

演習8-1 ビジネスロジックの実装方法の検討

問い1 ビジネスロジックをアプリケーションコードで実装することの是非

理想は様々な条件をテーブルの制約として定義することだと考える。しかしながら、複雑な条件（複数のデータの関係性を表すような条件、どちらがより多いとか。）をテーブルの制約で表現しきることはできないと考える。そういった場面ではアプリケーションコードで実装することは許されるべきではないかと思う。ただし、主キーや外部キーなどのSQLを実行するときに必要な制限に関してはDB側で実装すること。

また、データベース側で制限をかけた際に、不正なデータが入力されたときに返却されるエラーメッセージがDB側で決められているため、ユーザーにとってエラーの理由が分かりにくいこともある。

問い2 ビジネスロジックをデータベースの「トリガー」で実装することの是非

トリガー： 特定のテーブルに対する操作（挿入・更新・削除）を契機として、あらかじめ定義された処理を自動的に実行する機能のこと

必ずしも、意図したタイミングでトリガーが掛かるとは限らないのではないか、その際には適切なビジネスロジックを課すことができないと思う。また、ビジネスルールを制御するロジックがアプリケーション側と、データベース側のどちらにも存在するような状態は、なにか変更や問題があった際に両方を考慮して修正を行わなければならないため、管理や全体のロジックの理解が困難になり保守性が低いと感じる。

そのため、アプリケーション側で実現可能なロジックであればトリガーで実装するべきでないと考える。

演習8-2 一時テーブル

- ・一時テーブルとは、セッション単位で利用できるテーブルのことで、セッションが切れると自動的に一時テーブルは削除される。

- ・一時テーブルを利用するには、CREATE TEMPORARY TABLES権限（一時テーブルの作成・データの挿入・INDEXの追加などの操作ができる）を持つユーザーがCREATE TEMPORARY TABLE構文を実施する必要がある（mysql）

利用に関する是非

例えば、一時テーブルの中で沢山のデータの追加や削除が行われた場合、統計情報の取得のタイミングに気を付けなければならない、またその統計情報の取得にも時間がかかる（テーブルの量とかにも依るが）のでそういった時間も考慮しなければならないため、使いにくいのではないだろうか。一時テーブルを利用すること自体を良くないとは思わないが、あまり利用しない方が良いのではないかと感じる。