

Catalog Functions in ODBC Pro

By Noredidine Margoum, QA Engineer, 4D S.A.

TN 06-10

Introduction

このテクニカルノートでは、ODBC Pro プラグインコマンドを使用し、データベースのカタログにアクセスする方法について説明しています。

Catalog functions

Note: ここでは、データベースの「フィールド」を指して「カラム」という語を使用します。

データベースには普通、テーブルのリストや各テーブルのアクセス権限情報などにアクセスするための仕組みがあります。そのような情報はカタログと呼ばれています。

ODBC Pro プラグインのカタログコマンドを使用すれば、そのようなストラクチャ情報にアクセスすることができます。

カタログコマンドを使用することのメリットとしては、次のようなものが挙げられます：

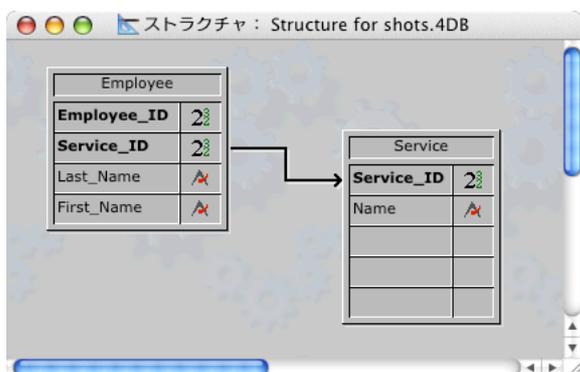
- **ODBC_SQLTables** および **ODBC_SQLColumns** を使用すれば、テーブルとカラムの情報が取得できるので、アプリケーションを実行しながら動的に SQL リクエストを組み立てることができる。
- **ODBC_SQLPrimaryKeys**、**ODBC_SQLForeignKeys**、**ODBC_SQLStatistics** を使用すれば、リレートやインデックスの情報が取得できるので、より効率的な SQL リクエストを実行することができる。

使用できるカタログコマンドとその説明は次のとおりです：

コマンド	説明
ODBC_SQLTables	カタログリスト、スキーマ、テーブル、テーブルタイプのリストをデータソースに対して返します。
ODBC_SQLColumns	カラムのリストをテーブルに対して返します。
ODBC_SQLStatistics	統計(レコード数、ユニークなインデックス値など)、関連付けられたインデックスのリストをテーブルに対して返しません。

- ODBC_SQLSpecialColumns** 単一のレコードを特定できるカラムのリストを返します。さらに、トランザクションでカラムが更新されると自動的に更新されるカラムのリストをそのテーブルに対して返します。
- ODBC_SQLPrimaryKeys** 主キーを構成するカラムのリストをテーブルに対して返します。主キーとは、テーブルで単一のレコードを特定することのできるカラムのことです。
- ODBC_SQLForeignKeys** 外部キーを構成するカラムのリストまたはそのテーブルに向かっている外部キーのリストをテーブルに対して返します。外部キーとは、他のテーブルの主キーを指しているカラムのことです。*
- ODBC_SQLTablePrivileges** テーブルのリスト、権限、権限を割り当てたユーザ、権限を割り当てられているユーザのリストを返します。
- ODBC_SQLColumnPrivileges** カラムのリスト、およびその権限をテーブルに対して返します。
- ODBC_SQLProcedures** ストアドプロシージャのリストをデータソースに対して返します。ストアドプロシージャとは、プロシージャと関数の両方を含みます。
- ODBC_SQLProcedureColumns** 入出力パラメータとカラムのリストをプロシージャに対して返します。パラメータのタイプ(**Input**、**Input/Output**、**Output**)も判別します。
- ODBC_SQLGetTypeInfo** サポートされるデータタイプをデータソースに対して返します。データタイプは、**CREATE TABLE**、**ALTER TABLE**などの**SQL** リクエストで使用されます。

*外部キーと主キーの関係は、図で表現すると理解しやすくなります：



上記のサンプルの場合、[Employee]テーブルには **Employee_ID** という主キー、[Services] テーブルには **Services_ID** という主キーが存在します。主キーはテーブルから単一のレコードを特定するために使用されます。主キーが複数のカラムで構成されるようなシステムも存在します。

この場合、外部キーは[Employee]テーブルの **Services_ID** だけになります。この外部キーは、[Employee]テーブルと[Services]テーブルの関係を定義しています。外部キーが複数のカラムで構成されるようなシステムも存在します。

すべてのカタログコマンドは、結果を並び替えられたカラムのリストで返します。カラムのデータを取得するには、4D のフィールドまたは変数を特定の位置にあるカラムにバインドしなくてはなりません。

Using the functions of the catalog

General use

```
C_LONGINT($0) `return code of the method
C_LONGINT($1;$vl_ConnectionID) `Connection ID
C_TEXT($2)
C_POINTER($3) ` pointers to array to retrieve data resulting from request
C_LONGINT($vl_StmtID) `statement handle
C_TEXT($vt_CatalogName) `Catalog name
C_TEXT($vt_SchemaName) `Schema name
C_TEXT($vt_TableName) `Table name
C_TEXT($vt_ColumnName) `Column name
$vl_ConnectionID:= $1
$vt_CatalogName:= " "
$vt_SchemaName:= " "
$vt_TableName:= $2
$vt_ColumnName:= " "
`allocatign the statement handle
vl_Status:= ODBC_SQLAllocStmt ($vl_ConnectionID;$vl_StmtID)
vl_Status:= ODBC_SQLColumnPrivileges
($vl_StmtID;$vt_CatalogName;$vt_SchemaName;$vt_TableName;$vt_ColumnName)
If ( (vl_Status= SQL_SUCCESS) | (vl_Status= SQL_SUCCESS_WITH_INFO) )
    vl_Status:= ODBC_SQLBindCol ($vl_StmtID;4;$3) `column name
    vl_Status:= ODBC_SQLBindCol ($vl_StmtID;5;$4) `name of assigner
    vl_Status:= ODBC_SQLBindCol ($vl_StmtID;6;$5)
    vl_Status:= ODBC_SQLBindCol ($vl_StmtID;7;$6) `privilée
    While ( (vl_Status#SQL_NO_DATA) & (vl_Status#SQL_ERROR) )
        vl_Status:= ODBC_SQLFetch ($vl_StmtID)
    End while
End if
```

```
vl_Status:= ODBC_SQLFreeStmt ($vl_StmtID;SQL_CLOSE )  
vl_Status:= ODBC_SQLFreeStmt ($vl_StmtID;SQL_DROP )
```

カタログコマンドは、上記の構造に従って実行します。

1. 接続ハンドル(**ODBC_SQLAllocStmt**)を使用して、倍長整数のステートメントハンドルを設定します。

2. ステートメントハンドルを第一パラメータにし、第二パラメータはコマンド特有のものを渡して、カタログコマンドを実行します。

3. カタログコマンドが正しく実行されると、結果として並び替えられたカラムのリストが返されます。

4. 結果として返されたカラムのそれぞれに対して、4D の変数またはフィールドを関連付けます。結果リストの中におけるカラムの位置は重要で、バインドの際に明示する必要があります。フィールドまたは変数をカラムとバインドするのは次のコードです：

```
ODBC_SQLBindCol(HandleRequete;PositionColonne;->VariableouChamp4D)
```

5. コマンドの実行結果データを取得するには、次のコマンドを使用します：

```
ODBC_SQLFetch
```

このコマンドは、エラーが発生するか、最後のデータが取り出されると実行を停止します。

6. リクエストを閉じて解放します。

Implementation example

```
C_LONGINT($0) `return parameter  
C_LONGINT($1;$vl_ConnectionID) `Connection ID  
C_TEXT($2) `Table name  
C_POINTER($3) `pointer to the arrays of the column names, and their types and sizes  
C_LONGINT($vl_StmtID) `statement handle  
C_TEXT($vt_CatalogName) `catalog name  
C_TEXT($vt_SchemaName) `schema name  
C_TEXT($vt_TableName) `table name  
C_TEXT($vt_ColumnName) `column name  
$vl_ConnectionID:= $1  
$vt_CatalogName:= " "  
$vt_SchemaName:= " "  
$vt_TableName:= $2  
$vt_ColumnName:= " "
```

```

`allocating the statement handle
vl_Status:= ODBC_SQLAllocStmt ($vl_ConnectionID;$vl_StmtID)
vl_Status:= ODBC_SQLColumns
($vl_StmtID;$vt_CatalogName;$vt_SchemaName;$vt_TableName;$vt_ColumnName)
If ( (vl_Status=SQL_SUCCESS) | (vl_Status=SQL_SUCCESS_WITH_INFO) )
    vl_Status:= ODBC_SQLBindCol ($vl_StmtID;4;$3) `column names
    vl_Status:= ODBC_SQLBindCol ($vl_StmtID;6;$4) `column types
    vl_Status:= ODBC_SQLBindCol ($vl_StmtID;8;$5) `column sizes
    While ( (vl_Status#SQL_NO_DATA) & (vl_Status#SQL_ERROR) )
        vl_Status:=ODBC_SQLFetch ($vl_StmtID)
    End while
Else
    $0:=SQL_ERROR
End if
vl_Status:= ODBC_SQLFreeStmt ($vl_StmtID;SQL_CLOSE)
vl_Status:= ODBC_SQLFreeStmt ($vl_StmtID;SQL_DROP)

```

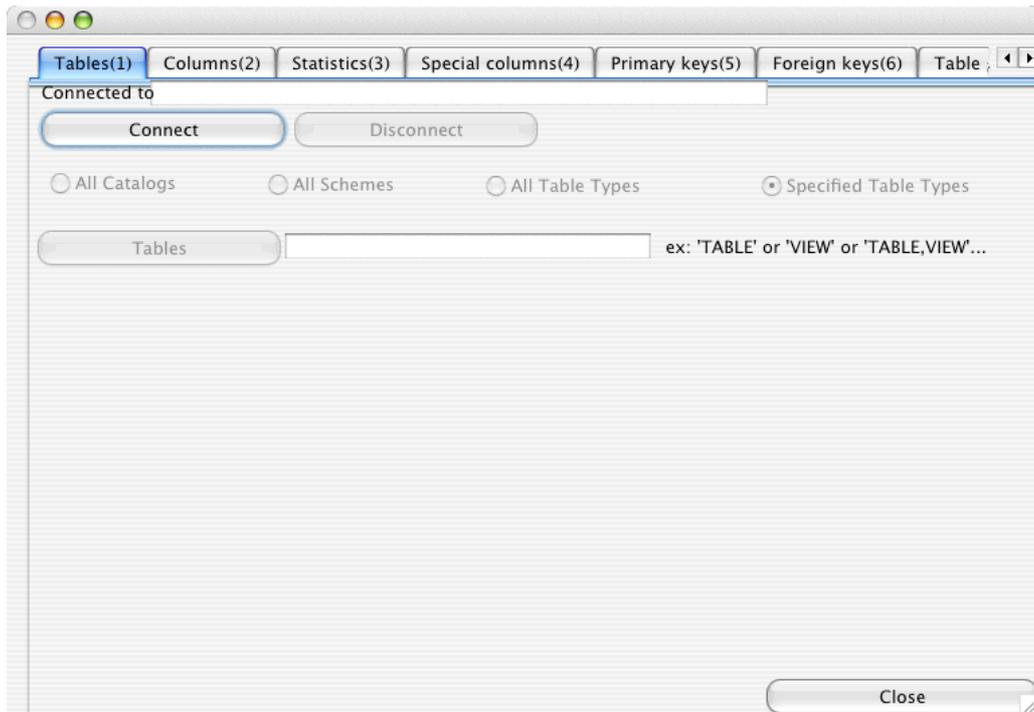
メソッドについての注釈：

上記のメソッドを実行すると、テーブルにあるカラムのリストがタイプおよびサイズとともに返されます。使用しているコマンドは **ODBC_SQLColumns** です。メソッドの構造は、前述の手順を踏まえた構成をしています。

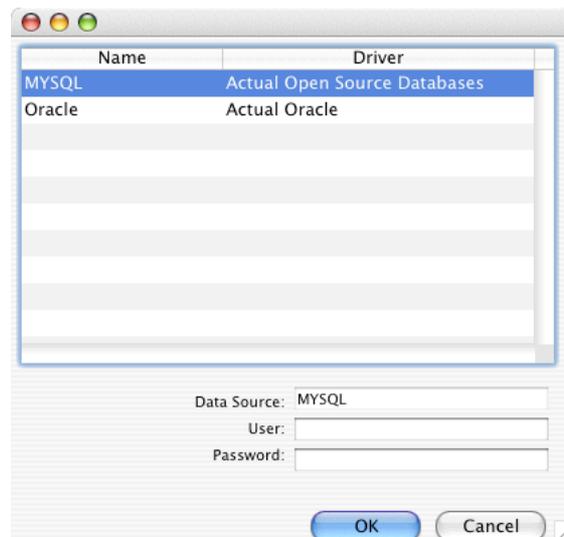
Using the Test database

テストデータベースには、すべてのカタログコマンドが実装されています。

データベースを起動するとメインダイアログ画面が表示され、ファイルメニュー/**Demo** を選択すれば、いつでもこの画面を表示することができます。



Connect ボタンをクリックすれば、接続先データソースを選択することができます。データソースは、ODBC データソースアドミニストレータで設定する必要があります。



データソースとの接続が確立されると、サンプルデータベースのさまざまな機能を使用することができます。

Conclusion

このテクニカルノートでは、ODBC Pro プラグインコマンドを使用し、データベースのカタログにアクセスする方法について説明しました。