

Action ZenからPostgreSQL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しての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かぎり、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。

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

PostgreSQL は、PostgreSQL Community Association of Canada のカナダにおける登録商標およびその他の国における商標です。

Fujitsu Enterprise Postgres は、富士通株式会社の商標または登録商標です。

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.

1. はじめに	1
2. PostgreSQL のセットアップ	2
2.1. PostgreSQL のインストール	2
2.1.1. Client モジュールのインストール	2
2.1.2. pgAdmin4	2
2.1.3. クライアントマシンの認証	2
2.2. データベースセットアップ	3
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	4
2.2.5. テーブル空間	4
2.3. xpa のセットアップ	5
2.3.1. データベースゲートウェイのインストール	5
2.3.2. Magic.ini の設定、または設定画面	5
3. プロジェクトの変更	8
3.1 テーブル定義	8
3.1.1. データベースの変更	8
3.1.2. データソース名	8
3.1.3. データソース特性 / オーナ名	9
3.1.4. データソース特性 / デフォルトの位置	10
3.1.5. データソース特性 / テーブルの存在チェック	10
3.1.6. カラム / DB カラム名	10
3.1.7. カラム / 文字型	11
3.1.8. カラム / UNICODE 型項目	11
3.1.9. カラム / 数値型項目	11
3.1.10. カラム / 論理型項目	12
3.1.11. カラム / 日付型項目	12
3.1.12. カラム / 時刻型項目	13
3.1.13. カラム / BLOB 型項目	13
3.1.14. インデックス / 重複不可インデックスの作成	13
3.1.15. インデックス / DB インデックス名	13
3.1.16. インデックス / タイプ	14
3.1.17. インデックス / セグメントのサイズ	15
3.2. プログラム	15
3.2.1. ロック方式とトランザクション開始	15
3.2.2. 重複不可データのチェック	17
3.2.3. レコード登録	18
3.2.4. データソース名の式	18
3.2.5. 関数	19
4. SQL インターフェースの利用	20
4.1. データ抽出	20
4.2. リンク結合	21
4.3. SQL コマンド	21
4.4. プログラムのソート	22
4.5. 日付型と時刻型の範囲	23
4.6. NULL 値	23
4.7. その他	24

1. はじめに

本書では、Actian Zen(旧 : Btrieve、または Pervasive PSQL、Actian PSQL) 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、PostgreSQL、または Fujitsu Enterprise Postgres 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。

PostgreSQL と Fujitsu Enterprise Postgres(以下 FEP) について、製品固有の機能の説明以外は、PostgreSQL と記述いたします。

Magic xpa サーバ製品 (Magic xpa Enterprise Server、Magic xpa RIA Server、および Magic xpa WebClient Server) は、Magic xpa Server と記述します。

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.PostgreSQL のセットアップ

PostgreSQL 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 PostgreSQL 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、pgAdmin4、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。

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

2.1.PostgreSQL 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、FEP、または PostgreSQL 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.Client モジュールのインストール

xpa 製品をインストールするマシンには、Actian Zen と同様、32bit、64bit のいずれかの Client 製品をインストールします。

FEP の場合、Magic xpa Enterprise Client では Fujitsu Enterprise Postgres Client (32bit)、Magic xpa Server (x64) では、Fujitsu Enterprise Postgres Client (64bit)、Magic xpa Server (x86) では、Fujitsu Enterprise Postgres Client (32bit) をインストールします。

Client 製品のインストール後、環境変数の Path に psqLODBC モジュールが存在するフォルダを追加します。FEP のデフォルトのフォルダは以下。

(32bit) C:\Program Files (x86)\Fujitsu\Fsep\v17\client32\odbc\bin

(64bit) C:\Program Files\Fujitsu\Fsep\v17\client64\odbc\bin

オープンソースの psqLODBC モジュールのフォルダは以下。

(32bit) C:\Program Files (x86)\psqLODBC\1700\bin

(64bit) C:\Program Files\psqLODBC\1700\bin

2.1.2.pgAdmin4

PostgreSQL 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。

FEP では、Fujitsu Enterprise Postgres WebAdmin(64bit) です。

Server 製品と同時にインストールすることを推奨します。

オープンソースの pgAdmin4 のデフォルトでは英語表記ですが、以下の設定で日本語に変更できます。

[File/Preferences/Miscellaneous/User language] から、「User language」を Japanese に変更。

(以後、日本語での設定を前提に説明します。)

2.1.3. クライアントマシンの認証

PostgreSQL では、サーバの pg_hba.conf ファイルに対して、接続するクライアントの IP アドレスと認証方法の設定が必要です。

詳細は、FEP のセットアップ関係の資料、または PostgreSQL のリファレンスを参照してください。

2.2. データベースセットアップ

アプリケーションの種類やユーザ数、規模によって、FEP または PostgreSQL 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。

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。

PostgreSQL では OS 上のファイルは「データベース」であり、その中で PostgreSQL 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 pgAdmin4 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します。pgAdmin4 より管理者アカウントでログインし、[ログイン/グループ]より新規ユーザを作成します。

一般的には、セキュリティ対策のため、ユーザの権限は必要最小限にすることが重要です。PostgreSQL をインストールした際に作成される postgres というスーパー権限を持つユーザを使うのではなく、それ以外にデータベースの管理者、データベースの利用者(アプリケーションユーザ)を作成してください。

以下では、次のようなユーザアカウントを使って説明します。

- postgres : インストール時に作成される最大の権限を持つユーザ。日常的な利用は避ける。
- xpaadmin : データベース・アプリケーション管理者用に新規作成したアカウント。システム管理を行うためのアカウント。
- xpauser : アプリケーションユーザ : Magic xpa から PostgreSQL データベースをアクセスするために新規作成したアカウント。

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。

PostgreSQL をインストールした際にデフォルトで作成される postgres でなく、新規に作成することが推奨されています。

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、PostgreSQL の [照合順序] と [文字の分類] に影響されます。FEP ではデフォルトで「C」に設定されますが、この場合 Actian Zen と互換性があります。

オープンソースの PostgreSQL ではデフォルトで「Japanese_Japan.932」が設定されるため、「C」に設定することを推奨します。

参考 : Japanese_Japan.932 の照合順序では、ひらがな、カタカナ、半角カタカナ、アルファベットの大文字・小文字や全角・半角の順序が変わります。日本語の直感に近い並び順にはなりますが、Magic xpa エンジン内部の操作は主にバイナリデータ(「C」設定と同じ)で行われるため、DBMS での処理と Magic xpa エンジンでの処理に違いが出てきて、予期せぬ動作になる可能性があります。

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には、ログインユーザに権限が必要です。権限はデータベース単位、スキーマ単位、テーブル単位に設定可能です。それぞれの単位でどのユーザにどのような権限を設定するかを細かく制御することができます。

例えば、異なるシステムの PostgreSQL のデータベースへのアクセスや、異なるログインユーザが作成するテーブルの場合などを考慮する必要があります。

ただし、デフォルトのデータベースや新規で作成するテーブルに対するアクセスが制限されることがありますので、細かな権限を必要としない Actian Zen のデータベースを移行する場合には、以下のような基本的な設定で可能です。この例ではすべてのテーブルに対して、作成を含むすべてのアクセスが可能です。

#	データベースとユーザの設定例	権限の設定
1	データベースの所有者 : xpaadmin スキーマ : public (デフォルト) または xpauser (所有者 xpaadmin)	[スキーマ /public/ プロパティ /セキュリティ] に対して、[受領者] に xpauser, [権限] に ALL を追加 
2	所有者 : xpaadmin スキーマ : xpauser (所有者 xpauser)	なし ※テーブルは xpauser スキーマに作成されます。
3	所有者 : xpauser スキーマ : public (デフォルト)	なし ※アプリケーションユーザ(xpauser)にデータベースの全ての権限が与えられるため、運用環境では非推奨。

2.2.5. テーブル空間

データベースのオブジェクトを物理的に格納するフォルダまたはファイルは、データベースの「テーブル空間」で設定されますが、データベースが PostgreSQL のサーバと同じ物理ディスクのドライブで管理する場合、特にデフォルトの「pg_default」を変更する必要はありません。

FEP では、パフォーマンス向上や運用管理の効率化を目的としたいくつかの独自機能や設定が可能になっています。

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

xpa のインストーラを実行して、PostgreSQL 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 mgposgreSQL.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。PostgreSQL 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]
MGDB11=Gateways¥MGPostgreSQL. dll
```

PostgreSQL ゲートウェイのライセンスの登録

実行環境では、PostgreSQL ゲートウェイを使用するためのライセンスを登録します。

PostgreSQL ゲートウェイを運用環境（実行版）で使うには、追加ライセンスの購入が必要です。ライセンスがない場合、PostgreSQL ゲートウェイは利用できません。

ライセンスが必要となる製品は、以下のものです。

- Magic xpa Enterprise Client
- Magic xpa Enterprise Server
- Magic xpa RIA Server
- Magic xpa WebClient Server

Magic xpa Enterprise Studio 製品（開発版）では、別途ライセンスは必要ありません。通常の Enterprise Studio 用ライセンスのみで PostgreSQL を使った開発・テストを行うことができます。

参考：[PostgreSQL Gateway のライセンスについて](#)

設定 /DBMS

ここでは、PostgreSQL に共通した設定を行います。



最大接続数

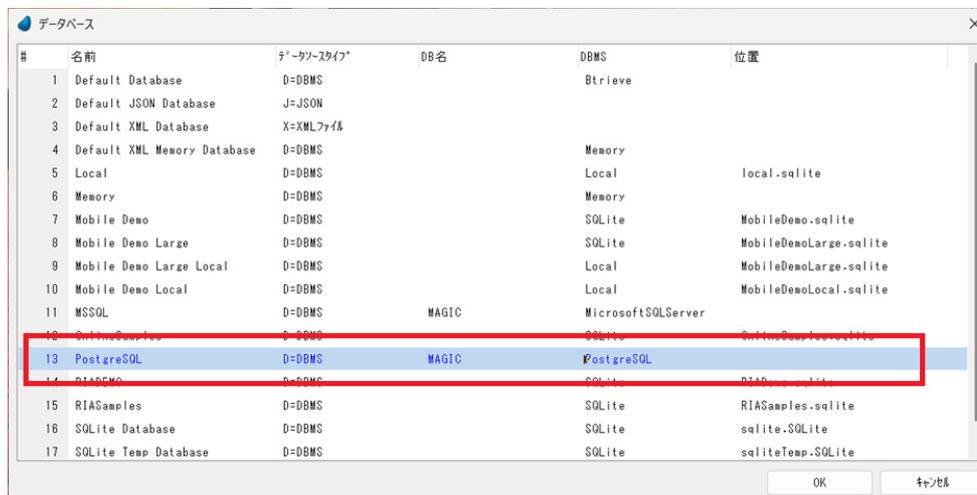
xpa の実行エンジンから PostgreSQL への最大接続数です。PostgreSQL から見るとセッション数に相当します。デフォルトは 0 ですが、この場合の最大数は 3 になります。クライアントサーバ型のオンラインシステムで実行クライアント数が 3 を越える場合、適切なクライアント数を設定する必要があります。

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](#)

設定 / データベース

PostgreSQL に作成したデータベースごとの設定を行います。



DB名

PostgreSQL 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

PostgreSQL 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

データベースサーバ

PostgreSQL が実行されるサーバ名、または IP アドレスを設定します。

ユーザ名

PostgreSQL のログインアカウント名を設定します。

ユーザパスワード

アカウントのパスワードを設定します。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

Ma 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、PostgreSQL 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。

設定 / データベース / データベース特性 / SQL タブ

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 が自動作成します。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 PostgreSQL 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 PostgreSQL のテーブルアクセスに適した設定に変更されます。

№	名前	データソース名	データベース
1	制御ファイル	PetshopControl	PostgreSQL
2	顧客ファイル	PetshopCustomer	PostgreSQL
3	商品ファイル	PetshopItem	PostgreSQL
4	受注ファイル	PetshopOrders	PostgreSQL
5	受注明細ファイル	PetshopOrdersDetails	PostgreSQL

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、PostgreSQL は、データベースのテーブル名です。テーブル名は PostgreSQL の以下のような規約に従って設定します。

№	名前	データソース名	データベース
1	制御ファイル	PetshopControl	PostgreSQL
2	顧客ファイル	PetshopCustomer	PostgreSQL
3	商品ファイル	PetshopItem	PostgreSQL
4	受注ファイル	PetshopOrders	PostgreSQL
5	受注明細ファイル	PetshopOrdersDetails	PostgreSQL

- 名前の最大サイズは 63 バイト
- . (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、_ が使用できます。
- PostgreSQL で決められた予約語 (CHECK, ORDER など) は使用できません。

- 大文字小文字の区別をつける場合、名前の両端を "(ダブルクォーテーション) で囲みます。
- 日本語 (2 バイト文字) は、推奨されていません。

この PostgreSQL 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。PostgreSQL は識別子を UTF-8 で扱えるので日本語名を使用してもエラーにはなりません。実務上、次のような理由により、一般に日本語名は推奨されていません。

- 名前の長さには 63 バイトの制限があります。UTF-8 では日本語一文字は 3 バイトになりますので、21 文字が限界になります。それより長くなった場合、PostgreSQL はエラーを出さずに、単純に切り捨てて処理をします。意図しない動作の原因になります。インデックス名 (後述) などは重複を避けるために慣例的にテーブル名と連結する場合がありますが、この制限にすぐに達してしまう可能性が高いです。
- 日本語は必ず “ (ダブルクォート) で囲む必要があります。Magic xpa だけから使う場合には PostgreSQL ゲートウェイが自動的に行うので動作に問題はありませんが、他のプログラムやツール等でアクセスする場合には煩雑になり、言語やツールの動作によっては思わぬ問題が発生することがあります。
- 他の DBMS などへの移植時の障害になる可能性があります。

詳細は、PostgreSQL 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

データソース特性: 制御ファイル

高度な設定 (A) SQL (Q) XML (X)

SQL データベース情報

オーナー名: xpauser

位置: D=デフォルト

インデックス: 0

デフォルトの位置: 1 制御キー

テーブルの存在チェック: D=テーブルの存在に依存

テーブルのタイプ: T=テーブル

ヒント: Yes

カール: D=デフォルト

配列のサイズ: 0

サイズ: 211

OK キャンセル

PostgreSQL のテーブルが格納されたスキーマ名を指定します。空欄の場合、ログイン名と同じスキーマがあればそのスキーマ名、なければデフォルトの **public** スキーマ名でアクセスされます。テーブル名が同じで [オーナ名] 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。複数のシステムで同じデータベースを使う場合等、それぞれのスキーマに分けることが可能になります。

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、PostgreSQL ではそのような内部データがありません。

従って、PostgreSQL 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、PostgreSQL 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

The screenshot shows the configuration interface for Actian Zen. On the left, there is a 'DB カラム名' (DB Column Name) section with a red box around the 'ControlKey' value. Below it, there is a note about column names in RDBMS. On the right, there is a table list with columns '#', '名前', and 'モデル'.

#	名前	モデル
1	制御キー	
2	消費税率	
3	最終受注番号	
4	顧客へのメッセージ	

Actian Zen にはカラム名の定義はありませんが、PostgreSQL のテーブルの各カラムにはカラム名が必要です。PostgreSQL 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 63 バイト
- . (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、_ が使用できます。
- PostgreSQL で決められた予約語 (CHECK, ORDER など) は使用できません。
- 大文字小文字の区別をつける場合、名前の両端を " (ダブルクォーテーション) で囲みます。
- 日本語 (2 バイト文字) は、推奨されていません。

詳細は、PostgreSQL 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 PostgreSQL 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	PostgreSQL のタイプ	備考
1	n, 1-2000	書式 +1	ZString	character varying	
2	n, 2001-32000	書式 +1	ZString	text	
3	n, 1-2000	書式	String	character varying	非推奨
4	n, 2001-32000	書式	String	text	非推奨
5	(任意)	書式 +2	String Memo	bytea	非推奨
6	(任意)	書式 +2	Magic Memo	bytea	非推奨

テーブルリポジトリで、「データベース」を Actian Zen から PostgreSQL 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。

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	PostgreSQL のタイプ
1	n, 1-2000	書式 *2+2	ZUnicode	character varying
2	n, >2000	書式 *2+2	ZUnicode	text

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	PostgreSQL のタイプ
1	n (1-4)	2	Signed Integer	smallint
2	n (5-9)	4	Signed Integer	integer
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	PostgreSQL のタイプ
1	(任意)	1	Integer Logical	boolean
2	(任意)	1	String Logical	bytea

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 PostgreSQL の `timestamp` 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	追加の設定	PostgreSQL のタイプ
1	YYYY/MM/DD	8	String Date	(なし)	date
2	YYYY/MM/DD	4	Integer Date	(なし)	integer
3	YYYY/MM/DD	8	String Date	タイプに CHAR(8) を設定	character(8)
4	YYYY/MM/DD	-	-	DateTime の一部に時刻型項目を設定	timestamp

#1 の `date` 型の場合、存在しない日付 (0000/00/00, 0000/01/01 など) は格納できません。この場合、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、`character` 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)

#4 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 PostgreSQL の `timestamp` 型に対応します。

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 `character` 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、`timestamp` 型でなく、`character` 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	PostgreSQL のタイプ	補足
1	HH:MM:SS	6	String Time	character	
2	HH:MM:SS	4	Integer Date	integer	
3	HH:MM:SS	-	-	timestamp	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、timestamp 型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	TEXT
3	B=バイナリ	(Binary Large Object)	BYTEA

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、PostgreSQL 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

PostgreSQL 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 PostgreSQL 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せん。

- 名前の最大サイズは 63 バイト
- . (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、_ が使用できます。
- PostgreSQL で決められた予約語 (CHECK, ORDER など) は使用できません。
- 大文字小文字の区別をつける場合、名前の両端を " (ダブルクォーテーション) で囲みます。
- 日本語 (2 バイト文字) は、推奨されていません。

詳細は、PostgreSQL のリファレンス等を参照してください。

異なるテーブルまたは同じテーブルで、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、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る場合は、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、PostgreSQL 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、PostgreSQL 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、PostgreSQL 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 PostgreSQL のインデックス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

#	名前	タイプ*	プライマリキー
2	顧客名	N=重複可	<input type="checkbox"/>

#	名前	サイズ*
1	2 顧客名	20

#	名前
1	顧客番号
2	顧客名
3	住所
4	割引率
5	条件

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。PostgreSQL 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

汎用(G) 動作(B) インタフェース(L) テーブル(D) オプション(Q) 拡張(A)

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

空のユーザー許可 : No

ビュー事前読込 : No

キャッシュ範囲 : S=メインメモリに依存

ロック方式 : O=入力時

エラー発生時 : R=復旧

PostgreSQL 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 PostgreSQL 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。

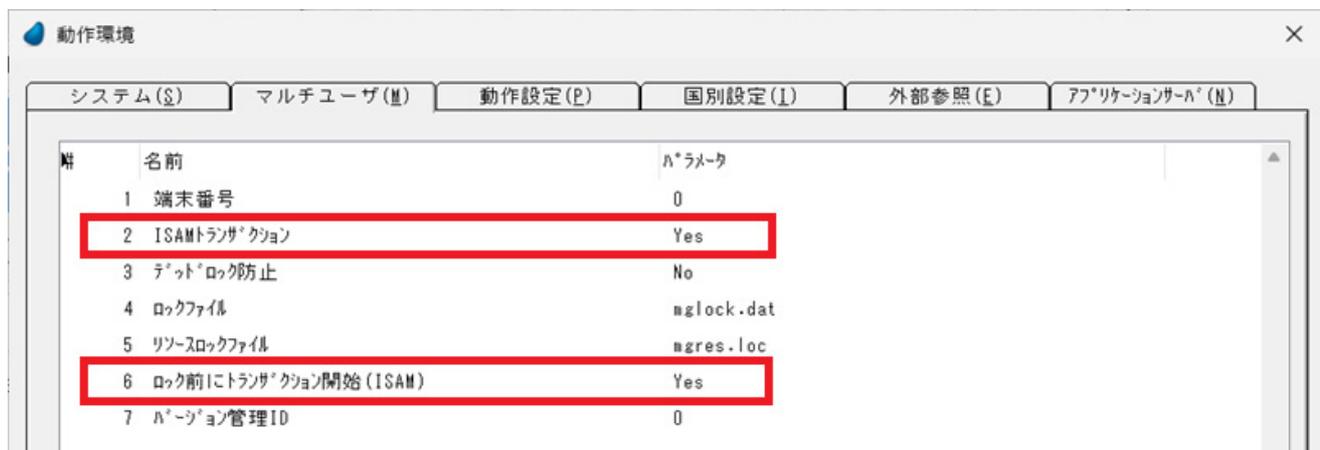
1. 親のプログラムの DB テーブルの設定とトランザクション開始と終了を確認します。DB テーブルの設定がないと、トランザクションの開始は行われません。
2. プログラムの中のコールコマンドを確認し、そこでコールされたプログラムのトランザクションの開始・終了を確認します。

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、PostgreSQL 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。

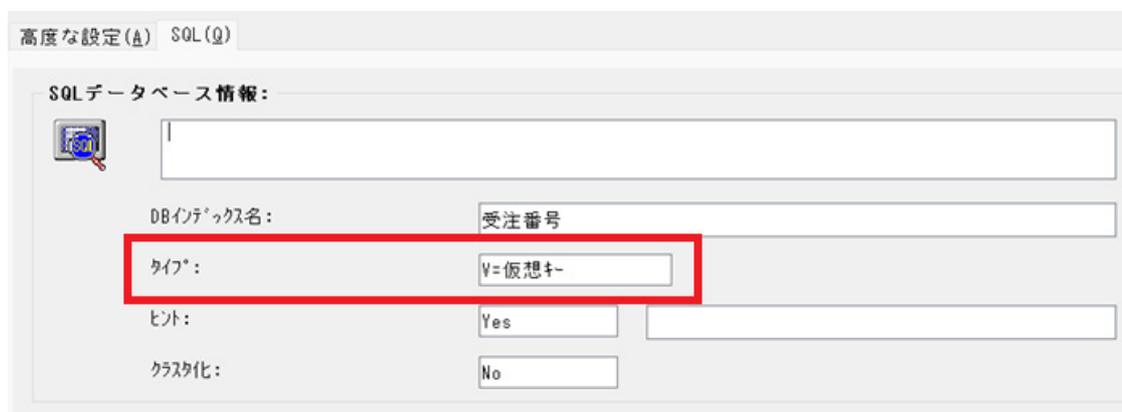
- [設定 / 動作環境 / マルチユーザタブ / ISAM トランザクション]
- [設定 / 動作環境 / マルチユーザタブ / ロック前にトランザクション開始 (ISAM)]



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 でデータの重複を判定します。しかし、PostgreSQL の場合は PostgreSQL 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。



インデックスのタイプに「仮想キー」を設定して、PostgreSQL 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、PostgreSQL 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、PostgreSQL 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、PostgreSQL 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

The screenshot shows the '特性: メインソース' (Properties: Main Source) dialog box. The 'データ' (Data) section is expanded, and the 'データソース名' (Data Source Name) field is highlighted with a red box. The value entered is '制御ファイル更新 ONL'. The table structure is visible in the background, showing columns C=カラム A through F with various data types and constraints.

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、PostgreSQL のテーブル名に変更します。

これも PostgreSQL のテーブル名に関する制限事項の対象になりますので、式を評価した結果が PostgreSQL として正しいテーブル名になるかを確認してください。

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 ALTER TABLE ... RENAME TO ... 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、PostgreSQL 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、PostgreSQL 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 PostgreSQL から最初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。

PostgreSQL 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、PostgreSQL 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 '%フイック%' ]
```

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

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 UA

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、PostgreSQL の SELECT 構文の INNER JOIN, または LEFT OUTER JOIN を実行します。これにより、PostgreSQL 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 PostgreSQL のステートメント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

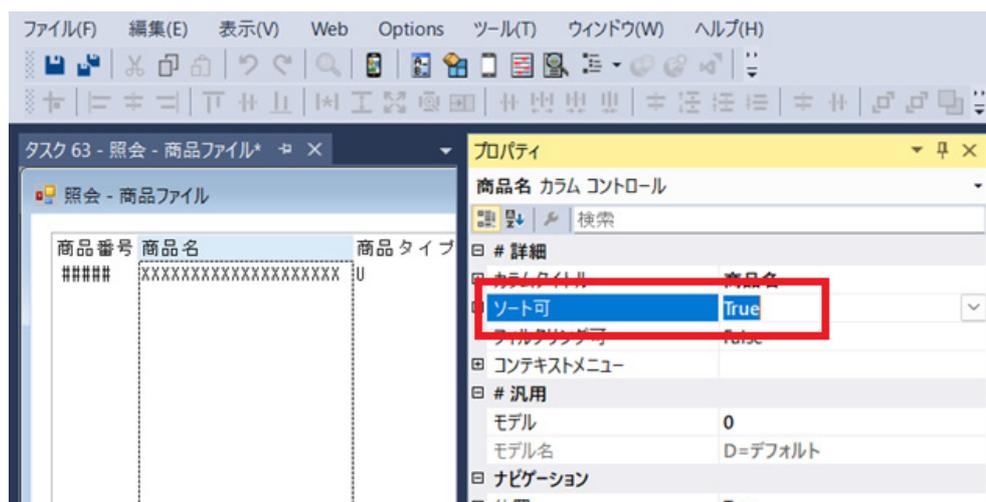


プログラムの [タスク環境 / ソート] は、Action Zen では読み込まれたデータをプログラムの中でソートするため、レコード件数によってはパフォーマンスが影響することがありました。PostgreSQL のプログラムでは、ソートに設定した項目を Order by 句に追加して実行するため、プログラム内でのソート処理は行いません。これにより [タスク環境 / ソート] は、メインテーブルの仮想キーのインデックスに指定した場合と同等の処理になります。

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、PostgreSQL の timestamp 型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。

PostgreSQL の timestamp 型の場合、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、timestamp 型は 0000/00/00 の日付が無効であること、timestamp 型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、PostgreSQL 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 Actian Zen configuration interface. On the left, there is a list of configuration options for NULL values. The 'NULL 値可' (NULL Allowed) option is highlighted with a red box and set to 'Yes'. Below it, other options like 'NULL 表示文字列', 'NULL デフォルト', and 'デフォルト値' are visible. On the right, a table definition window is open, showing a table with 8 columns. The columns are: 1. 顧客番号 (3 characters), 2. 顧客名 (7 characters), 3. 住所 (10 characters), 4. 割引率 (12 characters), 5. 条件 (9 characters), 6. 受注累計額 (18 characters), 7. 取引回数 (16 characters), and 8. メモ (20 characters).

#	名前	モデル
1	顧客番号	3 顧客
2	顧客名	7 顧客
3	住所	10 住所
4	割引率	12 その他
5	条件	9 条件
6	受注累計額	18 金額
7	取引回数	16 取引
8	メモ	20 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こともある。

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、PostgreSQL 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 ヘルプ / データベース情報](#)

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