

# テーブルのインデックス

主キーインデックス、外部キーインデックス、  
ユーザーインデックス

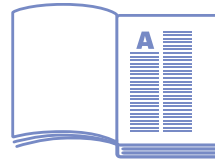
**GeneXus**<sup>TM</sup>

## インデックス

### データベースのインデックス



### 書籍のインデックス (索引)



データベースのインデックスは、書籍のインデックス (索引) に似ています。書籍の場合、全ページに目を通さなくても特定の内容を見つけることができます。

データベースの場合、インデックスはテーブル内の特定レコードへのクイックアクセスを可能にする構造であり、操作スピードを高め、テーブル全体を順番に確認せずにデータを探ることができます。

## 例

CUSTOMER

CustomerId	CustomerName
1	Jonathan
2	Alexander
3	Peter
4	Susan
5	Christopher
6	Ann

CUSTOMER

CustomerName に基づくインデックス

CustomerId	CustomerName
2	Alexander
6	Ann
5	Christopher
1	Jonathan
3	Peter
4	Susan

例として Customer テーブルを考えます。Christopher という名前に対応するレコードを探したい場合、1 行ずつ順番に確認する必要があります。これに対して、CustomerName に基づくインデックスが定義されている場合、テーブル全体を順番に確認しなくても、該当するレコードに直接アクセスできます。

インデックスの概念を理解できたところで、GeneXus によって自動的に作成されるインデックスと、開発者が必要に応じて独自のインデックスを作成する方法を見ていきます。

## GeneXus によって自動的に作成されるインデックス

Attraction トランザクション

Name	Type
Attraction	Attraction
AttractionId	Id
AttractionName	Name
AttractionPhoto	Image
CountryId	Id
CountryName	Name
CategoryId	Id
CategoryName	Name

Attraction テーブル

Attribute	Order
Attraction Indexes	
IAttraction	Primary Key
AttractionId	Ascending
IAttraction1	Foreign Key
CategoryId	Ascending
IAttraction2	Foreign Key
CountryId	Ascending

GeneXus で自動的に作成:

- 主キーインデックス
- 外部キーインデックス

GeneXus によって自動的に作成されるインデックスは、テーブルへの効率的なアクセスを可能にするだけでなく、効率的な参照整合性チェックを実行します。

Attraction トランザクションの構造を確認します。

- AttractionId が主キーです。
- CountryId と CategoryId が外部キーです。

[KB エクスプローラー] ウィンドウでは、トランザクションの下に、関連付けられているテーブルが表示されます。テーブルを開くと、[Indexes] タブに、GeneXus によって自動的に作成されたインデックスが表示されます。すべてのインデックスに、名前、インデックスが定義されている項目属性、そして順序があります。順序は昇順または降順であり、既定は昇順です。

**主キーインデックス:** 主キー AttractionId に対するインデックス IAttraction は、レコードの一意性を制御するために使用されます。つまり、2 つの観光名所が同じ識別子を持つことはありません。また、たとえば、その観光名所へのツアーを作成するタイミングで、ツアーで指定されている AttractionId の値が ATTRACTION テーブルに主キーとして存在していることのチェックにも使います。

主キーインデックスは次の用途で使用されます。

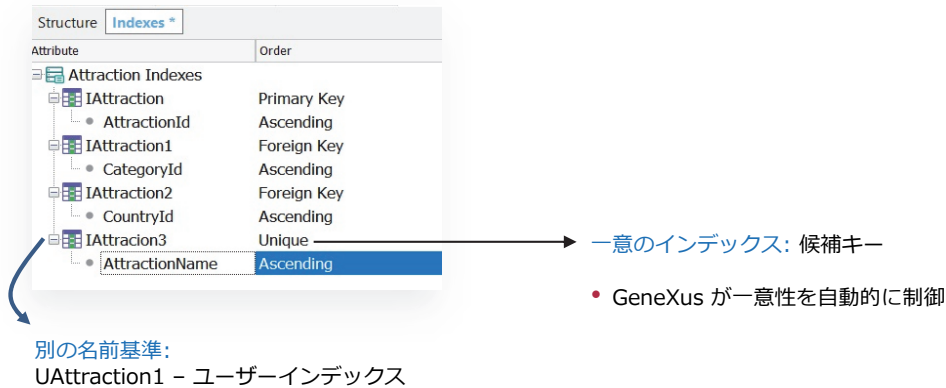
- 参照整合性チェックを効率的に実行する。
- 外部キーの値を挿入または変更するときに参照整合性チェックを実行する。

GeneXus では、トランザクションを識別する項目属性の主キーインデックスがすべて自動的に定義されます。

次に、IAttraction1 と IAttraction2 の各外部キーインデックスを見てみます。それぞれ外部キー CategoryId と CountryId が対象です。これらの外部キーインデックスも GeneXus によって自動的に作成され、効率的な参照整合性のチェックに使用されます。たとえば、識別子 2 のカテゴリを削除しようとする、このインデックスによって

値 CategoryId = 2 の関連レコードが Attraction にないことがチェックされます。

## ユーザーインデックス: 重複/一意



ここで、観光名所のリストをアルファベット順にする場合を考えます。観光名所の名前に対するインデックスがあると便利です。

そこで、AttractionName 項目属性に基づくユーザーインデックスを定義します。

このように、**ユーザーインデックス**は、開発者がさまざまな検索を効率的に行うために定義するものです。

ここで疑問が生じるかもしれません。なぜこのようなインデックスは GeneXus によって自動的に作成されないのでしょうか。それは単に参照整合性チェックに必要ではないからです。

テーブルは、必要に応じて任意の項目属性、または項目属性のセットで並べ替えが可能です。

インデックスは、繰り返しがある値について、重複を受け入れます。たとえば、繰り返される可能性のある人名を考えます。人名をアルファベット順にするためのインデックスが必要な場合、人名は繰り返しがありうるので、インデックスで重複する値を認める必要があります。

これに対して、インデックスを一意と指定した場合、インデックスを定義する項目属性、または項目属性のセットに値の繰り返しを認めないと定義することになります。値の一意性は GeneXus によって自動的に制御されます。

この場合、インデックスを定義する項目属性、または項目属性のセットを**候補キー**と呼びます。トランザクションの主キーではありませんが、主キーと同様に GeneXus によって一意性が制御されるからです。

## 実行時

The screenshot displays the GeneXus application interface during runtime. It features a table of attractions and a form for adding a new attraction. The table lists the following data:

Id	Name	Country Name	Category Name	Photo	City Name	UPDATE	DELETE
2	Eiffel Tower	France	Monument		Paris	UPDATE	DELETE
3	Great Wall	China			Beijing	UPDATE	DELETE
1	Louvre Museum	France	Museum		Paris	UPDATE	DELETE

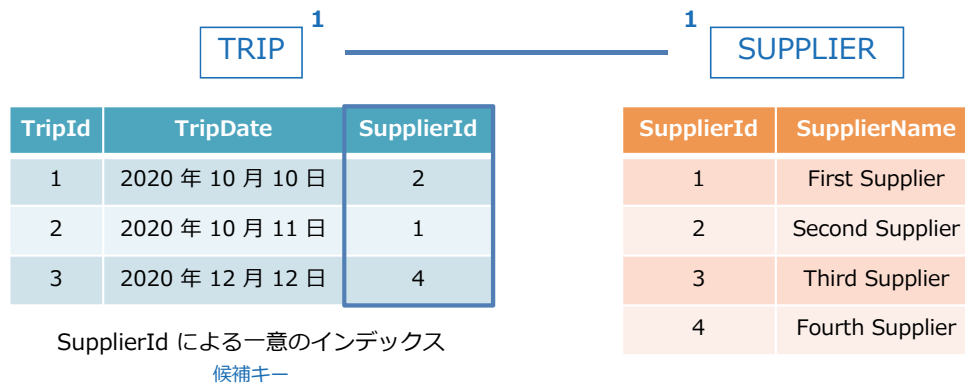
The form for adding a new attraction is shown below the table. It includes fields for Id, Name, Country Id, and Country Name. The Name field contains the text 'eiffel tower', and a yellow warning message 'Attraction Name already exists' is displayed next to it.

たとえば、エッフェル塔を入力するとします。

エッフェル塔という観光名所を新規に入力しようとする、すべて小文字で入力しても、GeneXus によって観光名所名の一意性が自動的に制御され「既に存在しています」というメッセージが表示されます。

同様に、対応する一意のインデックスを定義することで、同じ名前のカテゴリや国が入力されないよう制御できます。

## 1 対 1 の関係



この代理店では、とある 1 つのツアーを開催できる業者は 1 社のみで、一方、業者は 1 つのツアー以外、取りあつかえないという特殊なケースを考えてみましょう。

これは GeneXus でどのように表せるでしょうか。

1 対 1 の関係は、1 対 N の関係の一種と考えることができます。N は「多」を表しますが、この「多」には 1 も含まれるからです。

そこで、業者の Id 項目属性を Trip の外部キーとして追加します。これは 1 対 1 の関係であるため、逆方向も可能です。

ツアー業者が 1 社のみで、その業者はそのツアーのみを取り扱っているため、SupplierId の値が Trip で繰り返されていないことをチェックする必要があります。これはどのようにしたらよいでしょうか。

Trip で外部キー SupplierId に対して一意のインデックスを定義します。

この方法で、SupplierId 項目属性が **Trip の候補キー** となり、GeneXus によってその一意性が自動的に制御されます。

1 対 1 の関係については、また後ほど説明します。





動画	<a href="https://www.genexus.com/community-and-support-jp/training?ja">https://www.genexus.com/community-and-support-jp/training?ja</a>
ドキュメント	<a href="http://wiki.genexus.jp/">http://wiki.genexus.jp/</a>
認定資格	<a href="https://training.genexus.com/certifications">training.genexus.com/certifications</a>