

# SQL Server の移行

Visual Expert を使用して、SQL Server スキーマと T-SQL コードの準備、移行、最適化を行います。

## はじめに

SQL Server データベースを新しいバージョンに移行するのは、複雑な作業です。データを転送するだけでなく、大量の Transact-SQL (T-SQL) コードを調整し、時間の経過とともに蓄積された不要な要素をクリーンアップし、リグレッションが発生しないことを保証する必要があります。そのため、SQL Server の開発者や DBA は、プロジェクトのあらゆる段階で細心の注意を払う必要があります。

幸い、この作業を大幅に容易にする専用ツールがあります。[Visual Expert は](#)、SQL Server 移行の各段階を安心して進めるためのツールを組み合わせ提供しています。[Visual Expert](#) は、既存のコードの調査、理解、文書化に最適な、T-SQL コードの静的解析ツールです。

この記事では、SQL Server 移行の典型的な段階をレビューし、各段階で遭遇する問題と、このツールが特定の機能によってそれらの問題を克服する方法について見ていきます。

例として、主に内部 T-SQL コード（ストアドプロシージャ、トリガーなど）を含む SQL Server 2008/2012 データベースを SQL Server 2022 に移行する場合を取り上げます。

## 目次

Visual Expert を使用して、SQL Server スキーマと T-SQL コードの準備、移行、最適化を行います。 .....	1
はじめに .....	1
ステップ 1：コードおよびオブジェクトのボリュームの評価 .....	3
ステップ 2：コードのクリーンアップ .....	4
ステップ 3：置き換えるべき廃止された要素の検出 .....	5
ステップ 4：実施すべき変更の影響分析 .....	6
ステップ 5：変更の実施 .....	8
ステップ 6：移行済みデータベースの技術文書 .....	9
ステップ 7：移行後の最適化 .....	10
ステップ 8：Visual Expert による移行後の継続的なモニタリング .....	11
結論 .....	12
リソース： .....	12
フェーズ別概要表 .....	13
付録 A – 廃止または削除された要素の例 (SQL Server 2008/2012 → 2022) .....	14

# ステップ 1：コードおよびオブジェクトのボリュームの評価

目的：分析対象範囲を明確に定め、移行の負荷を見積もる。

課題：移行プロジェクトの最初のステップは、作業規模の評価です。

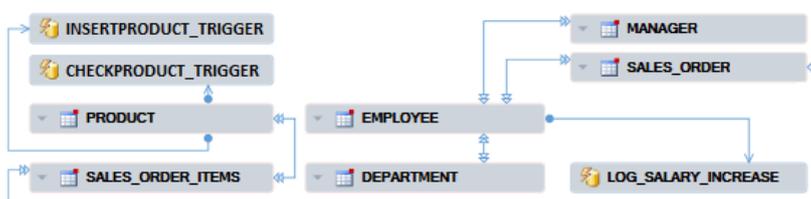
データベース内のすべてのオブジェクト（テーブル、ビュー、インデックスなど）と、すべての T-SQL コード（ストアドプロシージャ、関数、ビュー、トリガー）をリストアップする必要があります。このボリュームの推定は、作業負荷と必要なリソースの計画に役立ちます。

Database	
Tables	37
Stored Procedures	1
Views	8
Synonyms	2
Schemas	1
Transact SQL	
Functions	15
Procedures	52
Schemas	4
Triggers	9
Sequences	1

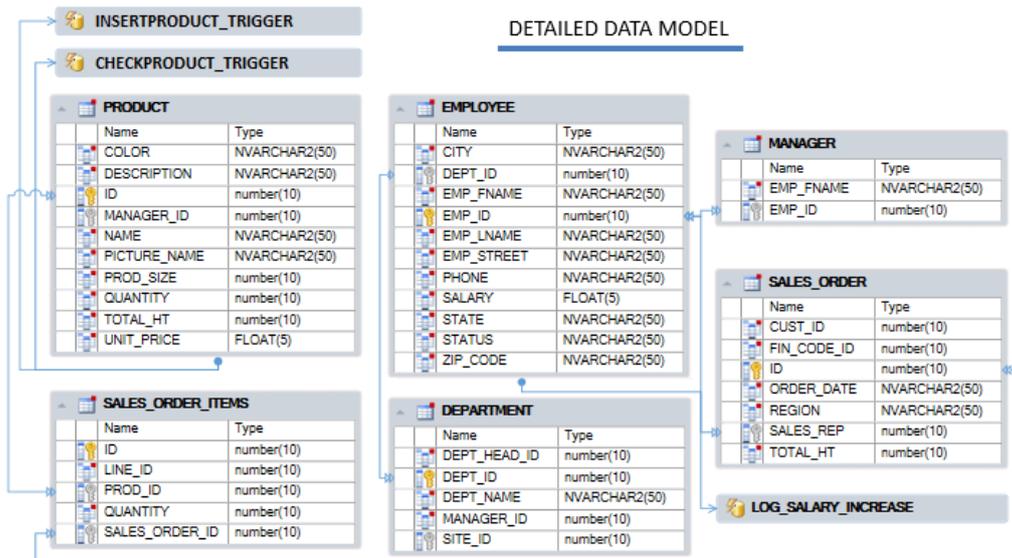
ツールがない場合、SQL Server カタログと T-SQL コードを手作業で閲覧することになり、これは面倒でエラーが発生しやすい作業です。SQL Server データベースには、何千ものオブジェクトと何百万行ものコードが含まれている場合があります。最初から正確な全体像を把握することは、アーキテクトやプロジェクトマネージャーにとって困難な課題です。

- Visual Expert は、コードとオブジェクトの自動インベントリを提供します。データベースを分析すると、このツールは詳細な**コードメトリクス**を計算できます：T-SQL コードの総行数（コメントと有効な命令を区別）、プロシージャ、関数、ビュー、トリガーなどの数。また、タイプ別にオブジェクトのリストを作成します。
- さらに、Visual Expert は**既存のデータベースのモデリングにも使用できます**。SQL Server のエンティティリレーションシップ図を自動的に生成し、**データモデルの複雑さ**（テーブルとリレーションシップの数、機能サブドメインへの分割など）を把握することができます。これにより、アーキテクトはモデルの一部が非常に密接（多くのテーブルが相互接続）であるか、あるいは逆に孤立しているかを確認することができます。これにより、移行において最も重要なモデル領域（密度の高い領域は、多くの場合、関連するロジックが多いことを意味します）を評価することができます。

COMPACT DATA MODEL (objects collapsed)



DETAILED DATA MODEL



## ステップ 2：コードのクリーンアップ

目的：未使用のオブジェクトを削除して、技術的負債とリスクを軽減する。

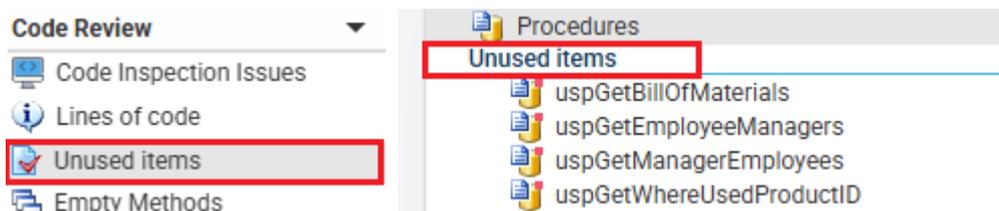
**課題：**移行前に、コードとオブジェクトを整理しておくことが賢明です。もはや使用されていない要素（呼び出されなくなった古いプロシージャ、忘れられた一時テーブルやテスト用テーブル、廃止されたビューや同義語など）を新しいバージョンに移行する意味があるでしょうか？

移行には無駄な時間がかかり、リスクも増大します（例：廃止されたプロシージャには、SQL Server 2022 と互換性のないコードが含まれている可能性がある）。

とはいえ、大規模なデータベースでは、未使用のオブジェクトを手作業で検出することは非常に困難です。開発者は、隠れた機能を破壊する恐れがあるため、コードの削除を躊躇します。予防措置として、多くの場合、不必要な労力とより重い技術的負債を犠牲にして、すべてを移行します。

Visual Expert は、SQL Server データベースの「デッドコード」のリストを作成することで、この整理を支援します。この情報をもとに、チームは SQL Server 2022 に移行しない**オブジェクト**を決定し、不要な要素を排除することができます。

- 依存関係の完全な静的分析により、Visual Expert が分析した他のプロシージャ、関数、トリガー、スクリプトから参照/呼び出されていない、**使用されていない可能性のある T-SQL オブジェクト**を



特定します。

- また、コードによって一度も使用されていないテーブル（SELECT/INSERT/UPDATE/DELETE クエリによって参照されたことがないテーブル）も特定できます。これは、そのテーブルがアプリケーションによって（もはや）使用されていない可能性があることを示唆しています。
- 最後に、クリーンアップすべきその他の要素（空のプロシージャ、統合すべき重複コードなど）も報告します。

## ステップ 3：置き換えるべき廃止された要素の検出

目的：移行前に置き換えるべき廃止された技術を特定すること。

課題：不要な要素に加え、チームは**必要な要素でありながら廃止された要素**も管理する必要があります。

時間の経過とともに、SQL Server は特定の機能を廃止し、新しい機能を導入しています。たとえば、**TEXT**、**NTEXT**、**IMAGE** などの一部のデータ型は、**VARCHAR(MAX)**、**NVARCHAR(MAX)**、**VARBINARY(MAX)** に置き換えられました。コードに関しては、一部の T-SQL の手法や関数が、より堅牢で最新の代替手段に徐々に置き換えられています（例：**@@IDENTITY** が **SCOPE\_IDENTITY()** に、**RAISERROR** が多くの場合 **THROW** に置き換えられています）。

これらの要素は、サポートが長期的に弱くなるレガシー技術を維持しないために、置き換えることをお勧めします。

難しいのは、ベーススキーマやコード内で、こうした廃止された要素が使われている場所を全部見つけること。見落とされやすいものもある（例：テクニカルテーブルに隠れた TEXT/NTEXT 型の列）。したがって、SQL Server の開発者や DBA は、必要な置換を計画するために、徹底的な検出を行う必要があります。何千行ものコードを手作業でスキャンしたり、SQL Server デクショナリをくまなく調べたりすることは現実的な解決策ではありません。そのため、ツールが必要となります。

**Visual Expert** は、依存関係を詳細に分析することで、この作業に優れています。

- 廃止された SQL Server 関数を参照するすべてのプロシージャ、トリガー、関数、または TEXT/NTEXT などの廃止された型に基づくテーブルや列を一覧表示することができます。
- Visual Expert には、コードの自動検査エンジンも搭載されており、そのルールにより、廃止された要素の使用について警告を發します。

Location	Non-compliant item	Description
dbo > uspGetEmployeeManagers	SQL Select	SELECT [EMP_cte].[Re
dbo > uspGetManagerEmployees	SQL Select	SELECT [EMP_cte].[Re

- Visual Expert は、既存のコードをベストプラクティスに基づいて精査する**互換性スキャナー**としての役割を果たし、テスト中に予期せぬ問題が発生するのを防ぎます。

データをモデリングすることで、Visual Expert は、SQL Server による改善を活用するために近代化することが**望ましい設計上の旧式な要素**を特定するのに役立ちます。例えば：

- 一部のテーブルでは、IDENTITY タイプの一意の識別子ではなく、複合識別子が使用されています。
- 一部の関係は制約によって宣言されていません（従来はコードによって管理されていました）。

- 関連性のない TEXT/NTEXT フィールドは、VARCHAR(MAX)/NVARCHAR(MAX) に変更できる可能性があります。

**Visual Expert** は、コードの変更に加えて、**構造上の変更**についても検討材料を提供します。その後、予定されている変更（新しいリレーションシップ、新しい型）をダイアグラムに注釈として追加することができます。これらの要素は、次のステップである影響分析で使用されます。

## ステップ 4：実施すべき変更の影響分析

目的：変更によるあらゆる影響を予測し、後退を防ぐ。

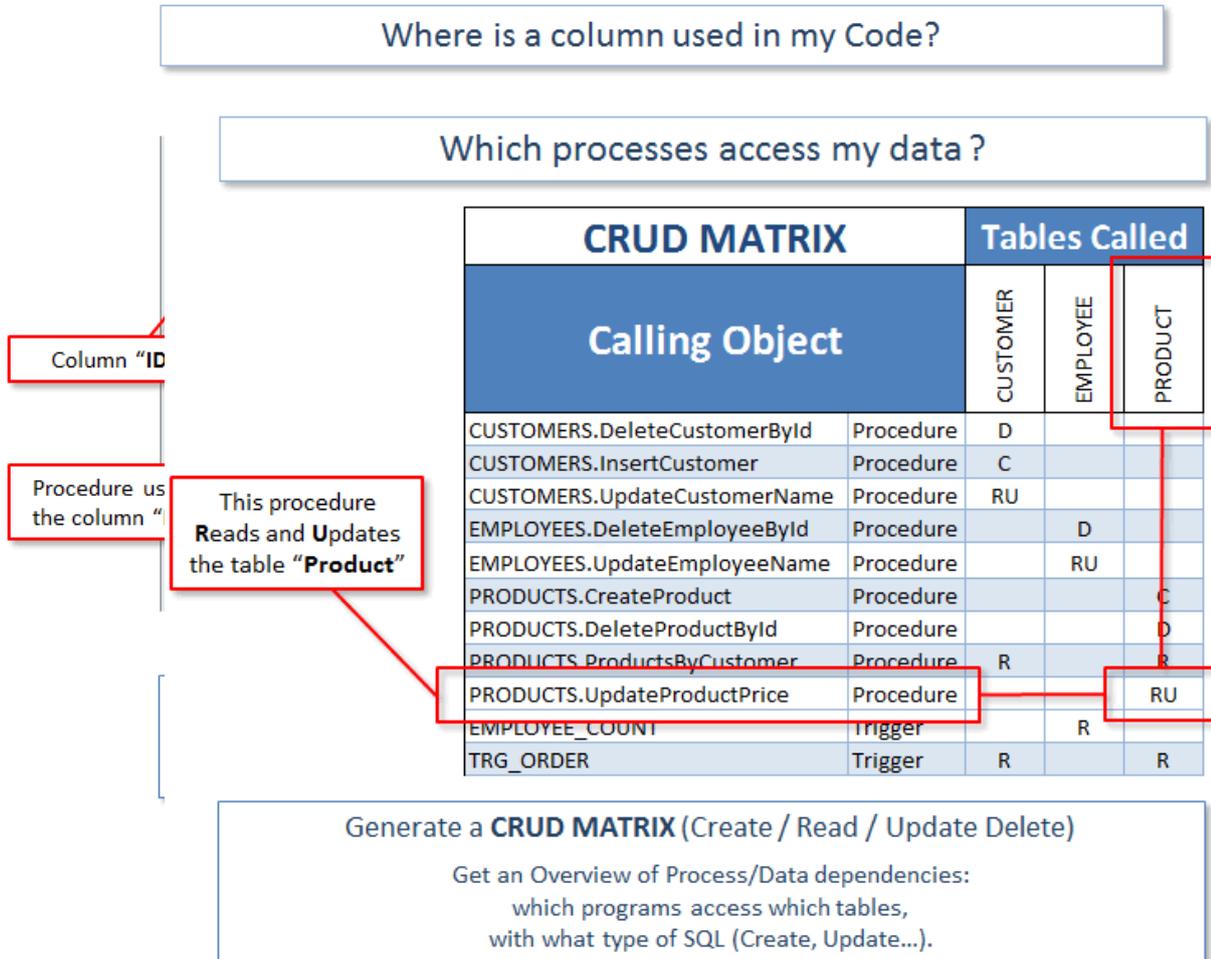
**変更すべき内容**（置換すべき関数、変更すべき列など）がわかったら、**その変更がどこに影響を与えるかを評価する必要があります**。これは、移行中に何も壊さないために非常に重要な影響分析です。

- 例えば、廃止された関数を置き換えるには、その関数を呼び出すすべてのプロシージャのコードを変更する必要があります。しかし、これらのプロシージャは他の場所で呼び出されている可能性があります。そのため、これらの変更が及ぼす影響をすべて確認する必要があります。
- 同様に、TEXT/NTEXT 型の列を VARCHAR(MAX)/NVARCHAR(MAX) に変更するには、その列を操作するプロシージャを変更し、上流/下流のコンポーネント（ビュー、インターフェースなど）を確認する必要があります。

ツールがない場合、これらの影響をマッピングするのは骨の折れる作業です。複数の依存関係（何が何を呼び出すか、どのテーブル/列を使用するか）を手作業で追跡する必要があります。チームにとって懸念されるのは、**間接的な参照を見逃し**、更新されていないコードの一部で、受け入れテストや本番環境において問題を発見することでしょう。

**Visual Expert** は、「X を変更するとどうなるか」といった質問に、**インタラクティブな影響分析**で答えます。

- オブジェクト（関数や列など）を選択して影響分析を実行すると、Visual Expert はそれを参照しているすべてのオブジェクトを一覧表示します。例えば、「列「顧客名」を変更すると、どのプロシージャ、トリガー、ビュー、関数が影響を受けるか？」や「関数 XYZ を呼び出しているのはどこか？どのスクリプトがそれを参照しているか？」といった質問です。



- このツールは、依存関係を E/R 図やマトリックスで表示することができます。例えば、その **CRUD マトリックス**は、データとそれを操作する関数を交差させ、各プロシージャによって実行される操作の種類（作成/読み取り/更新/削除）を明確に示します。
- Visual Expert は、手作業では数時間かかる作業をわずか数秒で実行します。これにより、チームは各変更の準備（必要な時間の見積もり、変更対象オブジェクトのリストアップ）を行い、見落としリスクも排除できます。隠れた依存関係も一切見落とされることはありません。

## ステップ 5：変更の実施

目的：変更を首尾一貫した安全な方法で適用すること。

分析段階の次は、**変更段階**（コードのリファクタリング、スキーマの変更）です。

**Visual Expert** は分析ツールであり、編集ツールではありません。したがって、変更は開発者が通常使用するコードエディタで行われます。ただし、Visual Expert は実装中に役立ちます。

- チームは、**変更を追跡するためのダッシュボード**としてこのツールを利用できます。変更を行った後、新しいコードを分析し、古い要素が一切存在しなくなったことを確認できます。この**品質チェック**は、実装中に定期的に行うことができます。
- Visual Expert は、初期スキーマと移行後スキーマを比較する**コード/スキーマ比較機能**も提供しています。このツールは、コードとオブジェクトの構造を理解し、**構造化された差異**レポート（削除または追加されたオブジェクト、プロシージャコードの変更、列のタイプ変更など）を生成します。このチェックリストにより、予定されていた各項目が確実に処理されていることを確認し、意図しない差異を検出することができます。

Compare different versions of your code

Select 2 versions of your code (Code Analysis)

Choose a Scope (all procedures in that case)

Objects changed listed here  
Green = new / Red = removed  
Blue = modified objects

For modified objects, changes are highlighted in the code  
Green = new / Red = removed

```
1 CREATE procedure ProductsByCustomer(@Customer_Id int)
2 AS
3 BEGIN
4 DECLARE @Cust_fname nvarchar(50)
5 DECLARE @Cust_lname nvarchar(50)
6 DECLARE @prod_name nvarchar(50)
7 DECLARE @prod_size nvarchar(50)
8 DECLARE @Prod_num int
9
10 DECLARE @r1 nvarchar(max)
11
12 DECLARE c1 cursor FOR
13 SELECT product.id Pid,
14        product.name prod_name,
15        sum (sales_order_items.quantity*unit_price)
16 FROM customer,
17        product,
18        sa
```

## ステップ 6：移行済みデータベースの技術文書

目的：完全かつ最新の、実用的なドキュメントを自動的に生成すること。

移行が完了したら、最新の**技術文書を作成する必要があります**。これは、メンテナンスのための現状把握と、実施された変更の追跡の両方に役立ちます。

しかし、スキーマや何千行ものコードを手作業で文書化することは、プロジェクト終了時には時間が不足していることが多いことを考えると、あまり現実的ではありません。この作業は時間がかかり、費用対効果が低く、エラーが発生しやすいものです。したがって、ドキュメント作成を自動化することが不可欠です。

デフォルトでは、ドキュメントには以下を含める必要があります。

1. データベースの構造（スキーマ、図）
2. データ辞書（列、型）
3. T-SQL コード（モジュール、パラメータ、呼び出しのリストと説明）、
4. 場合によっては、移行前後の比較情報

**Visual Expert** はコードのドキュメントを自動的に生成します。分析リポジトリから、アプリケーションを詳細に説明する **HTML ページ** を作成することができます。チームに新しく加入したメンバーは、コードの動作を推測する必要がなく、このリポジトリを参照することができます。

- 各プロシージャ/関数/トリガーについて、フォーマットされたソースコード、パラメータのリスト、説明、および**参照情報**（例：「このプロシージャは、このようなテーブルを呼び出し、このような他のオブジェクトによって呼び出されます…」）を明記します。これらの参照情報は、ドキュメント内でクリック可能なハイパーリンクとなっているため、ウェブサイトのよう**に閲覧**することができます。
- また、**特定のレポートを生成することも**できます。たとえば、移行中に変更されたすべてのオブジェクトのリストと新しいステータス、**CRUD マトリックス**、コンポーネント間の相互作用を示す呼び出し図などです。

**Visual Expert** は、**移行後のデータモデルのドキュメント作成にも**役立ちます。

- 移行が完了すると、**SQL Server のエンティティリレーションシップ図を更新し**、システム全体の**最新のスキーマを視覚的に**表現します。
- この図は調整が可能で、例えば、参照しやすいように**複数のテーマ別サブ図に分割**することができます。各サブ図は、**機能領域**（請求、顧客管理など）をカバーすることができます。
- 図が満足のものになったら、**Visual Expert は最終的なドキュメントに組み込むために、PDF または高解像度の画像として図をエクスポート**することができます。

データモデルの概要と詳細なコードのドキュメントを組み合わせることで、**Visual Expert** はデータベースの両面をカバーします。このツールを使用することで、チームに過大な負担をかけることなく、**プロジェクトの全体的な品質を向上させ**、最小限の労力で作業を行うことができます。

## ステップ 7：移行後の最適化

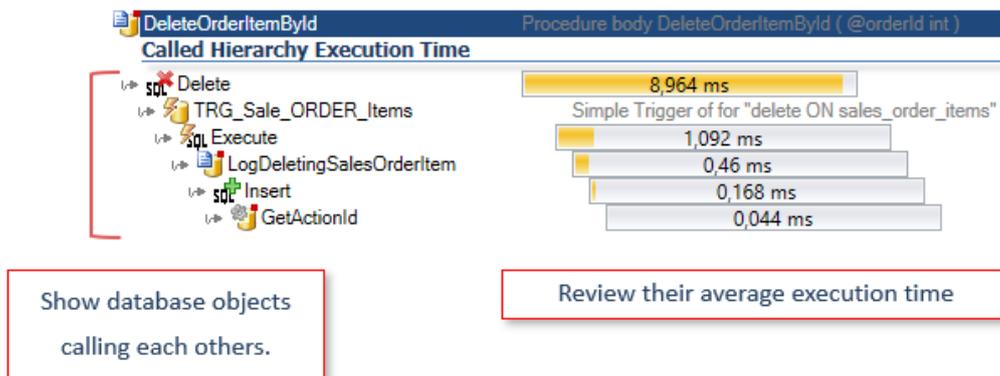
目的：移行後のデータベースのパフォーマンスと保守性を向上させること。

移行の成功は、バージョンアップだけで終わりではありません。データベースが最適に機能することも必要です。しかし、バージョンアップによって、**予想もしなかった問題**が発生する場合があります。移行は、以前のバージョンでは対処されなかったパフォーマンスの問題（インデックスの欠落、クエリの遅延など）に対処するチャンスでもあります。

こうした移行後の最適化は、時間がないために見逃されがちです。そのため、最も緊急性の高いものを特定する必要があります。何百ものクエリやプロシージャの中から、優先的に最適化すべきものはどれか？

**Visual Expert** には、SQL Server によって生成された実行統計情報を収集・分析する **パフォーマンス最適化** モジュールが組み込まれています。移行後にこれらの問題を修正することで、データベースのパフォーマンスと保守性が向上します。

- 平均実行時間が最も長いプロシージャ、または最も頻繁に実行されるプロシージャを特定することができます。また、プロシージャ内の各 SQL が消費する時間も表示できます。この情報により、たとえば、最もコストのかかる 10 件のクエリを特定することができます。
- また、コードの構造に関連するパフォーマンスも分析します。呼び出し関数のチェーンを含む処理を分解し、**実行時間付きの呼び出しグラフ**を生成します。これにより、処理を遅くしている関数を視覚的に特定することができます。



- Visual Expert はコードの品質もチェックし、**パフォーマンスに影響を与える特定の悪しき慣行**（例：SQL 結合の代わりに T-SQL でネストされたループを使用）を検出します。
- また、SQL クエリの実行を不必要に遅くしている、インデックスの欠落（インデックスがないにもかかわらず、SQL クエリの Where/Group by/Order by/Having 句で使用されているデータベースの列）を自動的に検出します。
- 最後に、Visual Expert は、検出した低速なクエリを分析することができます。SQL Server DBA は、クエリの実行プランを表示して、SQL Server による処理方法（インデックスの走査、完全スキャン、ハッシュ結合、ソートなど）を確認できます。これにより、分析から SQL の最適化へのアクションを迅速に移行することができます。

この機能の組み合わせにより、包括的な予防措置として、あるいは特定のボトルネックを解消するために、アプリケーションのパフォーマンスを向上させることができます。

また、Visual Expert を使用して、移行前後のパフォーマンス測定を行い、実施した最適化の効果を実際に評価することもできます。

## ステップ 8 : Visual Expert による移行後の継続的なモニタリング

目的 : T-SQL コードの品質とパフォーマンスを長期的に維持すること。

移行が完了したら、技術的な成果を維持することが重要です。Visual Expert は、SQL Server スキーマおよび T-SQL コードの変更を長期的に保護、文書化、および管理するのに役立ちます。

- **変更前の体系的な影響分析**

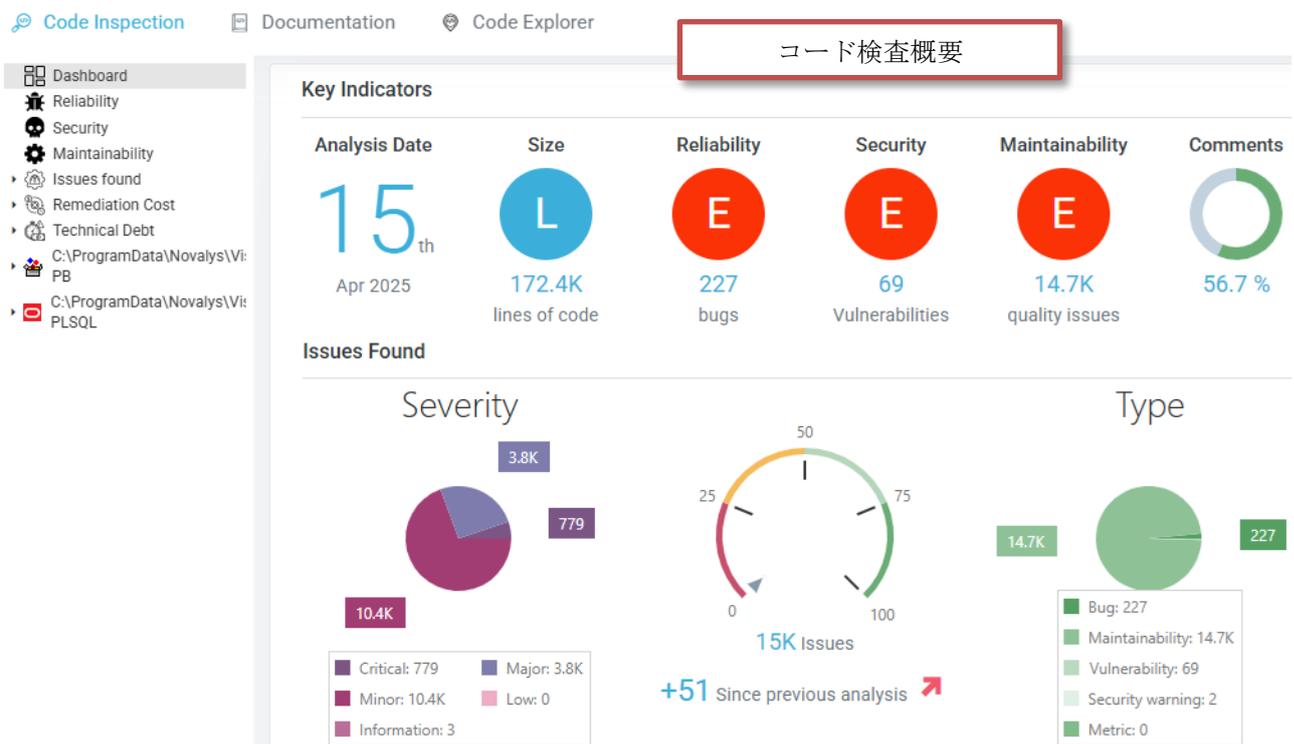
SQL Server データベースおよび T-SQL コードに大幅な変更を加える前には、必ず影響分析を実行することをお勧めします。この方法により、依存関係を網羅的に考慮し、リグレッションのリスクを最小限に抑えることができます。

- **定期的なパフォーマンスのチェック**

SQL プロシージャやクエリの実行時間を監視し、新たな問題を迅速に検出・対処することができます。例えば、パフォーマンスの低下を防ぐために新しいインデックスの作成を必要とするクエリの変更後に、こうした問題が発生する可能性があります。Visual Expert は、SQL Server の DBA が適切な措置を講じることができるよう、こうした問題を検出して通知します。

- **コードの品質とセキュリティの定期的なチェック**

Visual Expert は、コードを自動的かつ定期的に検査し、品質やセキュリティの問題の発生を検出することができます。たとえば、チームは、廃止されたデータ型や SQL Server 関数を習慣的に使用している場合があります。この定期的なチェックにより、長期的に高い品質とセキュリティレベルを維持することができます。



- スキーマおよびコードの変更の文書化された追跡

Visual Expert は、スキーマおよび T-SQL コードに加えられた変更を追跡するため、バージョン管理と変更のトレーサビリティが大幅に簡素化されます。

この継続的な追跡により、メンテナンスコストが削減され、将来の進化が確実なものとなり、チームは移行後の機能改善により一層集中することができます。

## 結論

SQL Server 2008/2012 から 2022 へのデータベース移行は複雑なプロジェクトですが、適切なツールを使用することで大幅に簡略化できます。Visual Expert を活用することで、開発チームと管理チームは、コード分析、スキーマ管理、スキーマモデリング、ドキュメント作成を網羅する**包括的なツールキット**を利用できます。

移行の各段階が容易になります：範囲の正確な特定、クリーンアップ、新機能への適応、制御された変更、技術文書化、継続的な最適化。

初期移行後も、Visual Expert は定期的な影響分析、随時パフォーマンスチェック、定期的なコード品質チェックにより、引き続き大きな支援を提供します。これらの機能により、SQL Server データベースの長期的な安定性、品質、パフォーマンスが保証され、移行プロジェクトの持続的な投資収益率が確保されます。

Visual Expert をコード編集ツールと組み合わせ、プロジェクトの各段階を厳密に管理することで、技術チームはリスクを軽減し、進化を確実にし、SQL Server データベースの将来の保守性を高めながら、野心的な移行を成功させることができます。

## リソース：

- [Visual Expert for SQL Server – 製品ページ](#)
- [Visual Expert for SQL Server – オンラインドキュメント](#)

## 段階別概要表 フェーズ別概要表

プロジェクトのフェーズ段階	Visual Expert の機能
1. ボリューム評価	<ul style="list-style-type: none"><li>- メトリクスの計算（コード行数、オブジェクト数）</li><li>- オブジェクトタイプ別インベントリ</li><li>- アプリケーション資産の概要表示</li><li>- ER 図（データモデル）の自動生成</li><li>- 複雑または孤立した領域の視覚的分析</li></ul>
2. コードのクリーンアップ	<ul style="list-style-type: none"><li>- 未使用（参照されていない）オブジェクトの特定</li><li>- コードからアクセスされたことのないテーブルの特定</li><li>- 重複、空、冗長なコードの検出</li></ul>
3. 廃止された要素の検出	<ul style="list-style-type: none"><li>- 廃止された関数/型の使用の検索</li><li>- コード分析ルールによる警告</li><li>- 自動化されたオブジェクトの特定</li><li>- ER モデルの分析による設計上の旧式化の検出</li><li>- モデルへの注釈付与による変更点の定義</li></ul>
4. 影響分析	<ul style="list-style-type: none"><li>- インタラクティブな影響分析（誰がどの要素を使用しているか？）</li><li>- CRUD マトリックス（どのテーブルを読み書きしているか？）</li><li>- 呼び出しおよび相互参照の図</li></ul>
5. 変更の実施	<ul style="list-style-type: none"><li>- 変更後の検証（すべてのポイントが適切に処理されているか）</li><li>- 移行前後のスキーマ/コードの比較</li></ul>
6. 技術文書	<ul style="list-style-type: none"><li>- コード文書の自動生成</li><li>- CRUD マトリックスの生成</li><li>- 呼び出し図の生成</li><li>- 移行後のドキュメント用に更新された ER 図</li><li>- データモデルのサブダイアグラムのエクスポートと作成</li></ul>
7. 移行後の最適化	<ul style="list-style-type: none"><li>- コードのパフォーマンス分析（平均時間、頻度）</li><li>- 関数呼び出しチェーンの分析</li><li>- 欠落インデックスの検出</li><li>- SQL クエリの分析とチューニング（実行計画）</li><li>- 最適化を反映したモデルの更新</li></ul>
8 - 移行後の継続的なモニタリング	<ul style="list-style-type: none"><li>- 変更前の体系的な影響分析</li><li>- 定期的なパフォーマンスのチェック</li><li>- 品質の定期的な検証</li><li>- スキーマおよびコードの変更の文書化</li></ul>

# 付録 A – 廃止または削除された要素の例 (SQL Server 2008/2012 → 2022)

この付録では、SQL Server のバージョンアップに伴い廃止または削除されたオブジェクト、構文、オプションの一般的な例を、SQL Server 2008/2012 データベースで頻繁に見られるものから紹介しています。Visual Expert によるインベントリ分析、影響分析、コード検査を補完し、バージョンアップの準備に役立つ実用的なチェックリストを提供することを目的としています。

## A.1 コンポーネントと機能 (プラットフォーム)

最近のバージョンで廃止されたもの

- 分散リプレイ (廃止)。
- Machine Learning Server (廃止)。
- Stretch Database (廃止済み、新規プロジェクトには導入不可)。

削除済み/利用不可 (例)

- SQL Server 32 ビットのインストール (機能が削除されました)。
- SQL Server Agent : ActiveX サブシステム (廃止)。
- 非常に古い互換性レベル (例: レベル 90) – 最近のバージョンではサポートされていません。

## A.2 データ型と関連オプション

- 従来の LOB タイプ: TEXT / NTEXT / IMAGE → VARCHAR(MAX) / NVARCHAR(MAX) / VARBINARY(MAX)。
- テーブルオプション「text in row」: 列を (N)VARCHAR(MAX) または VARBINARY(MAX) に移行して置き換える。
- TIMESTAMP (「rowversion」の意味): モデルおよびコードでは ROWVERSION 型を優先してください。

## A.3 LOB に関連する古い API および関数 (「テキストポインタ」)

- WRITETEXT / UPDATETEXT / READTEXT (非推奨)。
- TEXTPTR() および TEXTVALID() (非推奨)。

## A.4 システムプロシージャおよび管理コマンドの履歴

- sp\_attach\_db / sp\_attach\_single\_file\_db → CREATE DATABASE ... FOR ATTACH を使用してください。
- DEFAULT / RULE : CREATE DEFAULT / CREATE RULE および関連プロシージャ (sp\_bindefault、sp\_bindrule など) → DEFAULT および CHECK 制約を使用する。
- sp\_depends → 依存関係ビュー /DMV (例: sys.dm\_sql\_referencing\_entities、sys.dm\_sql\_referenced\_entities) を使用する。

- sp\_change\_users\_login → ALTER USER および最新のセキュリティオブジェクトを使用してユーザーを管理する。
- sp\_renamedb → ALTER DATABASE … MODIFY NAME を使用して名前を変更する。

#### A.5 近代化すべき Transact-SQL の慣習と構文（レガシーで遭遇するもの）

- INSERT/UPDATE/DELETE に対する SET ROWCOUNT → TOP（または適切な制限ロジック）を使用。
- WITH なしのテーブルヒント → WITH (… ) 構文を使用する。
- UPDATE/DELETE の FROM 内の NOLOCK / READUNCOMMITTED → 避けるべき（不整合のリスクあり）。これらのヒントを削除する。
- GROUP BY ALL（非推奨）。
- 番号付きプロシージャ（numbered procedures）（非推奨）。
- 3 部分または 4 部分で構成される列の参照 → 曖昧さを制限するため、2 部分で構成される名前（スキーマ.オブジェクト）を優先する。

#### A.6 バックアップ/復元およびセキュリティ（運用スクリプト）

- BACKUP/RESTORE の PASSWORD / MEDIAPASSWORD オプション：削除済みまたは削除予定。スクリプトではこれらのオプションの使用を避ける。
- 旧式の暗号化アルゴリズム（例：RC4）：非推奨。暗号化が必要な場合は、最新のアルゴリズム（例：AES）を使用してください。
- 古い SSL 設定：TLS を使用してください（SSL はサポートされなくなりました）。

#### A.7 バージョンアップ時の互換性と注意点（2008/2012 → 2022）

バージョンアップ時には、一部の旧互換性レベルや旧オプションがサポートされなくなりま  
す。Visual Expert によるオブジェクトのインベントリ、コードの検査、影響分析により、修  
正が必要な領域（関連するプロシージャ、トリガー、関数、ビュー）を迅速に特定し、置換  
の計画を立てることができます。

#### A.8 出典（Microsoft 参照）

- SQL Server 2022 で廃止されるデータベース エンジン機能 (Microsoft Learn)。
- SQL Server 2016/2017 で廃止されたデータベースエンジン機能 (Microsoft Learn)。
- SQL Server で廃止されたデータベース エンジン機能 (Microsoft Learn)。