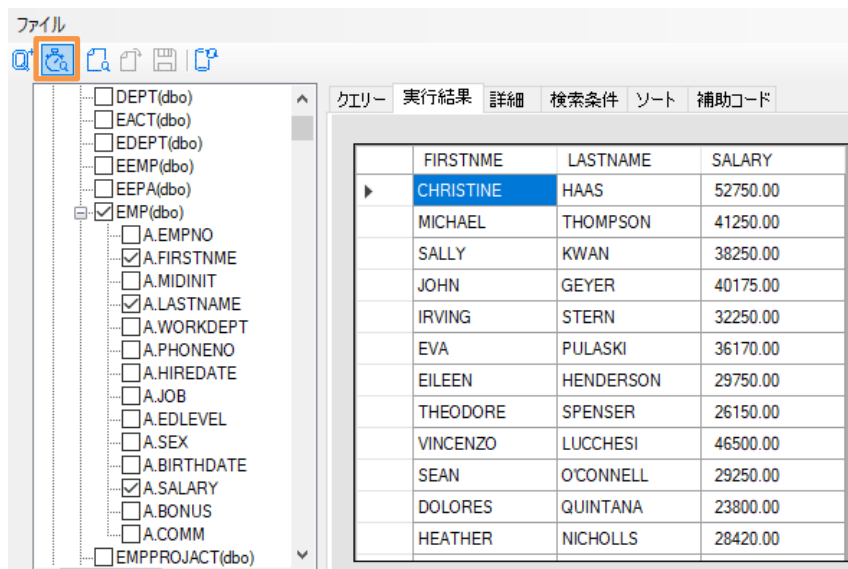
- ① EMP テーブルツリーから「A.FIRSTNAME」、「A.LASTNAME」、「A.SALARY」にチェックを入れます。[クエリー] タブ内のサンプルコードに選択したフィールドが付加されます。



カラム名に“A.”が付与されていますが、これは上記画像の6行目にて EMP テーブルのエイリアスを“A”としているためです。複数テーブルを JOIN する場合、エイリアスが“B”, “C”, ... と設定することができ、異なるテーブルに同名の項目が存在した場合でも、エイリアスを利用することで正しく取得先を識別することができます。

4) クエリーのテスト

- ① [クエリーの実行] アイコン  をクリックすると、[実行結果] タブ内に実行結果が表示されます。



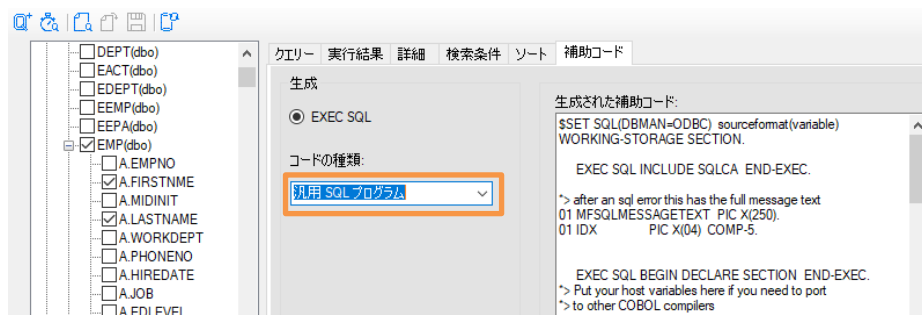
3.6. SQL プログラムの生成と埋め込み

1) SQL プログラムの生成

- ① OpenESQL アシスタントにて [補助コード] タブをクリックします。

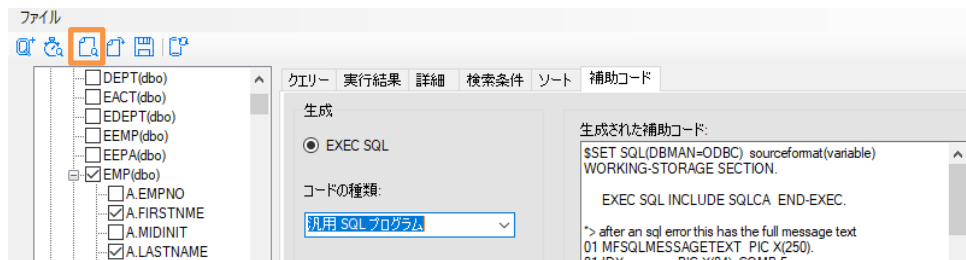


- ② [コードの種類] にて「汎用 SQL プログラム」を選択します。生成された補助コード欄に生成されたソースコードが表示されます。

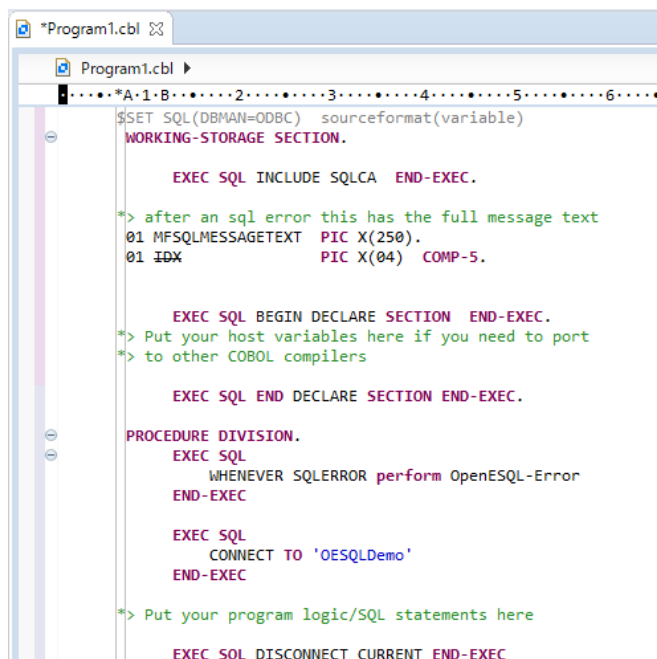


2) OpenESQL アシスタントが生成するプログラムコードを利用したデータベース接続プログラムの作成

- ① 「Program1.cbl」をエディタで開き、現在のコードをすべて削除します。
- ② OpenESQL アシスタントにて [補助コード] タブをクリックします。[一時ファイルにクエリーを追加] アイコン  をクリックし、クリップボードにソースコードをコピーします。



- ③ 「Program1.cbl」をエディタ上で開き、Ctrl+V で生成された補助コードを貼りつけます。



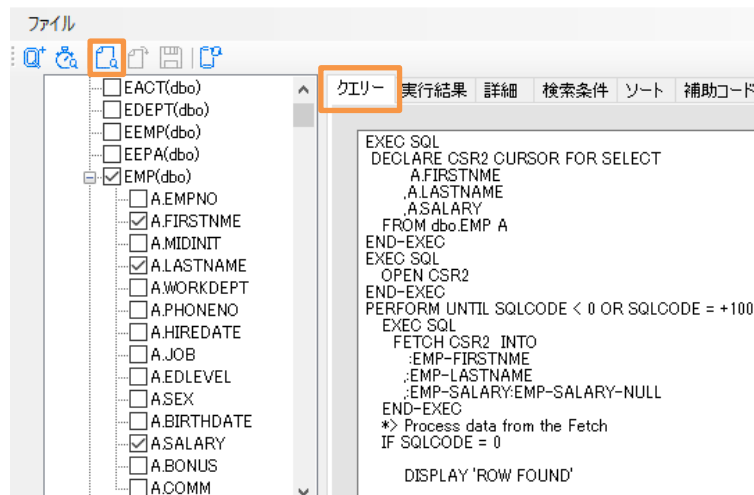
- ④ “CONNECT TO ‘OESQLDemo’”の末尾に “USER ‘データベースユーザーID.パスワード’ ” 句を記述します。以下の例では、データベースユーザーIDに “sa”, パスワードに “HogeHoge”を指定しています。

```
EXEC SQL
CONNECT TO 'OESQLDemo' USER 'sa.HogeHoge'
END-EXEC
```

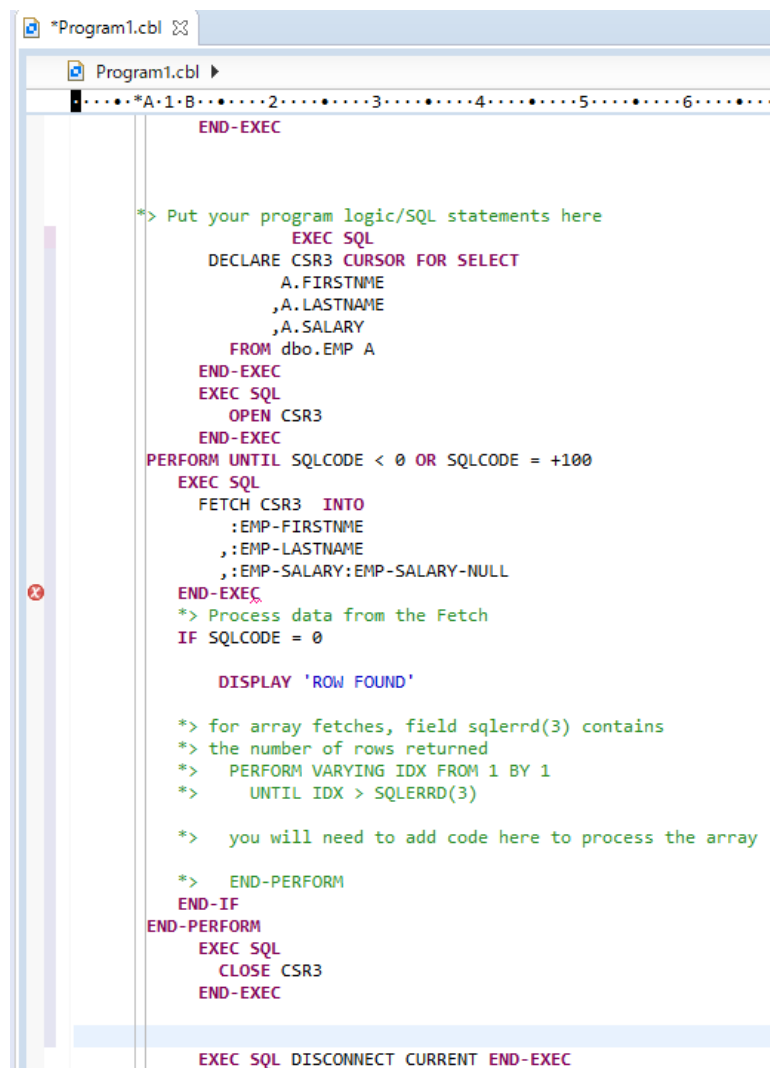
補足)

上記例は、SQL Server 認証が有効な場合の記述例です。環境に合わせ、**データベースユーザーID** にはデータベースにアクセス可能なログインユーザー、**パスワード**にはデータベースユーザーID のパスワードを設定してください。なお、認証方式や情報については、自社のデータベース管理者にお問い合わせください。

- ⑤ OpenESQL アシスタントにて[クエリー] タブをクリックします。[一時ファイルにクエリーを追加] アイコン  をクリックし、クリップボードにソースコードをコピーします。



- ⑥ Eclipse の COBOL エディタにて下記のコメント部分の下部に CTRL+V でコードを張り付けます。
 “*> Put your program logic/SQL statements here”



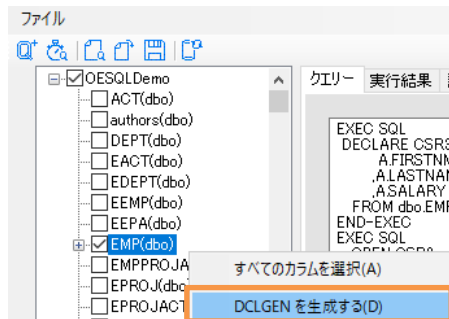
エラーが検出されますが、次の手順で対応しますので、ここでは無視してください。

- ⑦ Eclipse メニューより [ファイル(F)] > [保管(S)] を選択し、プログラムを保存します。

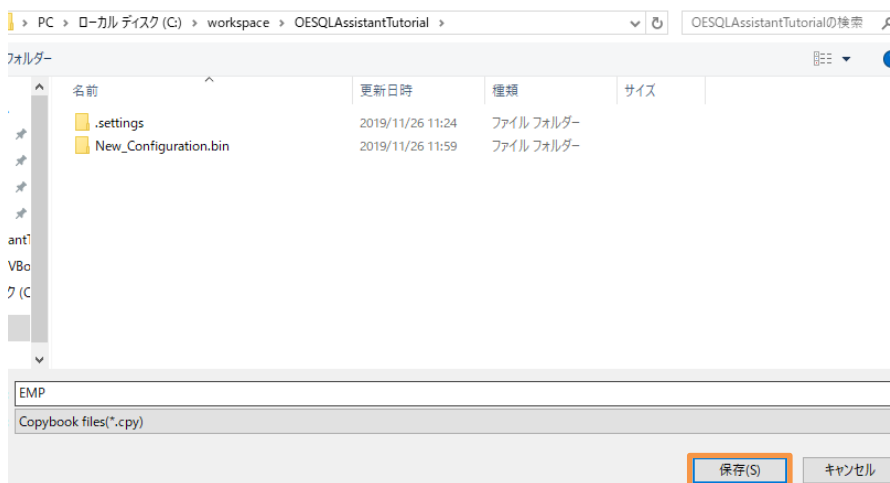
3.7. 追加のコードを記述

1) コピーブックの生成

- ① OpenESQL アシスタントの「EMP」テーブルツリー内にて右クリックし、コンテキストメニューから「DCLGEN を生成する(D)」を選択します。



- ② [Generate Host Variable Declarations] ダイアログが表示されるので OESQLAssistantTutorial プロジェクトフォルダ配下まで移動した上で、「EMP.cpy」という名称でコピーブックを保存します。



2) INCLUDE 命令の追加

- ① OpenESQL アシスタントにて「補助コード」タブをクリックします。「EXEC SQL INCLUDE」文が生成されているので「一時ファイルにクエリーを追加」アイコン  をクリックし、クリップボードにソースコードをコピーします。



- ② Eclipse の COBOL エディタにて EXEC SQL INCLUDE SQLCA に下に CTRL+V で貼り付けます。

```
*Program1.cbl
Program1.cbl
.....*A.1.B.....2.....3.....4.....5.....6.
$SET SQL(DBMAN=ODBC) sourceformat(variable)
WORKING-STORAGE SECTION.

EXEC SQL INCLUDE SQLCA END-EXEC.
EXEC SQL INCLUDE EMP END-EXEC.

*> after an sql error this has the full message text
01 MFSQLEMSGTEXT PIC X(250).
01 IDX          PIC X(04) COMP-5.

EXEC SQL BEGIN DECLARE SECTION END-EXEC.
*> Put your host variables here if you need to port
*> to other COBOL compilers

EXEC SQL END DECLARE SECTION END-EXEC.

PROCEDURE DIVISION.
```

- ③ Eclipse メニューより [ファイル(F)] > [保管(S)] を選択し、プログラムを保存します。

3) 表示フォーマットの追加

- ① Eclipse の COBOL エディタにて「EXEC SQL INCLUDE EMP END-EXEC」の下に下記の命令を追加します。

```
01 EDIT-PAY      PIC Z,ZZ99.99.
```

```
*Program1.cbl
Program1.cbl
.....*A.1.B.....2.....3.....4.....5.....6.
$SET SQL(DBMAN=ODBC) sourceformat(variable)
WORKING-STORAGE SECTION.

EXEC SQL INCLUDE SQLCA END-EXEC.
EXEC SQL INCLUDE EMP END-EXEC.
01 EDIT-PAY      PIC Z,ZZ99.99.

*> after an sql error this has the full message text
01 MFSQLEMSGTEXT PIC X(250).
01 IDX          PIC X(04) COMP-5.
```

- ② コードを下の方にスクロールして「DISPLAY 'ROW FOUND」命令を以下の2行で置き換えます。

```
MOVE EMP-SALARY TO EDIT-PAY
DISPLAY 'NAME: ' EMP-LASTNAME ', ' EMP-FIRSTNME '   SALARY: ' EDIT-PAY UPON
CONSOLE
```

修正前)

```
PERFORM UNTIL SQLCODE < 0 OR SQLCODE = +100
EXEC SQL
  FETCH CSR3 INTO
    :EMP-FIRSTNME
    ,:EMP-LASTNAME
    ,:EMP-SALARY:EMP-SALARY-NULL
END-EXEC
*> Process data from the Fetch
IF SQLCODE = 0

  DISPLAY 'ROW FOUND'

  *> for array fetches, field sqlerrd(3) contains
  *> the number of rows returned
  *> PERFORM VARYING IDX FROM 1 BY 1
  *>   UNTIL IDX > SQLERRD(3)

  *> you will need to add code here to process the array

  *> END-PERFORM
END-IF
```

修正後)

```
PERFORM UNTIL SQLCODE < 0 OR SQLCODE = +100
EXEC SQL
  FETCH CSR3 INTO
    :EMP-FIRSTNME
    ,:EMP-LASTNAME
    ,:EMP-SALARY:EMP-SALARY-NULL
END-EXEC
*> Process data from the Fetch
IF SQLCODE = 0

  MOVE EMP-SALARY TO EDIT-PAY
  DISPLAY 'NAME: ' EMP-LASTNAME ' , ' EMP-FIRSTNME ' SALARY: ' EDIT-PAY UPON CONSOLE

  *> for array fetches, field sqlerrd(3) contains
  *> the number of rows returned
  *> PERFORM VARYING IDX FROM 1 BY 1
  *> UNTIL IDX > SQLERRD(3)

  *> you will need to add code here to process the array

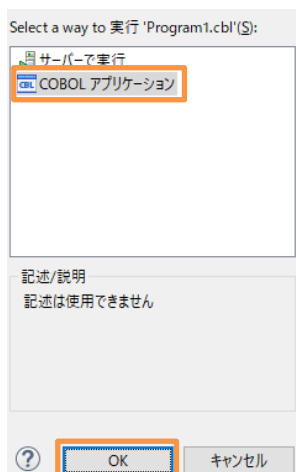
  *> END-PERFORM
END-IF
```

- ⑧ Eclipse メニューより [ファイル(F)] > [保管(S)] を選択し、プログラムを保存します。自動的にリビルドされます。

3.8. プログラムを実行し動作を確認

1) プログラムの実行

- ① Eclipse メニューより [実行(R)] > [実行(R)] をクリックします。
- ② 「COBOL アプリケーション」を選択し、[OK] をクリックします。



- ③ Eclipse の実行ボタンを押して、プログラムが実行され EMP テーブルの内容が出力されることを確認します。

```
C:\Windows\SYSTEM32\cmd.exe
NAME: PLANKA      , ELIZABETH      SALARY: 2,2250.00
NAME: YOSHIMURA , MASATOSHI     SALARY: 2,4680.00
NAME: SCOUTTEN   , MARILYN       SALARY: 2,1340.00
NAME: WALKER     , JAMES         SALARY: 2,0450.00
NAME: BROWN     , DAVID         SALARY: 2,7740.00
NAME: JONES     , WILLIAM       SALARY: 1,8270.00
NAME: LUTZ      , JENNIFER      SALARY: 2,9840.00
NAME: JEFFERSON , JAMES         SALARY: 2,2180.00
NAME: MARINO    , SALVATORE     SALARY: 2,8760.00
NAME: SMITH     , DANIEL        SALARY: 1,9180.00
NAME: JOHNSON   , SYBIL         SALARY: 1,7250.00
NAME: PEREZ     , MARIA         SALARY: 2,7380.00
NAME: SCHNEIDER , ETHEL         SALARY: 2,6250.00
NAME: PARKER    , JOHN          SALARY: 1,5340.00
NAME: SMITH     , PHILIP        SALARY: 1,7750.00
NAME: SETRIGHT  , MAUDE         SALARY: 1,5900.00
NAME: MEHTA     , RAMLAL        SALARY: 1,9950.00
NAME: LEE       , WING          SALARY: 2,5370.00
NAME: GOUNOT    , JASON         SALARY: 2,3840.00
NAME: HEMMINGER , DIAN          SALARY: 4,6500.00
NAME: ORLANDO   , GREG          SALARY: 2,9250.00
NAME: NATZ      , KIM           SALARY: 2,8420.00
NAME: YAMAMOTO  , KIYOSHI       SALARY: 2,4680.00
NAME: JOHN      , REBA          SALARY: 2,9840.00
NAME: MONTEVERDE , ROBERT        SALARY: 2,8760.00
NAME: SCHWARTZ  , EILEEN        SALARY: 2,6250.00
NAME: SPRINGER  , MICHELLE      SALARY: 1,5900.00
NAME: WONG      , HELENA        SALARY: 2,5370.00
NAME: ALONZO    , ROY           SALARY: 2,3840.00
続行するには何かキーを押してください . . .
```

WHAT'S NEXT

- 本チュートリアルで学習した技術の詳細については製品マニュアルをご参照ください。

免責事項

ここで紹介したソースコードは、機能説明のためのサンプルであり、製品の一部ではございません。ソースコードが実際に動作するか、御社業務に適合するかなどに関しまして、一切の保証はございません。ソースコード、説明、その他すべてについて、無謬性は保障されません。ここで紹介するソースコードの一部、もしくは全部について、弊社に断りなく、御社の内部に組み込み、そのままご利用頂いても構いません。本ソースコードの一部もしくは全部を二次的著作物に対して引用する場合、著作権法に基づき、適切な扱いを行ってください。