拠点CPT設計書（図解）

本ドキュメントは、WordPressを用いた多拠点サイト運用における「店舗カスタム投稿タイプ（CPT）設計」と「共通コンポーネント連携」の概念図および詳細を解説します。これにより、多拠点サイトが抱えがちな情報管理の課題を解決し、Local SEOに最適化された運用を実現するための基盤を構築することを目指します。

1. 設計の目的

- **情報の構造化**: 店舗情報を一貫性のあるデータとして管理し、属人化を防ぐ。
- **運用効率の向上**: 定型化された入力フォームにより、更新作業の負担を軽減し、ヒューマンエラーを削減する。
- **SEO品質の担保**: NAP情報の一貫性、構造化データの自動出力、適切な内部リンク設計により、検索エンジンからの評価を高める。
- **拡張性**: 将来的な店舗数増加や機能追加にも柔軟に対応できるシステム基盤を構築する。

2. 拠点CPT設計概要図

以下の図は、WordPressの管理画面から入力された店舗情報が、データベースにどのように格納され、最終的にフロントエンドでどのように表示・連携されるかを示しています。

（※別ファイルをダウンロードしてください）

図解の説明

- **WordPress Admin**: 管理画面から「店舗CPT」を通じて店舗情報が入力されます。ここでは「カスタムフィールド」と「カスタムタクソノミ」が重要な役割を果たします。
- **データ入力**: カスタムフィールドには、店舗の「正式店舗名」「住所」「電話番号」「営業時間」「緯度・経度」などが定型化された形式で入力されます。カスタムタクソノミでは、「エリア（都道府県、市区町村）」や「業態」といった分類情報が設定されます。
- **データベース**: 入力された店舗情報は、WordPressのデータベース（`wp_posts`テーブルにカスタム投稿タイプ`shop`として、`wp_term_relationships`テーブルにタクソノミ情報として）に構造化されたデータとして保存されます。
- **フロントエンド表示**: データベースから取得された店舗情報は、`single-shop.php`（店舗詳細ページ）や`archive-shop.php`（エリア別一覧ページ）といったテンプレートファイルを通じてユーザーに表示されます。また、`get_template_part`などの共通コンポーネントとして、フッターやサイドバーなどサイト内の様々な箇所に動的に表示されます。
- **コンポーネント/出力**: 店舗詳細ページでは、基本情報の表示、Google Mapの埋め込み、そしてLocal SEOに不可欠な「構造化データ（JSON-LD）」の自動出力が行われます。共通コ

ンポーネントは、フッターの店舗一覧やサイドバーの近隣店舗表示などに活用され、NAP情報の一貫性を保ちます。

3. データベース設計詳細

3.1. 店舗CPTのカスタムフィールド

Advanced Custom Fields (ACF) などのプラグインを利用して、以下のカスタムフィールドを店舗CPTに紐付けます。

フィールド名	フィールドタイプ	必須	説明	備考
shop_name	Text	はい	正式店舗名（NAPのN）	Googleビジネスプロフィールと一致させる
shop_zipcode	Text	はい	郵便番号	ハイフンなし、またはハイフンありで統一
shop_address	Text	はい	住所（NAPのA）	都道府県から建物名まで正確に記載。Googleビジネスプロフィールと一致させる
shop_tel	Text	はい	電話番号（NAPのP）	市外局番から記載。Googleビジネスプロフィールと一致させる
shop_hours	Wysiwyg Editor / Textarea	はい	営業時間	曜日ごとの詳細な記述を推奨
shop_holiday	Text	いいえ	定休日	不定休の場合はその旨を記載
shop_latitude	Number	はい	緯度	Google Maps API連携用
shop_longitude	Number	はい	経度	Google Maps API連携用
shop_image	Image	いいえ	店舗外観・内観画像	alt属性の自動生成を考慮

shop_description	Wysiwyg Editor	いいえ	店舗紹介文	ユーザーへの訴求ポイント

3.2. カスタムタクソノミ

店舗の分類と内部リンク設計のために、以下のカスタムタクソノミを定義します。

タクソノミ名	スラッグ	階層	説明	備考
エリア	area	はい	都道府県、市区町村、駅名など	親子関係を設定し、階層構造を構築
業態	category	いいえ	カフェ、レストラン、美容室など	店舗のサービス内容で分類

4. コンポーネント連携とデータフロー

4.1. NAP一貫性の確保

店舗CPTのカスタムフィールドに入力されたNAP情報（ shop_name , shop_address , shop_tel ）は、サイト内のあらゆる箇所で共通の関数やテンプレートパーツを通じて呼び出されるように設計します。これにより、情報更新時の修正漏れを防ぎ、サイト全体でのNAP一貫性を保ちます。

実装例:

PHP

```
<?php
// functions.php またはプラグイン内で定義
function get_shop_nap_info($post_id) {
    return [
        'name' => get_field('shop_name', $post_id),
        'address' => get_field('shop_address', $post_id),
        'phone' => get_field('shop_tel', $post_id),
    ];
}

// テンプレートファイルでの利用例
$current_shop_id = get_the_ID();
```

```
$nap_info = get_shop_nap_info($current_shop_id);
echo '<p>店名：' . esc_html($nap_info['name']) . '</p>';
echo '<p>住所：' . esc_html($nap_info['address']) . '</p>';
echo '<p>電話：' . esc_html($nap_info['phone']) . '</p>';
?>
```

4.2. 構造化データの自動出力

店舗詳細ページでは、`LocalBusiness`スキーマを用いた構造化データをJSON-LD形式で自動出力します。これにより、検索エンジンが店舗情報を正確に理解し、リッチリザルトやローカルパックでの表示に寄与する可能性があります。

実装例（抜粋）：

PHP

```
<?php
// single-shop.php またはヘッダー部分に記述
if (is_singular('shop')) {
    $shop_id = get_the_ID();
    $nap_info = get_shop_nap_info($shop_id);
    $latitude = get_field('shop_latitude', $shop_id);
    $longitude = get_field('shop_longitude', $shop_id);
    $hours = get_field('shop_hours', $shop_id);

    $schema_data = [
        "@context" => "http://schema.org",
        "@type" => "LocalBusiness",
        "name" => $nap_info['name'],
        "address" => [
            "@type" => "PostalAddress",
            "streetAddress" => $nap_info['address'], // 必要に応じて分割
            "addressLocality" => "", // 市区町村
            "addressRegion" => "", // 都道府県
            "postalCode" => get_field('shop_zipcode', $shop_id  )
        ],
        "telephone" => $nap_info['phone'],
        "geo" => [
            "@type" => "GeoCoordinates",
            "latitude" => $latitude,
            "longitude" => $longitude
        ],
        "openingHours" => $hours, // 適切な形式に変換
        "url" => get_permalink($shop_id)
    ];
    echo '<script type="application/ld+json">' . json_encode($schema_data, JSON
```

```
}
?>
```

4.3. 自動内部リンクの生成

カスタムタクソノミや緯度・経度情報を活用し、関連性の高い店舗への内部リンクを自動生成します。これにより、ユーザーの回遊性を高め、サイト全体のクローラビリティを向上させます。

- **近隣店舗リンク**: 現在の店舗の緯度・経度と、他の店舗の緯度・経度を比較し、距離が近い店舗を数件抽出して表示します。
- **エリア別・業態別リンク**: タクソノミのターム（例: 「東京都」「カフェ」）に紐づく店舗一覧ページへのリンクを生成します。
- **パンくずリスト**: `wp_get_post_terms` などの関数を用いて、店舗CPTのタクソノミ階層に基づいたパンくずリストを動的に生成します。

5. 運用上の注意点

- **スラッグ設計**: CPTやタクソノミのスラッグは、一度設定すると変更が困難なため、初期段階で慎重に決定してください。SEOへの影響も考慮し、シンプルかつ分かりやすいものを選びましょう。
- **重複コンテンツの回避**: エリア別一覧ページなどで、店舗詳細ページと内容が重複しすぎないよう、適切なコンテンツ（エリアの紹介文など）を追加することを検討してください。
- **更新フローの明確化**: 誰が、いつ、どの情報を更新するのか、運用ルールを明確にし、担当者への教育を徹底することで、情報の鮮度と一貫性を維持できます。

6. まとめ

多拠点サイトにおけるWordPress運用は、単なるウェブサイト制作の範疇を超え、データ管理とシステム設計の視点が不可欠です。本設計書が示すように、CPTとカスタムフィールド、タクソノミを適切に設計し、共通コンポーネントとして連携させることで、情報の属人化を防ぎ、長期的に安定した運用とLocal SEO効果の最大化が期待できます。

この設計図を基に、貴社の多拠点サイトがより効果的なマーケティング資産となることを願っています。