データベースの正規化

ここには請求書を表すテーブルがあります。これをデータベースに保存すると、スペースと効率が無駄になります。

ここに含まれる情報の一部は冗長であり、何度も記録する必要はありません。これは「非正規化」データベースと呼ばれます。

データベースを「正規化」するということは、情報を複数のテーブルに分割し、特定の規則に従ってそれらのテーブルに含まれる情報を平滑化することを意味します。

1NF

> - 各エントリは「アトミック」である必要があります(列には 1 つの情報のみが含まれる必要があります)。
> - すべて「同じ型」の値です(oat と int、円とドルを混在させることはありません)。

正規化前 :表のエラー 正規化後 :表NF1

ここにあります:

- 「姓」と「名」を分離し、「住所」、
- 「市」、「州」、「郵便番号」を分離し、「セント」単位だっ
- た価格を「ドル」単位に変更しました

2NF

> - テーブルは NF1 規則に準拠する必要があります。
> - 各非キー属性は、主キーを基準として完全に機能する必要があります。 （つまり、主キーを基準とした場合、各列は主キーと一意に関連付けられている必要があります。）

候補キーを含むテーブル

ここには、「主キー」または「候補キー」として機能できる 3 つの列があります。

「請求書番号」、「顧客番号」、「在庫番号」。

今回の場合、 「請求書番号」と「在庫番号」の「複合キー」を作成できます。

「主キー」ができたので、すべての情報が本当にこのキーに依存しているかどうかを確認しましょう。
よく見ると、重複している列が8つあり、それらは「請求書番号」にのみリンクされています。そのため、重複を避けるため、テーブルを2つに分割します。

候補キーを含むテーブル 候補キーを含むテーブル

請求書番号は両方に表示されるため、2 つのテーブル間のリンクを作成できます。

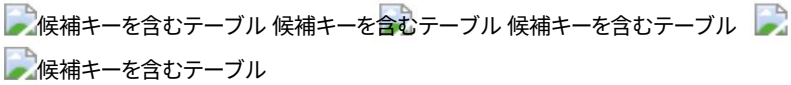
3NF

> - テーブルは NF1 および NF2 の規則に準拠する必要があります。

> - 非キー属性は、キー候補に推移的に依存することはできません。 （つまり、キー候補に依存する列がある場合は、新しいテーブルを作成する必要があります。）

前の例では、 「主キー」として機能しうる別の情報、「顧客番号」が見つかりました。これは「候補キー」です。ただし、顧客に関連するすべての情報は、この顧客番号に依存します。

また、 「商品名」とその「価格」は「商品番号」のみに依存していることにも注目してください。次に、これら2つのテーブルを2つに分割して、以下のデータを取得します。



そして、それらは常に異なるキーを介して相互にリンクされています。

正規化は最大 5 つ（およびオプションの正規化が 1 つ）ありますが、最初の 3 つを尊重することで、機能的で効率的なデータベースを実現できます。

正規化を減らすことが正常である特定のケースもいくつかあります。