

DBA

チームメンバー：渡邊諒（2442102）・田中誠真（2442054）

データベース環境と選定理由

採用技術:

DBMS: PostgreSQL 15.x

インフラ: Dockerコンテナ

選定理由:

堅牢性: ACID特性に準拠し、業務データの整合性を確実に守るため。

拡張性: 複雑なクエリや大量データの集計処理に強い。

データ永続化:

Docker Volume (postgres_data) をマウントし、コンテナのライフサイクルとは独立してデータを永続保存する構成を採用。

E-Rモデルとスキーマ設計

概要: 業務要件を分析し、以下の主要エンティティを定義。

主要テーブル:

Users: ユーザー認証情報(Django標準拡張)。

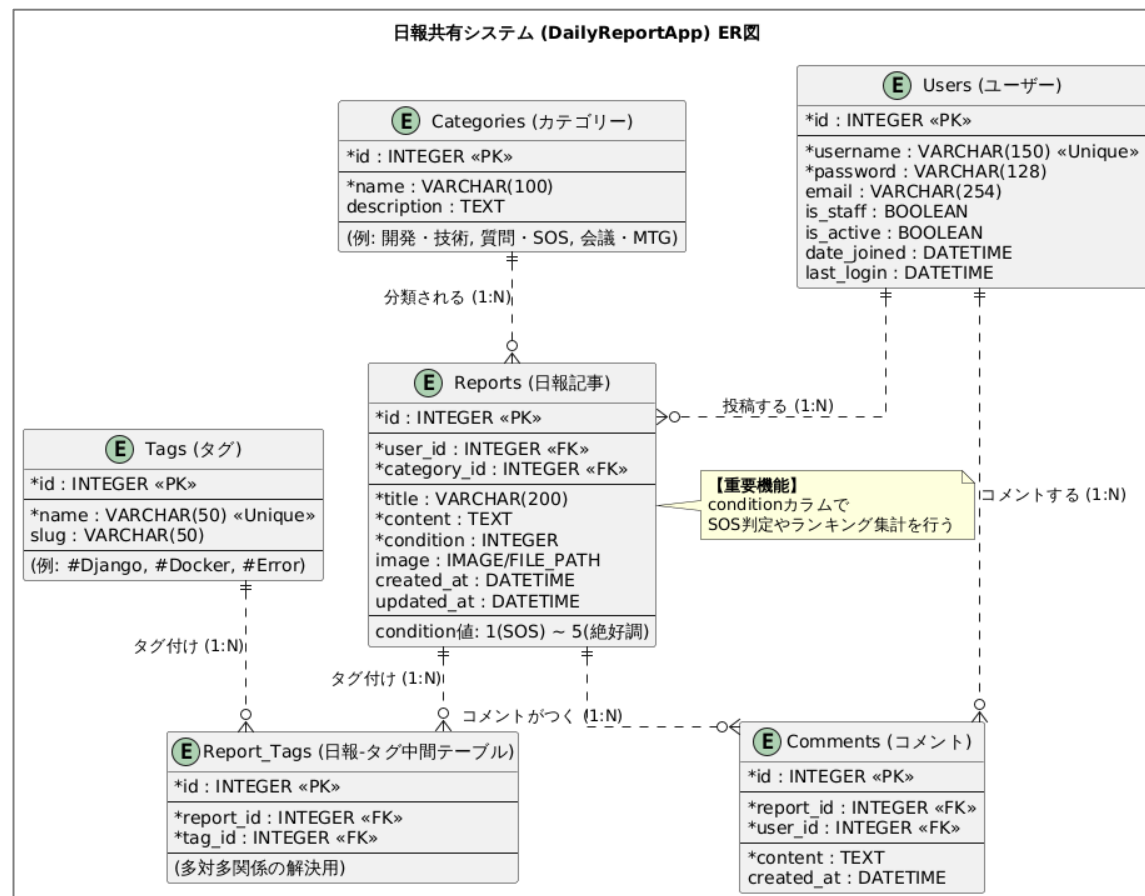
Reports: 日報のメインデータ。condition(数値)や image(パス)を保持。

Categories / Tags:

マスターデータとして独立させ、検索効率と保守性を向上。

E-Rモデルとスキーマ設計

ビジュアル:



正規化とリレーション設計

正規化への取り組み:

繰り返し項目(タグなど)や、重複する属性(カテゴリー名)を別テーブルに切り出し、第3正規形(3NF)まで正規化を実施。データの冗長性を排除。

多対多(Many-to-Many)の解決:

課題: 1つの日報に複数のタグがつき、1つのタグは複数の日報につく。

解決策: 中間テーブル Report_Tags を物理実装し、関連を管理。

これにより、タグごとの記事抽出や、記事ごとのタグ一覧取得を高速に行える設計とした。

データ整合性の担保 (Constraints)

実装した制約:

主キー制約 (Primary Key): 全テーブルにIDを設定し、行の一意性を保証。

外部キー制約 (Foreign Key):

Reports テーブルにおける user_id や category_id に設定。親データが存在しない子の登録を防ぎ、参照整合性を維持。

ユニーク制約 (Unique):

ユーザー名 (username) やタグ名 (name) の重複をデータベースレベルで禁止。

データ整合性の担保 (Constraints)

トランザクション:

記事投稿とタグ保存の処理において、原子性 (Atomicity) を保証するためトランザクション制御を適用。