

## 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

#### Visual Studio 編

##### 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繁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、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、製品搭載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

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

DBMS 製品	SQL Server 2022
---------	-----------------

- 開発クライアント ソフトウェア

OS	Windows 11
COBOL 製品	Visual COBOL 11.0 for Visual Studio 2022
DBMS 製品	ODBC Driver 17 for SQL Server

- チュートリアル用サンプルプログラム

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、任意のフォルダー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
- 3      チュートリアル手順
  - 3.1    データベースの準備
  - 3.2    Visual Studio プロジェクトの作成と設定
  - 3.3    ODBC データソースの設定
  - 3.4    クエリーの作成とテスト
  - 3.5    SQL 埋め込みプログラムの作成
  - 3.6    SQL 埋め込みプログラムの実行

### 3 チュートリアル手順

#### 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

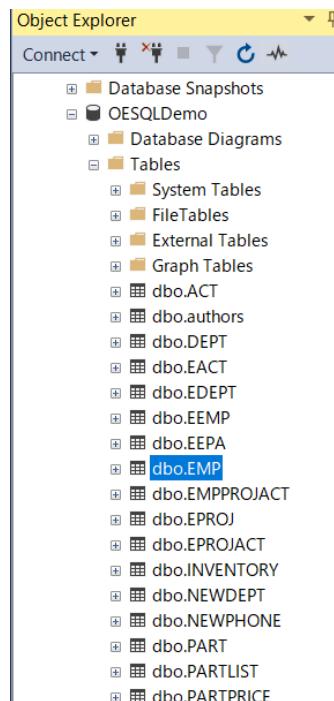
Microsoft SQL Server Management Studio を起動し、管理者でログインを行います。※Microsoft SQL Server Management Studio は別途ダウンロードが必要です。

##### 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] 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 Visual Studio プロジェクトの作成と設定

#### 1) Visual Studio の起動とプロジェクトの作成

- ① Visual Studio を起動します。
- ② 「新しいプロジェクトの作成(N)」を選択します。



- ③ 以下のフィルタを設定し、[コンソールアプリケーション] を選択し、[次へ(N)] をクリックします。

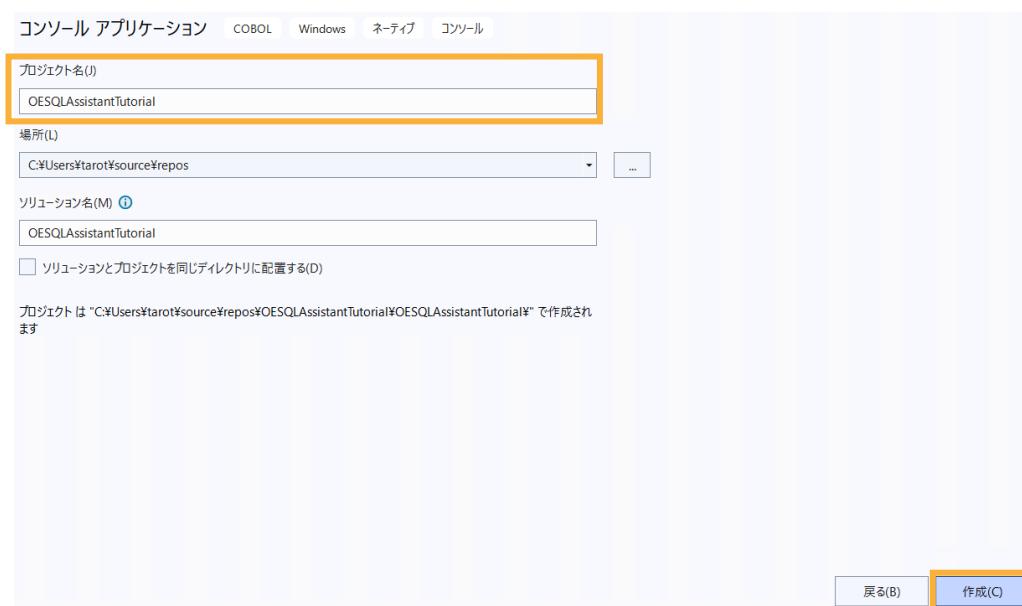
全ての言語 : COBOL

全てのプラットフォーム : Windows

全てのプロジェクトの種類 : コンソール

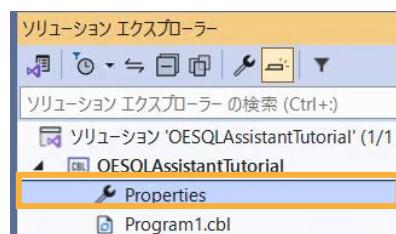


- ④ プロジェクト名に “OESQLAssistantTutorial” を入力し、[作成(C)] をクリックします。



## 2) OpenESQL のプロジェクト設定

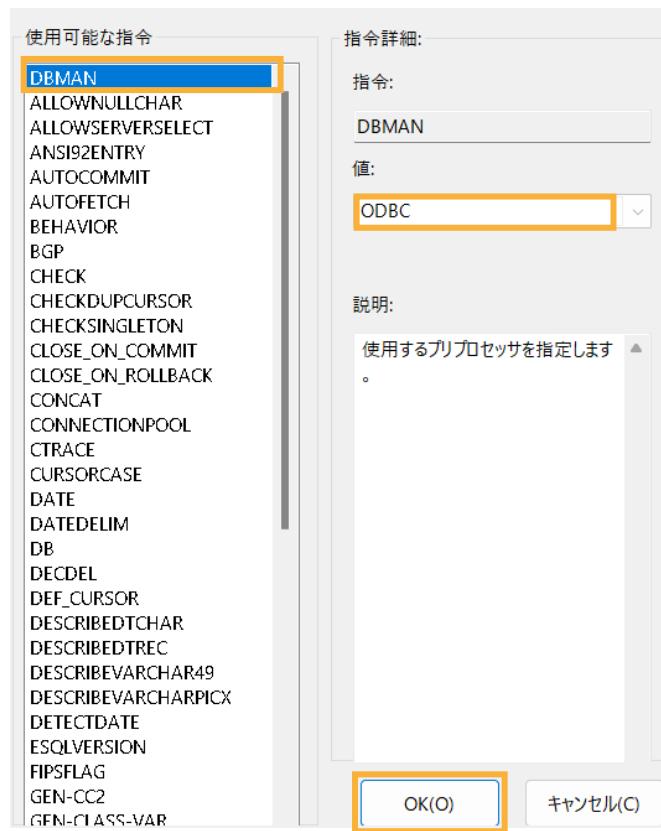
- ① OESQLAssistantTutorial プロジェクト配下の「Properties」を選択し、ダブルクリックします。



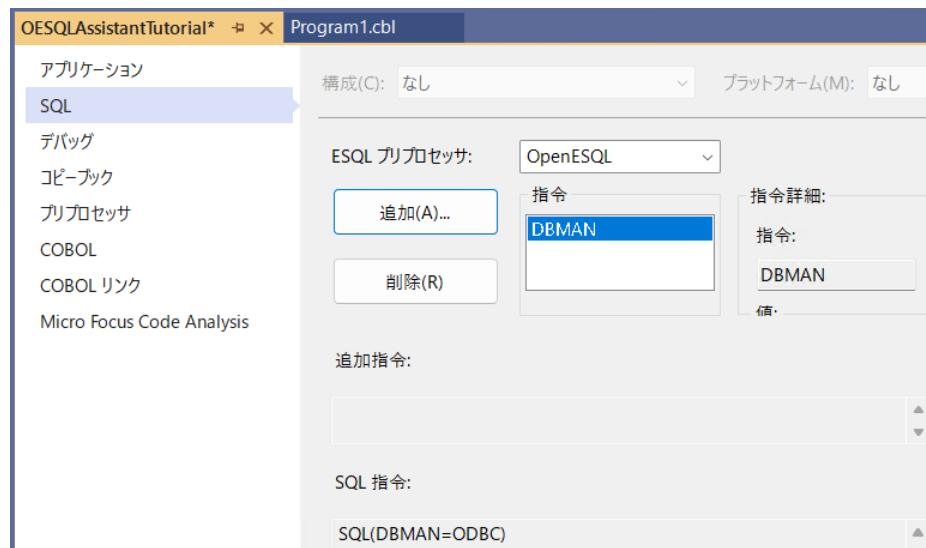
- ② 次に [SQL] タブを開き、[ESQL プリプロセッサ] に “OpenESQL” を選択した上で、[追加(A)] をクリックします。



- ③ 「使用可能な指令」欄より “DBMAN” を選択し、[OK] をクリックします。



DBMAN が追加されます。



④ ②、③ の手順を再度実施し、“BEHAVIOR”を追加します。

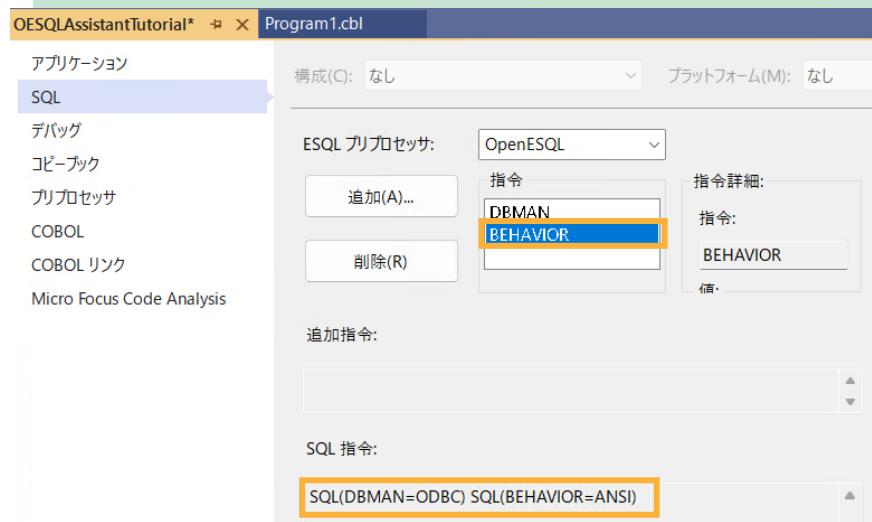
追加を行うと、DBMAN に続き、BEHAVIOR が指令欄に追加されます。



- ⑤ BEHAVIOR の値には “ANSI” を選択して、変更を保存します。

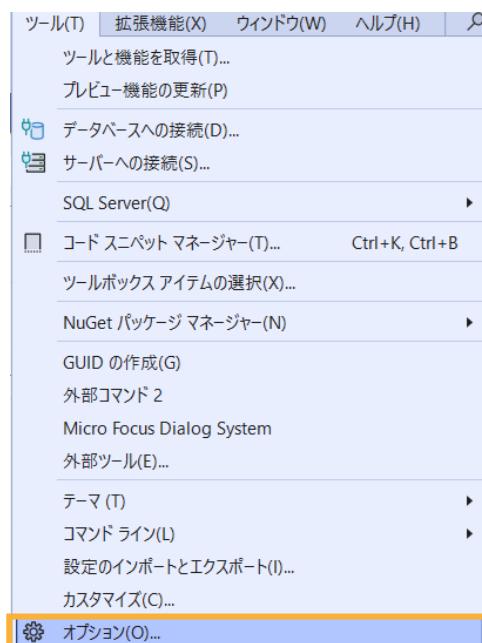
**補足)**

以下のように、値の設定箇所が見えにくくなっていることがあります。その場合は、[指令詳細] セクションの [指令] BEHAVIOR の編集不可フィールドをクリックの上、タブキーを押すことで、値が変更できる箇所にフォーカスが移動します。その後、↑、↓カーソルキーを用いて任意の値を選択できます。現在の選択値は、SQL 指令欄にて確認できます。



3) OpenESQL の設定変更

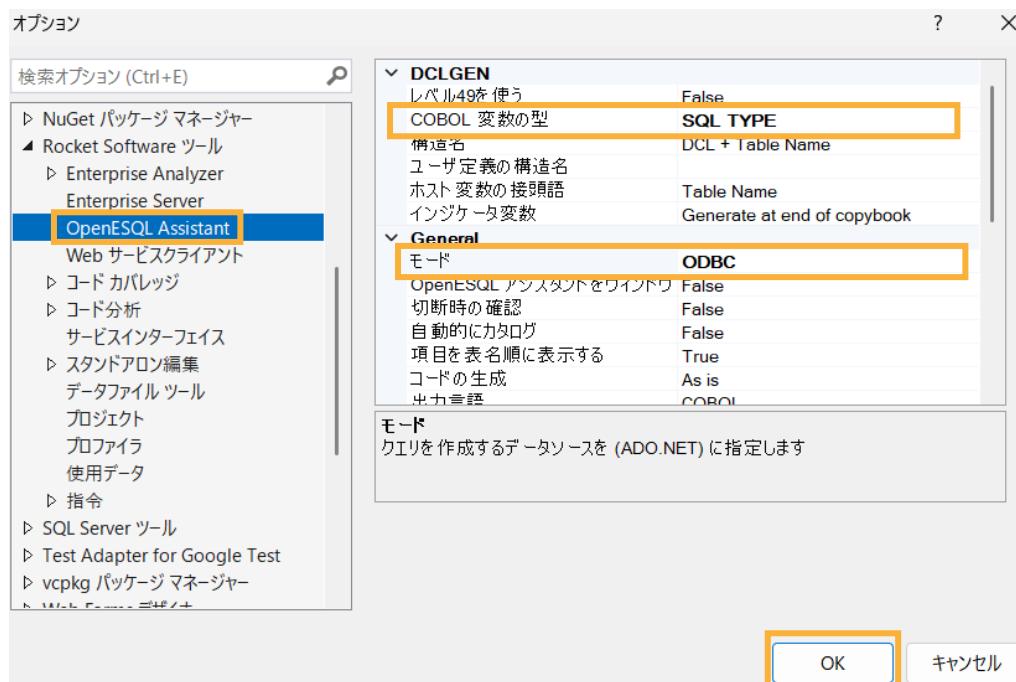
- ① Visual Studio のメニューより、[ツール(T)] > [オプション(O)] を選択します。



- ② ダイアログ左側より [Rocket Software ツール] > [OpenESQL Assistant] を選択し、以下の設定を行ったう  
えで、[OK] をクリックします。

COBOL 変数の型 : SQL TYPE

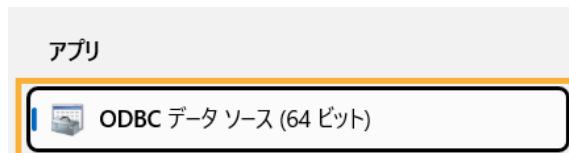
モード : ODBC



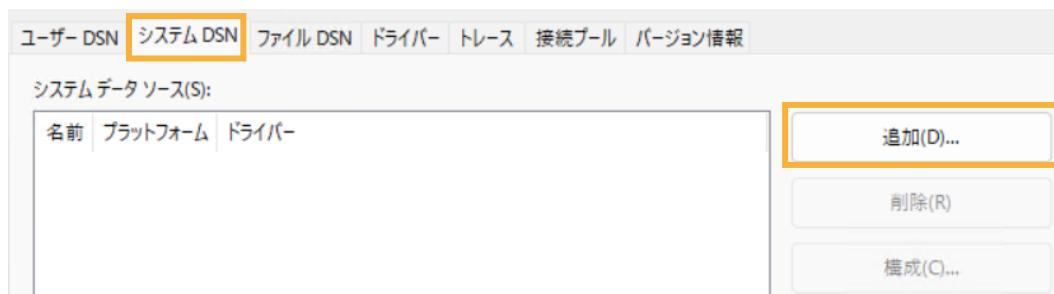
### 3.3 ODBC データソースの設定

1) データソースの設定と接続確認

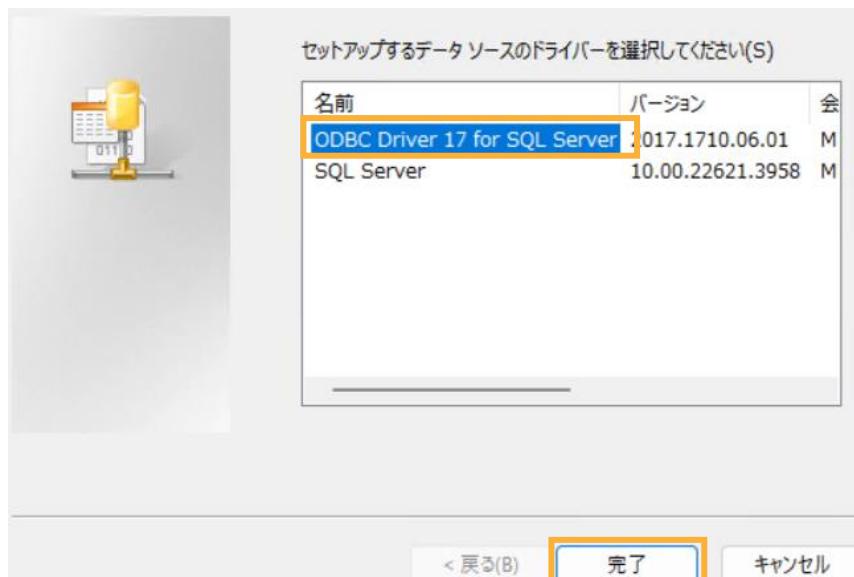
- ① スタートメニュー横の検索にて “ODBC” とタイプし、検索候補から「ODBC データソース（64 ビット）」を起動します。



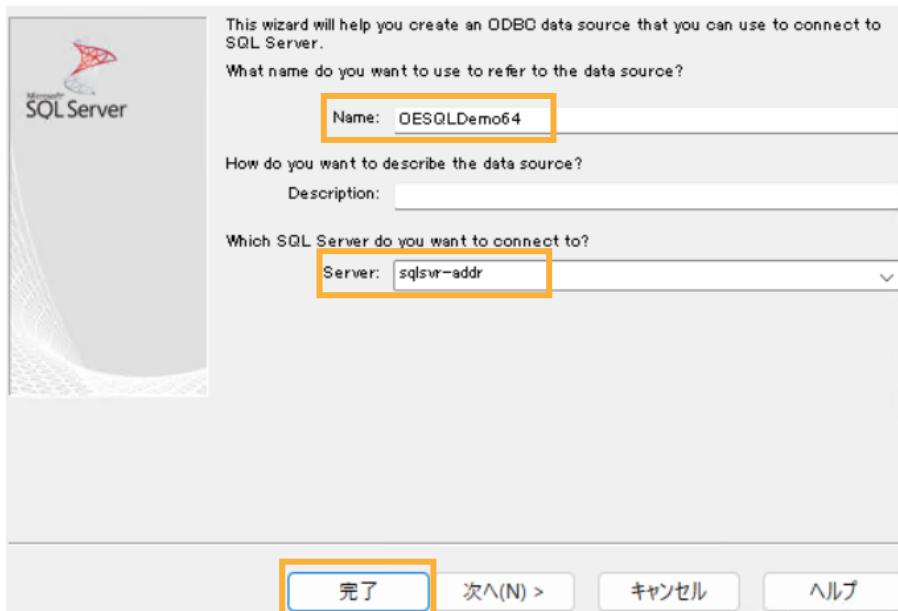
- ② ODBC データソースアドミニストレーター（64 ビット）画面より [システム DSN] タブをクリックし、[追加(D)…] をクリックします。



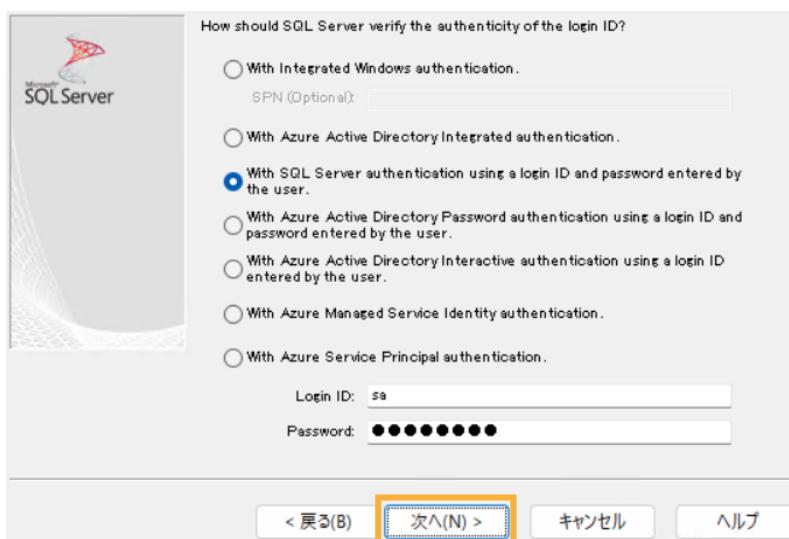
- ③ [データソースの新規作成] ダイアログが表示されるので「ODBC Driver 17 for SQL Server」を選択し、[完了] をクリックします。



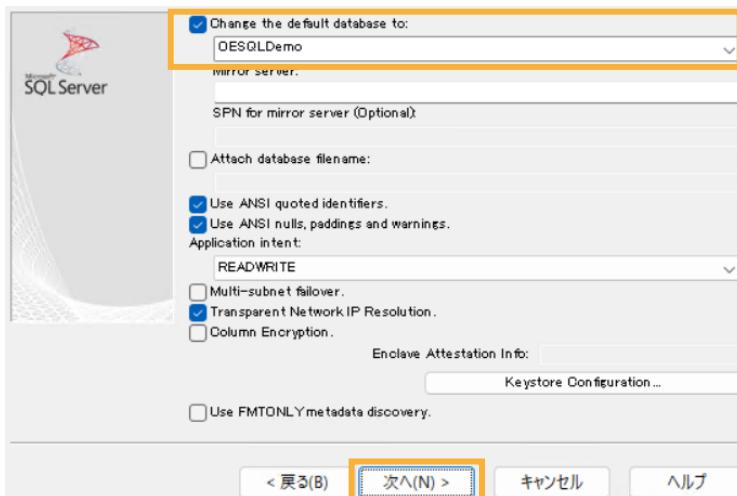
- ④ [Name] フィールドは、“OESQLDemo64” と入力し、[Server] フィールドは接続する SQL Server 名を指定し、[次へ(N)] をクリックします。もし、不明な場合は、自社の DBMS 管理者に確認してください。



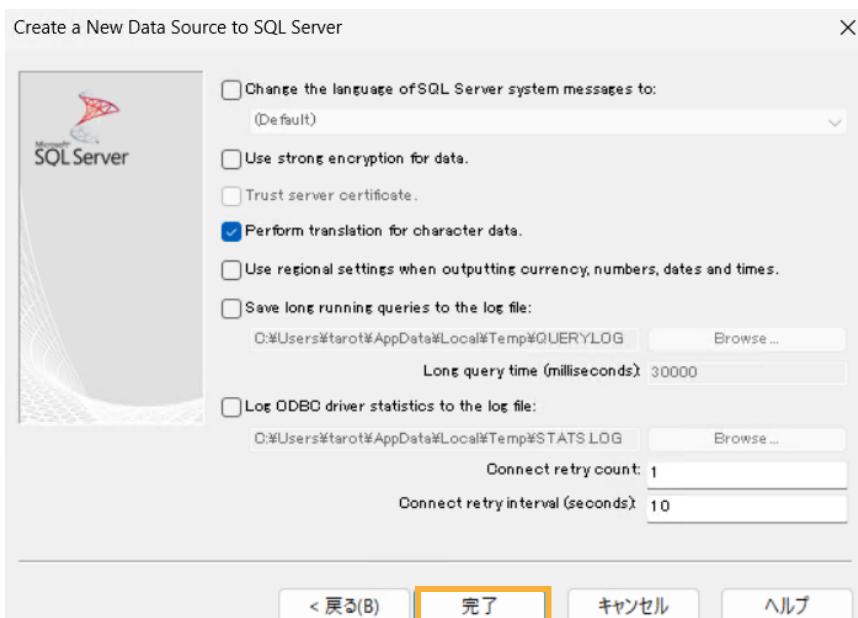
- ⑤ 設定されている認証方式を選択し [次へ(N)] をクリックします。不明な場合は、自社の DBMS 管理者に確認してください。（この例では SQL Server 認証で設定しています）



- ⑥ [Change the default database to] 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)] をクリックします。



⑦ [完了] をクリックして設定を保存します。



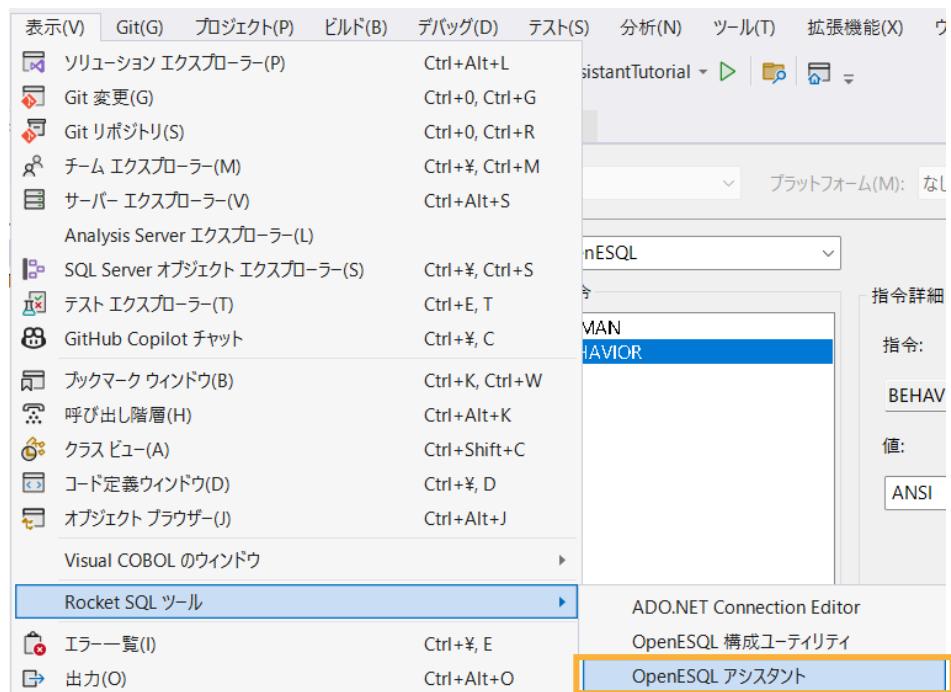
[Test Data Source] をクリックして、正常に接続できるかを確認してください。



[OK] 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（64 ビット）画面に戻ります。

### 3.4 クエリーの作成とテスト

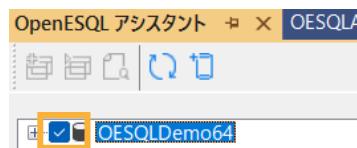
- 1) IDE のメニューより [表示(V)] > [Rocket SQL ツール] > [OpenESQL アシスタント] を選択します。



OpenESQL アシスタントが開き、「OESQLDemo64」が一覧に表示されていることを確認してください。



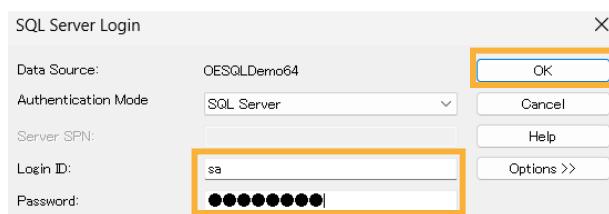
- 2) OpenESQL アシスタント上の「OESQLDemo64」にチェックを行います。



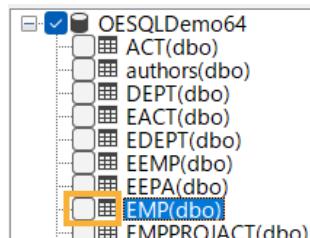
チェックした際に、以下のダイアログが表示された場合は、「OESQLDemo64」を選択し、[OK] をクリックします。



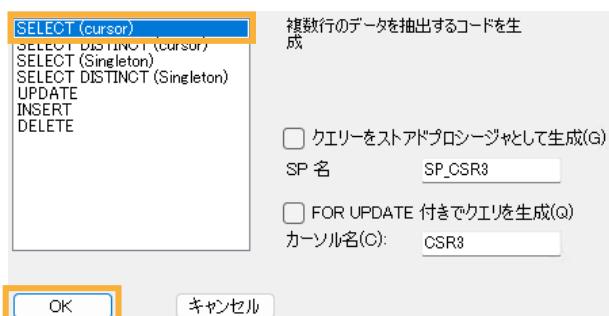
SQL Server 認証の場合、認証ダイアログが表示されますので、再度認証情報を入力し、[OK] をクリックします。



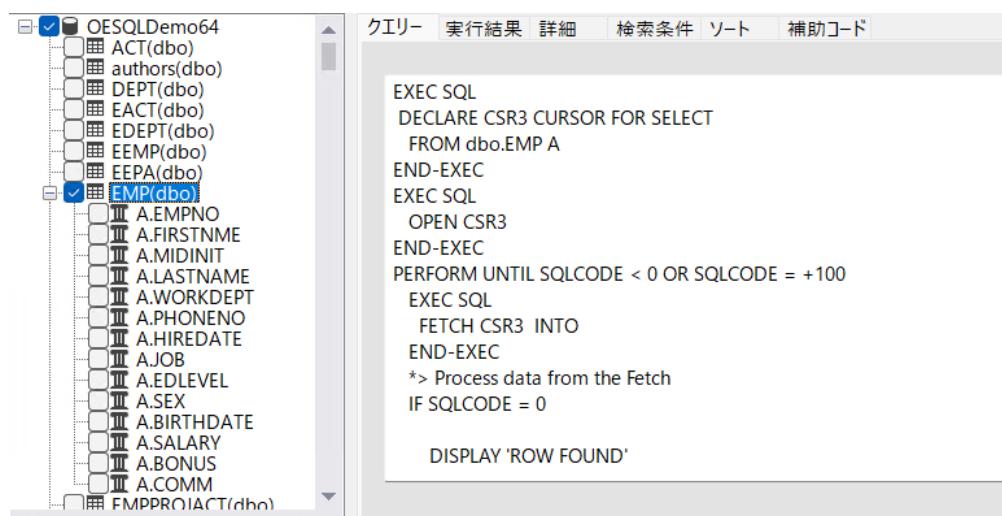
3) 「EMP(dbo)」にチェックを行います。



表示されたダイアログから作成するクエリーを選択できます。今回は、「SELECT (cursor)」を選択し、[OK] をクリックします。



EMP テーブルの項目一覧が表示され、右側にクエリーのテンプレートが表示されます。



```

EXEC SQL
DECLARE CSR3 CURSOR FOR SELECT
  FROM dbo.EMP A
END-EXEC
EXEC SQL
  OPEN CSR3
END-EXEC
PERFORM UNTIL SQLCODE < 0 OR SQLCODE = +100
  EXEC SQL
    FETCH CSR3 INTO
  END-EXEC
  *-> Process data from the Fetch
  IF SQLCODE = 0

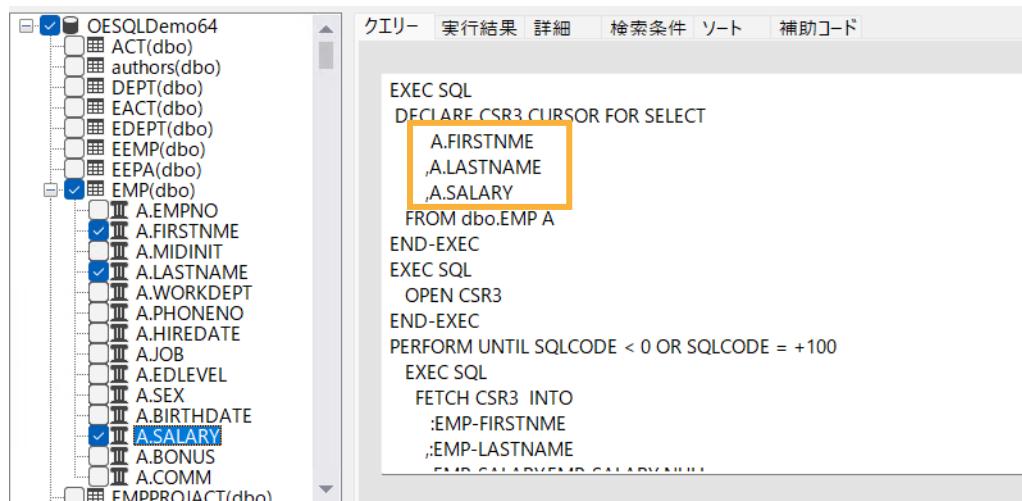
  DISPLAY 'ROW FOUND'

```

テーブル項目名が“A.” から始まっていますが、これはクエリーの 3 行目で 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) 「A.FIRSTNME」、「A.LASTNAME」、「A.SALARY」をチェックします。

チェックを行った項目が、SELECT 項目に追加されます。



```

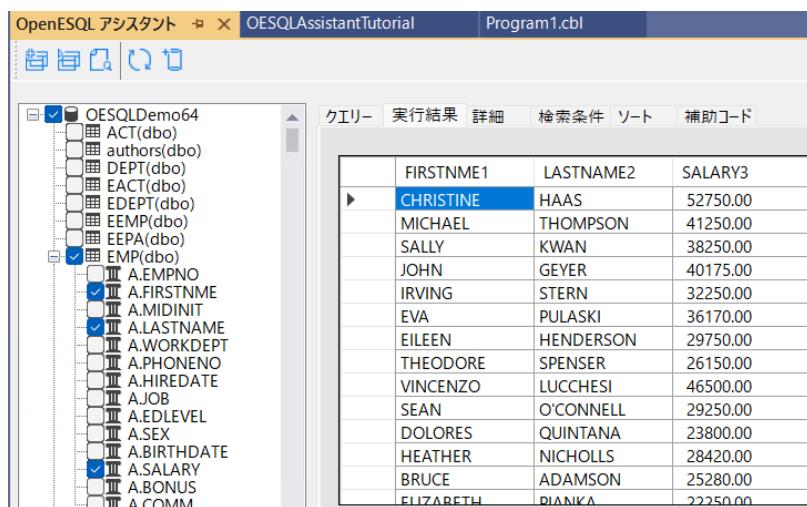
EXEC SQL
DECLARE CSR3 CURSOR FOR SELECT
  A.FIRSTNME
  ,A.LASTNAME
  ,A.SALARY
  FROM dbo.EMP A
END-EXEC
EXEC SQL
  OPEN CSR3
END-EXEC
PERFORM UNTIL SQLCODE < 0 OR SQLCODE = +100
  EXEC SQL
    FETCH CSR3 INTO
      :EMP-FIRSTNME
      ,:EMP-LASTNAME
      ,:EMP-SALARY
  END-EXEC

```

- 5) 「クエリーを実行する」アイコンをクリックして、プログラムを実行します。



「実行結果」タブに切り替わり、検索結果が表形式で表示されます。

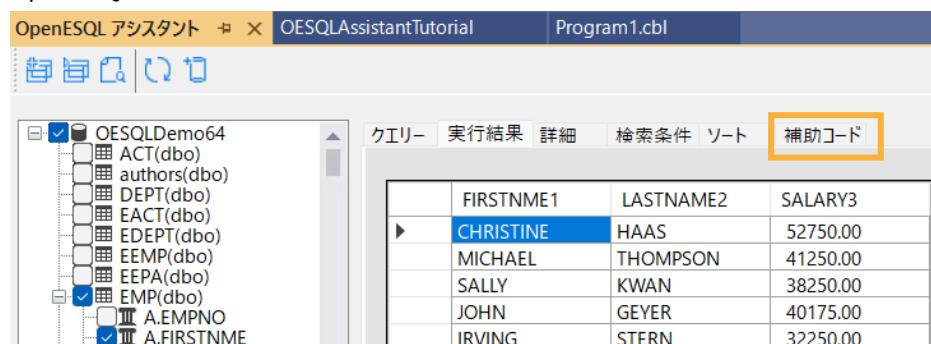


The screenshot shows the OpenESQL Assistant interface. On the left, there is a tree view of database objects under 'OESQLDemo64'. Several objects are selected, including tables like ACT(dbo), authors(dbo), DEPT(dbo), EACT(dbo), EDEPT(dbo), EEMP(dbo), EEPA(dbo), EMP(dbo), and various columns from these tables such as A.EMPNO, A.FIRSTNAME, A.MINIT, A.LASTNAME, A.WORKDEPT, A.PHONENO, A.HIREDATE, A.JOB, A.EDLEVEL, A.SEX, A.BIRTHDATE, A.SALARY, A.BONUS, and A.COMM. On the right, a query results grid displays data from the selected columns. The grid has columns FIRSTNAME1, LASTNAME2, and SALARY3. The data includes rows for CHRISTINE HAAS (52750.00), MICHAEL THOMPSON (41250.00), SALLY KWAN (38250.00), JOHN GEYER (40175.00), IRVING STERN (32250.00), EVA PULASKI (36170.00), EILEEN HENDERSON (29750.00), THEODORE SPENSER (26150.00), VINCENZO LUCCHESI (46500.00), SEAN O'CONNELL (29250.00), DOLORES QUINTANA (23800.00), HEATHER NICHOLLS (28420.00), BRUCE ADAMSON (25280.00), and ELIZABETH DIANKA (22250.00).

### 3.5 SQL 埋め込みプログラムの作成

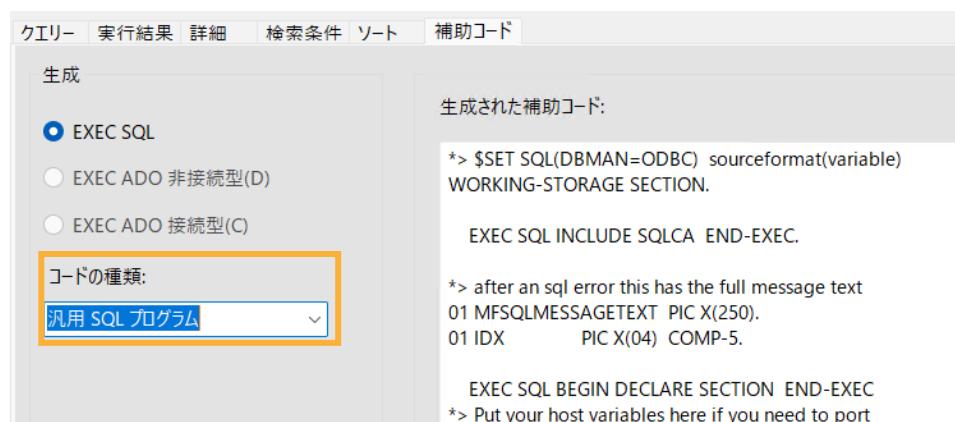
#### 1) OpenESQL アシスタントを利用した埋め込みプログラムの作成

- ① OpenESQL アシスタント上の「補助コード」タブを選択します。



The screenshot shows the OpenESQL Assistant interface with the 'Assistants' tab selected. The 'Auxiliary Code' tab is highlighted with a yellow box. On the left, the same tree view of database objects is shown. On the right, the query results grid is visible. The 'Auxiliary Code' tab panel contains a section for generating code. It has a radio button for 'EXEC SQL' which is selected, and two other options: 'EXEC ADO 非接続型(D)' and 'EXEC ADO 接続型(C)'. Below this, a dropdown menu labeled 'コードの種類:' is open, showing '汎用 SQL プログラム' which is also highlighted with a yellow box.

- ② 「コードの種類」に“汎用 SQL プログラム”を選択します。



The screenshot shows the 'Auxiliary Code' tab panel. The '生成' (Generate) section has 'EXEC SQL' selected. The 'コードの種類:' dropdown is set to '汎用 SQL プログラム'. To the right, the '生成された補助コード:' (Generated Auxiliary Code) section displays the generated SQL code. The code includes sections for setting the database manager to ODBC, defining working storage, executing SQL, and handling errors. It also includes declarations for host variables.

```

生成された補助コード:
*> $SET SQL(DBMAN=ODBC) sourceformat(variable)
WORKING-STORAGE SECTION.

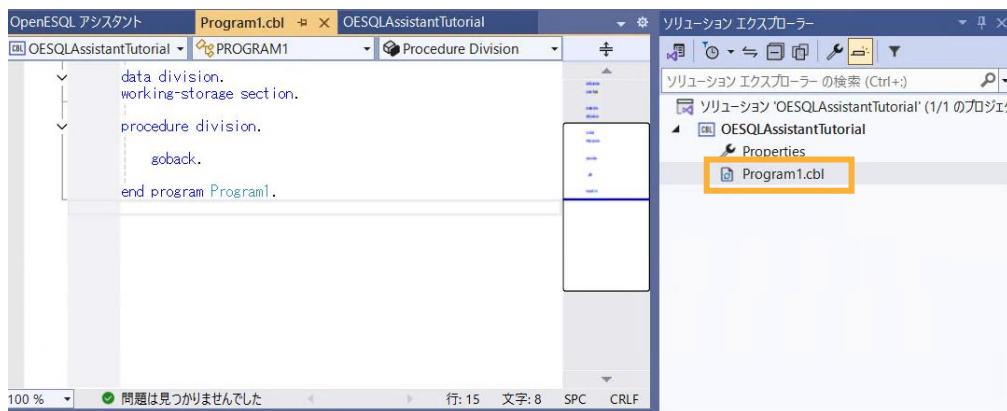
EXEC SQL INCLUDE SQLCA END-EXEC.

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

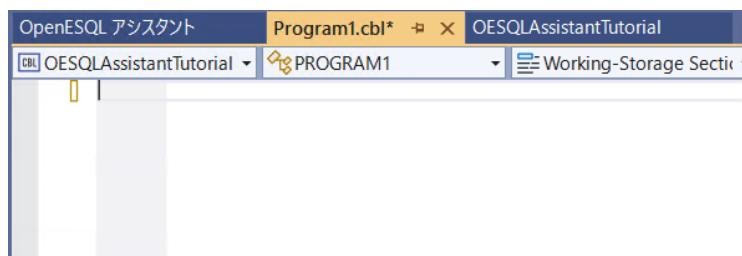
EXEC SQL BEGIN DECLARE SECTION END-EXEC
*> Put your host variables here if you need to port

```

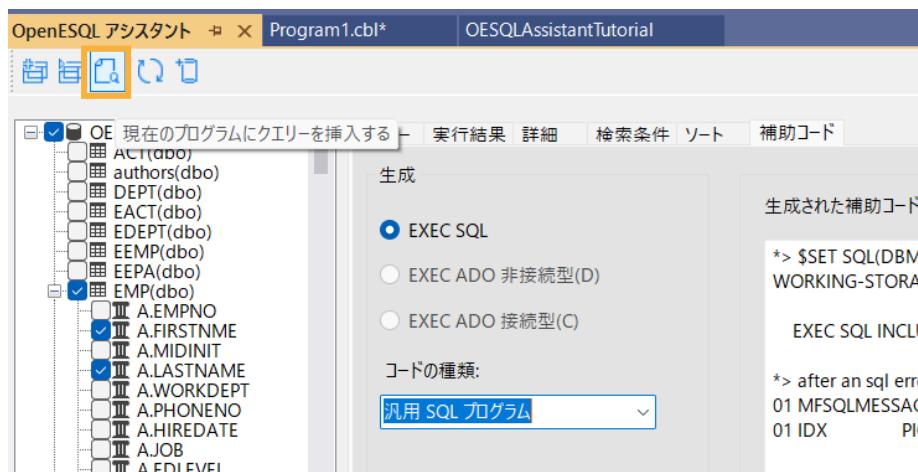
- ③ ソリューションエクスプローラーより、OESQLAssistantTutorial プロジェクト内の「Program1.cbl」をダブルクリックして開きます。



Program1.cbl の既存コードをすべて削除します。



- ④ OpenESQL アシスタントに戻り、「現在のプログラムにクエリーを挿入する」アイコンをクリックします。



さきほどの「Program1.cbl」に戻ると、プログラムが挿入されています。

OpenESQL アシスタント Program1.cbl\* OESQLAssistantTutorial

```
*> $SET SQL(DBMAN=ODBC) sourceformat(variable)
WORKING-STORAGE SECTION.

    EXEC SQL INCLUDE SQLCA END-EXEC.

*> after an sql error this has the full message text
01 MFSQMESSAGETEXT 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.

    EXEC SQL
        WHENEVER SQLERROR perform OpenESQL-Error
    END-EXEC

    EXEC SQL
        CONNECT TO 'OESQLDemo64'
    END-EXEC
*> Put your program logic/SQL statements here
```

- ⑤ IDE のメニューより [ファイル(F)] > [Program1.cbl の保存(S)] を選択します。
- ⑥ Program1.cbl の “CONNECT TO 'OESQLDemo'” の末尾に “USER ‘データベースユーザーID.パスワード’” 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。

補足)

以下の例は、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。

OpenESQL アシスタント Program1.cbl\* OESQLAssistantTutorial

```
*> $SET SQL(DBMAN=ODBC) sourceformat(variable)
WORKING-STORAGE SECTION.

    EXEC SQL INCLUDE SQLCA END-EXEC.

*> after an sql error this has the full message text
01 MFSQMESSAGETEXT 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.

    EXEC SQL
        WHENEVER SQLERROR perform OpenESQL-Error
    END-EXEC

    EXEC SQL
        CONNECT TO 'OESQLDemo64' USER 'sa.hogehoge'
    END-EXEC
```

- ⑦ Program1.cbl の “Put your program logic/SQL statements here” の次の空行をクリックします。

OpenESQL アシスタント Program1.cbl\* OESQLAssistantTutorial

```

CBL OESQLAssistantTutorial PROGRAM1
*> $SET SQL(DBMAN=ODBC) sourceformat(variable)
WORKING-STORAGE SECTION.

EXEC SQL INCLUDE SQLCA END-EXEC.

*> after an sql_error this has the full message text
01 MFSQMESSAGEGETTEXT 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.

EXEC SQL
  WHENEVER SQLERROR perform OpenESQL-Error
END-EXEC

EXEC SQL
  CONNECT TO 'OESQLDemo64' USER 'sa.hogehoge'
END-EXEC
*> Put your program logic/SQL statements here

EXEC SQL DISCONNECT CURRENT END-EXEC

```

- ⑧ 再度、OpenESQL アシスタントを開き、「クエリー」タブを選択の上、[現在のプログラムにクエリーを挿入する] をクリックします。

OpenESQL アシスタント Program1.cbl\* OESQLAssistantTutorial

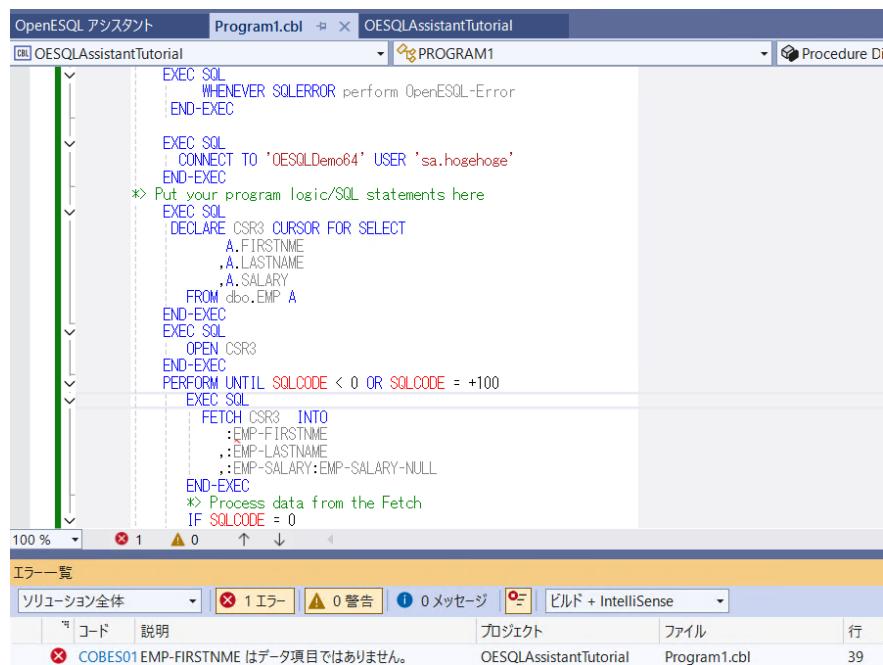
クエリー 実行結果 詳細 検索条件 ソート 補助コード

```

OESQLDemo64
  ACT(dbo)
  authors(dbo)
  DEPT(dbo)
  EACT(dbo)
  EDEPT(dbo)
  EEMP(dbo)
  EEPAP(dbo)
  EMP(dbo)
    A.EMPNO
    A.FIRSTNAME
    A.LASTNAME
    A.SALARY
  FROM dbo.EMP A

```

再度、「Program1.cbl」に戻ると、新たにコードが追加されています。現時点ではホスト変数項目が定義されていないため、エラーとなっていますが、次の手順で対応します。ここでは、この状態で変更を保存します。



```

OpenESQL アシスタント Program1.cbl OESQLAssistantTutorial
OESQLAssistantTutorial PROGRAM1 Procedure Di
EXEC SQL
WHENEVER SQLERROR perform OpenESQL-Error
END-EXEC

EXEC SQL
CONNECT TO 'OESQLDemo64' USER 'sa.hogehoge'
END-EXEC
*> Put your program logic/SQL statements here
EXEC SQL
DECLARE CSR3 CURSOR FOR SELECT
      A.FIRSTNAME
      ,A.LASTNAME
      ,A.SALARY
      FROM dbo.EMP A
END-EXEC
EXEC SQL
OPEN CSR3
END-EXEC
PERFORM UNTIL SQLCODE < 0 OR SQLCODE = +100
EXEC SQL
  FETCH CSR3 INTO
    :EMP-FIRSTNAME
    ::EMP-LASTNAME
    ::EMP-SALARY:EMP-SALARY-NULL
END-EXEC
*> Process data from the Fetch
IF SQLCODE = 0

```

エラー一覧

ソリューション全体 1 エラー 0 警告 0 メッセージ ビルド + IntelliSense

コード 説明 プロジェクト ファイル 行

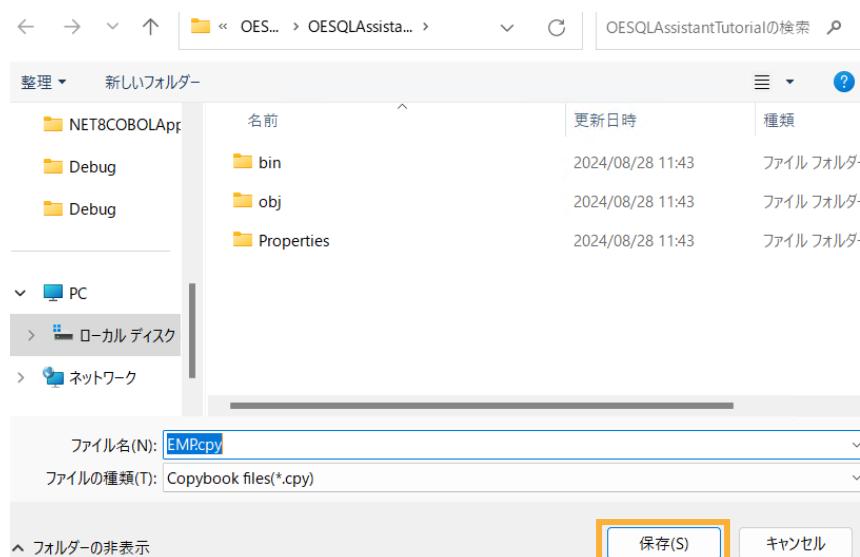
COBES01 EMP-FIRSTNAME はデータ項目ではありません。 OESQLAssistantTutorial Program1.cbl 39

## 2) ホスト変数項目の追加

- ① OpenESQL アシスタントを表示し、「EMP(dbo)」を選択し、マウスの右クリックにてコンテキストメニューを表示した上で、[DCLGEN を生成する(D)] を選択します。



- ② そのまま、[保存(S)] をクリックします。

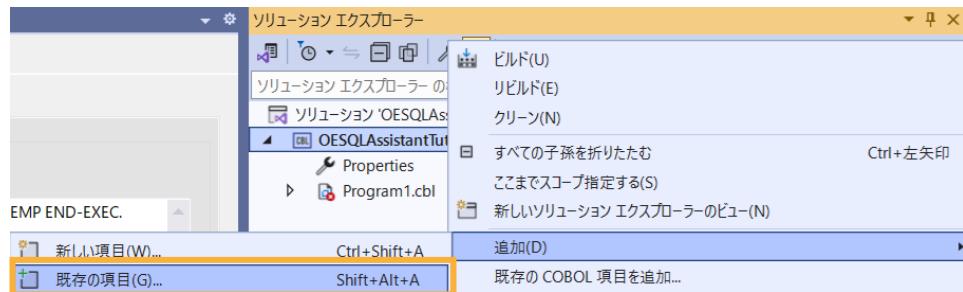


補足)

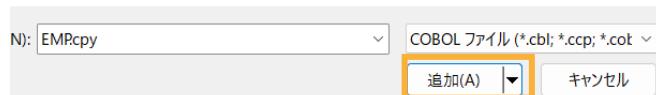
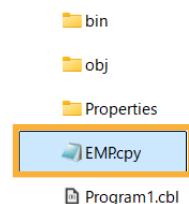
デフォルトのファイル名は、テーブル名が表示されます。

デフォルトの保存先は、開いているプロジェクト（今回は、OESQLAssistantTutorial）の直下です。

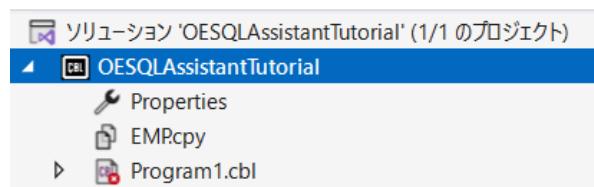
- ③ ソリューションエクスプローラーより、OpenESQLAssistantTutorial プロジェクトを選択し、マウスの右クリックにてコンテキストメニューを表示した上で、[追加(D)] > [既存の項目(G)] を選択します。



- ④ 「EMP」 COBOL コピー・ブックを選択し、[追加(A)] をクリックします。



EMP.cpy がプロジェクト配下に追加されます。



EMP.cpy をダブルクリックすると、以下のような COBOL プログラムが表示されます。

```
EMP.cpy ✎ X OpenESQL アシスタント Program1.cbl OESQLAssistantTutorial  
[CL] OESQLAssistantTutorial EMP DC  
*> *> DECLARE TABLE for EMP  
*>  
*> EXEC SQL DECLARE EMP TABLE  
( EMPNO char(6) NOT NULL  
, FIRSTNAME varchar(12) NOT NULL  
, MIDINIT char(1) NOT NULL  
, LASTNAME varchar(15) NOT NULL  
, WORKDEPT char(3)  
, PHONENO char(4)  
, HIREDATE date  
, JOB char(8)  
, EDLEVEL smallint  
, SEX char(1)  
, BIRTHDATE date  
, SALARY decimal(9,2)  
, BONUS decimal(9,2)  
, COMM decimal(9,2)  
) END-EXEC.  
*>  
*> COBOL HOST VARIABLES FOR TABLE EMP  
*>  
01 DCLEMP.  
03 EMP-EMPNO  
03 EMP-FIRSTNAME  
03 EMP-MIDINIT  
03 EMP-LASTNAME  
03 EMP-WORKDEPT  
03 EMP-PHONENO  
03 EMP-HIREDATE  
03 EMP-JOB  
03 EMP-EDLEVEL  
03 EMP-SEX  
03 EMP-BIRTHDATE  
03 EMP-SALARY  
03 EMP-BONUS  
03 EMP-COMM  
SQL TYPE IS  
CHAR(6).  
SQL TYPE IS  
CHAR(12).  
100 % 11 0
```

OpenESQL アシスタントが、選択したテーブル構造を基にホスト変数など、必要な定義が記述されています。

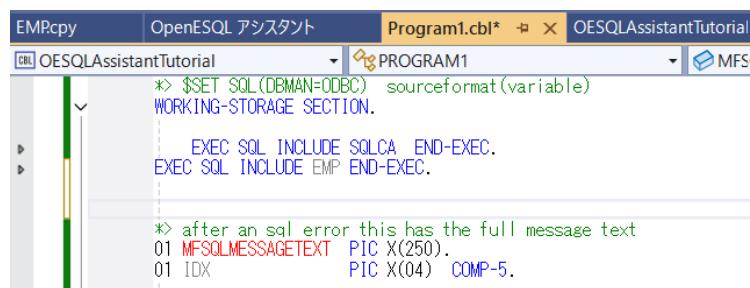
- ⑤ 上記コピー句を Program1.cbl の適切な位置に挿入するため、ソリューションエクスプローラーより「Program1.cbl」をダブルクリックします。そのうえで、"EXEC SQL INCLUDE SQLCA END-EXEC." 句の次の空行をクリックします。

```
EMPcpl | OpenESQL アシスタント | Program1.cbl ✘ X OESQLAssis...  
[CTRL] OESQLAssistantTutorial -> PROGRAM1  
*> $$SET SQL (DBMAN=ODBC) sourceformat(variable)  
WORKING-STORAGE SECTION.  
EXEC SQL INCLUDE SQLCA END-EXEC.  
/*> after an SQL error this has the full message text  
01 MFSQMESSAGEGETTEXT PIC X(250).  
01 IDX PIC X(04) COMP-5.
```

- ⑥ OpenESQL アシスタントを開き、「補助コード」タブを選択すると、EXEC SQL INCLUDE 句が指定されています。「現在のプログラムにクエリーを挿入する」アイコンをクリックします。



再度、Program1.cbl に戻ると、EMP コピーブックの INCLUDE が挿入されています。



```

EMPLCPY      OpenESQL アシスタント      Program1.cbl*  OESQLAssistantTutorial
[CHL] OESQLAssistantTutorial      [PROG] PROGRAM1      [MFS]
*> $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 MFSQMESSAGEGETTEXT PIC X(250).
01 IDX          PIC X(04)  COMP-5.

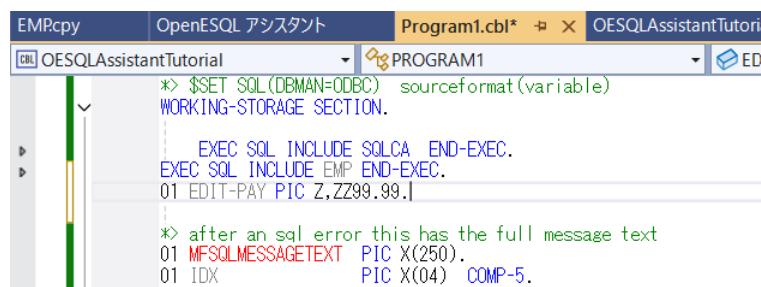
```

⑦ Program1.cbl の変更を保存します。

### 3) 実行時のコンソールログの追加

① Program1.cbl を開き、“ EXEC SQL INCLUDE EMP END-EXEC” の後に以下の行を追加します。

“01 EDIT-PAY PIC Z,ZZ99.99.”



```

EMPLCPY      OpenESQL アシスタント      Program1.cbl*  OESQLAssistantTutorial
[CHL] OESQLAssistantTutorial      [PROG] PROGRAM1      [ED]
*-> $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 MFSQMESSAGEGETTEXT PIC X(250).
01 IDX          PIC X(04)  COMP-5.

```

② “DISPLAY 'ROW FOUND'” 行を以下の2行と置換します。

“MOVE EMP-SALARY TO EDIT-PAY”

“ DISPLAY 'NAME: ' EMP-LASTNAME ', ' EMP-FIRSTNME ' SALARY: ' EDIT-PAY UPON  
CONSOLE”

#### 変更前

```

PERFORM UNTIL SQLCODE < 0 OR SQLCODE = +100
  EXEC SQL
    FETCH CSR3 INTO
      :EMP-FIRSTNAME
      ,: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
END-PERFORM

```

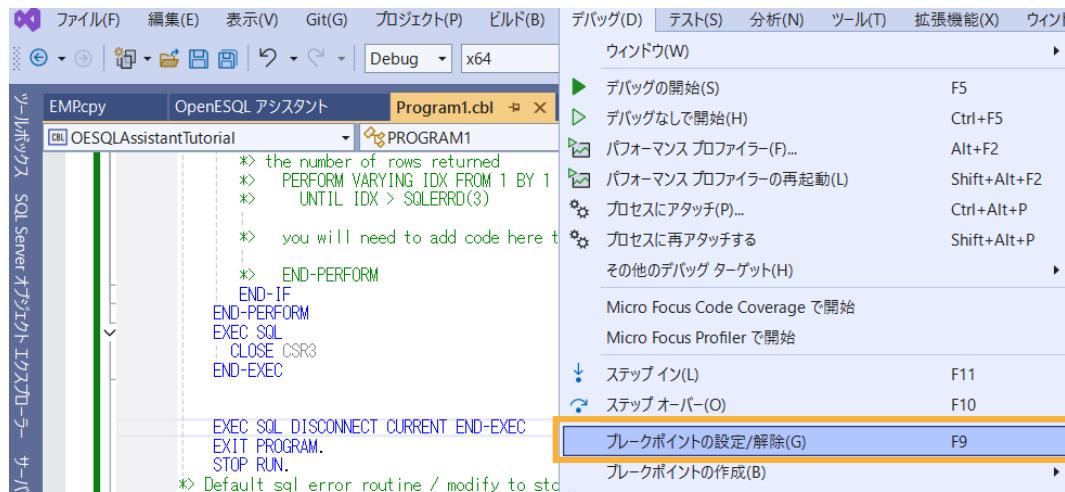
## 変更後

```
PERFORM UNTIL SQLCODE < 0 OR SQLCODE = +100
  EXEC SQL
    FETCH CSR3 INTO
      :EMP-FIRSTNAME
      ,: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-FIRSTNAME
           ' 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
END-PERFORM
```

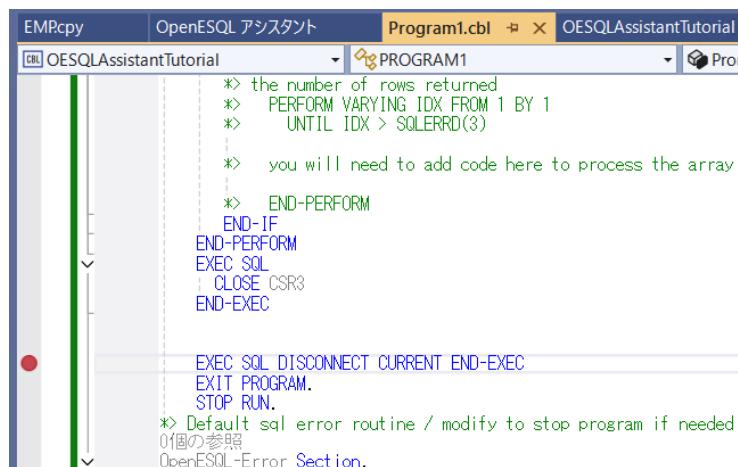
- ③ 変更を保存します。

### 3.6 SQL 埋め込みプログラムの実行

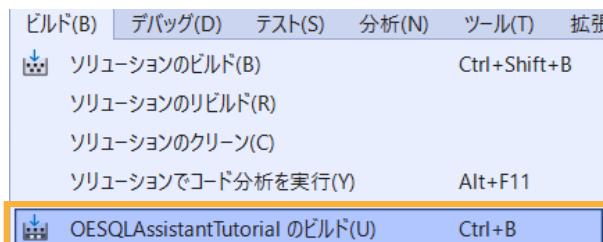
- 1) Program1.cbl を開き、" EXEC SQL DISCONNECT CURRENT END-EXEC" の行をクリックし、IDE のメニューより [デバッグ(D)] > [ブレークポイントの設定/解除(G)] を選択し、ブレークポイントを設定します。



ブレークポイントが設定された箇所には、赤丸が設定されます。

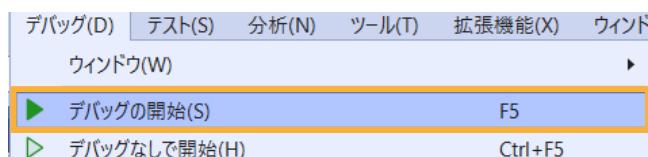


- 2) IDE のメニューより [ビルド(B)] > [OESQLAssistantTutorial のビルド(U)] をクリックするとビルドが正常に終了します。

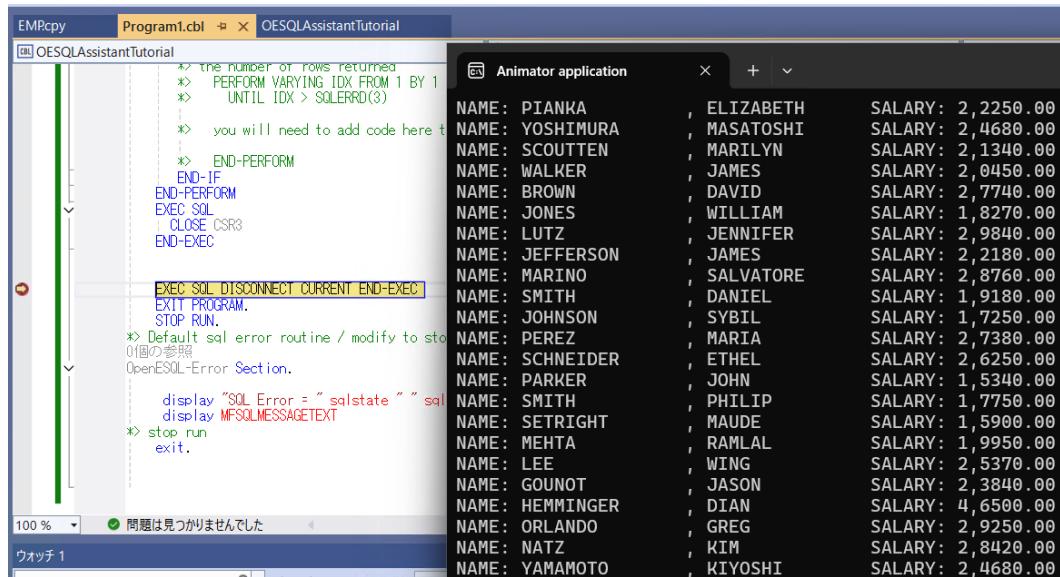


```
出力: ビルド
出力元(S): ビルド
1> * C:\Users\tarot\source\repos\OESQLAssistantTutorial\OESQLAssistantTutorial\Program1.cbl のコンパイル中
1> * Generating C:\obj\x64\Debug\Program1
1> * Data: 136688 Code: 5856 Literals: 784
1> COBOL コンパイル: 1 個 正常終了または最新の状態 0 個 失敗。
1> OESQLAssistantTutorial -> C:\Users\tarot\source\repos\OESQLAssistantTutorial\OESQLAssistantTutorial\bin\x64\
===== ビルド: 1 正常終了または最新の状態、0 失敗、0 スキップ =====
===== ビルドは 14:21 で完了し、02.314 秒 掛かりました =====
```

3) IDE のメニューより [デバッグ(D)] > [デバッグの開始(S)] をクリックして、デバッグ起動します。



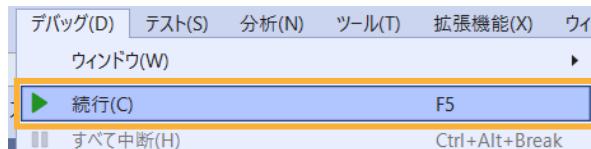
ブレークポイントで停止し、コンソール画面に検索結果が一覧で表示されます。



The screenshot shows the IDE interface with multiple windows. On the left, there's a code editor window for 'Program1.cbl' showing some assembly-like pseudocode. In the center, there's a search results window titled 'Animator application' displaying a list of names and salaries from a database. The results are as follows:

NAME		SALARY
PIANKA	, ELIZABETH	2,2250.00
YOSHIMURA	, MASATOSHI	2,4680.00
SCOUTTEN	, MARILYN	2,1340.00
WALKER	, JAMES	2,0450.00
BROWN	, DAVID	2,7740.00
JONES	, WILLIAM	1,8270.00
LUTZ	, JENNIFER	2,9840.00
JEFFERSON	, JAMES	2,2180.00
MARINO	, SALVATORE	2,8760.00
SMITH	, DANIEL	1,9180.00
JOHNSON	, SYBIL	1,7250.00
PEREZ	, MARIA	2,7380.00
SCHNEIDER	, ETHEL	2,6250.00
PARKER	, JOHN	1,5340.00
SMITH	, PHILIP	1,7750.00
SETRIGHT	, MAUDE	1,5900.00
MEHTA	, RAMLAL	1,9950.00
LEE	, WING	2,5370.00
GOUNOT	, JASON	2,3840.00
HEMMINGER	, DIAN	4,6500.00
ORLANDO	, GREG	2,9250.00
NATZ	, KIM	2,8420.00
YAMAMOTO	, KIYOSHI	2,4680.00

IDE のメニューより [デバッガ(D)] > [続行(C)] をクリックするとデバッグが終了します。



## 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、製品の一部ではありません。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、著作権法の精神に基づき、適切な扱いを行ってください。