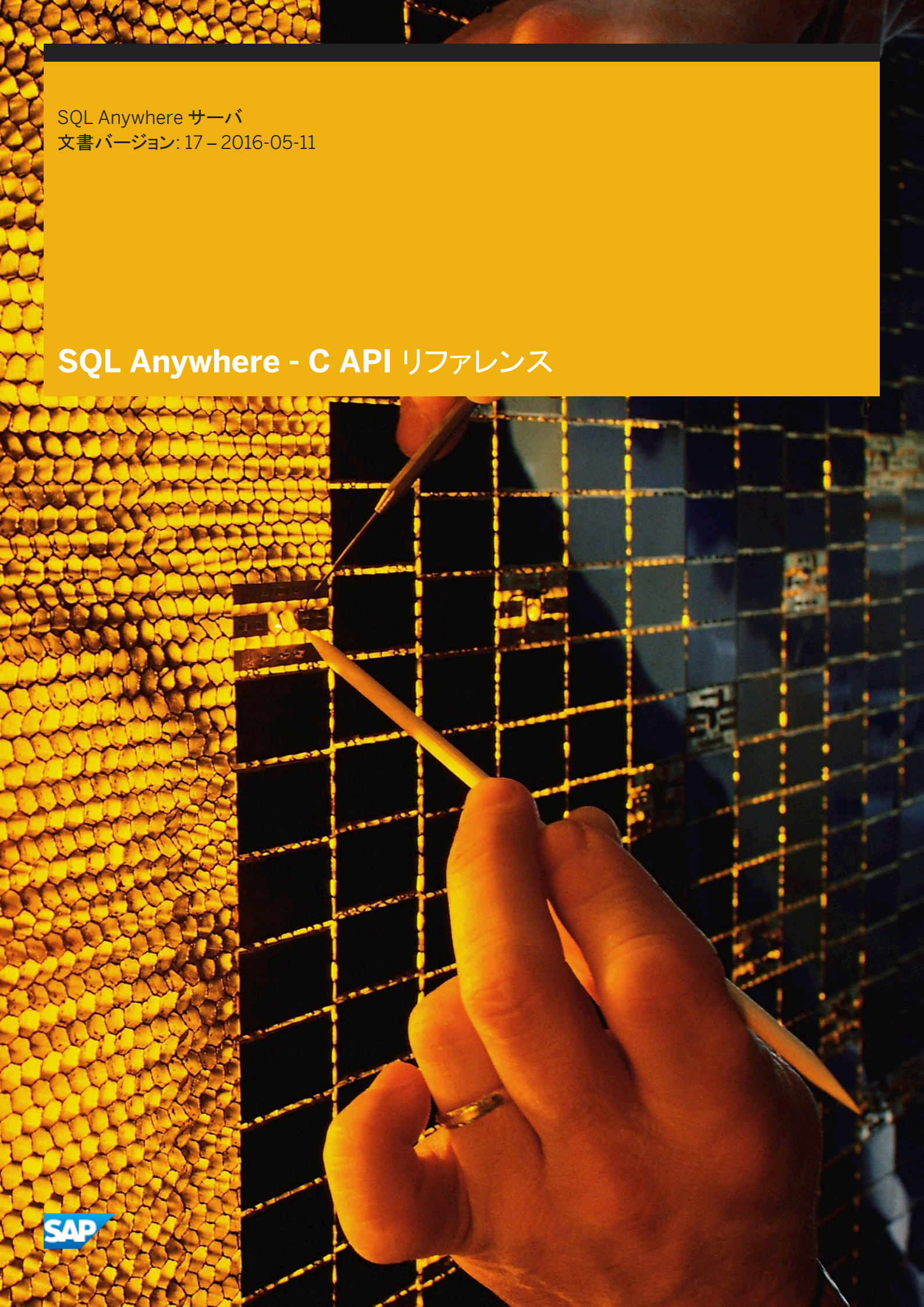

SQL Anywhere サーバ
文書バージョン: 17 – 2016-05-11

SQL Anywhere - C API リファレンス

目次

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

1 SQL Anywhere C API リファレンス

特定の C API 要素がヘッダファイルに定義されています。

ヘッダファイル

- `sacapi.h`
- `sacapidll.h`

備考

`sacapi.h` ヘッダファイルでは、SQL Anywhere C API エントリポイントを定義します。

`sacapidll.h` ヘッダファイルでは、C API ライブラリの初期化関数とファイナライズ関数を定義します。ソースファイルに `sacapidll.h` をインクルードし、`sacapidll.c` からソースコードをインクルードする必要があります。

> **注記**
>
> 主な **SQL Anywhere** マニュアルをお探しですか。マニュアルをローカルにインストールした場合は、Windows のスタートメニューを使用してアクセスするか (Microsoft Windows)、`C:¥Program Files¥SQL Anywhere 17¥Documentation` にナビゲートします。
>
> また、DocCommentXchange の Web で、主な SQL Anywhere API リファレンスマニュアルにアクセスすることもできます。http://dcx.sap.com

このセクションの内容:

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

SACAPI_ERROR_SIZE 変数 [67 ページ]
ビルドするバージョンをコマンドラインで指定しない場合、最新バージョンをビルドします。

SQLANY_API_VERSION_1 変数 [68 ページ]
バージョン 1 は、C/C++ API の初期バージョンです。

SQLANY_API_VERSION_2 変数 [68 ページ]
バージョン 2 では "_ex" 関数、および要求をキャンセルする機能が追加されました。

SQLANY_API_VERSION_3 変数 [68 ページ]
バージョン 3 より、"コールバック" 関数が導入されました。

SQLANY_API_VERSION_4 変数 [69 ページ]
バージョン 4 より、NCHAR サポートとワイド挿入が導入されました。

SQLANY_CALLBACK 変数 [69 ページ]
コールバック関数のタイプ。

1.1 sqlany_affected_rows(a_sqlany_stmt *) メソッド

準備文の実行の影響を受けるローの数を返します。

構文

```
public sacapi_i32 sqlany_affected_rows (a_sqlany_stmt * sqlany_stmt)
```

パラメータ

sqlany_stmt 準備および実行が成功したものの、結果セットが返されなかった文。たとえば、INSERT 文、UPDATE 文、または DELETE 文が実行された場合です。

戻り値

影響を受けたローの数。失敗した場合は -1。

関連情報

sqlany_execute(a_sqlany_stmt *) メソッド [19 ページ]
sqlany_execute_direct(a_sqlany_connection *, const char *) メソッド [20 ページ]

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

1.2 sqlany_bind_column(a_sqlany_stmt *, sacapi_u32, a_sqlany_data_value *) メソッド

ユーザが指定するバッファを準備文の結果セットカラムとしてバインドします。

構文

```
public sacapi_bool sqlany_bind_column (
    a_sqlany_stmt * sqlany_stmt,
    sacapi_u32 index,
    a_sqlany_data_value * value
)
```

パラメータ

sqlany_stmt sqlany_prepare() を使用して準備された文。

index カラムのインデックス。この数値は、0 ～ sqlany_num_cols() - 1 の間である必要があります。

value バインドされたバッファを記述する a_sqlany_data_value 構造体、または以前のバインド情報をクリアする場合は NULL。

戻り値

成功した場合は 1、失敗した場合は 0。

備考

フェッチしたローセットのサイズが 1 より大きい場合、バッファは、ローセットのすべてのローのデータを十分に格納できる大きさになっている必要があります。この関数は、値を NULL になるように指定することでカラムのバインドをクリアする場合にも使用できます。

関連情報

sqlany_clear_column_bindings(a_sqlany_stmt *) メソッド [12 ページ]
sqlany_set_rowset_size(a_sqlany_stmt *, sacapi_u32) メソッド [52 ページ]

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

1.3 sqlany_bind_param(a_sqlany_stmt *, sacapi_u32, a_sqlany_bind_param *) メソッド

ユーザが指定するバッファを準備文のパラメータとしてバインドします。

構文

```
public sacapi_bool sqlany_bind_param (
    a_sqlany_stmt * sqlany_stmt,
    sacapi_u32 index,
    a_sqlany_bind_param * param
)
```

パラメータ

sqlany_stmt sqlany_prepare() を使用して準備された文。
index パラメータのインデックス。この数値は、0 ～ sqlany_num_params() - 1 の間である必要があります。
param バインドされるパラメータの a_sqlany_bind_param 構造体の記述。

戻り値

成功した場合は 1、失敗した場合は 0。

関連情報

sqlany_describe_bind_param(a_sqlany_stmt *, sacapi_u32, a_sqlany_bind_param *) メソッド [17 ページ]

1.4 sqlany_cancel(a_sqlany_connection *) メソッド

接続の未処理の要求をキャンセルします。

構文

```
public void sqlany_cancel (a_sqlany_connection * sqlany_conn)
```


© 2016 SAP SE or an SAP affiliate company. All rights reserved.

パラメータ

sqlany_conn sqlany_connect() を使用して確立された接続の接続オブジェクト。

1.5 sqlany_clear_column_bindings(a_sqlany_stmt *) メソッド

sqlany_bind_column() を使用して定義したすべてのカラムバインドを削除します。

構文

```
public sacapi_bool sqlany_clear_column_bindings (a_sqlany_stmt * sqlany_stmt)
```

パラメータ

sqlany_stmt sqlany_prepare() を使用して準備された文。

戻り値

成功した場合は 1、失敗した場合は 0。

関連情報

sqlany_bind_column(a_sqlany_stmt *, sacapi_u32, a_sqlany_data_value *) メソッド [10 ページ]

1.6 sqlany_clear_error(a_sqlany_connection *) メソッド

最後に格納されたエラーコードをクリアします。

構文

```
public void sqlany_clear_error (a_sqlany_connection * sqlany_conn)
```


© 2016 SAP SE or an SAP affiliate company.All rights reserved.

パラメータ

sqlany_conn sqlany_new_connection() から返された接続オブジェクト。

関連情報

sqlany_new_connection(void) メソッド [39 ページ]

1.7 sqlany_client_version(char *, size_t) メソッド

現在のクライアントバージョンを返します。

構文

```
public sacapi_bool sqlany_client_version (
    char * buffer,
    size_t len
)
```

パラメータ

buffer クライアントバージョン文字列を格納するバッファ。

len 指定されたバッファの長さ。

戻り値

成功した場合は 1、失敗した場合は 0。

備考

このメソッドは、渡されたバッファに、クライアントライブラリのメジャー番号、マイナー番号、パッチ番号、ビルド番号を入れます。バッファは NULL で終了します。

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

1.8 sqlany_client_version_ex(a_sqlany_interface_context *, char *, size_t) メソッド

現在のクライアントバージョンを返します。

構文

```
public sacapi_bool sqlany_client_version_ex (
    a_sqlany_interface_context * context,
    char * buffer,
    size_t len
)
```

パラメータ

context sqlany_init_ex() を使用して作成されたオブジェクト。

buffer クライアントバージョン文字列を格納するバッファ。

len 指定されたバッファの長さ。

戻り値

成功した場合は 1、失敗した場合は 0。

備考

このメソッドは、渡されたバッファに、クライアントライブラリのメジャー番号、マイナー番号、パッチ番号、ビルド番号を入れます。バッファは NULL で終了します。

関連情報

sqlany_init_ex(const char *, sacapi_u32, sacapi_u32 *) メソッド [36 ページ]

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

1.9 sqlany_commit(a_sqlany_connection *) メソッド

現在のトランザクションをコミットします。

構文

```
public sacapi_bool sqlany_commit (a_sqlany_connection * sqlany_conn)
```

パラメータ

sqlany_conn コミット操作が実行される接続オブジェクト。

戻り値

成功した場合は 1、失敗した場合は 0。

関連情報

sqlany_rollback(a_sqlany_connection *) メソッド [46 ページ]

1.10 sqlany_connect(a_sqlany_connection *, const char *) メソッド

指定された接続オブジェクトと接続文字列を使用して、SQL Anywhere データベースサーバへの接続を作成します。

構文

```
public sacapi_bool sqlany_connect (
    a_sqlany_connection * sqlany_conn,
    const char * str
)
```


© 2016 SAP SE or an SAP affiliate company. All rights reserved.

パラメータ

sqlany_conn sqlany_new_connection() によって作成された接続オブジェクト。
str SQL Anywhere 接続文字列。

戻り値

接続が確立された場合は 1、接続が失敗した場合は 0。sqlany_error() を使用してエラーコードとエラーメッセージを取得します。

備考

まず、sqlany_new_connection() を使用して、指定された接続オブジェクトを割り付ける必要があります。

次の例は、失敗した接続試行のエラーコードを取得する方法を示します。

```
a_sqlany_connection * sqlany_conn;
sqlany_conn = sqlany_new_connection();
if( !sqlany_connect( sqlany_conn, "uid=dba;pwd=passwd" ) ) {
   char reason[SACAPI_ERROR_SIZE];
   sacapi_i32 code;
   code = sqlany_error( sqlany_conn, reason, sizeof(reason) );
   printf( "Connection failed. Code: %d Reason: %s¥n", code, reason );
} else {
   printf( "Connected successfully!¥n" );
   sqlany_disconnect( sqlany_conn );
}
sqlany_free_connection( sqlany_conn );
```

関連情報

sqlany_new_connection(void) メソッド [39 ページ]
sqlany_error(a_sqlany_connection *, char *, size_t) メソッド [18 ページ]

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

1.11 sqlany_describe_bind_param(a_sqlany_stmt *, sacapi_u32, a_sqlany_bind_param *) メソッド

準備文のバインドパラメータを記述します。

構文

```
public sacapi_bool sqlany_describe_bind_param (
    a_sqlany_stmt * sqlany_stmt,
    sacapi_u32 index,
    a_sqlany_bind_param * param
)
```

パラメータ

sqlany_stmt sqlany_prepare() を使用して準備された文。
index パラメータのインデックス。この数値は、0 ～ sqlany_num_params() - 1 の間である必要があります。
param 情報が格納された a_sqlany_bind_param 構造体。

戻り値

成功した場合は 1、失敗した場合は 0。

備考

この関数により、呼び出し元は準備文のパラメータに関する情報を判断できます。提供される情報の量は、準備文のタイプ (ストアドプロシージャまたは DML 文) によって決まります。パラメータの方向 (入力、出力、または入出力) に関する情報は常に提供されます。

関連情報

sqlany_bind_param(a_sqlany_stmt *, sacapi_u32, a_sqlany_bind_param *) メソッド [11 ページ]
sqlany_prepare(a_sqlany_connection *, const char *) メソッド [43 ページ]

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

1.12 sqlany_disconnect(a_sqlany_connection *) メソッド

すでに確立された SQL Anywhere 接続を切断します。

構文

```
public sacapi_bool sqlany_disconnect (a_sqlany_connection * sqlany_conn)
```

パラメータ

sqlany_conn sqlany_connect() を使用して確立された接続の接続オブジェクト。

戻り値

成功した場合は 1、失敗した場合は 0。

備考

コミットされていないトランザクションはすべてロールバックされます。

関連情報

sqlany_connect(a_sqlany_connection *, const char *) メソッド [15 ページ]
sqlany_new_connection(void) メソッド [39 ページ]

1.13 sqlany_error(a_sqlany_connection *, char *, size_t) メソッド

接続オブジェクトに最後に格納されたエラーコードとエラーメッセージを取得します。

構文

```
public sacapi_i32 sqlany_error (
    a_sqlany_connection * sqlany_conn,
    char * buffer,
```


© 2016 SAP SE or an SAP affiliate company.All rights reserved.

```
    size_t size
)
```

パラメータ

sqlany_conn sqlany_new_connection() から返された接続オブジェクト。
buffer エラーメッセージを格納するバッファ。
size 指定されたバッファのサイズ。

戻り値

最後のエラーコード。正の値は警告、負の値はエラー、0 は成功を示します。

関連情報

sqlany_connect(a_sqlany_connection *, const char *) メソッド [15 ページ]

1.14 sqlany_execute(a_sqlany_stmt *) メソッド

準備文を実行します。

構文

```
public sacapi_bool sqlany_execute (a_sqlany_stmt * sqlany_stmt)
```

パラメータ

sqlany_stmt sqlany_prepare() を使用して準備された文。

戻り値

文が正しく実行された場合は 1、失敗した場合は 0。

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

備考

sqlany_num_cols() を使用して、実行された文が結果セットを返したかどうか確認できます。

次の例は、結果セットを返さない文を実行する方法を示します。

```
a_sqlany_stmt *          stmt;
int                      i;
a_sqlany_bind_param   param;
stmt = sqlany_prepare( sqlany_conn, "insert into moe(id,value) values( ?,? )" );
if( stmt ) {
   sqlany_describe_bind_param( stmt, 0, &param );
   param.value.buffer = (char *)&i;
   param.value.type   = A_VAL32;
   sqlany_bind_param( stmt, 0, &param );
   sqlany_describe_bind_param( stmt, 1, &param );
   param.value.buffer = (char *)&i;
   param.value.type   = A_VAL32;
   sqlany_bind_param( stmt, 1, &param );
   for( i = 0; i < 10; i++ ) {
       if( !sqlany_execute( stmt ) ) {
               // call sqlany_error()
           }
   }
   sqlany_free_stmt( stmt );
}
```

関連情報

sqlany_prepare(a_sqlany_connection *, const char *) メソッド [43 ページ]

1.15 sqlany_execute_direct(a_sqlany_connection *, const char *) メソッド

文字列引数で指定された SQL 文を実行し、結果セットを返す可能性があります。

構文

```
public a_sqlany_stmt * sqlany_execute_direct (
    a_sqlany_connection * sqlany_conn,
    const char * sql_str
)
```

パラメータ

sqlany_conn sqlany_connect() を使用して確立された接続の接続オブジェクト。

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

sql_str SQL 文字列。SQL 文字列には、? のようなパラメータを含めることはできません。

戻り値

関数の実行が成功した場合はステートメントハンドル、失敗した場合は NULL。

備考

このメソッドを使用して、文を準備および実行できます。また、sqlany_prepare() に続けて sqlany_execute() を呼び出す代わりにも使用できます。

次の例は、結果セットを返す文を実行する方法を示します。

```
a_sqlany_stmt *   stmt;
stmt = sqlany_execute_direct( sqlany_conn, "select * from employees" );
if( stmt && sqlany_num_cols( stmt ) > 0 ) {
   while( sqlany_fetch_next( stmt ) ) {
       int i;
           for( i = 0; i < sqlany_num_cols( stmt ); i++ ) {
           // Get column i data
       }
   }
   sqlany_free_stmt( stmt  );
}
```

注記

この関数は、パラメータを持つ SQL 文の実行には使用できません。

関連情報

sqlany_fetch_absolute(a_sqlany_stmt *, sacapi_i32) メソッド [22 ページ]
sqlany_fetch_next(a_sqlany_stmt *) メソッド [23 ページ]
sqlany_num_cols(a_sqlany_stmt *) メソッド [41 ページ]
sqlany_get_column(a_sqlany_stmt *, sacapi_u32, a_sqlany_data_value *) メソッド [29 ページ]

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

1.16 sqlany_execute_immediate(a_sqlany_connection *, const char *) メソッド

指定された SQL 文を、結果セットを返さずにただちに実行します。

構文

```
public sacapi_bool sqlany_execute_immediate (
    a_sqlany_connection * sqlany_conn,
    const char * sql
)
```

パラメータ

sqlany_conn sqlany_connect() を使用して確立された接続の接続オブジェクト。

sql 実行される SQL 文を表す文字列。

戻り値

成功した場合は 1、失敗した場合は 0。

備考

この関数は、結果セットを返さない SQL 文に役立ちます。

1.17 sqlany_fetch_absolute(a_sqlany_stmt *, sacapi_i32) メソッド

結果セット内の現在のローを、指定されたロー番号に移し、現在のローから始まるデータのローをフェッチします。

構文

```
public sacapi_bool sqlany_fetch_absolute (
    a_sqlany_stmt * sqlany_stmt,
    sacapi_i32 row_num
)
```


© 2016 SAP SE or an SAP affiliate company.All rights reserved.

パラメータ

sqlany_stmt sqlany_execute() または sqlany_execute_direct() によって実行された文オブジェクト。
row_num フェッチされるローの番号。最初のローは 1、最後のローは -1。

戻り値

フェッチに成功した場合は 1、失敗した場合は 0。

備考

フェッチするローの数は、sqlany_set_rowset_size() 関数で設定されます。デフォルトでは、1 つのローが返されます。

関連情報

sqlany_execute_direct(a_sqlany_connection *, const char *) メソッド [20 ページ]
sqlany_execute(a_sqlany_stmt *) メソッド [19 ページ]
sqlany_error(a_sqlany_connection *, char *, size_t) メソッド [18 ページ]
sqlany_fetch_next(a_sqlany_stmt *) メソッド [23 ページ]
sqlany_set_rowset_size(a_sqlany_stmt *, sacapi_u32) メソッド [52 ページ]

1.18 sqlany_fetch_next(a_sqlany_stmt *) メソッド

結果セットから次のローのセットを返します。

構文

```
public sacapi_bool sqlany_fetch_next (a_sqlany_stmt * sqlany_stmt)
```

パラメータ

sqlany_stmt sqlany_execute() または sqlany_execute_direct() によって実行された文オブジェクト。

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

戻り値

フェッチに成功した場合は 1、失敗した場合は 0。

備考

結果オブジェクトを初めて作成した場合、現在のローポインタは最初のロー (すなわちロー 0) の前に設定されます。この関数が呼び出されると、フェッチされていない次のローにローポインタが進み、そのローから始まるデータのローがフェッチされます。フェッチするローの数は、sqlany_set_rowset_size() 関数で設定されます。デフォルトでは、1 つのローが返されます。

関連情報

sqlany_fetch_absolute(a_sqlany_stmt *, sacapi_i32) メソッド [22 ページ]
sqlany_execute_direct(a_sqlany_connection *, const char *) メソッド [20 ページ]
sqlany_execute(a_sqlany_stmt *) メソッド [19 ページ]
sqlany_error(a_sqlany_connection *, char *, size_t) メソッド [18 ページ]
sqlany_set_rowset_size(a_sqlany_stmt *, sacapi_u32) メソッド [52 ページ]

1.19 sqlany_fetched_rows(a_sqlany_stmt *) メソッド

フェッチしたローの数を返します。

構文

```
public sacapi_i32 sqlany_fetched_rows (a_sqlany_stmt * sqlany_stmt)
```

パラメータ

sqlany_stmt sqlany_prepare() を使用して準備された文。

戻り値

フェッチしたローの数。失敗した場合は -1。

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

備考

通常、フェッチしたローの数は、sqlany_set_rowset_size() 関数で指定したサイズと同じです。例外は、フェッチ位置から結果セットの最後までのローの数が指定値よりも少ない場合です。この場合、フェッチしたローの数は、指定したローセットサイズよりも少なくなります。最後のフェッチが失敗した場合、または文が実行されなかった場合、この関数は -1 を返します。文が実行されたものの、フェッチが実行されなかった場合、この関数は 0 を返します。

関連情報

sqlany_bind_column(a_sqlany_stmt *, sacapi_u32, a_sqlany_data_value *) メソッド [10 ページ]
sqlany_fetch_next(a_sqlany_stmt *) メソッド [23 ページ]
sqlany_fetch_absolute(a_sqlany_stmt *, sacapi_i32) メソッド [22 ページ]

1.20 sqlany_finalize_interface(SQLAnywhereInterface *) メソッド

C API DLL ライブラリをアンロードし、SQLAnywhereInterface 構造体をリセットします。

構文

```
public void sqlany_finalize_interface (SQLAnywhereInterface * api)
```

パラメータ

api ファイナライズする初期化された構造体。

備考

次の文を使用して関数プロトタイプをインクルードします。

```
#include "sacapidll.h"
```

このメソッドを使用して、SQL Anywhere C API DLL に関連付けられているリソースのファイナライズおよび解放を実行します。

sqlany_finalize_interface メソッドの使用例については、インストールされている SQL Anywhere の `sdk¥dbcapi¥examples` ディレクトリにある C API の例を参照してください。

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

1.21 sqlany_fini() メソッド

インタフェースをファイナライズします。

構文

```
public void sqlany_fini ()
```

備考

API によって割り付けられたリソースをすべて解放します。

関連情報

sqlany_init(const char *, sacapi_u32, sacapi_u32 *) メソッド [35 ページ]

1.22 sqlany_fini_ex(a_sqlany_interface_context *) メソッド

指定したコンテキストを使用して作成されたインタフェースをファイナライズします。

構文

```
public void sqlany_fini_ex (a_sqlany_interface_context * context)
```

パラメータ

context sqlany_init_ex() から返されたコンテキストオブジェクト。

関連情報

sqlany_init_ex(const char *, sacapi_u32, sacapi_u32 *) メソッド [36 ページ]

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

1.23 sqlany_free_connection(a_sqlany_connection *) メソッド

接続オブジェクトに関連付けられているリソースを解放します。

構文

```
public void sqlany_free_connection (a_sqlany_connection * sqlany_conn)
```

パラメータ

sqlany_conn sqlany_new_connection によって作成された接続オブジェクト。

関連情報

sqlany_new_connection(void) メソッド [39 ページ]

1.24 sqlany_free_stmt(a_sqlany_stmt *) メソッド

準備文オブジェクトに関連付けられているリソースを解放します。

構文

```
public void sqlany_free_stmt (a_sqlany_stmt * sqlany_stmt)
```

パラメータ

sqlany_stmt sqlany_prepare() または sqlany_execute_direct() の実行によって返された文オブジェクト。

関連情報

sqlany_prepare(a_sqlany_connection *, const char *) メソッド [43 ページ]
sqlany_execute_direct(a_sqlany_connection *, const char *) メソッド [20 ページ]

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

1.25 sqlany_get_batch_size(a_sqlany_stmt *) メソッド

バッチ実行のロー配列のサイズを取得します。

構文

```
public sacapi_u32 sqlany_get_batch_size (a_sqlany_stmt * sqlany_stmt)
```

パラメータ

sqlany_stmt sqlany_prepare() を使用して準備された文。

戻り値

ロー配列のサイズ。

関連情報

sqlany_set_batch_size(a_sqlany_stmt *, sacapi_u32) メソッド [48 ページ]

1.26 sqlany_get_bind_param_info(a_sqlany_stmt *, sacapi_u32, a_sqlany_bind_param_info *) メソッド

sqlany_bind_param() を使用してバインドされたパラメータに関する情報を取得します。

構文

```
public sacapi_bool sqlany_get_bind_param_info (
    a_sqlany_stmt * sqlany_stmt,
    sacapi_u32 index,
    a_sqlany_bind_param_info * info
)
```


© 2016 SAP SE or an SAP affiliate company.All rights reserved.

パラメータ

sqlany_stmt sqlany_prepare() を使用して準備された文。
index パラメータのインデックス。この数値は、0 ～ sqlany_num_params() - 1 の間である必要があります。
info バインドパラメータの情報を格納する sqlany_bind_param_info バッファ。

戻り値

成功した場合は 1、失敗した場合は 0。

関連情報

sqlany_bind_param(a_sqlany_stmt *, sacapi_u32, a_sqlany_bind_param *) メソッド [11 ページ]
sqlany_describe_bind_param(a_sqlany_stmt *, sacapi_u32, a_sqlany_bind_param *) メソッド [17 ページ]
sqlany_prepare(a_sqlany_connection *, const char *) メソッド [43 ページ]

1.27 sqlany_get_column(a_sqlany_stmt *, sacapi_u32, a_sqlany_data_value *) メソッド

現在のローの指定カラムのためにフェッチされた値を、指定されたバッファに格納します。

構文

```
public sacapi_bool sqlany_get_column (
    a_sqlany_stmt * sqlany_stmt,
    sacapi_u32 col_index,
    a_sqlany_data_value * buffer
)
```

パラメータ

sqlany_stmt sqlany_execute() または sqlany_execute_direct() によって実行された文オブジェクト。
col_index 取り出すカラムの数。カラムの数は、0 ～ sqlany_num_cols() - 1 の間です。
buffer a_sqlany_data_value オブジェクトには、ローセットの現在のローのカラム col_index のためにフェッチされた値が入ります。

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

戻り値

成功した場合は 1、失敗した場合は 0。パラメータのいずれかが無効であった場合、または SQL Anywhere データベースサーバから完全な値を取り出すために必要なメモリがない場合は、失敗する可能性があります。

備考

sqlany_fetch_absolute() 関数または sqlany_fetch_next() 関数を実行すると、ローセットが作成され、現在のローがローセットの最初のローに設定されます。現在のローは sqlany_set_rowset_pos() 関数で設定されます。

A_BINARY および A_STRING * データ型では、value->buffer は、結果セットに関連付けられている内部バッファをポイントします。ポインタバッファの内容は、新しいローがフェッチされたとき、または結果セットオブジェクトが解放されたときに変更されます。したがって、ポインタの内容を変更したり、内容に依存したりしないでください。ユーザはこれらのポインタから各自のバッファにデータをコピーする必要があります。

value->length フィールドは、value->buffer がポイントする有効な文字の数を示します。value->buffer で返されるデータは、NULL で終了しません。この関数は、SQL Anywhere データベースサーバからのすべての戻り値をフェッチします。たとえば、カラムに BLOB が含まれている場合、その値を保持するのに必要なメモリの割り付けはこの関数が行います。メモリの割り付けを行わない場合は、代わりに sqlany_get_data() を使用してください。

関連情報

sqlany_execute_direct(a_sqlany_connection *, const char *) メソッド [20 ページ]
sqlany_execute(a_sqlany_stmt *) メソッド [19 ページ]
sqlany_fetch_absolute(a_sqlany_stmt *, sacapi_i32) メソッド [22 ページ]
sqlany_fetch_next(a_sqlany_stmt *) メソッド [23 ページ]
sqlany_set_rowset_pos(a_sqlany_stmt *, sacapi_u32) メソッド [51 ページ]

1.28 sqlany_get_column_info(a_sqlany_stmt *, sacapi_u32, a_sqlany_column_info *) メソッド

カラムのメタデータ情報を取得し、a_sqlany_column_info 構造体にカラムに関する情報を格納します。

構文

```
public sacapi_bool sqlany_get_column_info (
    a_sqlany_stmt * sqlany_stmt,
    sacapi_u32 col_index,
    a_sqlany_column_info * buffer
)
```


© 2016 SAP SE or an SAP affiliate company.All rights reserved.

パラメータ

sqlany_stmt sqlany_prepare() または sqlany_execute_direct() によって作成された文オブジェクト。
col_index カラムの数は、0 ～ sqlany_num_cols() - 1 の間です。
buffer カラム情報を格納するカラム info 構造体。

戻り値

成功した場合は 1。カラムのインデックスが範囲外の場合、または文が結果セットを返さない場合は 0。

関連情報

sqlany_execute(a_sqlany_stmt *) メソッド [19 ページ]
sqlany_execute_direct(a_sqlany_connection *, const char *) メソッド [20 ページ]
sqlany_prepare(a_sqlany_connection *, const char *) メソッド [43 ページ]

1.29 sqlany_get_data(a_sqlany_stmt *, sacapi_u32, size_t, void *, size_t) メソッド

現在のローの指定されたカラムのためにフェッチしたデータを取り出し、指定されたバッファのメモリに格納します。

構文

```
public sacapi_i32 sqlany_get_data (
    a_sqlany_stmt * sqlany_stmt,
    sacapi_u32 col_index,
    size_t offset,
    void * buffer,
    size_t size
)
```

パラメータ

sqlany_stmt sqlany_execute() または sqlany_execute_direct() によって実行された文オブジェクト。
col_index 取り出すカラムの数。カラムの数は、0 ～ sqlany_num_cols() - 1 の間です。
offset 取得するデータの開始オフセット。
buffer バッファには、ローセットの現在のローのカラムの内容が入ります。コピーされるデータ型に応じてバッファポインタのアラインメントを適切に行う必要があります。

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

size バッファのサイズ (バイト単位)。指定したサイズが 2^31 - 1 より大きいと、関数は失敗します。

戻り値

指定されたバッファにコピーされたバイト数。この値は 2^31 - 1 を超えることはできません。0 は、コピーすべきデータが残っていないことを意味します。-1 は失敗を示します。

備考

sqlany_fetch_absolute() 関数または sqlany_fetch_next() 関数を実行すると、ローセットが作成され、現在のローがローセットの最初のローに設定されます。現在のローは sqlany_set_rowset_pos() 関数で設定されます。

関連情報

sqlany_execute(a_sqlany_stmt *) メソッド [19 ページ]
sqlany_execute_direct(a_sqlany_connection *, const char *) メソッド [20 ページ]
sqlany_fetch_absolute(a_sqlany_stmt *, sacapi_i32) メソッド [22 ページ]
sqlany_fetch_next(a_sqlany_stmt *) メソッド [23 ページ]
sqlany_set_rowset_pos(a_sqlany_stmt *, sacapi_u32) メソッド [51 ページ]

1.30 sqlany_get_data_info(a_sqlany_stmt *, sacapi_u32, a_sqlany_data_info *) メソッド

現在のローにあるフェッチ済みデータに関する情報を取得します。

構文

```
public sacapi_bool sqlany_get_data_info (
    a_sqlany_stmt * sqlany_stmt,
    sacapi_u32 col_index,
    a_sqlany_data_info * buffer
)
```

パラメータ

sqlany_stmt sqlany_execute() または sqlany_execute_direct() によって実行された文オブジェクト。

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

col_index カラムの数は、0 ～ sqlany_num_cols() - 1 の間です。
buffer データ情報バッファには、ローセットの現在のローのデータに関するメタデータが入ります。

戻り値

成功した場合は 1、失敗した場合は 0。指定したパラメータのいずれかが無効であった場合、失敗が返されます。

備考

sqlany_fetch_absolute() 関数または sqlany_fetch_next() 関数を実行すると、ローセットが作成され、現在のローがローセットの最初のローに設定されます。現在のローは sqlany_set_rowset_pos() 関数で設定されます。

関連情報

sqlany_execute(a_sqlany_stmt *) メソッド [19 ページ]
sqlany_execute_direct(a_sqlany_connection *, const char *) メソッド [20 ページ]
sqlany_fetch_absolute(a_sqlany_stmt *, sacapi_i32) メソッド [22 ページ]
sqlany_fetch_next(a_sqlany_stmt *) メソッド [23 ページ]
sqlany_set_rowset_pos(a_sqlany_stmt *, sacapi_u32) メソッド [51 ページ]

1.31 sqlany_get_next_result(a_sqlany_stmt *) メソッド

複数の結果セットクエリのうちの次の結果セットに進みます。

構文

```
public sacapi_bool sqlany_get_next_result (a_sqlany_stmt * sqlany_stmt)
```

パラメータ

sqlany_stmt sqlany_execute() または sqlany_execute_direct() によって実行された文オブジェクト。

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

戻り値

文が次の結果セットに進んだ場合は 1、それ以外の場合は 0。

備考

クエリ (ストアドプロシージャの呼び出しなど) が複数の結果セットを返す場合、この関数は現在の結果セットから次へ進みます。

次の例は、複数の結果セットクエリのうちの次の結果セットに進む方法を示します。

```
stmt = sqlany_execute_direct( sqlany_conn, "call my_multiple_results_procedure()" );
if( result ) {
   do {
       while( sqlany_fetch_next( stmt ) ) {
          // get column data
       }
   } while( sqlany_get_next_result( stmt ) );
   sqlany_free_stmt( stmt );
}
```

関連情報

sqlany_execute_direct(a_sqlany_connection *, const char *) メソッド [20 ページ]
sqlany_execute(a_sqlany_stmt *) メソッド [19 ページ]

1.32 sqlany_get_rowset_size(a_sqlany_stmt *) メソッド

sqlany_fetch_absolute() 関数と sqlany_fetch_next() 関数でフェッチされるローセットのサイズを取得します。

構文

```
public sacapi_u32 sqlany_get_rowset_size (a_sqlany_stmt * sqlany_stmt)
```

パラメータ

sqlany_stmt sqlany_prepare() を使用して準備された文。

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

戻り値

ローセットのサイズ。文が結果セットを返さない場合は 0。

関連情報

sqlany_set_rowset_size(a_sqlany_stmt *, sacapi_u32) メソッド [52 ページ]
sqlany_fetch_absolute(a_sqlany_stmt *, sacapi_i32) メソッド [22 ページ]
sqlany_fetch_next(a_sqlany_stmt *) メソッド [23 ページ]

1.33 sqlany_init(const char *, sacapi_u32, sacapi_u32 *) メソッド

インタフェースを初期化します。

構文

```
public sacapi_bool sqlany_init (
    const char * app_name,
    sacapi_u32 api_version,
    sacapi_u32 * version_available
)
```

パラメータ

app_name API を使用しているアプリケーションに名前を付ける文字列。たとえば、"PHP"、"PERL"、"RUBY" など。
api_version コンパイルされたアプリケーションのバージョン。
version_available サポートされている API の最大バージョンを返すオプションの引数。

戻り値

成功した場合は 1、失敗した場合は 0。

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

備考

次の例は、SQL Anywhere C API DLL を初期化する方法を示します。

```
sacapi_u32 api_version;
if( sqlany_init( "PHP", SQLANY_API_VERSION_1, &api_version ) ) {
   printf( "Interface initialized successfully!¥n" );
} else {
   printf( "Failed to initialize the interface! Supported version=%d¥n",
api_version );
}
```

関連情報

sqlany_fini() メソッド [26 ページ]

1.34 sqlany_init_ex(const char *, sacapi_u32, sacapi_u32 *) メソッド

コンテキストを使用してインタフェースを初期化します。

構文

```
public a_sqlany_interface_context * sqlany_init_ex (
    const char * app_name,
    sacapi_u32 api_version,
    sacapi_u32 * version_available
)
```

パラメータ

app_name 使用される API に "PHP"、"PERL"、"RUBY" などの名前を付ける文字列。

api_version アプリケーションで使用されている現在の API バージョン。通常、SQLANY_API_VERSION_* マクロの 1 つである必要があります。

version_available サポートされている API の最大バージョンを返すオプションの引数。

戻り値

成功した場合はコンテキストオブジェクト、失敗した場合は NULL。

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

関連情報

sqlany_fini_ex(a_sqlany_interface_context *) メソッド [26 ページ]

1.35 sqlany_initialize_interface(SQLAnywhereInterface *, const char *) メソッド

SQLAnywhereInterface オブジェクトを初期化し、DLL を動的にロードします。

構文

```
public int sqlany_initialize_interface (
    SQLAnywhereInterface * api,
    const char * optional_path_to_dll
)
```

パラメータ

api 初期化する API structure 構造体。
optional_path_to_dll SQL Anywhere C API DLL へのパスを指定するオプションの引数。

戻り値

初期化に成功した場合は 1、失敗した場合は 0。

備考

次の文を使用して関数プロトタイプをインクルードします。

```
#include "sacapidll.h"
```

この関数は SQL Anywhere C API DLL を動的にロードし、DLL のすべてのエントリポイントを検索します。SQLAnywhereInterface 構造体のフィールドに入力されたデータは、DLL の該当する関数をポイントします。オプションのパス引数が NULL の場合は、SQLANY_DLL_PATH 環境変数がチェックされます。SQLANY_DLL_PATH 環境変数が設定されている場合、ライブラリは環境変数が指定する DLL をロードします。失敗した場合、インタフェースが DLL を直接ロードします (環境が正しく設定されているかによって決まります)。

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

sqlany_initialize_interface メソッドの使用例については、インストールされている SQL Anywhere の `sdk¥dbcapi¥examples` ディレクトリにある C API の例を参照してください。

1.36 sqlany_make_connection(void *) メソッド

指定された DBLIB SQLCA ポインタに基づいて接続オブジェクトを作成します。

構文

```
public a_sqlany_connection * sqlany_make_connection (void * arg)
```

パラメータ

arg DBLIB SQLCA オブジェクトへの void * ポインタ。

戻り値

接続オブジェクト。

関連情報

sqlany_new_connection(void) メソッド [39 ページ]
sqlany_execute(a_sqlany_stmt *) メソッド [19 ページ]
sqlany_execute_direct(a_sqlany_connection *, const char *) メソッド [20 ページ]
sqlany_execute_immediate(a_sqlany_connection *, const char *) メソッド [22 ページ]
sqlany_prepare(a_sqlany_connection *, const char *) メソッド [43 ページ]

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

1.37 sqlany_make_connection_ex(a_sqlany_interface_context *, void *) メソッド

指定された DBLIB SQLCA ポインタとコンテキストに基づいて接続オブジェクトを作成します。

構文

```
public a_sqlany_connection * sqlany_make_connection_ex (
    a_sqlany_interface_context * context,
    void * arg
)
```

パラメータ

context sqlany_init_ex() によって作成された有効なコンテキストオブジェクト。
arg DBLIB SQLCA オブジェクトへの void * ポインタ。

戻り値

接続オブジェクト。

関連情報

sqlany_init_ex(const char *, sacapi_u32, sacapi_u32 *) メソッド [36 ページ]
sqlany_execute(a_sqlany_stmt *) メソッド [19 ページ]
sqlany_execute_direct(a_sqlany_connection *, const char *) メソッド [20 ページ]
sqlany_execute_immediate(a_sqlany_connection *, const char *) メソッド [22 ページ]
sqlany_prepare(a_sqlany_connection *, const char *) メソッド [43 ページ]

1.38 sqlany_new_connection(void) メソッド

接続オブジェクトを作成します。

構文

```
public a_sqlany_connection * sqlany_new_connection (void)
```


© 2016 SAP SE or an SAP affiliate company. All rights reserved.

戻り値

接続オブジェクト

備考

API 接続オブジェクトを作成してからデータベース接続を確立する必要があります。接続オブジェクトからエラーが取得される場合があります。各接続で一度に処理できる要求は 1 つだけです。また、接続オブジェクトに一度にアクセスできるスレッドは 1 つに限られます。複数のスレッドが同時に接続オブジェクトへのアクセスを試みると、不確定な動作や障害が発生します。

関連情報

sqlany_connect(a_sqlany_connection *, const char *) メソッド [15 ページ]
sqlany_disconnect(a_sqlany_connection *) メソッド [18 ページ]

1.39 sqlany_new_connection_ex(a_sqlany_interface_context *) メソッド

コンテキストを使用して接続オブジェクトを作成します。

構文

```
public a_sqlany_connection * sqlany_new_connection_ex (a_sqlany_interface_context
* context)
```

パラメータ

context sqlany_init_ex() から返されたコンテキストオブジェクト。

戻り値

接続オブジェクト

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

関連情報

sqlany_connect(a_sqlany_connection *, const char *) メソッド [15 ページ]
sqlany_disconnect(a_sqlany_connection *) メソッド [18 ページ]
sqlany_init_ex(const char *, sacapi_u32, sacapi_u32 *) メソッド [36 ページ]

1.40 sqlany_num_cols(a_sqlany_stmt *) メソッド

結果セット内のカラム数を返します。

構文

```
public sacapi_i32 sqlany_num_cols (a_sqlany_stmt * sqlany_stmt)
```

パラメータ

sqlany_stmt sqlany_prepare() または sqlany_execute_direct() によって作成された文オブジェクト。

戻り値

結果セット内のカラム数。失敗した場合は -1。

関連情報

sqlany_execute(a_sqlany_stmt *) メソッド [19 ページ]
sqlany_execute_direct(a_sqlany_connection *, const char *) メソッド [20 ページ]
sqlany_prepare(a_sqlany_connection *, const char *) メソッド [43 ページ]

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

1.41 sqlany_num_params(a_sqlany_stmt *) メソッド

準備文で必要とされるパラメータ数を返します。

構文

```
public sacapi_i32 sqlany_num_params (a_sqlany_stmt * sqlany_stmt)
```

パラメータ

sqlany_stmt sqlany_prepare() の実行によって返された文オブジェクト。

戻り値

必要とされるパラメータ数。文オブジェクトが有効でない場合は -1。

関連情報

sqlany_prepare(a_sqlany_connection *, const char *) メソッド [43 ページ]

1.42 sqlany_num_rows(a_sqlany_stmt *) メソッド

結果セット内のロー数を返します。

構文

```
public sacapi_i32 sqlany_num_rows (a_sqlany_stmt * sqlany_stmt)
```

パラメータ

sqlany_stmt sqlany_execute() または sqlany_execute_direct() によって実行された文オブジェクト。

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

戻り値

結果セット内のロー数。ロー数が推定値の場合は負の値を返します。また、その推定値が、返された整数の絶対値となります。ロー数が正確な値の場合は正の値を返します。

備考

デフォルトでは、この関数は推定値のみを返します。正確なロー数が返されるようにするには、接続の row_counts オプションを設定します。

関連情報

sqlany_execute_direct(a_sqlany_connection *, const char *) メソッド [20 ページ]
sqlany_execute(a_sqlany_stmt *) メソッド [19 ページ]

1.43 sqlany_prepare(a_sqlany_connection *, const char *) メソッド

指定された SQL 文字列を準備します。

構文

```
public a_sqlany_stmt * sqlany_prepare (
    a_sqlany_connection * sqlany_conn,
    const char * sql_str
)
```

パラメータ

sqlany_conn sqlany_connect() を使用して確立された接続の接続オブジェクト。
sql_str 準備される SQL 文。

戻り値

SQL Anywhere 文オブジェクトのハンドル。sqlany_execute() で文オブジェクトを使用して、文を実行できます。

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

備考

sqlany_execute() が呼び出されるまで実行は行われません。返された文オブジェクトは、sqlany_free_stmt() を使用して解放する必要があります。

次の例は、SELECT SQL 文字列を準備する方法を示します。

```
char * str;
a_sqlany_stmt * stmt;
str = "select * from employees where salary >= ?";
stmt = sqlany_prepare( sqlany_conn, str );
if( stmt == NULL ) {
   // Failed to prepare statement, call sqlany_error() for more info
}
```

関連情報

sqlany_free_stmt(a_sqlany_stmt *) メソッド [27 ページ]
sqlany_connect(a_sqlany_connection *, const char *) メソッド [15 ページ]
sqlany_execute(a_sqlany_stmt *) メソッド [19 ページ]
sqlany_num_params(a_sqlany_stmt *) メソッド [42 ページ]
sqlany_describe_bind_param(a_sqlany_stmt *, sacapi_u32, a_sqlany_bind_param *) メソッド [17 ページ]
sqlany_bind_param(a_sqlany_stmt *, sacapi_u32, a_sqlany_bind_param *) メソッド [11 ページ]

1.44 sqlany_register_callback(a_sqlany_connection *, a_sqlany_callback_type, SQLANY_CALLBACK_PARM) メソッド

コールバックルーチンを登録します。

構文

```
public sacapi_bool sqlany_register_callback (
    a_sqlany_connection * sqlany_conn,
    a_sqlany_callback_type index,
    SQLANY_CALLBACK_PARM callback
)
```

パラメータ

sqlany_conn sqlany_connect() を使用して確立された接続の接続オブジェクト。

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

index コールバックのタイプを下に列挙します。

callback コールバックルーチンのアドレス。

戻り値

成功した場合は 1、失敗した場合は 0。

備考

この関数はコールバック関数の登録に使用できます。

```
A callback type can be any one of the following:
   CALLBACK_START
   CALLBACK_WAIT
   CALLBACK_FINISH
   CALLBACK_MESSAGE
   CALLBACK_CONN_DROPPED
   CALLBACK_DEBUG_MESSAGE
   CALLBACK_VALIDATE_FILE_TRANSFER
The following example shows a simple message callback routine and how to register
it.
void SQLANY_CALLBACK messages(
   a_sqlany_connection *sqlany_conn,
   a_sqlany_message_type msg_type,
   int sqlcode,
   unsigned short length,
   char *msg )
{
   size_t  mlen;
   char    mbuffer[80];
   mlen = __min( length, sizeof(mbuffer) );
   strncpy( mbuffer, msg, mlen );
   mbuffer[mlen] = '¥0';
   printf( "Message is ¥"s".
", mbuffer );
   sqlany_sqlstate( sqlany_conn, mbuffer, sizeof( mbuffer ) );
   printf( "SQLCode(%d) SQLState(¥"s")
", sqlcode, mbuffer );
}
```

1.45 sqlany_reset(a_sqlany_stmt *) メソッド

文を準備されたステータス状態にリセットします。

構文

```
public sacapi_bool sqlany_reset (a_sqlany_stmt * sqlany_stmt)
```


© 2016 SAP SE or an SAP affiliate company. All rights reserved.

パラメータ

sqlany_stmt sqlany_prepare() を使用して準備された文。

戻り値

成功した場合は 1、失敗した場合は 0。

関連情報

sqlany_prepare(a_sqlany_connection *, const char *) メソッド [43 ページ]

1.46 sqlany_rollback(a_sqlany_connection *) メソッド

現在のトランザクションをロールバックします。

構文

```
public sacapi_bool sqlany_rollback (a_sqlany_connection * sqlany_conn)
```

パラメータ

sqlany_conn ロールバック操作が実行される接続オブジェクト。

戻り値

成功した場合は 1、失敗した場合は 0。

関連情報

sqlany_commit(a_sqlany_connection *) メソッド [15 ページ]

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

1.47 sqlany_send_param_data(a_sqlany_stmt *, sacapi_u32, char *, size_t) メソッド

データをバインドパラメータの一部として送信します。

構文

```
public sacapi_bool sqlany_send_param_data (
    a_sqlany_stmt * sqlany_stmt,
    sacapi_u32 index,
    char * buffer,
    size_t size
)
```

パラメータ

sqlany_stmt sqlany_prepare() を使用して準備された文。
index パラメータのインデックス。これは 0 ～ sqlany_num_params() - 1 の間の数です。
buffer 送信されるデータ。
size 送信するバイト数。

戻り値

成功した場合は 1、失敗した場合は 0。

備考

このメソッドを使用して、バインドパラメータの大量のデータをチャンク単位で送信できます。このメソッドは、バッチサイズが 1 の場合にのみ使用可能です。

関連情報

sqlany_prepare(a_sqlany_connection *, const char *) メソッド [43 ページ]

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

1.48 sqlany_set_batch_size(a_sqlany_stmt *, sacapi_u32) メソッド

バッチ実行のロー配列のサイズを設定します。

構文

```
public sacapi_bool sqlany_set_batch_size (
    a_sqlany_stmt * sqlany_stmt,
    sacapi_u32 num_rows
)
```

パラメータ

sqlany_stmt sqlany_prepare() を使用して準備された文。
num_rows バッチ実行のローの数。値は 1 以上でなければなりません。

戻り値

成功した場合は 1、失敗した場合は 0。

備考

バッチサイズは INSERT 文の場合にのみ使用されます。デフォルトのバッチサイズは 1 です。1 より大きな値はワイド挿入を示します。

関連情報

sqlany_bind_param(a_sqlany_stmt *, sacapi_u32, a_sqlany_bind_param *) メソッド [11 ページ]
sqlany_get_batch_size(a_sqlany_stmt *) メソッド [28 ページ]

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

1.49 sqlany_set_column_bind_type(a_sqlany_stmt *, sacapi_u32) メソッド

カラムのバインド型を設定します。

構文

```
public sacapi_bool sqlany_set_column_bind_type (
    a_sqlany_stmt * sqlany_stmt,
    sacapi_u32 row_size
)
```

パラメータ

sqlany_stmt sqlany_prepare() を使用して準備された文。

row_size ローのバイトサイズ。値 0 はカラムワイズバインド、正の値はローワイズバインドを示します。

戻り値

成功した場合は 1、失敗した場合は 0。

備考

デフォルト値は 0 で、カラムワイズバインドを示します。ゼロ以外の値はローワイズバインドを示し、ローを格納するデータ構造のバイトサイズを指定します。カラムは、連続する値配列の最初の要素にバインドされます。次の要素へのアドレスのオフセットはバインド型に基づいて計算されます。

- カラムワイドバインド - カラム型のバイトサイズ
- ローワイドバインド - ローのサイズ

関連情報

sqlany_bind_column(a_sqlany_stmt *, sacapi_u32, a_sqlany_data_value *) メソッド [10 ページ]

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

1.50 sqlany_set_param_bind_type(a_sqlany_stmt *, size_t) メソッド

パラメータのバインド型を設定します。

構文

```
public sacapi_bool sqlany_set_param_bind_type (
    a_sqlany_stmt * sqlany_stmt,
    size_t row_size
)
```

パラメータ

sqlany_stmt sqlany_prepare() を使用して準備された文。
row_size ローのバイトサイズ。値 0 はカラムワイズバインド、正の値はローワイズバインドを示します。

戻り値

成功した場合は 1、失敗した場合は 0。

備考

デフォルト値は 0 で、カラムワイズバインドを示します。ゼロ以外の値はローワイズバインドを示し、ローを格納するデータ構造のバイトサイズを指定します。パラメータは、連続する値配列の最初の要素にバインドされます。次の要素へのアドレスのオフセットはバインド型に基づいて計算されます。

- カラムワイドバインド - パラメータ型のバイトサイズ
- ローワイドバインド - ローのサイズ

関連情報

sqlany_bind_param(a_sqlany_stmt *, sacapi_u32, a_sqlany_bind_param *) メソッド [11 ページ]

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

1.51 sqlany_set_rowset_pos(a_sqlany_stmt *, sacapi_u32) メソッド

フェッチされたローセットの現在のローを設定します。

構文

```
public sacapi_bool sqlany_set_rowset_pos (
    a_sqlany_stmt * sqlany_stmt,
    sacapi_u32 row_num
)
```

パラメータ

sqlany_stmt sqlany_prepare() を使用して準備された文。
row_num ローセット内のロー番号。有効な値は 0 から sqlany_fetched_rows() - 1 までの範囲です。

戻り値

成功した場合は 1、失敗した場合は 0。

備考

sqlany_fetch_absolute() 関数または sqlany_fetch_next() 関数を実行すると、ローセットが作成され、現在のローがローセットの最初のローに設定されます。sqlany_get_column()、sqlany_get_data()、sqlany_get_data_info() の各関数は、現在のローでデータを取得するために使用されます。

関連情報

sqlany_set_rowset_size(a_sqlany_stmt *, sacapi_u32) メソッド [52 ページ]
sqlany_get_column(a_sqlany_stmt *, sacapi_u32, a_sqlany_data_value *) メソッド [29 ページ]
sqlany_get_data(a_sqlany_stmt *, sacapi_u32, size_t, void *, size_t) メソッド [31 ページ]
sqlany_get_data_info(a_sqlany_stmt *, sacapi_u32, a_sqlany_data_info *) メソッド [32 ページ]
sqlany_fetch_absolute(a_sqlany_stmt *, sacapi_i32) メソッド [22 ページ]
sqlany_fetch_next(a_sqlany_stmt *) メソッド [23 ページ]

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

1.52 sqlany_set_rowset_size(a_sqlany_stmt *, sacapi_u32) メソッド

sqlany_fetch_absolute() 関数と sqlany_fetch_next() 関数でフェッチされるローセットのサイズを設定します。

構文

```
public sacapi_bool sqlany_set_rowset_size (
    a_sqlany_stmt * sqlany_stmt,
    sacapi_u32 num_rows
)
```

パラメータ

sqlany_stmt sqlany_prepare() を使用して準備された文。
num_rows ローセットのサイズ。値は 1 以上でなければなりません。

戻り値

成功した場合は 1、失敗した場合は 0。

備考

ローセットのデフォルトサイズは 1 です。num_rows に 1 より大きな値を設定すると、ワイドフェッチになります。

関連情報

sqlany_bind_column(a_sqlany_stmt *, sacapi_u32, a_sqlany_data_value *) メソッド [10 ページ]
sqlany_fetch_absolute(a_sqlany_stmt *, sacapi_i32) メソッド [22 ページ]
sqlany_fetch_next(a_sqlany_stmt *) メソッド [23 ページ]
sqlany_get_rowset_size(a_sqlany_stmt *) メソッド [34 ページ]

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

1.53 sqlany_sqlstate(a_sqlany_connection *, char *, size_t) メソッド

現在の SQLSTATE を取得します。

構文

```
public size_t sqlany_sqlstate (
    a_sqlany_connection * sqlany_conn,
    char * buffer,
    size_t size
)
```

パラメータ

sqlany_conn sqlany_new_connection() から返された接続オブジェクト。
buffer 現在の 5 文字 SQLSTATE を格納するバッファ。
size バッファのサイズ。

戻り値

バッファにコピーされたバイト数。

関連情報

sqlany_error(a_sqlany_connection *, char *, size_t) メソッド [18 ページ]

1.54 a_sqlany_callback_type 列挙

コールバックタイプの列挙。

構文

```
enum a_sqlany_callback_type
```


© 2016 SAP SE or an SAP affiliate company. All rights reserved.

メンバー

メンバー名	説明
CALLBACK_START	この関数は、データベース要求がサーバに送信される直前に呼び出されます。 CALLBACK_START は、Windows オペレーティングシステムでのみ使用されます。
CALLBACK_WAIT	この関数は、データベースサーバまたはクライアントライブラリがデータベース要求を処理しているあいだ、インタフェースライブラリによって繰り返し呼び出されます。
CALLBACK_FINISH	この関数は、データベース要求に対する応答を DBLIB インタフェース DLL が受け取ったあとに呼び出されます。 CALLBACK_FINISH は、Windows オペレーティングシステムでのみ使用されます。
CALLBACK_MESSAGE	この関数は、要求の処理中にサーバからメッセージを受け取ったときに呼び出されます。 メッセージは、SQL MESSAGE 文を使用してクライアントアプリケーションからデータベースサーバに送信できます。実行時間が長いデータベースサーバ文によってメッセージも生成できます。
CALLBACK_CONN_DROPPED	この関数は、DROP CONNECTION 文を通じた活性タイムアウトのため、またはデータベースサーバがシャットダウンされているために、データベースサーバが接続を切断しようとするときに呼び出されます。 複数の接続を区別できるように、接続名 conn_name が渡されます。接続が無名の場合は、値が NULL になります。
CALLBACK_DEBUG_MESSAGE	この関数はデバッグメッセージごとに 1 回呼び出され、デバッグメッセージのテキストを含む NULL で終了する文字列が渡されます。 デバッグメッセージは、LogFile のファイルに記録されるメッセージです。デバッグメッセージをこのコールバックに渡すには、LogFile 接続パラメータを使用する必要があります。通常、この文字列の末尾の NULL 文字の直前に改行文字 () が付いています。
CALLBACK_VALIDATE_FILE_TRANSFER	この関数は、ファイル転送に検証が必要なときに呼び出されます。 ストアドプロシージャからなどの間接文の実行中にクライアントのデータ転送が要求された場合、クライアントライブラリは、クライアントアプリケーションで検証コールバックが登録されており、コールバックからの応答が転送許可を示していないかぎり転送を許可しません。

備考

Embedded SQL コールバック型に対応するコールバックタイプ。

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

関連情報

sqlany_register_callback(a_sqlany_connection *, a_sqlany_callback_type, SQLANY_CALLBACK_PARM) メソッド [44 ページ]

1.55 a_sqlany_data_direction 列挙

データ方向の列挙です。

構文

```
enum a_sqlany_data_direction
```

メンバー

メンバー名	説明	値
DD_INVALID	無効なデータ方向。	0x0
DD_INPUT	入力専用のホスト変数。	0x1
DD_OUTPUT	出力専用のホスト変数。	0x2
DD_INPUT_OUTPUT	入出力するホスト変数。	0x3

1.56 a_sqlany_data_type 列挙

渡されている、または取り出されているデータ型を指定します。

構文

```
enum a_sqlany_data_type
```


© 2016 SAP SE or an SAP affiliate company.All rights reserved.

メンバー

メンバー名	説明
A_INVALID_TYPE	無効なデータ型。
A_BINARY	バイナリデータ。バイナリデータは現状のままで処理され、文字セット変換は実行されません。
A_STRING	文字列データ。文字セット変換が実行されるデータ。
A_DOUBLE	Double データ。float 値を含みます。
A_VAL64	64 ビット整数値。
A_UVAL64	64 ビット符号なし整数
A_VAL32	32 ビット整数値。
A_UVAL32	32 ビット符号なし整数
A_VAL16	16 ビット整数値。
A_UVAL16	16 ビット符号なし整数
A_VAL8	8 ビット整数値。
A_UVAL8	8 ビット符号なし整数。

1.57 a_sqlany_message_type enumeration 列挙

MESSAGE コールバックに対するメッセージタイプの列挙。

構文

```
enum a_sqlany_message_type
```

メンバー

メンバー名	説明
MESSAGE_TYPE_INFO	メッセージタイプは INFO でした。
MESSAGE_TYPE_WARNING	メッセージタイプは WARNING でした。
MESSAGE_TYPE_ACTION	メッセージタイプは ACTION でした。
MESSAGE_TYPE_STATUS	メッセージタイプは STATUS でした。

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

メンバー名	説明
MESSAGE_TYPE_PROGRESS	メッセージタイプは PROGRESS でした。 このタイプのメッセージは、BACKUP DATABASE や LOAD TABLE などの実行時間が長いデータベースサーバ文によって生成されます。

関連情報

sqlany_register_callback(a_sqlany_connection *, a_sqlany_callback_type, SQLANY_CALLBACK_PARM) メソッド [44 ページ]

1.58 a_sqlany_native_type 列挙

サーバによって記述された値のネイティブ型の列挙。

構文

```
enum a_sqlany_native_type
```

メンバー

メンバー名	説明	値
DT_BIGINT	64 ビット符号付き整数。	608
DT_BINARY	2 バイトの長さフィールドを持つ可変長バイナリデータ。データを送信する場合は、長さフィールドに値を設定してください。データを送信するときの最大長は 32767 バイトです。データをフェッチする場合は、データベースサーバが長さフィールドに値を設定します。データをフェッチするときの最大長は 32765 バイトです。	524
DT_BIT	8 ビット符号付き整数。	624
DT_DATE	有効な日付データを含み、NULL で終了する文字列。	384
DT_DECIMAL	パック 10 進数 (独自フォーマット)。	484
DT_DOUBLE	8 バイト浮動小数点数。	480

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

メンバー名	説明	値
DT_FIXCHAR	CHAR 文字セット内のブランクが埋め込まれた固定長文字列。最大長は 32767 で、バイト単位で指定します。データは NULL で終了しません。	452
DT_FLOAT	4 バイト浮動小数点数。	482
DT_INT	32 ビット符号付き整数。	496
DT_LONGBINARY	長いバイナリデータ。	528
DT_LONGNVARCHAR	NCHAR 文字セット内の長い可変長文字列。	640
DT_LONGVARCHAR	CHAR 文字セット内の長い可変長文字列。	456
DT_NFIXCHAR	NCHAR 文字セット内のブランクが埋め込まれた固定長文字列。最大長は 32767 で、バイト単位で指定します。データは NULL で終了しません。	632
DT_NOTYPE	データ型なし。	0
DT_NSTRING	NCHAR 文字セット内の NULL で終了する文字列。データベースがブランクを埋め込まれた文字列で初期化されると、文字列にブランクが埋め込まれます。	628
DT_NVARCHAR	NCHAR 文字セット内の 2 バイトの長さフィールドを持つ可変長文字列。データを送信する場合は、長さフィールドに値を設定してください。データを送信するときの最大長は 32767 バイトです。データをフェッチする場合は、データベースサーバが長さフィールドに値を設定します。データをフェッチするときの最大長は 32765 バイトです。データは NULL で終了せず、ブランクも埋め込まれません。	636
DT_SMALLINT	16 ビット符号付き整数。	500
DT_STRING	CHAR 文字セット内の NULL で終了する文字列。データベースがブランクを埋め込まれた文字列で初期化されると、文字列にブランクが埋め込まれます。	460
DT_TIME	有効な時間データを含み、NULL で終了する文字列。	388
DT_TIMESTAMP	有効なタイムスタンプを含み、NULL で終了する文字列。	392
DT_TINYINT	8 ビット符号付き整数。	604
DT_UNSBIGINT	64 ビット符号なし整数です。	620
DT_UNSINT	32 ビット符号なし整数です。	612
DT_UNSSMALLINT	16 ビット符号なし整数です。	616

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

メンバー名	説明	値
DT_VARCHAR	CHAR 文字セット内の 2 バイトの長さフィールドを持つ可変長文字列。データを送信する場合は、長さフィールドに値を設定してください。データを送信するときの最大長は 32767 バイトです。データをフェッチする場合は、データベースサーバが長さフィールドに値を設定します。データをフェッチするときの最大長は 32765 バイトです。データは NULL で終了せず、ブランクも埋め込まれません。	448

備考

Embedded SQL データ型に対応する値タイプ。

関連情報

sqlany_get_column_info(a_sqlany_stmt *, sacapi_u32, a_sqlany_column_info *) メソッド [30 ページ]
a_sqlany_column_info 構造体 [61 ページ]

1.59 a_sqlany_bind_param 構造体

パラメータと準備文をバインドするために使用されるバインドパラメータ構造体。

構文

```
typedef struct a_sqlany_bind_param
```

メンバー

a_sqlany_bind_param のすべてのメンバー (継承されたメンバーも含みます) を以下に示します。

変数

変更子とタイプ	変数	説明
public a_sqlany_data_direction	direction	データの方向(入力、出力、入出力)。

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

変更子とタイプ	変数	説明
public a_sqlany_data_value	value	データの実際の値。
public char *	name	バインドパラメータの名前。sqlany_describe_bind_param() によってのみ使用されます。

備考

a_sqlany_bind_param 構造体の使用例については、インストールされている SQL Anywhere の `sdk¥dbcapi¥examples` ディレクトリにあるサンプルファイルを参照してください。

関連情報

sqlany_execute(a_sqlany_stmt *) メソッド [19 ページ]

1.60 a_sqlany_bind_param_info 構造体

現在のバインドパラメータに関する情報を取得します。

構文

```
typedef struct a_sqlany_bind_param_info
```

メンバー

_sqlany_bind_param_info のすべてのメンバー (継承されたメンバーも含みます) を以下に示します。

変数

変更子とタイプ	変数	説明
public char *	name	パラメータの名前へのポインタ。
public a_sqlany_data_direction	direction	パラメータの方向。
public a_sqlany_data_value	input_value	バインドされる入力値に関する情報。
public a_sqlany_data_value	output_value	バインドされる出力値に関する情報。
public a_sqlany_native_type	native_type	データベースのカラムのネイティブタイプ。

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

変更子とタイプ	変数	説明
public unsigned short	precision	精度。
public unsigned short	scale	位取り。
public size_t	max_size	このカラムに格納できるデータ値の最大サイズ。

備考

sqlany_get_bind_param_info() を使用して、この構造体にデータを移植できます。

a_sqlany_bind_param_info 構造体の使用例については、インストールされている SQL Anywhere の `sdk¥dbcapi¥examples` ディレクトリにあるサンプルファイルを参照してください。

関連情報

sqlany_execute(a_sqlany_stmt *) メソッド [19 ページ]

1.61 a_sqlany_column_info 構造体

カラムのメタデータ情報を返します。

構文

```
typedef struct a_sqlany_column_info
```

メンバー

a_sqlany_column_info のすべてのメンバー (継承されたメンバーも含みます) を以下に示します。

変数

変更子とタイプ	変数	説明
public char *	name	カラムの名前 (NULL で終了)。 結果セットのオブジェクトが解放されていない場合は、文字列を参照できます。

© 2016 SAP SE or an SAP affiliate company. All rights reserved.

変更子とタイプ	変数	説明
public a_sqlany_data_type	type	カラムのデータ型。
public a_sqlany_native_type	native_type	データベースのカラムのネイティブタイプ。
public unsigned short	precision	精度。
public unsigned short	scale	位取り。
public size_t	max_size	このカラムに格納できるデータ値の最大サイズ。
public sacapi_bool	nullable	カラムの値に NULL を指定できるかどうかを示します。
public char *	table_name	テーブルの名前 (NULL で終了)。 結果セットのオブジェクトが解放されていない場合は、文字列を参照できます。
public char *	owner_name	所有者の名前 (NULL で終了)。 結果セットのオブジェクトが解放されていない場合は、文字列を参照できます。
public sacapi_bool	is_bound	カラムがユーザバッファにバインドされているかどうかを示します。
public a_sqlany_data_value	binding	バインドされたカラムに関する情報。

備考

sqlany_get_column_info() を使用して、この構造体にデータを移植できます。

a_sqlany_column_info 構造体の使用例については、インストールされている SQL Anywhere の `sdk¥dbcapi¥examples` ディレクトリにあるサンプルファイルを参照してください。

- dbcapi_isql.cpp

1.62 a_sqlany_data_info 構造体

結果セット内のカラム値に関するメタデータ情報を返します。

構文

```
typedef struct a_sqlany_data_info
```


© 2016 SAP SE or an SAP affiliate company.All rights reserved.

メンバー

a_sqlany_data_info のすべてのメンバー (継承されたメンバーも含みます) を以下に示します。

変数

変更子とタイプ	変数	説明
public a_sqlany_data_type	type	カラムに格納されているデータ型
public sacapi_bool	is_null	最後にフェッチされたデータが NULL かどうかを示します。 このフィールドは、フェッチ操作が成功した後のみ有効です。
public size_t	data_size	フェッチ可能なバイトの総数。 このフィールドは、フェッチ操作が成功した後のみ有効です。

備考

sqlany_get_data_info() を使用して、フェッチ操作によって最後に取得されたデータに関する情報をこの構造体に移植できます。

a_sqlany_data_info 構造体の使用例については、インストールされている SQL Anywhere の `sdk¥dbcapi¥examples` ディレクトリにあるサンプルファイルを参照してください。

関連情報

sqlany_get_data_info(a_sqlany_stmt *, sacapi_u32, a_sqlany_data_info *) メソッド [32 ページ]

1.63 a_sqlany_data_value 構造体

データ値の属性に関する説明を返します。

構文

```
typedef struct a_sqlany_data_value
```


© 2016 SAP SE or an SAP affiliate company. All rights reserved.

メンバー

a_sqlany_data_value のすべてのメンバー (継承されたメンバーも含みます) を以下に示します。

変数

変更子とタイプ	変数	説明
public char *	buffer	ユーザが指定するデータバッファへのポインタ。
public size_t	buffer_size	バッファのサイズ。
public size_t *	length	バッファ内の有効なバイト数へのポインタ。この値は buffer_size 未満である必要があります。
public a_sqlany_data_type	type	データ型。
public sacapi_bool *	is_null	最後にフェッチされたデータが NULL かどうかを示すポインタ。
public sacapi_bool	is_address	バッファ値が実際の値へのポインタかどうかを示します。

備考

a_sqlany_data_value 構造体の使用例については、インストールされている SQL Anywhere の `sdk¥dbcapi¥examples` ディレクトリにあるサンプルファイルを参照してください。

- dbcapi_isql.cpp
- fetching_a_result_set.cpp
- send_retrieve_full_blob.cpp
- preparing_statements.cpp

1.64 SQLAnywhereInterface 構造体

SQL Anywhere C API インタフェース構造体。

構文

```
typedef struct SQLAnywhereInterface
```

メンバー

SQLAnywhereInterface のすべてのメンバー (継承されたメンバーも含みます) を以下に示します。

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

変数

変更子とタイプ	変数	説明
public void *	dll_handle	DLL ハンドル。
public int	initialized	初期化されているかどうかを識別するためのフラグ。
public void *	sqlany_init	sqlany_init() 関数へのポインタ。
public void *	sqlany_fini	sqlany_fini() 関数へのポインタ。
public void *	sqlany_new_connection	sqlany_new_connection() 関数へのポインタ。
public void *	sqlany_free_connection	sqlany_free_connection() 関数へのポインタ。
public void *	sqlany_make_connection	sqlany_make_connection() 関数へのポインタ。
public void *	sqlany_connect	sqlany_connect() 関数へのポインタ。
public void *	sqlany_disconnect	sqlany_disconnect() 関数へのポインタ。
public void *	sqlany_execute_immediate	sqlany_execute_immediate() 関数へのポインタ。
public void *	sqlany_prepare	sqlany_prepare() 関数へのポインタ。
public void *	sqlany_free_stmt	sqlany_free_stmt() 関数へのポインタ。
public void *	sqlany_num_params	sqlany_num_params() 関数へのポインタ。
public void *	sqlany_describe_bind_param	sqlany_describe_bind_param() 関数へのポインタ。
public void *	sqlany_bind_param	sqlany_bind_param() 関数へのポインタ。
public void *	sqlany_send_param_data	sqlany_send_param_data() 関数へのポインタ。
public void *	sqlany_reset	sqlany_reset() 関数へのポインタ。
public void *	sqlany_get_bind_param_info	sqlany_get_bind_param_info() 関数へのポインタ
public void *	sqlany_execute	sqlany_execute() 関数へのポインタ。
public void *	sqlany_execute_direct	sqlany_execute_direct() 関数へのポインタ。
public void *	sqlany_fetch_absolute	sqlany_fetch_absolute() 関数へのポインタ。
public void *	sqlany_fetch_next	sqlany_fetch_next() 関数へのポインタ。
public void *	sqlany_get_next_result	sqlany_get_next_result() 関数へのポインタ。
public void *	sqlany_affected_rows	sqlany_affected_rows() 関数へのポインタ。
public void *	sqlany_num_cols	sqlany_num_cols() 関数へのポインタ。

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

変更子とタイプ	変数	説明
public void *	sqlany_num_rows	sqlany_num_rows() 関数へのポインタ。
public void *	sqlany_get_column	sqlany_get_column() 関数へのポインタ
public void *	sqlany_get_data	sqlany_get_data() 関数へのポインタ。
public void *	sqlany_get_data_info	sqlany_get_data_info() 関数へのポインタ。
public void *	sqlany_get_column_info	sqlany_get_column_info() 関数へのポインタ。
public void *	sqlany_commit	sqlany_commit() 関数へのポインタ
public void *	sqlany_rollback	sqlany_rollback() 関数へのポインタ。
public void *	sqlany_client_version	sqlany_client_version() 関数へのポインタ。
public void *	sqlany_error	sqlany_error() 関数へのポインタ。
public void *	sqlany_sqlstate	sqlany_sqlstate() 関数へのポインタ。
public void *	sqlany_clear_error	sqlany_clear_error() 関数へのポインタ。
public void *	sqlany_init_ex	sqlany_init_ex() 関数へのポインタ。
public void *	sqlany_fini_ex	sqlany_fini_ex() 関数へのポインタ。
public void *	sqlany_new_connection_ex	sqlany_new_connection_ex() 関数へのポインタ。
public void *	sqlany_make_connection_ex	sqlany_make_connection_ex() 関数へのポインタ。
public void *	sqlany_client_version_ex	sqlany_client_version_ex() 関数へのポインタ。
public void *	sqlany_cancel	sqlany_cancel() 関数へのポインタ。
public void *	sqlany_register_callback	sqlany_register_callback() 関数へのポインタ。
public void *	sqlany_set_batch_size	sqlany_set_batch_size() 関数へのポインタ。
public void *	sqlany_set_param_bind_type	sqlany_set_param_bind_type() 関数へのポインタ。
public void *	sqlany_get_batch_size	sqlany_get_batch_size() 関数へのポインタ。
public void *	sqlany_set_rowset_size	sqlany_set_rowset_size() 関数へのポインタ。
public void *	sqlany_get_rowset_size	sqlany_get_rowset_size() 関数へのポインタ。
public void *	sqlany_set_column_bind_type	sqlany_set_column_bind_type() 関数へのポインタ。
public void *	sqlany_bind_column	sqlany_bind_column() 関数へのポインタ。

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

変更子とタイプ	変数	説明
public void *	sqlany_clear_column_bindings	sqlany_clear_column_bindings() 関数へのポインタ。
public void *	sqlany_fetched_rows	sqlany_fetched_rows() 関数へのポインタ。
public void *	sqlany_set_rowset_pos	sqlany_set_rowset_pos() 関数へのポインタ。

関連情報

sqlany_initialize_interface(SQLAnywhereInterface *, const char *) メソッド [37 ページ]

1.65 _sacapi_entry_ 変数

使用中のランタイム呼び出し規則 (Windows のみ)。

構文

```
#define _sacapi_entry_
```

1.66 SACAPI_ERROR_SIZE 変数

ビルドするバージョンをコマンドラインで指定しない場合、最新バージョンをビルドします。

構文

```
#define SACAPI_ERROR_SIZE
```

備考

最小エラーバッファサイズを返します。

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

1.67 SQLANY_API_VERSION_1 変数

バージョン 1 は、C/C++ API の初期バージョンです。

構文

```
#define SQLANY_API_VERSION_1
```

備考

この機能を使用するには、_SACAPI_VERSION を 1 以上として定義する必要があります。

1.68 SQLANY_API_VERSION_2 変数

バージョン 2 では "_ex" 関数、および要求をキャンセルする機能が追加されました。

構文

```
#define SQLANY_API_VERSION_2
```

備考

この機能を使用するには、_SACAPI_VERSION を 2 以上として定義する必要があります。

1.69 SQLANY_API_VERSION_3 変数

バージョン 3 より、"コールバック" 関数が導入されました。

構文

```
#define SQLANY_API_VERSION_3
```


© 2016 SAP SE or an SAP affiliate company.All rights reserved.

備考

この機能を使用するには、_SACAPI_VERSION を 3 以上として定義する必要があります。

1.70 SQLANY_API_VERSION_4 変数

バージョン 4 より、NCHAR サポートとワイド挿入が導入されました。

構文

```
#define SQLANY_API_VERSION_4
```

備考

この機能を使用するには、_SACAPI_VERSION を 4 以上として定義する必要があります。

1.71 SQLANY_CALLBACK 変数

コールバック関数のタイプ。

構文

```
#define SQLANY_CALLBACK
```


© 2016 SAP SE or an SAP affiliate company. All rights reserved.

2 このマニュアルの印刷、再生、および再配布

次の条件に従うかぎり、このマニュアルの全部または一部を使用、印刷、再生、配布することができます。

1. ここに示したものとそれ以外のすべての版権と商標の表示をすべてのコピーに含めること。
2. マニュアルに変更を加えないこと。
3. SAP 以外の人間がマニュアルの著者または情報源であるかのように示す一切の行為をしないこと。

ここに記載された情報は事前の通知なしに変更されることがあります。

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

重要免責事項および法的情報

コードサンプル

この文書に含まれるソフトウェアコード及び / 又はコードライン / 文字列 (「コード」) はすべてサンプルとしてのみ提供されるものであり、本稼動システム環境で使用することが目的ではありません。「コード」は、特定のコードの構文及び表現規則を分かりやすく説明及び視覚化することのみを目的としています。SAP は、この文書に記載される「コード」の正確性及び完全性の保証を行いません。更に、SAP は、「コード」の使用により発生したエラー又は損害が SAP の故意又は重大な過失が原因で発生させたものでない限り、そのエラー又は損害に対して一切責任を負いません。

アクセシビリティ

この SAP 文書に含まれる情報は、公開日現在のアクセシビリティ基準に関する SAP の最新の見解を表明するものであり、ソフトウェア製品のアクセシビリティ機能の確実な提供方法に関する拘束力のあるガイドラインとして意図されるものではありません。SAP は、この文書に関する一切の責任を明確に放棄するものです。ただし、この免責事項は、SAP の意図的な違法行為または重大な過失による場合は、適用されません。さらに、この文書により SAP の直接的または間接的な契約上の義務が発生することは一切ありません。

ジェンダーニュートラルな表現

SAP 文書では、可能な限りジェンダーニュートラルな表現を使用しています。文脈により、文書の読者は「あなた」と直接的な呼ばれ方をされたり、ジェンダーニュートラルな名詞 (例: 「販売員」又は「勤務日数」) で表現されます。ただし、男女両方を指すとき、三人称単数形の使用が避けられない又はジェンダーニュートラルな名詞が存在しない場合、SAP はその名詞又は代名詞の男性形を使用する権利を有します。これは、文書を分かりやすくするためです。

インターネットハイパーリンク

SAP 文書にはインターネットへのハイパーリンクが含まれる場合があります。これらのハイパーリンクは、関連情報を見いだすヒントを提供することが目的です。SAP は、この関連情報の可用性や正確性又はこの情報が特定の目的に役立つことの保証は行いません。SAP は、関連情報の使用により発生した損害が、SAP の重大な過失又は意図的な違法行為が原因で発生したものでない限り、その損害に対して一切責任を負いません。すべてのリンクは、透明性を目的に分類されています (http://help.sap.com/disclaimer を参照)。

© 2016 SAP SE or an SAP affiliate company.All rights reserved.

go.sap.com/registration/contact.html

© 2016 SAP SE or an SAP affiliate company.All rights reserved.
本書のいかなる部分も、SAP SE 又は SAP の関連会社の明示的な許可なくして、いかなる形式でも、いかなる目的にも複製又は伝送することはできません。本書に記載された情報は、予告なしに変更されることがあります。SAP SE 及びその頒布業者によって販売される一部のソフトウェア製品には、他のソフトウェアベンダーの専有ソフトウェアコンポーネントが含まれています。製品仕様は、国ごとに変わる場合があります。
これらの文書は、いかなる種類の表明又は保証もなしで、情報提供のみを目的として、SAP SE 又はその関連会社によって提供され、SAP 又はその関連会社は、これら文書に関する誤記脱落等の過失に対する責任を負うものではありません。SAP 又はその関連会社の製品及びサービスに対する唯一の保証は、当該製品及びサービスに伴う明示的保証がある場合に、これに規定されたものに限られます。本書のいかなる記述も、追加の保証となるものではありません。
本書に記載される SAP 及びその他の SAP の製品やサービス、並びにそれらの個々のロゴは、ドイツ及びその他の国における SAP SE（又は SAP の関連会社）の商標若しくは登録商標です。本書に記載されたその他すべての製品およびサービス名は、それぞれの企業の商標です。
商標に関する詳細の情報や通知については、http://www.sap.com/corporate-en/legal/copyright/index.epx をご覧ください。