

データベース設計書

TechFeed Hub - データベース設計書

1. 概要

1.1 プロジェクト概要

TechFeed Hubは、企業・個人のテックブログ情報を効率的に収集・閲覧できるWebサービスです。エンジニアの学習・情報収集体験の向上を目的としています。

1.2 データベース設計方針

- ・ **セキュリティ・バイ・デザイン**: 設計段階からセキュリティ要件を組み込み
- ・ **スケーラビリティ**: 大量の記事データとユーザーデータに対応
- ・ **パフォーマンス**: 高速な記事検索・フィルタリング
- ・ **運用性**: 監視・トラブルシューティングを考慮した設計
- ・ **柔軟性**: 企業ブログと個人ブログの両方に対応
- ・ **可観測性**: 包括的な監査ログとモニタリング

1.3 技術スタック・セキュリティ要件

- ・ **データベース**: PostgreSQL (Amazon RDS)
- ・ **主キー**: ULID (セキュリティと性能の両立)
- ・ **暗号化**: 保存時・転送時暗号化
- ・ **パスワード**: bcrypt (cost=12)
- ・ **セッション**: 安全なトークン管理
- ・ **監査**: 包括的なセキュリティログ

2. セキュリティ設計原則

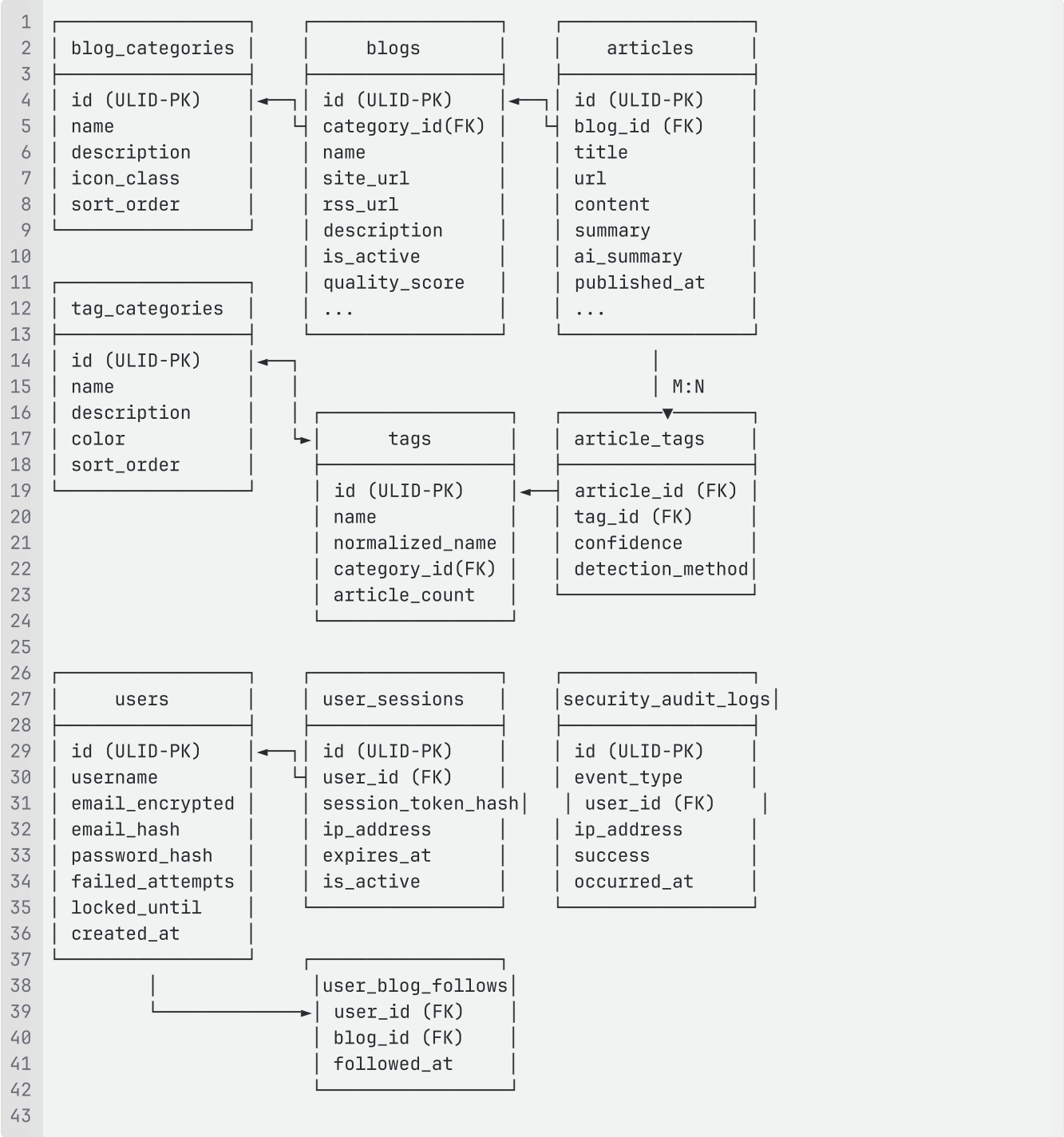
2.1 適用原則

- ・ **最小権限の原則**: 不要なアクセス権限の排除
- ・ **防御の多層化**: 複数のセキュリティレイヤー
- ・ **ゼロトラスト**: 全入力を検証・制限
- ・ **暗号化**: 個人情報・機密情報の保護
- ・ **監査**: 全重要操作のログ記録

2.2 ULID採用理由

- ・ **セキュリティ**: 推測不可能なID
- ・ **性能**: 時系列順でのインデックス効率化
- ・ **分散対応**: 重複なしでの生成保証

3. ER図



4. テーブル設計詳細

4.1 ULID生成関数

```

1  -- ULID生成関数 (PostgreSQL)
2  CREATE EXTENSION IF NOT EXISTS "pgcrypto";
3
4  CREATE OR REPLACE FUNCTION generate_ulid() RETURNS CHAR(26) AS $$
5  DECLARE
6      -- Crockford Base32エンコーディング文字セット
7      encoding BYTEA = '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ';
8      timestamp BIGINT;
9      unix_time BIGINT;
10     output TEXT = '';
11 BEGIN
12     -- 現在のUnixタイムスタンプ (ミリ秒)
13     unix_time = FLOOR(EXTRACT(EPOCH FROM NOW()) * 1000);
14
15     -- タイムスタンプ部分 (10文字)
16     FOR i IN 0..9 LOOP
17         output = CHR(GET_BYTE(encoding, (unix_time >> (45 - (i * 5))) & 31)) || output;
18     END LOOP;
19
20     -- ランダム部分 (16文字)
21     FOR i IN 0..15 LOOP
22         output = output || CHR(GET_BYTE(encoding, FLOOR(RANDOM() * 32)::INT));
23     END LOOP;
24
25     RETURN output;
26 END;
27 $$ LANGUAGE plpgsql VOLATILE;
28

```

4.2 ブログ管理テーブル群

4.2.1 blog_categories (ブログカテゴリマスタ)

カラム名	データ型	制約	説明
id	CHAR(26)	PK, DEFAULT generate_ulid()	カテゴリID (ULID)
name	VARCHAR(50)	NOT NULL, UNIQUE	カテゴリ名
description	TEXT	-	カテゴリ説明
icon_class	VARCHAR(50)	-	アイコンクラス (FontAwesome 等)
sort_order	INTEGER	DEFAULT 0	表示順序
created_at	TIMESTAMP WITH TIME	DEFAULT NOW()	作成日時

	ZONE		
created_by	CHAR(26)	-	作成者ID

4.2.2 blogs (ブログ情報)

カラム名	データ型	制約	説明
id	CHAR(26)	PK, DEFAULT generate_ulid()	ブログID (ULID)
name	VARCHAR(255)	NOT NULL	ブログ名
site_url	VARCHAR(500)	NOT NULL	サイト URL
rss_url	VARCHAR(500)	NOT NULL, UNIQUE	RSS/Atomフィード URL
rss_url_hash	CHAR(64)	NOT NULL, UNIQUE	RSS URLのSHA-256ハッシュ
description	TEXT	-	ブログ説明
favicon_url	VARCHAR(500)	-	ファビコンURL
category_id	CHAR(26)	FK	カテゴリID
language	VARCHAR(10)	DEFAULT 'ja'	言語コード
country	VARCHAR(10)	DEFAULT 'JP'	国コード
quality_score	DECIMAL(3,2)	DEFAULT 0	品質スコア (0.00-5.00)
content_frequency	VARCHAR(20)	DEFAULT 'UNKNOWN'	更新頻度
is_active	BOOLEAN	DEFAULT true	アクティブフラグ
fetch_interval_minutes	INTEGER	DEFAULT 60	取得間隔 (分)
fetch_timeout_seconds	INTEGER	DEFAULT 30	タイムアウト (秒)

max_articles_per_fetch	INTEGER	DEFAULT 50	1回の取得記事上限
last_fetched_at	TIMESTAMP WITH TIME ZONE	-	最終取得日時
last_successful_fetch_at	TIMESTAMP WITH TIME ZONE	-	最終成功取得日時
last_error_message	TEXT	-	最終エラーメッセージ
consecutive_error_count	INTEGER	DEFAULT 0	連続エラー回数
total_articles_count	INTEGER	DEFAULT 0	総記事数
avg_fetch_duration_ms	INTEGER	DEFAULT 0	平均取得時間 (ms)
success_rate_percentage	DECIMAL(5,2)	DEFAULT 100.00	成功率 (%)
created_at	TIMESTAMP WITH TIME ZONE	DEFAULT NOW()	作成日時
updated_at	TIMESTAMP WITH TIME ZONE	DEFAULT NOW()	更新日時
created_by	CHAR(26)	-	作成者ID
updated_by	CHAR(26)	-	更新者ID

4.2.3 blog_fetch_logs (ブログ取得履歴)

カラム名	データ型	制約	説明
------	------	----	----

id	CHAR(26)	PK, DEFAULT generate_ulid()	ログID (ULID)
blog_id	CHAR(26)	NOT NULL, FK	ブログID
request_id	CHAR(26)	NOT NULL	リクエストトレーシングID
started_at	TIMESTAMP WITH TIME ZONE	NOT NULL	開始日時
completed_at	TIMESTAMP WITH TIME ZONE	-	完了日時
duration_ms	INTEGER	-	処理時間 (ms)
status	VARCHAR(20)	NOT NULL	ステータス
articles_found	INTEGER	DEFAULT 0	発見記事数
articles_processed	INTEGER	DEFAULT 0	処理記事数
http_status_code	INTEGER	-	HTTPステータスコード
response_size_bytes	INTEGER	-	レスポンスサイズ
error_type	VARCHAR(50)	-	エラー種別
error_message	TEXT	-	エラーメッセージ
lambda_request_id	VARCHAR(100)	-	Lambda実行ID
lambda_memory_used_mb	INTEGER	-	使用メモリ (MB)
created_at	TIMESTAMP WITH TIME	DEFAULT NOW()	作成日時

	ZONE		
--	------	--	--

4.3 タグ管理テーブル群

4.3.1 tag_categories (タグカテゴリ)

カラム名	データ型	制約	説明
id	CHAR(26)	PK, DEFAULT generate_ulid()	タグカテゴリID (ULID)
name	VARCHAR(50)	NOT NULL, UNIQUE	カテゴリ名
description	TEXT	-	説明
color	VARCHAR(7)	DEFAULT '#6B7280'	表示色 (HEX)
sort_order	INTEGER	DEFAULT 0	表示順序
created_at	TIMESTAMP WITH TIME ZONE	DEFAULT NOW()	作成日時
created_by	CHAR(26)	-	作成者ID

4.3.2 tags (タグマスタ)

カラム名	データ型	制約	説明
id	CHAR(26)	PK, DEFAULT generate_ulid()	タグID (ULID)
name	VARCHAR(100)	NOT NULL, UNIQUE	タグ名
normalized_name	VARCHAR(100)	NOT NULL, UNIQUE	正規化名 (検索用)
category_id	CHAR(26)	FK	カテゴリID
description	TEXT	-	説明

official_url	VARCHAR(500)	-	公式サイトURL
article_count	INTEGER	DEFAULT 0	関連記事数
trending_score	DECIMAL(8,2)	DEFAULT 0	トレンドスコア
is_verified	BOOLEAN	DEFAULT false	検証済みフラグ
is_deprecated	BOOLEAN	DEFAULT false	廃止技術フラグ
created_at	TIMESTAMP WITH TIME ZONE	DEFAULT NOW()	作成日時
updated_at	TIMESTAMP WITH TIME ZONE	DEFAULT NOW()	更新日時
created_by	CHAR(26)	-	作成者ID
updated_by	CHAR(26)	-	更新者ID

4.4 記事管理テーブル群

4.4.1 articles (記事メイン)

カラム名	データ型	制約	説明
id	CHAR(26)	PK, DEFAULT generate_ulid()	記事ID (ULID)
blog_id	CHAR(26)	NOT NULL, FK	ブログID
title	VARCHAR(500)	NOT NULL	記事タイトル
url	VARCHAR(1000)	NOT NULL	記事URL
url_hash	CHAR(64)	NOT NULL, UNIQUE	URL SHA-256ハッシュ (重複排除用)
content	TEXT	-	記事本文
summary	TEXT	-	元サイト要約

ai_summary	TEXT	-	AI生成要約
excerpt	TEXT	-	冒頭抜粋
author	VARCHAR(255)	-	著者名
author_email_hash	CHAR(64)	-	著者メールハッシュ
language	VARCHAR(10)	DEFAULT 'ja'	言語コード
published_at	TIMESTAMP WITH TIME ZONE	NOT NULL	公開日時
updated_at_source	TIMESTAMP WITH TIME ZONE	-	元記事更新日時
view_count	INTEGER	DEFAULT 0	閲覧回数
click_count	INTEGER	DEFAULT 0	クリック回数
bookmark_count	INTEGER	DEFAULT 0	ブックマーク回数
share_count	INTEGER	DEFAULT 0	シェア回数
sentiment_score	DECIMAL(3,2)	-	感情スコア (-1.0～1.0)
difficulty_level	SMALLINT	-	難易度 (1～5)
reading_time_minutes	INTEGER	-	推定読了時間
content_fetched_at	TIMESTAMP WITH TIME ZONE	-	本文取得日時
content_fetch_status	VARCHAR(20)	DEFAULT 'PENDING'	本文取得状況

created_at	TIMESTAMP WITH TIME ZONE	DEFAULT NOW()	作成日時
updated_at	TIMESTAMP WITH TIME ZONE	DEFAULT NOW()	更新日時

4.5 ユーザー管理テーブル群

4.5.1 users (ユーザー基本情報)

カラム名	データ型	制約	説明
id	CHAR(26)	PK, DEFAULT generate_ulid()	ユーザーID (ULID)
username	VARCHAR(50)	NOT NULL, UNIQUE	ユーザー名
email_encrypted	BYTEA	NOT NULL	暗号化されたメ ールアドレス
email_hash	CHAR(64)	NOT NULL, UNIQUE	メールアドレス 検索用ハッシュ
password_hash	VARCHAR(255)	NOT NULL	bcryptパスワー ドハッシュ
display_name	VARCHAR(100)	-	表示名
avatar_url	VARCHAR(500)	-	アバターURL
bio	TEXT	-	プロフィール
website_url	VARCHAR(500)	-	WebサイトURL
timezone	VARCHAR(50)	DEFAULT 'Asia/Tokyo'	タイムゾーン
language	VARCHAR(10)	DEFAULT 'ja'	言語設定
theme	VARCHAR(20)	DEFAULT 'system'	テーマ設定

email_notifications	BOOLEAN	DEFAULT true	メール通知
weekly_digest	BOOLEAN	DEFAULT true	週次ダイジェスト
trending_alerts	BOOLEAN	DEFAULT false	トレンドアラート
failed_login_attempts	INTEGER	DEFAULT 0	連続ログイン失敗回数
locked_until	TIMESTAMP WITH TIME ZONE	-	アカウントロック解除日時
last_password_change	TIMESTAMP WITH TIME ZONE	DEFAULT NOW()	最終パスワード変更日時
password_reset_token_hash	CHAR(64)	-	パスワードリセットトークンハッシュ
password_reset_expires_at	TIMESTAMP WITH TIME ZONE	-	リセットトークン有効期限
email_verification_token_hash	CHAR(64)	-	メール認証トークンハッシュ
email_verified_at	TIMESTAMP WITH TIME ZONE	-	メール認証完了日時
last_login_at	TIMESTAMP WITH TIME ZONE	-	最終ログイン

last_activity_at	TIMESTAMP WITH TIME ZONE	-	最終活動日時
articles_read_count	INTEGER	DEFAULT 0	記事読了数
bookmarks_count	INTEGER	DEFAULT 0	ブックマーク数
is_active	BOOLEAN	DEFAULT true	アクティブフラグ
is_admin	BOOLEAN	DEFAULT false	管理者フラグ
created_at	TIMESTAMP WITH TIME ZONE	DEFAULT NOW()	作成日時
updated_at	TIMESTAMP WITH TIME ZONE	DEFAULT NOW()	更新日時
created_by	CHAR(26)	-	作成者ID
updated_by	CHAR(26)	-	更新者ID

4.5.2 user_sessions (セッション管理)

カラム名	データ型	制約	説明
id	CHAR(26)	PK, DEFAULT generate_ulid()	セッションID (ULID)
user_id	CHAR(26)	NOT NULL, FK	ユーザーID
session_token_hash	CHAR(64)	NOT NULL, UNIQUE	セッショントークンハッシュ
refresh_token_hash	CHAR(64)	-	リフレッシュトークンハッシュ
ip_address	INET	NOT NULL	IPアドレス

user_agent_hash	CHAR(64)	NOT NULL	User-Agentハッシュ
device_fingerprint	CHAR(64)	-	デバイスフィンガープリント
expires_at	TIMESTAMP WITH TIME ZONE	NOT NULL	セッション有効期限
refresh_expires_at	TIMESTAMP WITH TIME ZONE	-	リフレッシュトークン有効期限
last_activity_at	TIMESTAMP WITH TIME ZONE	DEFAULT NOW()	最終アクティビティ
is_active	BOOLEAN	DEFAULT true	アクティブフラグ
revoked_at	TIMESTAMP WITH TIME ZONE	-	無効化日時
revoked_reason	VARCHAR(100)	-	無効化理由
created_at	TIMESTAMP WITH TIME ZONE	DEFAULT NOW()	作成日時

4.6 セキュリティ監査テーブル

4.6.1 security_audit_logs (セキュリティ監査ログ)

カラム名	データ型	制約	説明
id	CHAR(26)	PK, DEFAULT generate_ulid()	監査ログID (ULID)
event_type	VARCHAR(50)	NOT NULL	イベント種別

event_category	VARCHAR(30)	NOT NULL	イベントカテゴリ
severity	VARCHAR(10)	NOT NULL DEFAULT 'INFO'	重要度
user_id	CHAR(26)	FK	対象ユーザーID
target_type	VARCHAR(50)	-	対象リソース種別
target_id	CHAR(26)	-	対象リソースID
session_id	CHAR(26)	-	セッションID
request_id	CHAR(26)	NOT NULL	リクエストトレーシングID
ip_address	INET	NOT NULL	IPアドレス
user_agent_hash	CHAR(64)	NOT NULL	User-Agentハッシュ
action_description	TEXT	NOT NULL	操作内容
old_values_hash	CHAR(64)	-	変更前データハッシュ
new_values_hash	CHAR(64)	-	変更後データハッシュ
success	BOOLEAN	NOT NULL	成功フラグ
error_code	VARCHAR(50)	-	エラーコード
error_message	TEXT	-	エラーメッセージ
processing_time_ms	INTEGER	-	処理時間 (ms)
occurred_at	TIMESTAMP WITH TIME	DEFAULT NOW()	発生日時

5. インデックス設計

5.1 主要インデックス

```

1  -- ブログ関連 (セキュリティ考慮)
2  CREATE INDEX idx_blogs_active_fetch ON blogs(is_active, last_fetched_at) WHERE is_active = true;
3  CREATE UNIQUE INDEX idx_blogs_rss_hash ON blogs(rss_url_hash);
4  CREATE INDEX idx_blog_fetch_logs_blog_time ON blog_fetch_logs(blog_id, started_at DESC);
5  CREATE INDEX idx_blog_fetch_logs_status ON blog_fetch_logs(status, started_at DESC);
6
7  -- 記事関連 (性能最適化)
8  CREATE INDEX idx_articles_blog_published_covering ON articles(blog_id, published_at DESC, id);
9  CREATE INDEX idx_articles_published_desc ON articles(published_at DESC) WHERE published_at IS NOT NULL;
10 CREATE UNIQUE INDEX idx_articles_url_hash ON articles(url_hash);
11
12 -- フルテキスト検索 (セキュアな検索)
13 CREATE INDEX idx_articles_title_search ON articles USING GIN(to_tsvector('simple', title));
14 CREATE INDEX idx_articles_content_search ON articles USING GIN(to_tsvector('japanese', COALESCE(title, content)));
15
16 -- ユーザー関連 (セキュリティ重要)
17 CREATE UNIQUE INDEX idx_users_email_hash ON users(email_hash);
18 CREATE UNIQUE INDEX idx_users_username_lower ON users(LOWER(username));
19 CREATE INDEX idx_users_failed_attempts ON users(failed_login_attempts) WHERE failed_login_attempts > 0;
20 CREATE INDEX idx_users_locked ON users(locked_until) WHERE locked_until IS NOT NULL;
21
22 -- セッション管理 (セキュリティクリティカル)
23 CREATE UNIQUE INDEX idx_sessions_token_hash ON user_sessions(session_token_hash);
24 CREATE INDEX idx_sessions_user_active ON user_sessions(user_id, is_active, last_activity_at DESC);
25 CREATE INDEX idx_sessions_expires ON user_sessions(expires_at) WHERE is_active = true;
26 CREATE INDEX idx_sessions_cleanup ON user_sessions(expires_at, is_active) WHERE is_active = false;
27
28 -- 監査ログ関連 (セキュリティ分析用)
29 CREATE INDEX idx_audit_user_time ON security_audit_logs(user_id, occurred_at DESC);
30 CREATE INDEX idx_audit_event_severity ON security_audit_logs(event_type, severity, occurred_at DESC);
31 CREATE INDEX idx_audit_ip_time ON security_audit_logs(ip_address, occurred_at DESC);
32 CREATE INDEX idx_audit_failed_auth ON security_audit_logs(event_category, success, occurred_at DESC)
33     WHERE event_category = 'AUTH' AND success = false;
34

```

5.2 パーティショニング戦略

```

1  -- 監査ログの月次パーティション (データ量対応)
2  CREATE TABLE security_audit_logs_y2024m01 PARTITION OF security_audit_logs
3      FOR VALUES FROM ('2024-01-01') TO ('2024-02-01');
4
5  -- 記事の月次パーティション (将来的な大量データ対応)
6  CREATE TABLE articles_y2024m01 PARTITION OF articles
7      FOR VALUES FROM ('2024-01-01') TO ('2024-02-01');
8
9  -- 自動パーティション作成関数
10 CREATE OR REPLACE FUNCTION create_monthly_partitions()
11 RETURNS void AS $$
12 DECLARE
13     start_date date;

```

```

14     end_date date;
15     table_name text;
16 BEGIN
17     -- 来月のパーティションを事前作成
18     start_date := date_trunc('month', CURRENT_DATE + interval '1 month');
19     end_date := start_date + interval '1 month';
20
21     -- 監査ログパーティション
22     table_name := 'security_audit_logs_y' || to_char(start_date, 'YYYY') || 'm' || to_char(st
23 EXECUTE format('CREATE TABLE IF NOT EXISTS %I PARTITION OF security_audit_logs FOR VALUES
24                 table_name, start_date, end_date);
25
26     -- 記事パーティション
27     table_name := 'articles_y' || to_char(start_date, 'YYYY') || 'm' || to_char(start_date, '
28 EXECUTE format('CREATE TABLE IF NOT EXISTS %I PARTITION OF articles FOR VALUES FROM (%L)
29                 table_name, start_date, end_date);
30 END;
31 $$ LANGUAGE plpgsql;
32

```

6. セキュリティ制約・トリガー

6.1 セキュリティ制約

```

1  -- パスワード強度チェック
2  ALTER TABLE users ADD CONSTRAINT users_password_strength_check
3      CHECK (LENGTH(password_hash) >= 60); -- bcryptは60文字以上
4
5  -- セッション有効期限チェック
6  ALTER TABLE user_sessions ADD CONSTRAINT sessions_expires_check
7      CHECK (expires_at > created_at);
8
9  -- 監査ログの改ざん防止
10 ALTER TABLE security_audit_logs ADD CONSTRAINT audit_immutable_check
11     CHECK (occurred_at <= NOW());
12
13 -- アカунトロック期間制限
14 ALTER TABLE users ADD CONSTRAINT users_lock_duration_check
15     CHECK (locked_until IS NULL OR locked_until > NOW());
16

```

6.2 セキュリティトリガー

```

1  -- パスワード変更時の監査ログ生成
2  CREATE OR REPLACE FUNCTION log_password_change()
3  RETURNS TRIGGER AS $$
4  BEGIN
5      IF OLD.password_hash != NEW.password_hash THEN
6          INSERT INTO security_audit_logs (
7              event_type, event_category, severity, user_id,
8              action_description, success, request_id
9          ) VALUES (
10              'PASSWORD_CHANGE', 'AUTH', 'WARN', NEW.id,
11              'User password changed', true, generate_ulid()
12          );
13     END IF;
14     RETURN NEW;

```



```
15 END;
16 $$ LANGUAGE plpgsql SECURITY DEFINER;
17
18 CREATE TRIGGER trigger_password_change_audit
19     AFTER UPDATE ON users
20     FOR EACH ROW EXECUTE FUNCTION log_password_change();
21
22 -- ログイン失敗回数の自動リセット
23 CREATE OR REPLACE FUNCTION reset_failed_attempts_on_success()
24 RETURNS TRIGGER AS $$
25 BEGIN
26     -- セッション作成時に失敗回数をリセット
27     UPDATE users SET
28         failed_login_attempts = 0,
29         locked_until = NULL
30     WHERE id = NEW.user_id AND failed_login_attempts > 0;
31     RETURN NEW;
32 END;
33 $
34
```