

データアクセス層の品質確保に向けた SQL 精査の取り組み SQL Inspection for Quality Assurance of Data Access Layer

七澤 尚資 堀野 智久 遠藤 浩†

Naosuke Nanasawa, Tomohisa Horino, Hiroshi Endoh

1. はじめに

リレーショナルデータベースを用いた業務アプリケーションの開発では、開発効率や保守性の向上のために業務層とデータアクセス層を分離した開発方式を採用している。データアクセス層の開発では、データアクセス時の性能問題や検索条件の設定不具合等を早期に検出するため、事前に定められた機能面・非機能面のチェックルールに基づいて SQL(Structured Query Language)の設計内容を精査する。従来の精査作業は開発者による属人的な目視確認で行われており、効率化や精度向上が課題だった。

本稿では業務アプリケーションで実行する SQL の品質確保において、SQL の設計内容の精査作業をツールで効率化した取り組みを述べる。

2. データアクセス層の開発方式

弊社では、業務アプリケーションの開発において 3 階層アーキテクチャ(図 1)を採用し、プレゼンテーション層(P 層)には画面表示処理、ファンクション層(F 層)には業務処理、データアクセス層(D 層)にはデータベースアクセス処理を実装する。データベースアクセス処理と業務処理を分離することで、業務間でのデータベースアクセス処理の共有が容易となる。また、開発下流工程でデータベースアクセス処理の設計変更や性能改善等の改修作業が発生した場合に、修正範囲を D 層内に閉じることができる。

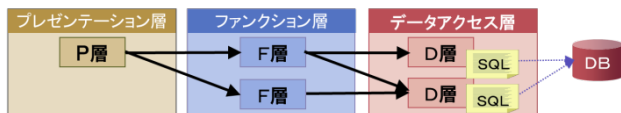


図 1: 3 階層アーキテクチャ

D 層の開発は、開発者が DDL(Data Definition Language)や DML(Data Manipulation Language)の設計情報を専用ツール(SQL 定義ツール) [1]に入力することで、ツールが D 層プログラムを自動生成する(図 2)。開発者によるコーディングが不要のため、SQL の標準化や D 層プログラムの品質向上が可能である。

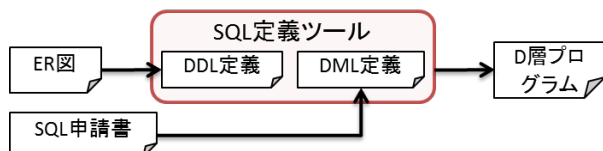


図 2: D 層プログラムの生成方式

3. データアクセス層の開発の流れ

D 層開発ではシステム全体で統制をとるために専任要員での体制確保を推奨している。D 層開発の流れは次の通りである(図 3)。

- ① P/F 層開発チームは SQL を申請する。
- ② D 層開発チームは申請された SQL を精査し、内容に問題がある場合は申請者との個別レビューで開発可否を審査する。
- ③ SQL の設計情報を SQL 定義ツールに入力し、D 層プログラムの作成とテストを行う。
- ④ 作成した D 層を P/F 層開発チームへ配布する。

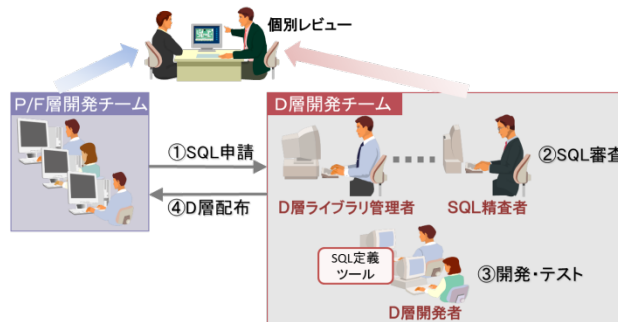


図 3: D 層開発の流れ

D 層開発における SQL 審査では、申請 SQL の設計内容を精査し、コーディング基準に違反した SQL や実運用で性能上のリスクがある SQL を早期に抽出する[2]。D 層は複数の F 層から共有され不具合発生時の影響が大きいため、設計段階で内容を精査して品質確保を図っている。

4. SQL 精査の内容

SQL の精査では申請 SQL のコーディング内容、既存 D 層との重複・類似をチェックする(表 1)。SQL のチェック後は、チェック結果を確認して開発可否を審査する(図 4)。

表 1: SQL 精査のチェック内容

No	種別	チェック内容
1	コーディング	コーディング基準に違反していないこと
2	重複・類似	代替可能な D 層が存在しないこと

コーディングのチェックでは、申請 SQL がコーディング基準に違反していないかを確認する。コーディング基準では性能や保守性を確保するための記述ルールを定めており、社内ノウハウをベースとした標準規則と、案件固有の規則で構成されている。チェックの結果、業務上必要だが性能問題等につながる可能性がある場合は、申請者との個別レビューで実現方式の妥当性や性能要件達成の見通しを確認し、未達の場合は実現方式の代替手段を検討する。性

†株式会社日立製作所 ICT事業統括本部 システム&サービス事業管理統括本部 生産技術本部, Software Engineering Division. Systems & Services Business Management Division. Information & Communication Technology Business Division. Hitachi, Ltd.

能上のリスクがある SQL を申請内容通りに実装する場合は性能の重点管理業務と位置付けてプロジェクトで管理し、開発時の性能検証状況をフォローする。

重複・類似のチェックでは、同一の結果が得られる SQL が既に開発済みかを確認し、開発済みならば該当 SQL を返却して D 層の共有化を促す。SQL の記述が部分的に異なる類似 SQL については、代替候補の提示や申請 SQL の再確認依頼を行うことで更なる共有化を検討する。

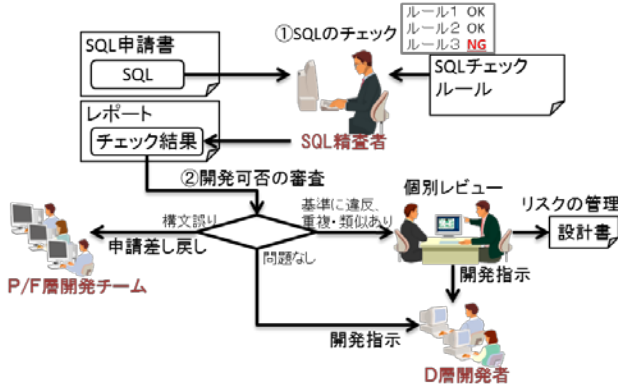


図 4：SQL 審査の流れ

SQL 精査では SQL を机上でチェックする作業に時間を要するため、作業の効率化が課題だった。

5. SQL 精査の自動化

SQL の精査作業を効率化するため、SQL 定義ツールの精査機能で作業を自動化した。精査機能では申請 SQL を構文解析し、単純な文字列だけでなく、句や述語を意識した構文の評価を行う。

構文解析では SQL をトークン(構成する単語の最小単位)に分解し、句や述語をノードとした構文木を作成する。構文木の特定のノードから評価対象のトークンを検索し、SQL を構文ベースで比較する (図 5)。

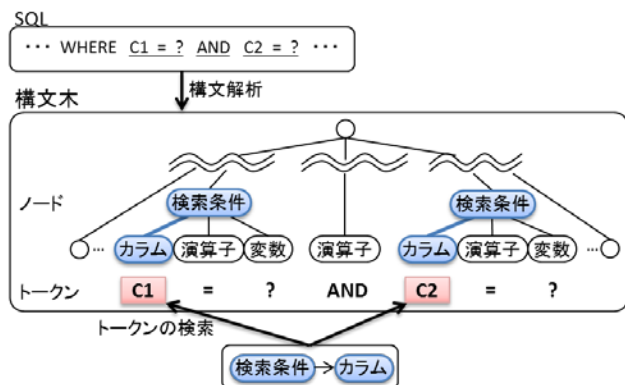


図 5：自動化の実現方式

コーディングのチェックでは、精査対象 SQL の構文木からコーディング基準の違反構文を探索して違反の有無を評価した (図 6)。精査機能では社内ノウハウをベースとした標準規則のチェックを自動化し、案件固有の規則は特定の構文で任意のカラムの利用有無を評価する機能を実装した。



図 6：コーディングのチェック

重複・類似のチェックでは、申請 SQL と既に開発された SQL の構文木をトークン単位で比較し、同一の場合は重複と判定する (図 7)。構文が部分的に異なる SQL は類似とみなし、類似の D 層を相違点とともに代替候補として提示する。これにより、開発者が D 層の共通化や申請 SQL の妥当性確認に必要な情報を提供する。

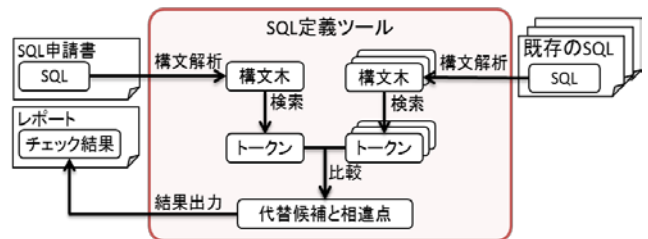


図 7：重複・類似のチェック

6. 自動化の評価

SQL の精査作業について、精査機能による作業時間削減率を試算した (表 2)(表 3)。実際の開発案件で試算したところ、案件のチェックルール 27 項目のうち 13 項目を自動化し、精査に関する作業時間を従来と比べて 59% 削減できた (表 3)。

表 2：チェック毎の作業時間削減率

No	種別	作業時間		作業時間削減率
		精査機能未使用	精査機能使用	
1	コーディング	35 秒	3 秒	91%
2	重複・類似	242 秒	17 秒	93%

表 3：精査作業全体の作業時間削減率

No	チェックルール		作業時間		作業時間削減率
	全体	自動化	精査機能未使用	精査機能使用	
1	27 項目	13 項目	653 秒	268 秒	59%

7. まとめ

D 層開発における SQL の精査作業はツールによる自動化で効率化できる見通しが得られた。

今後は実際の開発現場における作業効率および品質確保の適用効果を評価し、開発現場の要望を軸に自動化可能なチェックルールを拡充して D 層開発の更なる効率化を図っていく。

参考文献

- [1] 堀野智久、他『データベースセントリックに行うモデルベースの開発手法』、情報処理学会第 67 回全国大会。
- [2] 堀野智久、他『大規模プロジェクトにおけるデータアクセス機能の効率的な開発方式』、情報処理学会第 70 回全国大会。