Koala::ODBC

```
// ------------------------------------------------------
//【読み込みｸｴﾘ】を発行する。

    CResultSet* CStatement::ExecuteReadQuery( const wchar_t* p_read_query_, SQLINTEGER fetch_row_count
    {

        // ------------------
        // ｸｴﾘを発行する。

        SQLRETURN result = ::SQLExecDirectW( h_stmt, (SQLWCHAR*) p_read_query_, SQL_NTS ); // 文字列の

        if IsUnSuccess( result )
        {

#ifdef _DEBUG
            ODBC::ShowError( HT_Statement, h_stmt, L"読み込みｸｴﾘ発行時のｴﾗｰ", NULL ); // ｴﾗｰﾀﾞｲｱﾛｸﾞ
#endif
            return NULL;
        }

        // ------------------
        //【結果ｾｯﾄ】を作成する。

        CResultSet* p_res = new CResultSet( h_stmt, fetch_row_count_ ); // 列ｽｷｰﾏの読み取り、列ﾊﾞｯﾌｧの

        // ------------------

        return p_res;
    }


// ------------------------------------------------------
//【書き込みｸｴﾘ】を発行する。

    SQLINTEGER CStatement::ExecuteWriteQuery( const wchar_t* p_write_query_ ){ ... }
// ------------------------------------------------------
//【記述子ﾊﾝﾄﾞﾙ】を取得する。

// ・ｽﾃｰﾄﾒﾝﾄﾊﾝﾄﾞﾙが割り当てられると、同時に暗黙的に割り当てられる。( そして、ｽﾃｰﾄﾒﾝﾄﾊﾝﾄﾞﾙが解放されると
```

haseham

ODBC名前空間のｸﾗｽ

[CDriverManager] ... ODBCﾄﾞﾗｲﾊﾞﾏﾈｰｼﾞｬ

[CDataSource] ... ﾃﾞｰﾀｿｰｽ

[CStatement] ... ｽﾃｰﾄﾒﾝﾄ (SQL文)
 |
 +- [CProcedure] ... ﾌﾟﾘﾍﾟｱｰﾄﾞ・ｽﾃｰﾄﾒﾝﾄ
 +- [CCatalog] ... ｶﾀﾛｸﾞ

[CResultSet]　... 結果ｾｯﾄ (ｱｸｾｽ速度重視)
 |
 +- [CIndexSet] ... ｲﾝﾃﾞｯｸｽﾃｰﾌﾞﾙ
 +- [CAuthoritySet] ... 権限ﾃｰﾌﾞﾙ
 +- [CCatalogSet] ... ｶﾀﾛｸﾞﾃｰﾌﾞﾙ
 +- [CProcSet] ... ﾌﾟﾛｼｰｼﾞｬﾃｰﾌﾞﾙ
 +- [CProcColumnSet] ... ﾌﾟﾛｼｰｼﾞｬ列ﾃｰﾌﾞﾙ
 +- [CValueTypeSet] ... ﾃﾞｰﾀ型ﾃｰﾌﾞﾙ

[CBuffer] ... ﾊﾞｯﾌｧの基底ｸﾗｽ
 |
 +- [CColumnBuffer] ... 列ﾊﾞｯﾌｧ (行ｾｯﾄ)
 +- [CParamBuffer] ... ﾊﾟﾗﾒｰﾀﾊﾞｯﾌｧ

[CTable] ... ﾃｰﾌﾞﾙ (ﾒﾓﾘ節約重視)
 |
 +- [CIndexTable] ... ｲﾝﾃﾞｯｸｽﾃｰﾌﾞﾙ
 +- [CAuthorityTable] ... 権限ﾃｰﾌﾞﾙ
 +- [CCatalogTable] ... ｶﾀﾛｸﾞﾃｰﾌﾞﾙ
 +- [CProcTable] ... ﾌﾟﾛｼｰｼﾞｬﾃｰﾌﾞﾙ
 +- [CProcColumnTable] ... ﾌﾟﾛｼｰｼﾞｬ列ﾃｰﾌﾞﾙ
 +- [CValueTypeTable] ... ﾃﾞｰﾀ型ﾃｰﾌﾞﾙ

[CDescriptor] ... 記述子の基底ｸﾗｽ
 |
 +- [CBufferDesc] ... ﾊﾞｯﾌｧ系記述子の基底ｸﾗｽ
 | |

| +- [CColumnBufferDesc] ... 列ﾊﾞｯﾌｧの記述子
| +- [CParamBufferDesc] ... ﾊﾟﾗﾒｰﾀﾊﾞｯﾌｧの記述子
|
+- [CSchemaDesc] ... ｽｷｰﾏ系記述子の基底ｸﾗｽ
    |
    +- [CResultSetSchemaDesc] ... 結果ｾｯﾄのｽｷｰﾏ記述子
    +- [CQuerySchemaDesc] ... SQL文のｽｷｰﾏ記述子

・ｽｸﾛｰﾙｶｰｿﾙに対応している。
・ﾊﾞﾙｸ操作に対応している。
・ﾄﾗﾝｻﾞｸｼｮﾝに対応している。

MySQL向けに**WKB**に対応

構造体

LinearLing

Point

LineString

Polygon

MultiPoint

MultiLineString

MultiPolygon

Geometry

列挙体

GeometryType

ByteOrder

ﾛｺﾞとか作ってみた

日時を文字列に変換する

```
  //【日付値】の【文字列表現】を取得する。( 2014/01/01 ) [11]
Int32 GetDateText( const Date* p_date_, wchar_t* p_dest_array_ );

  //【日付値】の【文字列表現】を取得する。( 2014年01月01日 ) [12]
Int32 GetDateTextJ( const Date* p_date_, wchar_t* p_dest_array_ );

  //【日時値】の【文字列表現】を取得する。( 03:04:05 ) [9]
Int32 GetTimeText( const Time* p_time_, wchar_t* p_dest_array_ );

  //【日時値】の【文字列表現】を取得する。( 03時04分05秒 ) [10]
Int32 GetTimeTextJ( const Time* p_time_, wchar_t* p_dest_array_ );

  //【日時値】の【文字列表現】を取得する。( 2014/01/01 03:04:05 ) [20]
Int32 GetDateTimeText( const DateTime* p_datetime_, wchar_t* p_dest_array_ );

  //【日時値】の【文字列表現】を取得する。( 2014年01月01日 03時04分05秒 ) [22]
Int32 GetDateTimeTextJ( const DateTime* p_datetime_, wchar_t* p_dest_array_ );
```

環境属性と接続属性

環境属性	ODBCﾃﾞｰﾀ型	列挙体	説明
SQL_ATTR_CONNECTION_POOLING	SQLUINTEGER	ConnectPoolMode	接続ﾌﾟｰﾙﾓｰﾄﾞ
SQL_ATTR_CP_MATCH	SQLUINTEGER	ConnectPoolMatchLevel	接続ﾌﾟｰﾙﾏｯﾁﾚﾍﾞﾙ

接続属性	ODBCﾃﾞｰﾀ型	列挙体	説明
SQL_ATTR_TXN_ISOLATION	SQLINTEGER	IsolationLevel	ﾄﾗﾝｻﾞｸｼｮﾝ分離ﾚﾍﾞﾙ
SQL_ATTR_CONNECTION_DEAD	SQLINTEGER		接続中か否か
SQL_ATTR_ASYNC_DBC_FUNCTIONS_ENABLE	SQLUINTEGER		非同期実行が有効か否か
SQL_ATTR_LOGIN_TIMEOUT	SQLINTEGER		接続時の待機時間（秒単位）
SQL_AUTOCOMMIT	SQLUINTEGER		自動ｺﾐｯﾄが有効か否か。
SQL_ATTR_ACCESS_MODE	SQLINTEGER	AccessMode	ｱｸｾｽﾓｰﾄﾞ。（ 更新可能か ）
SQL_ATTR_CURRENT_CATALOG	文字列		ｶﾚﾝﾄ・ｶﾀﾛｸﾞ名
SQL_ATTR_DISCONNECT_BEHAVIOR	SQLUINTEGER	QuitMode	切断ﾓｰﾄﾞ
SQL_ATTR_ODBC_CURSORS	SQLUINTEGER	CursorEmulateMode	ｶｰｿﾙｴﾐｭﾚｰﾄﾓｰﾄﾞ
SQL_ATTR_QUIET_MODE	HWND		ﾀﾞｲｱﾛｸﾞの親ｳｨﾝﾄﾞｳ
SQL_ATTR_TRACE	SQLUINTEGER	TraceMode	ﾄﾚｰｽﾓｰﾄﾞ
SQL_ATTR_TRACEFILE	文字列		ﾄﾚｰｽ出力先のﾛｸﾞﾌｧｲﾙ名。
SQL_ATTR_TRANSLATE_LIB	文字列		ﾄﾗﾝｽﾚｰﾄﾗｲﾌﾞﾗﾘのﾌｧｲﾙ名。
SQL_ATTR_TRANSLATE_OPTION	SQLUINTEGER		ﾄﾗﾝｽﾚｰﾄﾗｲﾌﾞﾗﾘのDLLに渡すｵﾌﾟｼｮﾝ。
SQL_ATTR_ANSI_APP	SQLINTEGER	AppliCharType	ｱﾌﾟﾘ側の文字ｺｰﾄﾞがANSIか。

環境属性	Get	Set	MySQL
SQL_ATTR_CONNECTION_POOLING	○	○	×
SQL_ATTR_CP_MATCH	○	○	×

接続属性	Get	Set	MySQL
SQL_ATTR_TXN_ISOLATION	○	○	変更不可
SQL_ATTR_CONNECTION_DEAD	○	×	○
SQL_ATTR_ASYNC_DBC_FUNCTIONS_ENABLE	○	○	×
SQL_ATTR_LOGIN_TIMEOUT	○	○	×
SQL_AUTOCOMMIT	○	○	○
SQL_ATTR_ACCESS_MODE	○	○	○
SQL_ATTR_CURRENT_CATALOG	○	×	○
SQL_ATTR_DISCONNECT_BEHAVIOR	○	○	○
SQL_ATTR_ODBC_CURSORS	○	○	○
SQL_ATTR_QUIET_MODE	○	○	○
SQL_ATTR_TRACE	○	○	○
SQL_ATTR_TRACEFILE	○	○	○
SQL_ATTR_TRANSLATE_LIB	○	○	×
SQL_ATTR_TRANSLATE_OPTION	○	○	×
SQL_ATTR_ANSI_APP	○	○	×

ｽﾃｰﾄﾒﾝﾄ属性

ｽﾃｰﾄﾒﾝﾄ属性	旧名称	ODBCﾃﾞｰﾀ型	Koalaの列挙型
SQL_ATTR_MAX_LENGTH	SQL_MAX_LENGTH	SQLULEN	
SQL_ATTR_MAX_ROWS	SQL_MAX_ROWS	SQLULEN	
SQL_ATTR_ASYNC_ENABLE		SQLULEN	AsyncMode
SQL_ATTR_QUERY_TIMEOUT	SQL_QUERY_TIMEOUT	SQLULEN	
SQL_ATTR_RETRIEVE_DATA	SQL_RETRIEVE_DATA	SQLULEN	AutoFetchMode
SQL_ATTR_ENABLE_AUTO_IPD		SQLULEN	
SQL_ATTR_CURSOR_SCROLLABLE		SQLULEN	
SQL_ATTR_CURSOR_SENSITIVITY		SQLULEN	
SQL_ATTR_CONCURRENCY		SQLULEN	ConcurrencyMode
SQL_ATTR_CURSOR_TYPE	SQL_CURSOR_TYPE	SQLULEN	CursorType
SQL_ATTR_KEYSET_SIZE	SQL_KEYSET_SIZE	SQLULEN	
SQL_ATTR_SIMULATE_CURSOR	SQL_SIMULATE_CURSOR	SQLULEN	SimulateCursorMode
SQL_ATTR_USE_BOOKMARKS	SQL_USE_BOOKMARKS	SQLULEN	BookmarkMode
SQL_ATTR_FETCH_BOOKMARK_PTR		SQLLEN *	
SQL_ATTR_NOSCAN	SQL_NOSCAN	SQLULEN	ScanEscapeMode
SQL_ATTR_PARAM_BIND_OFFSET_PTR		SQLULEN *	
SQL_ATTR_PARAM_BIND_TYPE		SQLULEN	BindBufferDirection
SQL_ATTR_PARAM_OPERATION_PTR		SQLUSMALLINT*	ParamSkipModeに変換
SQL_ATTR_PARAM_STATUS_PTR		SQLUSMALLINT*	ResultParamStateに変換
SQL_ATTR_PARAMS_PROCESSED_PTR		SQLULEN *	
SQL_ATTR_PARAMSET_SIZE		SQLULEN	
SQL_ATTR_ROW_BIND_OFFSET_PTR		SQLULEN *	
SQL_ATTR_ROW_BIND_TYPE	SQL_BIND_TYPE	SQLULEN	BindBufferDirection
SQL_ATTR_ROW_NUMBER	SQL_ROW_NUMBER	SQLULEN	
SQL_ATTR_ROW_OPERATION_PTR		SQLUSMALLINT*	RowSkipModeに変換
SQL_ATTR_ROW_STATUS_PTR		SQLUSMALLINT*	ResultRowStateに変換
SQL_ATTR_ROWS_FETCHED_PTR		SQLULEN *	
SQL_ATTR_ROW_ARRAY_SIZE	SQL_ROWSET_SIZE	SQLULEN	
SQL_ATTR_APP_ROW_DESC		SQLHDESC	
SQL_ATTR_APP_PARAM_DESC		SQLHDESC	

SQL_ATTR_IMP_ROW_DESC		SQLHDESC	
SQL_ATTR_IMP_PARAM_DESC		SQLHDESC	

ｽﾃｰﾄﾒﾝﾄ属性	Statement	Procedure	MySQL
SQL_ATTR_MAX_LENGTH	○	継承	○
SQL_ATTR_MAX_ROWS	○	継承	○
SQL_ATTR_ASYNC_ENABLE	○	継承	設定無効
SQL_ATTR_QUERY_TIMEOUT	○	継承	設定無効
SQL_ATTR_RETRIEVE_DATA	○	継承	設定無効
SQL_ATTR_ENABLE_AUTO_IPD	使用しない	○	使用しないのみ
SQL_ATTR_CURSOR_SCROLLABLE	○	継承	○
SQL_ATTR_CURSOR_SENSITIVITY	○	継承	設定無効
SQL_ATTR_CONCURRENCY	○	継承	設定不可
SQL_ATTR_CURSOR_TYPE	○	継承	前方と静的のみ
SQL_ATTR_KEYSET_SIZE	○	継承	設定無効
SQL_ATTR_SIMULATE_CURSOR	○	継承	○
SQL_ATTR_USE_BOOKMARKS	○	継承	○
SQL_ATTR_FETCH_BOOKMARK_PTR	○	継承	○
SQL_ATTR_NOSCAN	使用しない	○	設定無効
SQL_ATTR_PARAM_BIND_OFFSET_PTR	使用しない	○	○
SQL_ATTR_PARAM_BIND_TYPE	使用しない	○	設定無効
SQL_ATTR_PARAM_OPERATION_PTR	使用しない	○	○
SQL_ATTR_PARAM_STATUS_PTR	使用しない	○	○
SQL_ATTR_PARAMS_PROCESSED_PTR	使用しない	○	○
SQL_ATTR_PARAMSET_SIZE	使用しない	○	○
SQL_ATTR_ROW_BIND_OFFSET_PTR	○	継承	○
SQL_ATTR_ROW_BIND_TYPE	○	継承	設定無効
SQL_ATTR_ROW_NUMBER	○	継承	設定不可
SQL_ATTR_ROW_OPERATION_PTR	○	継承	○
SQL_ATTR_ROW_STATUS_PTR	○	継承	○
SQL_ATTR_ROWS_FETCHED_PTR	○	継承	○
SQL_ATTR_ROW_ARRAY_SIZE	○	継承	○
SQL_ATTR_APP_ROW_DESC	×	×	設定無効

SQL_ATTR_APP_PARAM_DESC	×	○	設定無効
SQL_ATTR_IMP_ROW_DESC	×	×	設定不可
SQL_ATTR_IMP_PARAM_DESC	×	○	設定不可

ｽﾃｰﾄﾒﾝﾄ属性	取得ﾀｲﾐﾝｸﾞ	設定ﾀｲﾐﾝｸﾞ	ﾃﾞﾌｫﾙﾄ値
SQL_ATTR_MAX_LENGTH	いつでも可	いつでも可	0
SQL_ATTR_MAX_ROWS	いつでも可	いつでも可	0
SQL_ATTR_ASYNC_ENABLE	いつでも可	いつでも可	SQL_ASYNC_ENABLE_ON
SQL_ATTR_QUERY_TIMEOUT	いつでも可	いつでも可	0
SQL_ATTR_RETRIEVE_DATA	いつでも可	いつでも可	SQL_RD_ON
SQL_ATTR_ENABLE_AUTO_IPD	いつでも可	発行前	無効な値
SQL_ATTR_CURSOR_SCROLLABLE	いつでも可	発行前	SQL_NONSCROLLABLE
SQL_ATTR_CURSOR_SENSITIVITY	いつでも可	発行前	SQL_UNSPECIFIED
SQL_ATTR_CONCURRENCY	いつでも可	発行前	SQL_CONCUR_READ_ONLY
SQL_ATTR_CURSOR_TYPE	いつでも可	発行前	SQL_CURSOR_FORWARD_ONLY
SQL_ATTR_KEYSET_SIZE	いつでも可	いつでも可	0
SQL_ATTR_SIMULATE_CURSOR	いつでも可	発行前	SQL_SC_NON_UNIQUE
SQL_ATTR_USE_BOOKMARKS	いつでも可	発行前	SQL_UB_OFF
SQL_ATTR_FETCH_BOOKMARK_PTR	いつでも可	いつでも可	NULL
SQL_ATTR_NOSCAN	いつでも可	いつでも可	SQL_NOSCAN_ON
SQL_ATTR_PARAM_BIND_OFFSET_PTR	いつでも可	いつでも可	NULL
SQL_ATTR_PARAM_BIND_TYPE	いつでも可	発行前	SQL_PARAM_BIND_BY_COLUMN
SQL_ATTR_PARAM_OPERATION_PTR	いつでも可	いつでも可	NULL
SQL_ATTR_PARAM_STATUS_PTR	いつでも可	いつでも可	NULL
SQL_ATTR_PARAMS_PROCESSED_PTR	いつでも可	いつでも可	NULL
SQL_ATTR_PARAMSET_SIZE	いつでも可	いつでも可	1
SQL_ATTR_ROW_BIND_OFFSET_PTR	いつでも可	いつでも可	NULL
SQL_ATTR_ROW_BIND_TYPE	いつでも可	発行前	SQL_BIND_BY_COLUMN
SQL_ATTR_ROW_NUMBER	いつでも可	いつでも可	Fetch後に1
SQL_ATTR_ROW_OPERATION_PTR	いつでも可	いつでも可	NULL
SQL_ATTR_ROW_STATUS_PTR	いつでも可	いつでも可	NULL
SQL_ATTR_ROWS_FETCHED_PTR	いつでも可	いつでも可	NULL

SQL_ATTR_ROW_ARRAY_SIZE	いつでも可	いつでも可	1
SQL_ATTR_APP_ROW_DESC	ﾊﾞｲﾝﾄﾞ後	ー	自動割り当て
SQL_ATTR_APP_PARAM_DESC	ﾊﾞｲﾝﾄﾞ後	ー	自動割り当て
SQL_ATTR_IMP_ROW_DESC	発行後	ー	自動割り当て
SQL_ATTR_IMP_PARAM_DESC	SQLｾｯﾄ後	ー	自動割り当て

基本的なｽﾃｰﾄﾒﾝﾄ属性	
SQL_ATTR_MAX_LENGTH	列値の最大ﾃﾞｰﾀｻｲｽﾞ。
SQL_ATTR_MAX_ROWS	結果ｾｯﾄの最大行数。
SQL_ATTR_ASYNC_ENABLE	非同期実行を有効にするか否か。
SQL_ATTR_QUERY_TIMEOUT	ｸｴﾘ発行後の待機時間。(秒単位)
SQL_ATTR_RETRIEVE_DATA	Fetch時に移動のみ行うか否か。
SQL_ATTR_ENABLE_AUTO_IPD	SQL文の記述子を読み取るか否か。

ｶｰｿﾙ関連	
SQL_ATTR_CURSOR_SCROLLABLE	ｽｸﾛｰﾙｶｰｿﾙを有効にするか否か。
SQL_ATTR_CURSOR_SENSITIVITY	他のｶｰｿﾙによる変更を反映するか否か。
SQL_ATTR_CONCURRENCY	ｶｰｿﾙの同時制御ｵﾌﾟｼｮﾝ。ﾛｯｸするかとか。
SQL_ATTR_CURSOR_TYPE	使用するｶｰｿﾙの種類。(前方、ｷｰｾｯﾄ、動的、静的)
SQL_ATTR_KEYSET_SIZE	ｷｰｾｯﾄのｻｲｽﾞ。
SQL_ATTR_SIMULATE_CURSOR	ｶｰｿﾙ行への変更操作を、対応するﾕﾆｰｸ行にのみ行うか。

ﾌﾞｯｸﾏｰｸ関連	
SQL_ATTR_USE_BOOKMARKS	ﾌﾞｯｸﾏｰｸを有効にするか否か。
SQL_ATTR_FETCH_BOOKMARK_PTR	ﾌﾞｯｸﾏｰｸ変数へのﾎﾟｲﾝﾀ。

ﾊﾟﾗﾒｰﾀﾊﾞｯﾌｧ関連	
SQL_ATTR_NOSCAN	ﾌﾟﾘﾍﾟｱｰﾄﾞ・ｽﾃｰﾄﾒﾝﾄに対するESCAPE節の読み取りを行うか否か。
SQL_ATTR_PARAM_BIND_OFFSET_PTR	ﾊﾟﾗﾒｰﾀﾊﾞｯﾌｧ上のﾊﾞｲﾝﾄﾞ位置。
SQL_ATTR_PARAM_BIND_TYPE	ﾊﾟﾗﾒｰﾀﾊﾞｯﾌｧのﾊﾞｲﾝﾄﾞ方向。(列方向か、行方向)
SQL_ATTR_PARAM_OPERATION_PTR	行ｾｯﾄに対するﾊﾞﾙｸ操作ｽｷｯﾌﾟﾌﾗｸﾞ配列へのﾎﾟｲﾝﾀ。
SQL_ATTR_PARAM_STATUS_PTR	ﾊﾟﾗﾒｰﾀﾊﾞｯﾌｧの行状況配列へのﾎﾟｲﾝﾀ。

SQL_ATTR_PARAMS_PROCESSED_PTR	処理されたﾊﾟﾗﾒｰﾀｾｯﾄ数を受け取る変数へのﾎﾟｲﾝﾀ。
SQL_ATTR_PARAMSET_SIZE	ﾊﾟﾗﾒｰﾀの総数。

列ﾊﾞｯﾌｧ関連	
SQL_ATTR_ROW_BIND_OFFSET_PTR	列ﾊﾞｯﾌｧ上のﾊﾞｲﾝﾄﾞ位置。
SQL_ATTR_ROW_BIND_TYPE	列ﾊﾞｯﾌｧのﾊﾞｲﾝﾄﾞ方向。(列方向か、 行方向)
SQL_ATTR_ROW_NUMBER	結果ｾｯﾄ上における現在のｶﾚﾝﾄ行の行番号。
SQL_ATTR_ROW_OPERATION_PTR	ﾊﾟﾗﾒｰﾀｾｯﾄに対する処理ｽｷｯﾌﾟﾌﾗｸﾞ配列へのﾎﾟｲﾝﾀ。
SQL_ATTR_ROW_STATUS_PTR	列ﾊﾞｯﾌｧの行状況配列へのﾎﾟｲﾝﾀ。
SQL_ATTR_ROWS_FETCHED_PTR	取り出された行ｾｯﾄ数を受け取る変数へのﾎﾟｲﾝﾀ。
SQL_ATTR_ROW_ARRAY_SIZE	Fetchで取り出す行数。

記述子ﾊﾝﾄﾞﾙ関連	
SQL_ATTR_APP_ROW_DESC	列ﾊﾞｯﾌｧの記述子ﾊﾝﾄﾞﾙ。
SQL_ATTR_APP_PARAM_DESC	ﾊﾟﾗﾒｰﾀﾊﾞｯﾌｧの記述子ﾊﾝﾄﾞﾙ。
SQL_ATTR_IMP_ROW_DESC	結果ｾｯﾄの記述子ﾊﾝﾄﾞﾙ。
SQL_ATTR_IMP_PARAM_DESC	ﾌﾟﾘﾍﾟｱｰﾄﾞ・ｽﾃｰﾄﾒﾝﾄの記述子ﾊﾝﾄﾞﾙ。

記述子ﾌｨｰﾙﾄﾞ

列名		
記述子ﾌｨｰﾙﾄﾞ	列挙体のﾒﾝﾊﾞ名	説明
SQL_DESC_CATALOG_NAME	DRFT_CatalogName	ｶﾀﾛｸﾞ名
SQL_DESC_SCHEMA_NAME	DRFT_SchemaName	ｽｷｰﾏ名
SQL_DESC_BASE_TABLE_NAME	DRFT_BaseTableName	基本ﾃｰﾌﾞﾙ名
SQL_DESC_BASE_COLUMN_NAME	DRFT_BaseColumnName	基本列名
SQL_DESC_NAME	DRFT_Alias	別名or列名
SQL_DESC_LABEL	DRFT_Label	ﾗﾍﾞﾙ
SQL_DESC_UNNAMED	DRFT_UnColumnName	別名or列名か否か
SQL_DESC_TABLE_NAME	DRFT_TableName	列の属するﾃｰﾌﾞﾙ名
SQL_DESC_TYPE	DRFT_Type	SQLﾃﾞｰﾀ型
SQL_DESC_TYPE_NAME	DRFT_TypeName	SQLﾃﾞｰﾀ型名
SQL_DESC_LOCAL_TYPE_NAME	DRFT_LocalTypeName	DB固有のﾃﾞｰﾀ型名
SQL_DESC_CONCISE_TYPE	DRFT_SimpleDateType	日時期間系の簡略ﾃﾞｰﾀ型

列のｻｲｽﾞ・精度		
記述子ﾌｨｰﾙﾄﾞ	列挙体のﾒﾝﾊﾞ名	説明
SQL_DESC_PRECISION	DRFT_Precision	文字数or要素数or有効桁数
SQL_DESC_DISPLAY_SIZE	DRFT_DisplaySize	文字列表記時の文字数
SQL_DESC_SCALE	DRFT_Scale	実数の小数部のｹﾀ数
SQL_DESC_FIXED_PREC_SCALE	DRFT_HasFixedScale	固定少数部があるか否か
SQL_DESC_NUM_PREC_RADIX	DRFT_Radix	基数
SQL_DESC_DATETIME_INTERVAL_CODE	DRFT_IntervalCode	日時型のｻﾌﾞｺｰﾄﾞ
SQL_DESC_DATETIME_INTERVAL_PRECISION	DRFT_IntervalPrecision	日時期間型か否か

列の特性		
記述子ﾌｨｰﾙﾄﾞ	列挙体のﾒﾝﾊﾞ名	説明
SQL_DESC_UNSIGNED	DRFT_IsUnsignedValue	符号なしであるか否か
SQL_DESC_NULLABLE	DRFT_AceeptNull	NULL値を許容するか否か
SQL_DESC_CASE_SENSITIVE	DRFT_CaseAlpha	大文字小文字を区別するか否か
SQL_DESC_AUTO_UNIQUE_VALUE	DRFT_IsUniqueValue	ﾕﾆｰｸ列であるか否か

SQL_DESC_UPDATABLE	DRFT_CanUpdateValue	更新可能な列であるか否か
SQL_DESC_SEARCHABLE	DRFT_CompareSpec	検索・比較が可能な列であるか否か

ﾊﾞｯﾌｧ		
記述子ﾌｨｰﾙﾄﾞ	列挙体のﾒﾝﾊﾞ名	説明
SQL_DESC_PARAMETER_TYPE	DRFT_ParamType	ﾊﾟﾗﾒｰﾀﾀｲﾌﾟ（ 入力、出力、入出力 ）
SQL_DESC_DATA_PTR	DRFT_BufferPtr	ﾃﾞｰﾀを格納するﾊﾞｯﾌｧへのﾎﾟｲﾝﾀ
SQL_DESC_LENGTH	DRFT_Length	ﾊﾞｯﾌｧｻｲｽﾞ
SQL_DESC_OCTET_LENGTH	DRFT_MaxLength	ﾃﾞｰﾀ長の最大長
SQL_DESC_OCTET_LENGTH_PTR	DRFT_DataLengthPtr	ﾃﾞｰﾀ長を受け取るﾊﾞｯﾌｧへのﾎﾟｲﾝﾀ
SQL_DESC_INDICATOR_PTR	DRFT_IsNullValuePtr	NULL値ﾌﾗｸﾞを受け取るﾊﾞｯﾌｧへのﾎﾟｲﾝﾀ

その他		
記述子ﾌｨｰﾙﾄﾞ	列挙体のﾒﾝﾊﾞ名	説明
SQL_DESC_LITERAL_PREFIX	DRFT_LiteralBeginChar	ﾘﾃﾗﾙの開始記号
SQL_DESC_LITERAL_SUFFIX	DRFT_LiteralEndChar	ﾘﾃﾗﾙの終了記号

記述子ﾍｯﾀﾞ

記述子ﾍｯﾀﾞ	列挙体	説明
SQL_DESC_ALLOC_TYPE	DHFT_AllocType	暗黙的な記述子ﾊﾝﾄﾞﾙか否か
SQL_DESC_BIND_TYPE	DHFT_BindType	ﾊﾞｯﾌｧのﾊﾞｲﾝﾄﾞ方向
SQL_DESC_ARRAY_SIZE	DHFT_ArraySize	Fetchで取り出す行数
SQL_DESC_COUNT	DHFT_MostColumnNum	最大列番号（間に欠番あり）
SQL_DESC_ARRAY_STATUS_PTR	DHFT_ArrayStatusPtr	行状況配列へのﾎﾟｲﾝﾀ
SQL_DESC_ROWS_PROCESSED_PTR	DHFT_RowsProcessedPtr	処理した行数変数へのﾎﾟｲﾝﾀ
SQL_DESC_BIND_OFFSET_PTR	DHFT_BindOffsetPtr	ﾊﾞｲﾝﾄﾞｵﾌｾｯﾄ変数へのﾎﾟｲﾝﾀ

連続しない通番を連続させる

ODBCのAPIには、
SQLﾃﾞｰﾀ型の定数が定義されていますが、
値は連続しておらず、ﾃｰﾌﾞﾙ化しょうとすると
そこそこ大きな配列となります。

SQLﾃﾞｰﾀ型	定数値
Char	1
VarChar	12
LongVarChar	-1
WideChar	-8
VarWideChar	-9
LongVarWideChar	-10
Binary	-2
VarBinary	-4
Decimal	3
Numeric	2
Bit	-7
TinyInt	-6
SmallInt	5
Int	4
BigInt	-5
Real	7
Float	6
Double	8
Date	91
Time	92
TimeStamp	93
Month	102
Year	101
YearToMonth	107
Day	103

Hour	104
Minute	105
Second	106
DayToHour	108
DayToMinute	109
DayToSecond	110
HourToMinute	111
HourToSecond	112
MinuteToSecond	113
GUID	-11

これをなるべく減らす方法はないものかと考えた末、
次の方法に落ち着きました。

```
void view( const char* p_name_, const int db_type_ )
{

   // 負数を消すために、最低値(-12)の絶対値を加算している。
   BYTE plus12 = db_type_ + 12;

   BYTE up_u4bit = (plus12 >> 3) & 15; // 正数部の上位4bitを取り出す。
   BYTE down_3bit = plus12 & 7; // 下位3bitを取り出す。

   // 次に、正数部上位4bitごとに、
   // ﾃｰﾌﾞﾙの要素番号のｵﾌｾｯﾄを格納しておく。(ｵﾌｾｯﾄﾃｰﾌﾞﾙ)


    char offset_table [] = { -1, // [0]
      7, // [1]
    15, // [2]
    20, // [3]
      0, // [4] 未使用
      0, // [5] 未使用
```

```
      0, // [6] 未使用
      0, // [7] 未使用
      0, // [8] 未使用
      0, // [9] 未使用
      0, // [10] 未使用
      0, // [11] 未使用
     14, // [12]
     22, // [13]
     23, // [14]
     31 }; // [15]

    int offset = offset_table[ up_u4bit ]; // ﾃｰﾌﾞﾙ上でのｵﾌｾｯﾄ値を取り出す。
    int table_index = offset + down_3bit; // ﾃｰﾌﾞﾙ配列の要素番号を求める。
    printf( "[%s] i = %d \n", name_, table_index ); // ｺﾝｿｰﾙに出力する。
}
```

```
int main(void)
{

 view("Char", 1 ); //
 view("VarChar", 12 ); //
 view("LongVarChar", -1 ); //
 view("WideChar", -8 ); //
 view("VarWideChar", -9 ); //
 view("LongVarWideChar", -10 ); //
 view("Binary", -2 ); //
 view("VarBinary", -4 ); //
 view("Decimal", 3 ); //
 view("Numeric", 2 ); //
 view("Bit", -7 ); //
 view("TinyInt",-6  ); //
 view("SmallInt", 5 ); //
 view("Int", 4 ); //
 view("BigInt", -5 ); //
 view("Real", 7 ); //
 view("Float", 6 ); //
```

```
 view("Double", 8 ); //
 view("Date", 91 ); //
 view("Time", 92 ); //
 view("TimeStamp",  93); //
 view("Month", 102 ); //
 view("Year", 101 ); //
 view("YearToMonth", 107 ); //
 view("Day", 103 ); //
 view("Hour", 104 ); //
 view("Minute", 105 ); //
 view("Second", 106 ); //
 view("DayToHour", 108 ); //
 view("DayToMinute", 109 ); //
 view("DayToSecond", 110 ); //
 view("HourToMinute", 111 ); //
 view("HourToSecond", 112 ); //
 view("MinuteToSecond", 113 ); //
 view("GUID", -11 ); //

 return 0;
}
```

これを出力した結果がこちら。↓

```
[GUID] i = 0
[LongVarWideChar] i = 1
[VarWideChar] i = 2
[WideChar] i = 3
[Bit] i = 4
[TinyInt] i = 5
[BigInt] i = 6
[VarBinary] i = 7
```

[Binary] i = 9
[LongVarChar] i = 10

[Char] i = 12
[Numeric] i = 13
[Decimal] i = 14
[Int] i = 15
[SmallInt] i = 16
[Float] i = 17
[Real] i = 18
[Double] i = 19
[VarChar] i = 20
Date] i = 21
[Time] i = 22
[TimeStamp] i = 23
[Year] i = 24
[Month] i = 25
[Day] i = 26
[Hour] i = 27
[Minute] i = 28
[Second] i = 29
[YearToMonth] i = 30
[DayToHour] i = 31
[DayToMinute] i = 32
[DayToSecond] i = 33
[HourToMinute] i = 34
[HourToSecond] i = 35
[MinuteToSecond] i = 36

まあ正確にはこれは、
わかりやすいように、並べなおしたものなんですが、
かなり圧縮されており、ほとんど連番になっていることがわかると思います。

ｵﾌｾｯﾄをずらすことによって、

空き要素が生じないように工夫されています。

ｵﾌｾｯﾄﾃｰﾌﾞﾙが必要になりますが、
ﾃｰﾌﾞﾙを常駐させておくには妥当なｻｲｽﾞに
収まったのではないかと思います。

ﾋﾞｯﾄ演算を使えば、これをさらに圧縮することもできます。