

# Action Zen から SQL Server への 移行ガイド Magic xpa



OUTPERFORM THE FUTURE™

本マニュアルに記載の内容は、将来予告なしに変更することがあります。これらの情報について MSE (Magic Software Enterprises Ltd.) および MSJ (Magic Software Japan K.K.) は、いかなる責任も負いません。

本書の内容につきましては、万全を期して作成していますが、万一誤りや不正確な記述があったとしても、MSE および MSJ はいかなる責任、債務も負いません。

MSE および MSJ は、この製品の商業価値や特定の用途に対する適合性の保証を含め、この製品に関する明示的、あるいは黙示的な保証は一切していません。

本書に記載のソフトウェアは、製品の使用許諾契約書に記載の条件に同意をされたライセンス所有者に対してのみ供給されるものです。同ライセンスの許可する条件のもとでのみ、使用または複製することが許されます。当該ライセンスが特に許可している場合を除いては、いかなる媒体へも複製することはできません。

ライセンス所有者自身の個人使用目的で行う場合を除き、MSE または MSJ の書面による事前の許可なしでは、いかなる条件下でも、本書のいかなる部分も、電子的、機械的、撮影、録音、その他のいかなる手段によっても、コピー、検索システムへの記憶、電送を行うことはできません。

サードパーティ各社商標の引用は、MSE および MSJ の製品に対する互換性に関する情報提供のみを目的としてなされるものです。

本書において、説明のためにサンプルとして引用されている会社名、製品名、住所、人物は、特に断り書きのない限り、すべて架空のものであり、実在のものについて言及するものではありません。

Magic は Magic Software Enterprises Ltd. のイスラエルその他の国での商標または登録商標です。

Magic xpa Enterprise Studio、Magic xpa Enterprise Client、Magic xpa Enterprise Server、Magic xpa RIA Server、および Magic xpa WebClient Server は Magic Software Japan K.K. の商標です。

Actian、Actian Zen は、Actian Corporation およびその子会社の商標または登録商標です。

Microsoft、SQL Server は、Microsoft Corporation の登録商標です。また、Windows は Microsoft Corporation の商標です。

一般に、会社名、製品名は各社の商標または登録商標です。

MSE および MSJ は、本製品の使用またはその使用によってもたらされる結果に関する保証や告知は一切していません。この製品のもたらす結果およびパフォーマンスに関する危険性は、すべてユーザーが責任を負うものとします。

この製品を使用した結果、または使用不可能な結果生じた間接的、偶発的、副次的な損害（営利損失、業務中断、業務情報の損失などの損害も含む）に関し、事前に損害の可能性が勧告されていた場合であっても、MSE および MSJ、その管理者、役員、従業員、代理人は、いかなる場合にも一切責任を負いません。

**初版** 2026年2月6日

**第2版** 2026年3月2日

Copyright 2026 Magic Software Enterprises Ltd. and Magic Software Japan K.K. All rights reserved.

<b>1. はじめに</b> .....	<b>1</b>
<b>2. SQL Server のセットアップ</b> .....	<b>2</b>
<b>2.1. SQL Server のインストール</b> .....	<b>2</b>
2.1.1. 認証モード .....	2
2.1.2. OLE DB Driver .....	2
2.1.3. SQL Server Management Studio (SSMS) .....	2
<b>2.2. データベースのセットアップ</b> .....	<b>3</b>
2.2.1. ログインユーザ .....	3
2.2.2. データベースの作成 .....	3
2.2.3. 照合順序 .....	3
2.2.4. データベースのユーザとアクセス権限の設定 .....	3
<b>2.3. xpa のセットアップ</b> .....	<b>4</b>
2.3.1. データベースゲートウェイのインストール .....	4
2.3.2. Magic.ini の設定、または設定画面 .....	4
<b>3. プロジェクトの変更</b> .....	<b>7</b>
<b>3.1 テーブル定義</b> .....	<b>7</b>
3.1.1. データベースの変更 .....	7
3.1.2. データソース名 .....	7
3.1.3. データソース特性 / オーナ名 .....	8
3.1.4. データソース特性 / デフォルトの位置 .....	8
3.1.5. データソース特性 / テーブルの存在チェック .....	8
3.1.6. カラム / DB カラム名 .....	9
3.1.7. カラム / 文字型 .....	9
3.1.8. カラム / UNICODE 型項目 .....	10
3.1.9. カラム / 数値型項目 .....	10
3.1.10. カラム / 論理型項目 .....	10
3.1.11. カラム / 日付型項目 .....	10
3.1.12. カラム / 時刻型項目 .....	11
3.1.13. カラム / BLOB 型項目 .....	12
3.1.14. インデックス / 重複不可インデックスの作成 .....	12
3.1.15. インデックス / DB インデックス名 .....	12
3.1.16. インデックス / タイプ .....	13
3.1.17. インデックス / セグメントのサイズ .....	13
<b>3.2. プログラム</b> .....	<b>14</b>
3.2.1. ロック方式とトランザクション開始 .....	14
3.2.2. 重複不可データのチェック .....	16
3.2.3. レコード登録 .....	16
3.2.4. データソース名の式 .....	17
3.2.5. 関数 .....	17
<b>4. SQL インターフェースの利用</b> .....	<b>19</b>
4.1. データ抽出 .....	19
4.2. リンク結合 .....	20
4.3. SQL コマンド .....	20
4.4. プログラムのソート .....	21
4.5. 日付型と時刻型の範囲 .....	22
4.6. NULL 値 .....	22
4.7. その他 .....	23



# 1. はじめに

本書では、Actian Zen(旧 Btrieve, または Actian Zen) のデータベースに対する ISAM アクセスのアプリケーションから、Microsoft SQL Server (以下 SQL Server) 対応アプリケーションに移行するための基本的なポイントを整理します。

この文書では、移行以外に関する説明は割愛しますので、詳細な仕様に関してはリファレンス等で確認してください。

ここに書かれた Actian Zen の機能は、Magic xpa(以下 xpa) アプリケーションで提供された Btrieve ゲートウェイを通じたアクセスを行った場合に限られます。Actian Zen のリレーショナルアクセスによる動作については対象外とさせていただきます。

Magic xpa のバージョンは、4.11.1 を対象としています。データベースシステムは、動作環境に準じたバージョンを対象としています。

## 2. SQL Server のセットアップ

SQL Server データベース対応のアプリケーションを開発、運用を行う場合、事前に SQL Server アーキテクチャに関する基本知識、および SQL コマンドを習得しておく必要があります。また、SQL Server Management Studio、またはそれと同等の機能を有するデータベース管理アプリケーションの操作等にも慣れておく必要があります。

Magic xpa については、インストーラを実行して MS-SQL ゲートウェイを追加します。

### 2.1. SQL Server のインストール

アプリケーションの種類やユーザ数、規模によって、SQL Server の適切な製品を選択してください。ここではインストール時の xpa に関する設定を説明します。

#### 2.1.1. 認証モード

SQL Server のインストール時に認証モードの選択があります。

Windows 認証モードは、ログインするクライアントのアカウントが SQL Server のアカウント名になります。この場合、xpa の [設定/データベース/データベース特性] のユーザ ID とパスワードの設定は無効です。

混合モードは、Windows 認証モードに加えて、SQL Server 認証が有効になります。

この場合、xpa の [設定/データベース/データベース特性] のユーザ ID とパスワードを設定した場合は、SQL Server に登録したログインアカウントがユーザ名になります。ユーザ ID とパスワードの設定をしない場合は Windows 認証として動作するため、Windows のログインユーザ名が SQL Server のユーザ名になります。

#### 2.1.2. OLE DB Driver

xpa 製品をインストールするマシンには、OLE DB Driver を追加してインストールします。

インストーラは、Microsoft の公式サイトから入手してください。

マシンの Windows OS が x64 の場合は x64 および Arm64 のインストーラ、x86(32bit) の場合は x86 のインストーラを実行します。

2026 年 1 月現在、Microsoft OLE DB Driver 19 for SQL Server と Microsoft OLE DB Driver 18 for SQL Server のメジャーバージョンが公開されています。xpa のデフォルトでセットアップされるゲートウェイ (mgmssql.dll) は Microsoft OLE DB Driver 18 for SQL Server に対応しています。

Microsoft OLE DB Driver 19 for SQL Server の環境で xpa を使う場合、Magic.ini のゲートウェイの設定 (後述) を mgmssql19.dll に変更します。(※ Magic xpa 4.11.1 以降)

(「2.3.2. Magic.ini の設定、または設定画面」参照)

#### 2.1.3. SQL Server Management Studio (SSMS)

SQL Server の新規データベースの作成ユーザ管理、SQL Statement の実行等を行う管理ツールです。

データベースをセットアップするために有効なツールなので、Server インストール後にインストールします。

## 2.2. データベースのセットアップ

Actian Zen の ISAM アクセスでは、データベースの各テーブルは OS 上のファイルに相当するため、Actian Zen は Windows のユーザとフォルダまたはファイルの権限を管理します。

SQL Server では OS 上のファイルは「データベース」であり、その中で SQL Server がアプリケーションのテーブルを管理します。そのため、データベース管理者が先にデータベースを作成し、データベースのテーブルにアクセスするユーザとそのアクセス権限を設定します。

データベースの操作はコマンドプロンプトから実行することも可能ですが、一般的に SQL Server Management Studio(SSMS)で行います。

### 2.2.1. ログインユーザ

「データベース」へのアクセスユーザを作成する前に SQL Server システムへのログインユーザを作成します。SSMS より管理者アカウントでログインし、[セキュリティ/ログイン] より新規ユーザを作成します。

### 2.2.2. データベースの作成

アプリケーションのデータベースには、データベースのテーブル、インデック、アクセスするユーザアカウント、アクセス権限、プロシージャ等が含まれます。

データベース作成の前に Actian Zen の各テーブルのレコードサイズ、レコード総数、一定期間で登録するレコード数などを収集し、それに基づく情報整理を行った後にデータベースのファイルサイズの設定を行うことを推奨します。

### 2.2.3. 照合順序



プログラムのデータ抽出処理とソート処理は、SQL Server の「照合順序」に影響されます。

日本語環境のデフォルトでは「JAPANESE\_CI\_AS」が設定されますが、この場合大文字と小文字、ひらがなとカタカナの区別がありません。したがって異なる文字種の範囲大小で絞り込む、あるいはインデックスで文字型項目の順序で表示すると、Actian Zen とは異なるデータが取得されることがあります。例えば「あ」と「ア」が同じ文字として処理されるので、「あ」で抽出したデータに「ア」が混在することがあります。

Actian Zen と同等の動作にするには、データベースの [オプション/照合順序] を「JAPANESE\_BIN2」に設定します。

### 2.2.4. データベースのユーザとアクセス権限の設定

データベースのテーブルにアクセスするためのユーザとアクセス権限を設定します。

データベースのユーザは、ログインユーザと紐付いたユーザの設定であり、ユーザ毎にデータ照会 (Select)、作成 (Insert)、修正 (Update)、削除 (Delete) の各処理、あるいはテーブルの作成 (Create)、削除 (Drop) の権限を設定します。

各権限をまとめたロールを設定することもできます。

#	書式	ロール
1	照会のみ	db_datareader
2	レコードの作成、修正、削除	db_datawriter
3	テーブルの作成、変更、削除	db_ddladmin
4	すべての権限	db_owner

## 2.3.xpa のセットアップ

### 2.3.1. データベースゲートウェイのインストール

Magic xpa のインストーラを実行して、MSSQL ゲートウェイを追加します。

インストールフォルダの配下の gateway フォルダに mgmssql.dll または mgmssql19.dll がゲートウェイのファイルです。

Actian Zen と同様、実行エンジンにより x86(32 ビット) と x64(64 ビット) ではファイルが異なります。

### 2.3.2. Magic.ini の設定、または設定画面

詳細はリファレンスを参照してください。SQL Server を利用するための項目を説明します。

#### **MAGIC.INI の Gateways セクション**

デフォルトでは以下のように設定されます。

```
[MAGIC_GATEWAYS]
MGDB20=Gateways¥MGMSSQL. dll
```

Microsoft OLE DB Driver 19 for SQL Server の環境では、以下の設定に変更します。(※Magic xpa 4.11.1 以降)  
MGDB20=Gateways¥MGMSSQL19. dll

#### **設定 /DBMS**

ここでは、SQL Server に共通した設定を行います。

DBMS特性: MicrosoftSQLServer

**DBMSの設定**

DBMSの設定では、DBMSに接続する時の追加情報を定義します。

照合順序ファイル:

最大接続数:

分離レベル:

テーブルの存在チェック

OK キャンセル

## 最大接続数

xpaの実行エンジンとSQL Serverとの最大接続数です。SQL Server から見るとセッション数に相当します。デフォルトは 0 ですが、この場合の最大数は3になります。クライアントサーバ型のオンラインシステムで実行クライアント数が3を越える場合、適切なクライアント数を設定する必要があります。

サーバ実行版のシステムでは、各インスタンス毎で使用する最大スレッド数またはユーザ数の3倍を目安として設定します。

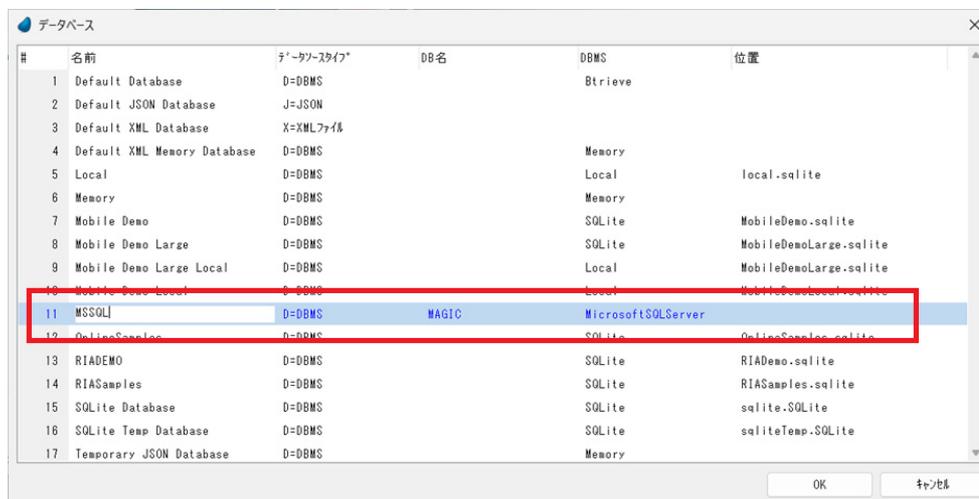
参考：[DBMS 特性](#)

## 分離レベル

複数クライアントからの同時アクセス時の動作を制御します。デフォルトは、READ UNCOMMITTED モードでダーティリードを許可しますが、このパラメータを1に設定すると READ COMMITTED モードに変更されます。

## 設定 / データベース

SQL Server に作成したデータベースごとの設定を行います。



### DB名

SQL Server のデータベース名を設定します。  
( 図の「MAGIC」は、新規に作成したデータベース )

### 位置

SQL Server では無効です。

## 設定 / データベース / データベース特性 / ログインタブ

データベース特性: MSSQL

ログイン(L) オプション(O) SQL(Q)

データベースログオン

ここでは、ログインするデータベースサーバ名とユーザID、パスワードを定義します。

データベースサーバ名: SQLServer

ユーザ名: xpouser

ユーザパスワード: %Password1%

接続文字列:

OK キャンセル

### データベースサーバ

SQL Server が実行されるサーバを設定します。

### ユーザ名

SQL Server のログインアカウント名。空欄の場合は、Windows 認証として Windows のログインアカウントが SQL Server のアカウントになります。

### ユーザパスワード

アカウントのパスワード。シークレット名を設定することを推奨します。

## 設定 / データベース / データベース特性 / オプションタブ

### M a g i c ロック

xpa によるロックメカニズムを使用するか否かを設定します。レコード単位のロックについては、SQL Server のロックメカニズムの有効を推奨しますので、「N=なし」か「T=テーブル」を推奨します。

「N=なし」の場合、テーブルロックの動作が Actian Zen の場合と異なり、プログラムのテーブルの [共有] を「R: 読込」または「N: なし」に設定したプログラムの実行時に待ち状態になることがあります。

### テーブルの存在チェック

実行時、アクセス先の SQL テーブルが存在するかどうかのチェックを行うか否かを指定します。

この特性が「有効」な場合、チェックを実行し、テーブルがない場合には、そのテーブルを xpa が作成します。なお、この機能を有効にすると、チェックのために処理速度が低下する場合があります。

## 3. プロジェクトの変更

Actian Zen のアプリケーションと SQL Server のアプリケーションにおける、テーブル定義とプログラムの設定の違いを説明します。

### 3.1 テーブル定義

#### 3.1.1. データベースの変更

データリポジトリのテーブルのデータベース欄を変更することによって、カラムの特性値、インデックスの特性値が自動的に SQL Server のテーブルアクセスに適した設定に変更されます。

順	名前	データソース名	データベース
1	制御ファイル	PetshopControl	MSSQL
2	顧客ファイル	PetshopCustomer	MSSQL
3	商品ファイル	PetshopItem	MSSQL
4	受注ファイル	PetshopOrders	MSSQL
5	受注明細ファイル	PetshopOrdersDetails	MSSQL

#### 3.1.2. データソース名

Actian Zen のデータソース名は Windows 上のファイル名ですが、SQL Server は、データベースのテーブル名です。テーブル名は SQL Server の以下のような規約に従って設定します。

順	名前	データソース名	データベース
1	制御ファイル	PetshopControl	MSSQL
2	顧客ファイル	PetshopCustomer	MSSQL
3	商品ファイル	PetshopItem	MSSQL
4	受注ファイル	PetshopOrders	MSSQL
5	受注明細ファイル	PetshopOrdersDetails	MSSQL

- 名前の最大サイズは 128 文字
- .(ドット) はスキーマ名とテーブル名の区切文字として認識されるため、使用できません。
- 記号は、@\_ \$# が使用可能。ただし、@, @#, ## を接頭文字にすると、特別な意味があるため、通常は使用しない。
- SQL Server で決められた予約語 (CHECK, ORDER など) は使用できません。

この SQL Server の命名規約は、テーブル名、カラム名、インデックス名等にも適用されます。詳細は、SQL Server のリファレンス等を参照してください。

### 3.1.3. データソース特性 / オーナ名

SQL Server のテーブルに設定されたスキーマ名を指定します。空欄の場合、データベースのユーザ ID をスキーマ名としてアクセスされます。

SQL Server ではユーザごとに「規定のスキーマ」が設定されています。ユーザが作成されたテーブルに対して、そのスキーマ名がテーブルのスキーマ名として設定されます。

テーブル名が同じでスキーマ名が異なる 2 つのテーブルは、別々のテーブルとして管理されます。

SQL Server ではデータベースの作成者、あるいは管理者が `db_owner` として設定され、そのスキーマ名が `dbo` に設定されます。スキーマ名の細かな管理を行っていなかった Actian Zen のデータベースを移行する場合、追加するユーザの「規定のスキーマ」を `dbo` に設定し、さらにオーナー名に `[dbo]` を設定しておくことで、移行をスムーズに行うことができます。

### 3.1.4. データソース特性 / デフォルトの位置

プログラムでレコードを 1 件読み込んで更新または削除を行う場合、読み込み時には `SELECT`、更新・削除時には `UPDATE` または `DELETE` コマンドが実行されます。

ここでは、対象レコードを一意に識別するために必要なカラムが含まれるインデックスを指定します。

Actian Zen では「ポジション」と呼ばれる内部データでレコード位置を識別しますが、SQL Server ではそのような内部データがありません。

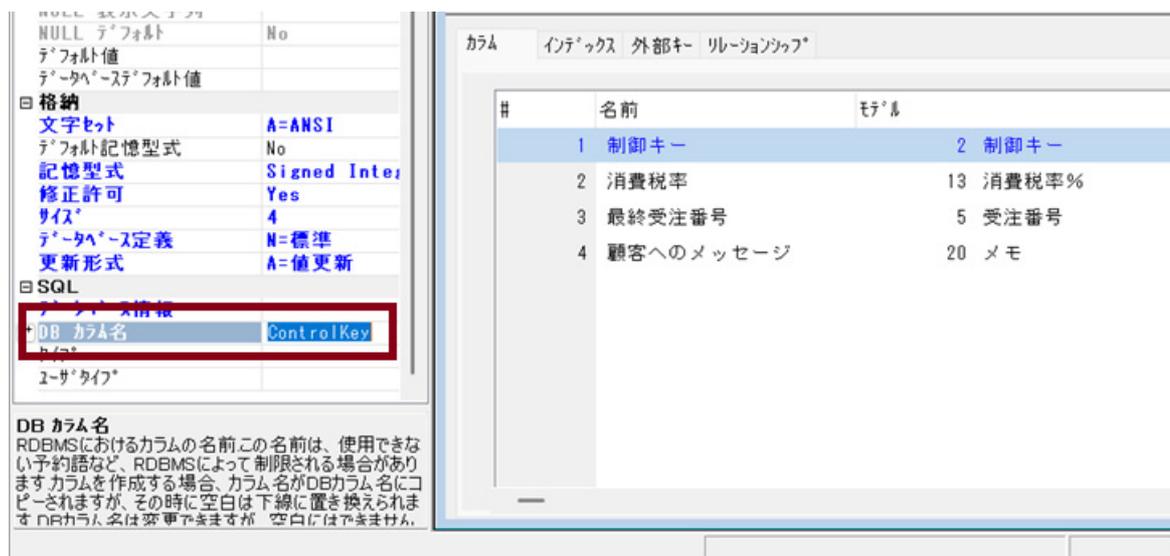
従って、SQL Server では [ デフォルトの位置 ] に、テーブルに設定された重複不可インデックスを設定します。

### 3.1.5. データソース特性 / テーブルの存在チェック

[ 設定 / データベース ] のパラメータと同様、Actian Zen では、プログラムでテーブルアクセス時にテーブルが存在しなかった場合、自動的にテーブルが作成されますが、SQL Server ではタスク開始時に `SELECT` コマンドで存在チェックを行い、存在しない場合には自動生成します。多くのテーブルを設定するプログラムでは、パフォーマンスを低下する恐れがあるため、実行環境でテーブルが存在することが自明の場合、この設定をオフにして存在確認のための `SELECT` 文の発行を抑止します。

この場合、存在しないテーブルにアクセスするとエラーが返ります。

### 3.1.6. カラム /DB カラム名



Actian Zen にはカラム名の定義はありませんが、SQL Server のテーブルの各カラムにはカラム名が必要です。SQL Server のカラム名は、各テーブルのカラム一覧の名前でなく、[カラム特性 /DB カラム名] で設定します。テーブル名と同様、次の命名規約があります。

- 名前の最大サイズは 128 文字
- . (ドット) はスキーマ名とテーブル名の区切文字として認識されるため、使用できません。
- 記号は、@\_ \$# が使用可能。ただし、@, @, @, #, ## を接頭文字にすると、特別な意味があるため、通常は使用しない。
- SQL Server で決められた予約語 (CHECK, ORDER など) は使用できません。

詳細は、SQL Server のリファレンス等を参照してください。

Actian Zen のテーブルを変更した場合や新規カラムを定義した場合は、カラムの名前が DB カラム名に設定されます。カラムの名前が SQL Server の命名規約に違反していた場合、エラーが発生しますので、確認する必要があります。

また、xpa アプリケーションからテーブルを作成した場合は、その DB カラム名で作成されますが、存在するテーブルにアクセスする際に [DB カラム名] と実際のカラムに設定されたカラムの名前に相違がある場合もエラーが発生します。

また 1 つのテーブルで同じ名前が別のカラムに設定されている場合もエラーになります。

### 3.1.7. カラム / 文字型

Actian Zen から移行した場合、書式の桁数によって表のように設定されます。

#	書式	サイズ	記憶型式	MS-SQL Server のタイプ	備考
1	n, 1-8000	書式 +1	ZString	CHAR	
2	n, 8001-32700	書式 +1	ZString	TEXT	
3	n, 1-8000	書式	String	CHAR	非推奨
4	n, 8001-32700	書式	String	TEXT	非推奨
5	(任意)	書式+2	String Memo	VARBINARY	非推奨
6	(任意)	書式+2	Magic Memo	VARBINARY	非推奨

テーブルリポジトリで、「データベース」を Actian Zen から SQL Server に変更した場合、カラムの書式によって、表の 1 または 2 が設定されます。

Actian Zen では [ 記憶型式 ] が String, String Memo, Magic Memo の場合も、データリポジトリの [ データベース ] の変更によって、「ZString」に変更されます。

テーブルリポジトリでなく、[ 設定 / データベース ] で変更した場合、[ 記憶型式 ] は変更されずに、3, または 4 になりますが、この場合プログラムから登録された日本語データは、SQL Server では 16 進表記のデータで登録されるため推奨されません。

### 3.1.8. カラム / UNICODE 型項目

Actian Zen から移行した場合、書式の桁数によって表のように設定されます。

#	書式	サイズ	記憶型式	MS-SQL Server のタイプ
1	n, 1-4000	書式 *2+2	ZUnicode	NVARCHAR
2	n, >4000	書式 *2+2	ZUnicode	NTEXT

### 3.1.9. カラム / 数値型項目

Actian Zen から移行した場合、書式の桁数によって表のように設定されます。

#	書式	サイズ	記憶型式	MS-SQL Server のタイプ
1	n (1-4)	2	Signed Integer	SMALLINT
2	n (5-9)	4	Signed Integer	INT
3	n (10-15)	8	Float	DOUBLE PRECISION
4	n (>=16)	n+1	String Number	NUMERIC

#3. の DOUBLE PRECISION は、8 バイトの浮動小数点の表現で格納されますので、小数点以下の桁数が多い場合は丸めによって誤差が生じ、場合によってはデータの更新に失敗することがあります。

この場合、[ カラム特性 / タイプ ] に numeric(n,m) (n: 最大桁数、m: 小数点以下の桁数) を指定します。

### 3.1.10. カラム / 論理型項目

Actian Zen から移行した場合、表の #1 がデフォルトで設定されます。

#	書式	サイズ	記憶型式	MS-SQL Server のタイプ
1	(任意)	1	Integer Logical	BIT
2	(任意)	2	Integer Logical	SMALLINT

### 3.1.11. カラム / 日付型項目

Actian Zen では、内部では数値のデータ (0001/01/01 を基点とする日数) ですが、デフォルトでは SQL Server の DATETIME タイプとして定義されます。

Actian Zen から移行した場合、表の #1 がデフォルトで設定されます。

#	書式	サイズ	記憶型式	タイプ	MS-SQL Server のタイプ
1	YYYY/MM/DD	8	String Date	(なし)	DATETIME
2	YYYY/MM/DD	4	Integer Date	(なし)	INT
3	YYYY/MM/DD	8	String Date	CHAR (8)	CHAR (8)

Actian Zen での日付型と、SQL Server の DATETIME 型では次のような違いがあります。

1. 存在しない日付 (0000/00/00, 0000/01/01 など) が格納できない。
2. 日付データに加えて、時刻データが含まれる。

1. について、アプリケーションで 0000/00/00 等を日付データとして利用している場合は、次のような方法で対応します。

- (a) 表の 3 の設定に変更して、CHAR 型として格納する。
  - (b) NULL 値許可 Yes として、NULL 計算値を 0000/00/00 とする。
- (b) の場合、NULL 値の動作を理解しておく必要があります。(「4.6. NULL 値」参照)

2. について、日付型項目の次に時刻型項目が定義されている場合、[カラム特性 /Datetime の一部] に、その時刻型項目を設定することにより、日付型項目と時刻型項目が SQL Server の DATETIME カラムに対応します。

日付型項目のみ定義されているテーブルでは、(a) の設定のように CHAR 型として定義すれば、アプリケーションの動作の互換性が保たれますが、プログラムを「4. SQL インターフェースの利用」の 4.1 ~ 4.4 のように拡張した場合、DATETIME 型でなく、CHAR 型として考慮する必要があります。

### 3.1.12. カラム / 時刻型項目

Actian Zen から移行した場合、表の #1 がデフォルトで設定されます。

#	書式	サイズ	記憶型式	MS-SQL Server のタイプ	追加の設定
1	HH:MM:SS	6	String Time	CHAR	
2	HH:MM:SS	4	Integer Date	INT	
3	HH:MM:SS	-	-	DATETIME	日付型項目の [Datetime の一部] に設定

3 の設定の場合、日付型項目と組み合わせて、DATETIME カラムとして設定することができます。(「4.5. 日付型と時刻型の範囲」参照)

この場合、23:59:59 を超える不正な時刻は格納できません。

### 3.1.13. カラム /BLOB 型項目

Actian Zen から移行した場合、[内容]によって、それぞれ #1,#2,#3 で設定されます。

#	内容	記憶型式	PostgreSQL のタイプ
1	A=Ansi	ANSI Blob	TEXT
2	U=Unicode	Unicode Blob	NTEXT
3	B= バイナリ	(Binary Large Object)	IMAGE

### 3.1.14. インデックス / 重複不可インデックスの作成

Actian Zen ではインデックスが定義されていない、または重複可のインデックスのみのテーブルを作成できますが、SQL Server では重複不可インデックスが必須です。その重複不可インデックスを [テーブル特性 / 位置インデックス] に設定する必要があります。

### 3.1.15. インデックス / DB インデックス名

The screenshot shows a dialog box titled 'インデックス特性: 制御キー'. It has a tab '高度な設定(A) SQL(Q)'. Under 'SQL データベース情報:', there is a search icon and a text box. Below that are fields for 'DB インデックス名:' (ControlKey), 'タイプ:' (R=実キー), 'ヒント:' (Yes), and 'クラス化:' (No). At the bottom are 'OK' and 'キャンセル' buttons.

SQL Server のインデックスオブジェクトの名前は、インデックス一覧の名前でなく、[インデックス特性 / DB インデックス名] で設定します。インデックスのタイプを「R=実キー」に設定した場合に有効です。

Actian Zen のテーブルを SQL Server に変更した場合や新規のインデックスを定義した場合は、インデックスの名前が DB インデックス名に設定されます。テーブル名。DB カラム名と同じ命名規約があります。

- 名前の最大サイズは 128 文字
- .(ドット) はスキーマ名とテーブル名の区切文字として認識されるため、使用できません。
- 記号は、@\_ \$# が使用可能。ただし、@,@@,##,## を接頭文字にすると、特別な意味があるため、通常は使用しない。
- SQL Server で決められた予約語 (CHECK, ORDER など) は使用できません。

詳細は、SQL Server のリファレンス等を参照してください。

1 つのテーブルで同じ名前が別の DB インデックス名に設定されている場合もエラーになります。プログラムの実行時にテーブルが作成された場合は、その DB インデックスが作成されますが、存在するテーブルにアクセスする場合にはインデックスの定義チェックは行われません。

### 3.1.16. インデックス / タイプ

Actian Zen でのインデックスは、データベースファイルの一部として定義されていますが、SQL Server ではテーブルと異なるオブジェクトとして存在します。

xpa のプログラムに設定したインデックスの機能は、(1) 照会プログラムの Select ステートメントの Order by に対して、インデックスに構成するカラムが付加されること、(2) レコード登録時に重複データのチェックを行うことです。

(1) について、メインテーブルの範囲パラメータ、またはリンクテーブルの位置付パラメータを設定したプログラムでは、SQL Server がインデックスのオブジェクトを使って最適なパフォーマンスで処理されますが、テーブルの作成と削除やレコードの挿入と削除を頻繁に行うテーブルでは、インデックスが多いほど負荷がかかります。

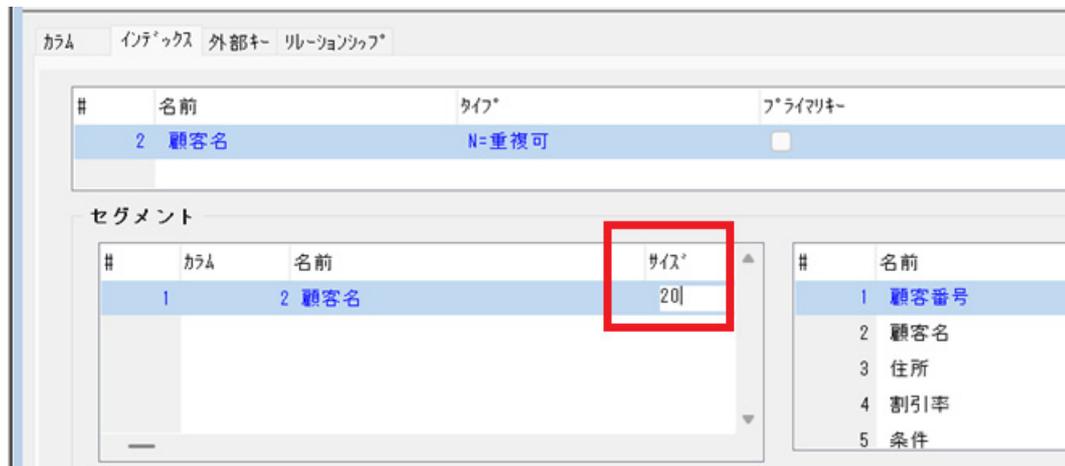
従って、プログラムに必要な機能やテーブルのレコード数に応じて、SQL Server のインデックスオブジェクトが必要かどうかを判断してください。

テーブル作成時に、定義したインデックスを SQL Server のインデックスのオブジェクトとして作成するには、[インデックス / インデックス特性 / タイプ] を「R= 実キー」に設定します。

インデックスのオブジェクトが必要ない場合は「V= 仮想キー」に設定します。

「V= 仮想キー」に設定した場合、(2) の機能は無効になり、重複データのチェックが行われませんので注意してください。(「3.2.2. 重複不可データのチェック」参照)

### 3.1.17. インデックス / セグメントのサイズ



Actian Zen では、インデックスの項目が文字型項目の場合、セグメントのサイズがカラムのサイズと異なる設定を行うことがありますが、SQL Server では「サイズ」は無効です。Actian Zen のテーブル定義で「サイズ」がデフォルトの値と異なる場合、修正してください。

## 3.2. プログラム

### 3.2.1. ロック方式とトランザクション開始

The screenshot shows a configuration window with the following settings:

- トランザクション**
  - トランザクションモード: P=物理
  - トランザクション開始: L=レコードロック時
- 管理**
  - 空のテーブルへの許可: No
  - ビュー事前読込: No
  - キャッシュ範囲: S=メインソースに依存
  - ロック方式: O=入力時
  - エラー発生時: R=復旧

SQL Server を使ったオンラインのプログラムで [レコードロック] が有効な場合、[トランザクション開始] を [ロック方式] のタイミングより前に設定する必要があります。

#	[ロック方式]	有効な [トランザクション開始] の設定
1	I= 即時	I= タスク前処理の前 G= グループ (バッチタスクのみ)
2	O= 入力時	I= タスク前処理の前 G= グループ (バッチタスクのみ) L= レコードロック時 P= レコード前処理の前
3	B= 更新時	I= タスク前処理の前 G= グループ (バッチタスクのみ) L= レコードロック時 P= レコード前処理の前 S= レコード後の前 U= レコード更新前

### トランザクション変更による影響

トランザクションを有効にした場合、プログラムで更新したレコードのデータベースへの書き込みのタイミングが変わります。さらにそのタスクで更新されるデータだけでなく、コールされるタスクで更新されるデータのタイミングも考慮する必要があります。このために、正しくデータ更新やロールバックが行われないことがありますので、次の点に注意してください。

## 「物理」と「遅延」のどちらか一方のトランザクションモードにあわせる

ここではオンラインプログラムの「物理」トランザクションモードを前提にしています。新規プログラムの作成時は、デフォルトでオンラインタスクは「遅延」モードに設定されるため、実際の動作と合わないことがあります。また「物理」トランザクションモードのタスクから「遅延」トランザクションモードのタスクをコールするとエラーが発生します。

このような問題を避けるため、以下の MAGIC.INI の設定によって、デフォルトのトランザクションモードを変更することができます。

```
[MAGIC_SPECIALS]
```

```
SpecialDefaultTransactionMode=P
```

## トランザクションの開始と終了(コミット)、およびネスト

トランザクションの開始・終了の単位で SQL Server への実際のデータの書き込みが行われるため、タスクの構成が複雑な場合は、プログラムで行う更新と実際のデータの書き込み順序が変わる可能性があります。このため、次のような順序でトランザクションの開始と終了がいつ行われるかを確認します。

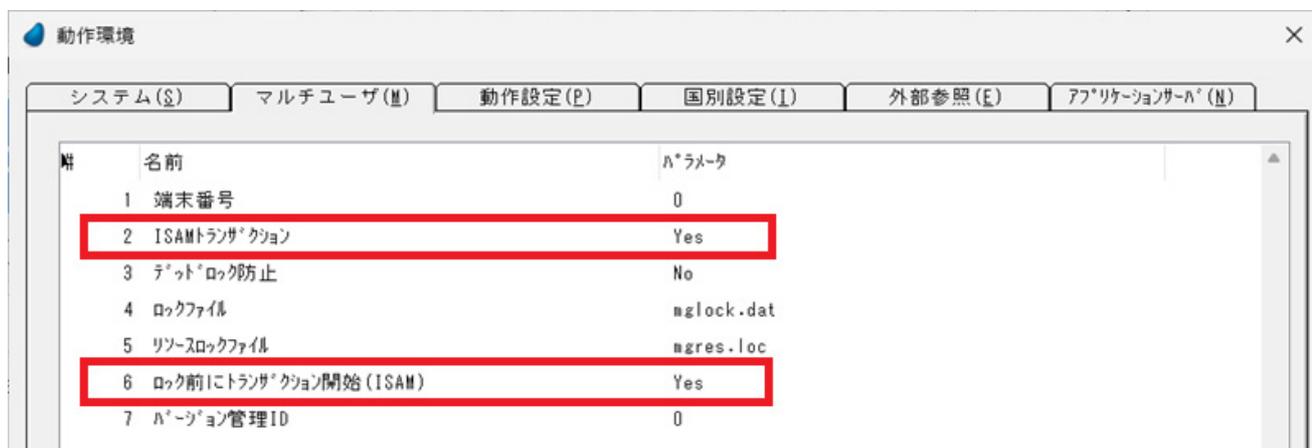
1. 親のプログラムの DB テーブルの設定とトランザクション開始と終了を確認します。DB テーブルの設定がないと、トランザクションの開始は行われません。
2. プログラムの中のコールコマンドを確認し、そこでコールされたプログラムのトランザクションの開始・終了を確認します。

例えば、親がレコード前処理でトランザクションを開始した場合、タスク前処理でコールされたタスクは別のトランザクションとして実行されますが、レコード前処理でコールされたタスクは同じトランザクションとして実行されます。

ユーザイベントハンドラの処理テーブルの場合は [強制終了] パラメータの設定により変わることがあります。例えば、「レコード」の場合はレコード後処理の後でコールコマンドが実行されるので、トランザクションが終了(コミット)してから、コールプログラムが実行されます。

トランザクション設定を有効にしていない Actian Zen のアプリケーションの場合、SQL Server への移行前に、以下の設定を Yes にしてトランザクション有効時の動作を確認することも検討してください。この設定を行ってもプログラムが自動的に修正されることはありませんが、プログラムチェックによって、設定の変更箇所を特定することができます。

- [設定 / 動作環境 / マルチユーザタブ / ISAM トランザクション]
- [設定 / 動作環境 / マルチユーザタブ / ロック前にトランザクション開始 (ISAM)]



### 3.2.2. 重複不可データのチェック

Actian Zen のテーブルの場合、重複不可インデックスを定義したテーブルで重複したデータを登録した場合、xpa アプリケーションでデータの重複を判定します。しかし、SQL Server の場合は SQL Server からのエラーによって判定します。それによって次の点を考慮する必要があります。

#### 仮想キーでは重複エラーが発生しない。

高度な設定(A) SQL(Q)

SQLデータベース情報:

DBインデックス名: 受注番号

タイプ: V=仮想キー

ヒント: Yes

クラス化: No

インデックスのタイプに「仮想キー」を設定して、SQL Server のインデックスオブジェクトがない場合、重複エラーが発生しません。

この状態で重複したレコードを登録した場合、SQL Server のテーブルには、同じインデックスの値のレコードが登録された状態になります。その状態でデータを読み込むと、同じインデックスの値の複数レコードの中の最初に読み込まれた 1 レコードが処理されるため、意図した処理が行われません。

#### 同一レコードのカラム間の移動時に重複エラーが発生しない。

Actian Zen では、インデックスセグメントのカラムに重複した値を入力して、次のカラムに移動した直後に重複エラーが発生しますが、SQL Server のデフォルトの設定では、すべてのカラムを入力してレコードを格納する直前にエラーが発生します。

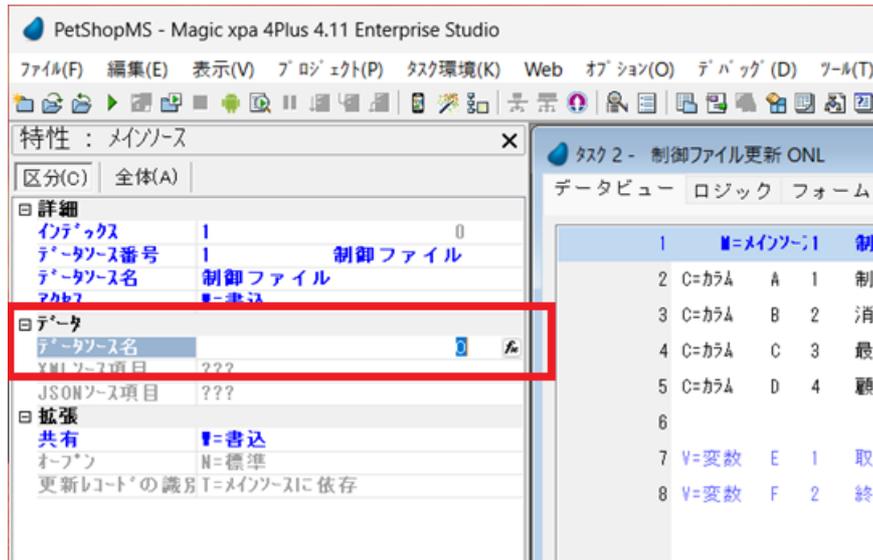
### 3.2.3. レコード登録

メインテーブルの登録モード、リンク書出、リンク登録を設定したプログラムの実行時に、そのテーブルに定義されたカラムをすべてセレクト項目に定義しないとエラーになることがあります。

レコード登録では、SQL Server に対して、Insert ステートメントを実行しますが、セレクト項目に定義されていないカラムに対して、NULL 値、あるいはデータベースデフォルト値を格納しようとします。Actian Zen から移行したカラムには、デフォルトで NULL 値が許可されていないため、エラーが発生します。これを解決するためには、次のいずれかの変更を行います。

- レコード登録を行うプログラムで、すべてのカラムをセレクト項目に設定する。
- [カラム特性/NULL 値可] を Yes に変更して、テーブルの定義を変更する。  
(NULL 値可について、「4.6. NULL 値」を参照)
- [カラム特性/データベースデフォルト値] に任意の値を設定して、テーブルの定義を変更する。

### 3.2.4. データソース名の式



プログラムのメインテーブル、またはリンクテーブルの [データソース名] に Actian Zen のファイル名を設定していた場合、SQL Server のテーブル名に変更します。

### 3.2.5. 関数

データベース処理を行う次の関数は置き換えが必要になります。外部のユーティリティ等を実行している場合も確認が必要です。

#### File 関数

File 関数 (FileExist, FileDelete, FileCopy, FileSize, FileRename) で Actian Zen のデータファイルを指定している場合、DB 関数を指定します。

#	File 関数	DB 関数
1	FileExist(ファイル名)	DBExist(番号, データソース名)(※1)
2	FileDelete(ファイル名)	DBDel(番号, データソース名)(※1)
3	FileRename(元ファイル, 新ファイル)	(※2)
4	FileCopy(元ファイル, 新ファイル)	DBCOPY(番号, 元データソース, 新データソース名)(※1)
5	FileSize(ファイル名)	DBSize(番号, データソース名)(※1, ※3)

※1 番号はデータリポジトリのテーブル番号

※2 対応する DB 関数はありません。SQL コマンドタスクで EXEC sp\_rename ステートメントなどを実行します。

※3 レコードサイズ(カラムのサイズの総数)×レコード数の値が返ります。

## DB 関数

DBExist, DBDel, DBName, DBSize, DBCopy 関数の第 2 引数に Actian Zen のデータファイル名を指定している場合、SQL Server のテーブル名に変更します。

## Actian Zen のデータファイルを操作するコマンド

OS コマンドや CALL UDF、CALLDLL などによって、Actian Zen を操作するユーティリティ等を実行している場合は、削除して別のコマンドに置き換えます。

## 4. SQL インターフェースの利用

Actian Zen の ISAM アクセスの機能で作成したアプリケーションは、SQL インターフェースによる次のような機能を組み込むことによって、パフォーマンスの改善やプログラムを簡略化することができます。

### 4.1. データ抽出

データ抽出を設定するのは、Actian Zen ではセレクト項目の範囲と [ 範囲 / 位置付 ] ダイアログの範囲式ですが、SQL Server では、SQL Where 句を設定することができます。

セレクト項目の「範囲」に値を設定した場合は Actian Zen 側でデータが抽出されます。一方、[ 範囲 / 位置付 ] ダイアログの「範囲式」を使用すると、Actian Zen ではなく Magic xpa 側で抽出処理が行われます。

このため、クライアントの動作環境や SQL Server から最初を取得されるレコード数、および範囲条件によっては、処理に時間を要することがあります。

SQL Server 利用時にパフォーマンスを安定させるには、セレクト項目の「範囲」で指定できない条件を SQL Where 句に設定してください。これにより、SQL Server 側で絞り込まれた適切な件数のレコードのみが取得されるようになります。

#### DB SQL の例



範囲 SQL Where句 式

Magic SQL

式:

DB SQL:

```
[:C LIKE '%フイッジョ%']
```

逆引き辞典：[独自の SQL ステートメントをデータベースに送るには](#)

プログラム例：[タグ（複数グループ）を持つデータのサンプル](#)

## 4.2. リンク結合

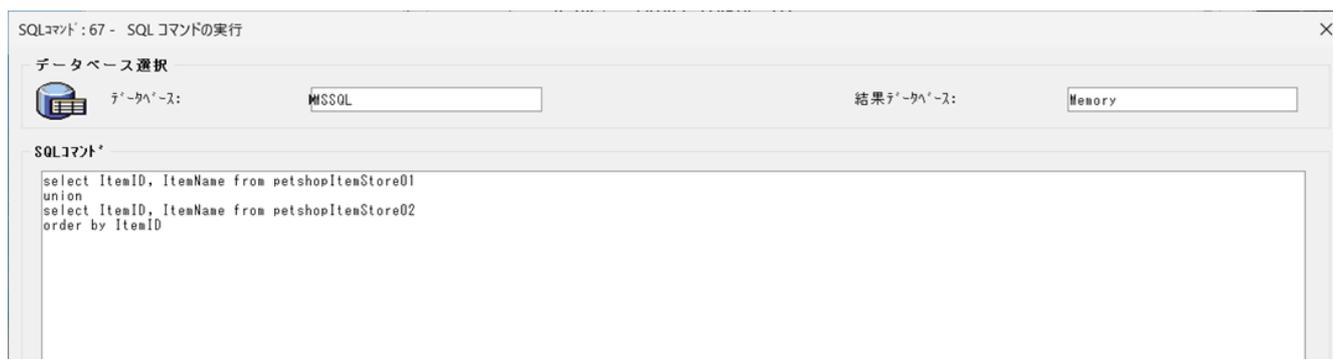
ID	コード	説明	データ型
1	M=メインソース	5 受注明細ファイル	インテック1
2	C=カラム A	1 受注番号	[5] N=数値 3Z
3	C=カラム B	2 明細番号	[6] N=数値 3Z
4	C=カラム C	3 商品番号	[4] N=数値 5Z
5	C=カラム D	4 商品タイプ	[19] A=文字 UA
6	C=カラム E	5 数量	[14] N=数値 N5CZ
7	C=カラム F	6 単価	[17] N=数値 N7CZ
8	C=カラム G	7 合計	[18] N=数値 N8CZ
10	I=結合リンク 3	商品ファイル	インテック1
11	C=カラム H	1 商品番号	[4] N=数値 5Z 位置付2 終12
12	C=カラム I	2 商品名	[8] A=文字 20
13	C=カラム J	3 商品タイプ	[10] A=文字 11

リンクコマンドの照会モードを結合モード、または外部結合モードに変更することができます。これは、SQL Server の SELECT 構文の INNER JOIN, または LEFT OUTER JOIN を実行します。これにより、SQL Server に送られる SQL ステートメントが軽減されるのでパフォーマンスの向上が期待できますし、リンクテーブルのセレクト項目を SQL Where 句の条件に設定することもできます。

参考：[結合リンク](#)

## 4.3. SQL コマンド

プログラムのメインテーブルとリンクテーブルで設定できない SQL Server のステートメントや、SQL Server のストアドプロシージャを実行することができます。



参考：[SQL コマンド \(埋め込み SQL\)](#)

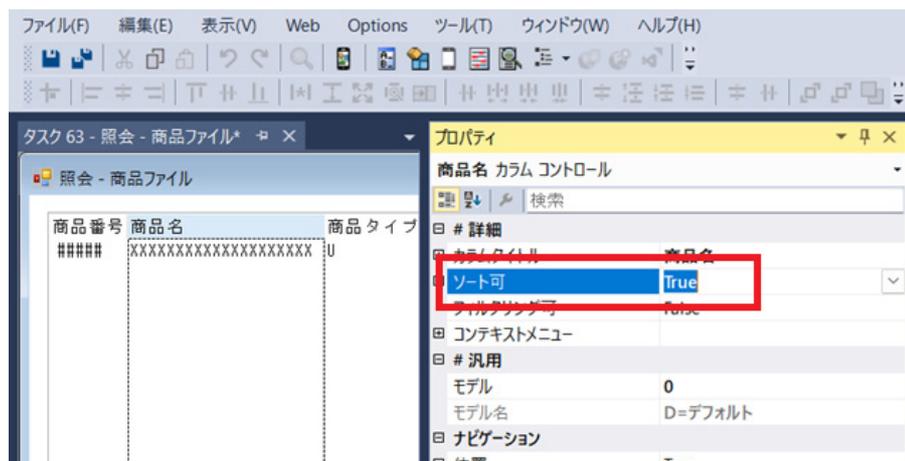
## 4.4. プログラムのソート



プログラムの [タスク環境 / ソート] は、Actian Zen では読み込まれたデータをプログラムの中でソートするため、レコード件数によってはパフォーマンスが影響することがありました。SQL Server のプログラムでは、ソートに設定した項目を **Order by** 句に追加して実行するため、プログラム内でのソート処理は行いません。これにより [タスク環境 / ソート] は、メインテーブルの仮想キーのインデックスに指定した場合と同等の処理になります。ただし、TEXT 型 (書式 8000 を超える文字型項目等) では、使用できません。また、実行時のフォーム上のテーブルの任意のカラムヘッダをクリックした際のソート処理も同様です。



カラムヘッダのソートを有効にするのは、フォームデザイナー上でカラムの [プロパティ / ソート可] を True に設定します。



## 4.5. 日付型と時刻型の範囲

日付型項目と時刻型項目が設定されたテーブルに対して、SQL Server の DATETIME タイプとして定義した場合、2つの項目を組み合わせた範囲条件の設定が簡略になります。

Actian Zen では2つの項目に分かれているため、デフォルトでは、日付型項目の最大値と最小値と、時刻型項目の最大値と最小値の AND 条件で抽出されるので、範囲式で設定するか2つの項目を合わせた変数を定義して範囲の設定などを行うこととなります。

SQL Server の DATETIME タイプの場合、複数の文字型、数値型の設定と同様、それぞれの [範囲] 大小に最大値、最小値を設定することで可能です。

ただし、DATETIME タイプは 0000/00/00 の日付が無効であること、DATETIME 以外に類似したタイプがあることを考慮に入れる必要があります。

## 4.6. NULL 値

Actian Zen ではカラムにスペースや 0 を挿入すると、20H や 0 を格納しますが、SQL Server では NULL 値を格納することができます。NULL 値はテーブル作成時のカラム特性の NULL 値可パラメータで設定します。

The screenshot shows the 'NULL 値可' (NULL Allowable) checkbox checked and highlighted with a red box. Below it, the 'NULL 表示文字列' (NULL Display String) is set to 'Yes'. The table 'インテックス 外部キー リレーションシップ' is shown with columns like '顧客番号', '顧客名', '住所', '割引率', '条件', '受注累計額', '取引回数', and 'メモ'.

NULL 値可が Yes のカラムは、レコード登録時に格納する値がないときにはセレクトコマンドとして定義する必要がないなどのメリットがありますが、次のような動作の違いが発生することがあります。

- セレクトコマンドの範囲や位置付けに 0 や "(スペース)" を設定していた場合、NULL 値のデータがこの条件に満たさないのデータが取得できないことがある。
- (数値項目)=0、(文字項目)="などの定義式の結果が、項目が NULL 値の場合に 'FALSE'LOG が返る。
- (数値項目)+1、(文字項目)&'\*'などの定義式の結果が、1 や '\*' でなく、NULL 値が返ることがある。
- NULL 値を含むカラムで構成されたインデックスで絞込みをしたバッチ処理のパフォーマンスが相対的に遅くなる可能性がある。

## 4.7. その他

データベース特性 / データベース情報、テーブル特性 / データベース情報、カラム特性 / データベース情報などにも、SQL Server の機能を有効にする設定がありますので、詳細はリファレンスを参照してください。

参考：[xpa 4Plus 4.11 ヘルプ / データベース情報](#)

Copyright (C) 2005-2026 Magic Software Japan K.K. All Rights Reserved.