

テーブルのインデックス

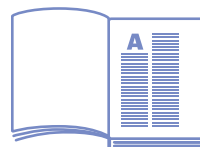
主キーインデックス、外部キーインデックス、ユーザーインデックス

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)



一度、GeneXus に特化した話ではなく、一般的な「データベースにおけるインデックス」について説明し、この後の説明で認識の相違が無いようにします。

データベースにおける「インデックス」は、書籍の「インデックス（索引）」に似ています。

もし、書籍にインデックスがない場合、読みたい内容が記載されたページを見つけるためには、すべてのページに目を通す必要があります。

一方、インデックスがあれば、この中から読みたい内容が記載されたページを特定し、効率的に書籍内を読むことができます。

データベースの場合、インデックスがあることで、テーブル内の特定レコードへアクセスが容易になり、データベースを対象にした操作速度を向上させ、テーブル内の全件を順番に参照する必要がなくなります。

インデックスに基づく参照

CUSTOMER
インデックスなし

CustomerId	CustomerName
1	ジョナサン
2	アレクサンダー
3	ピーター
4	スーザン
5	クリストファー
6	アン

CUSTOMER
CustomerName に基づくインデックス

CustomerId	CustomerName
1	ジョナサン
2	アレクサンダー
3	ピーター
4	スーザン
5	クリストファー
6	アン

例えば、6 件のデータが登録された Customer というテーブルがあったとします。この中で、CustomerName に、「クリストファー」というデータが登録されたレコードを探す場合、インデックスがない場合には、1 行ずつ順番に参照し、内容を確認する必要があります。

もし、CustomerName に基づくインデックスが定義されている場合、1 件目から参照する必要はなく、直接該当するデータを参照することができます。

GeneXus では、「自動的に作成されるインデックス」と「開発者が作成するインデックス」の 2 種類があります。これらについて詳細を説明していきます。

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

名前	タイプ
Attraction	Attraction
AttractionId	Id
AttractionName	Name
CountryId	Id
CountryName	Name
CityId	Id
CityName	Name
CategoryId	Id
CategoryName	Name
AttractionPhoto	Image

ATTRACTION テーブル

項目属性	順序
Attraction インデックス	
IAttraction	主キー
AttractionId	昇順
IAttraction2	外部キー
CategoryId	昇順
IAttraction1	外部キー
CountryId	昇順
CityId	昇順

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

それでは、まずは GeneXus によって「自動的に作成されるインデックス」についてです。GeneXus が、インデックスを作成する理由は、テーブルのアクセスを効率化するためだけでなく、参照整合性のチェックも効率化するためです。

作成されたインデックスは、ナレッジベース内のテーブルオブジェクトを開き、[Indexes] エLEMENTで確認できます。
インデックスの定義には、名前と種類があり、紐づくものとして、項目属性がある。項目属性には、順序が指定でき、インデックスにより、昇順で並び替えるか、降順で並び替えるかを指定できる。
この既定値は、昇順です。

自動的に作成されるインデックスは、次の2種類です。

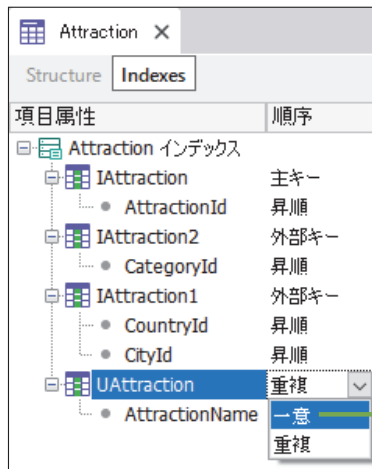
主キーインデックス：

主キー項目属性が自動的に紐づくインデックスです。
このインデックスによって、テーブルに登録された主キー項目属性の一意性が制御されます。
また、新しいレコードの登録処理を開始した際に、入力された主キーの値が登録済みであるかチェックを行う際にも、このインデックスが利用されます。
その他、別のテーブルから参照される際にも利用され、参照整合性チェックの効率化が行われます。
別のテーブルから参照されるケースとして、このテーブルの主キー項目属性が、別のテーブルで、外部キーとして利用されている場合、入力された値が、このテーブルで登録されたものか、参照整合性のチェックを行います。

外部キーインデックス：

外部キー項目属性が自動的に紐づくインデックスです。
このインデックスによって、紐づけられた項目属性が、主キーのテーブルで、データの削除が行われた場合、外部キーとして利用されていないか、参照整合性のチェックを行うために利用します。

ユーザーインデックス: 重複/一意



一意のインデックス: 候補キー

- GeneXus が一意性を自動的に制御



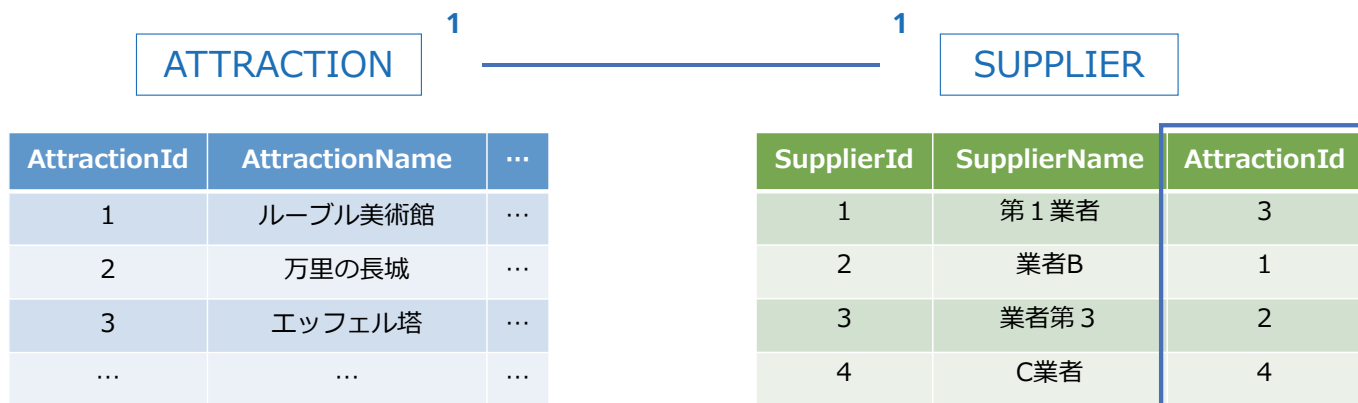
前述の通り、インデックスは、開発者によって作成することも可能です。
このような開発者によって作成するインデックスは、「ユーザーインデックス」と呼ばれます。

任意の項目属性（またはその組み合わせ）に対し、インデックスを作成し、参照の効率化を行うことができます。
また、ユーザーインデックスに設定できる種類として、「重複」（既定値）と「一意」があります。

「重複」の場合、定義されたインデックスは、参照の効率化に利用されるものとなります。
「一意」の場合、定義されたインデックスは、参照の効率化に加え、紐づけられた項目属性（またはその組み合わせ）が、テーブル内で一意となるように制限を行います。
つまり、主キーインデックスと同様の制御が行われるようになります。

一意のインデックスに紐づけられた項目属性は、主キーではないですが、主キーのように一意の値しかとらないため、「候補キー」と呼ばれます。
この結果、一意のインデックスが設定された項目に、既に登録済みの値を入力した場合、自動でエラーが発生するようになります。

1 対 1 の関係



AttractionId による一意のインデックス

この一意のユーザーインデックスを利用することで、1 対 1 の関係を実装することができます。

実装するためには、外部参照キーが定義されている側で、ユーザーインデックスを作成し、外部参照の項目属性を紐づけ、一意とします。

この結果、外部参照する際に、選択した値は、他のレコードで利用されている場合、存在している旨のエラーとなり、登録できなくなります。

その結果、2 つのテーブル間には、関係性ができますが、外部参照として利用される回数は、1 度だけとなるため、1 対 1 の関係となります。

*GeneXus*TM

training.genexus.com
wiki.genexus.com