

SPECIFICHE DATABASE CMS/E-COMMERCE LARAVEL

Documento Tecnico per Implementazione

INTRODUZIONE

Il presente documento descrive le specifiche complete per la creazione del database di un sistema CMS/E-commerce sviluppato in Laravel. Il sistema deve gestire contenuti editoriali (pagine e articoli), prodotti e-commerce, media library e ottimizzazione SEO, ispirandosi alle funzionalità di WordPress per la parte CMS e Magento per l'e-commerce.

OBIETTIVI DEL SISTEMA

Il database deve supportare le seguenti funzionalità principali:

1. **Gestione Utenti:** Sistema completo di autenticazione con ruoli e permessi granulari
 2. **CMS Flessibile:** Creazione e gestione di pagine, articoli e contenuti personalizzati
 3. **E-commerce Completo:** Catalogo prodotti, varianti, inventario, carrello e ordini , gestione sconti (programmati e coupon)
 4. **Media Library:** Gestione centralizzata di immagini, documenti e file multimediali
 5. **SEO Avanzato:** Metadati, sitemap, redirects e ottimizzazione per motori di ricerca
 6. **Multi-lingua:** Supporto per contenuti in più lingue
 7. **Configurazione:** Sistema flessibile di impostazioni e opzioni
-

CONVENZIONI E STANDARD

Prima di iniziare l'implementazione, è necessario rispettare i seguenti standard Laravel:

- **Nomi tabelle:** sempre in minuscolo, snake_case e plurale (es: `order_items`)
- **Chiavi primarie:** sempre chiamate `id`, tipo BIGINT UNSIGNED con AUTO_INCREMENT

- **Timestamp automatici:** ogni tabella deve avere `created_at` e `updated_at` (TIMESTAMP NULL)
- **Soft Delete:** dove indicato, aggiungere `deleted_at` (TIMESTAMP NULL)
- **Chiavi esterne:** formato `nome_tabella_id` (es: `users_id`, `products_id`)
- **Charset:** UTF8MB4 per supportare emoji e caratteri speciali
- **Collation:** `utf8mb4_unicode_ci`

Tipi di dato standard da utilizzare:

- **ID e riferimenti:** BIGINT UNSIGNED
- **Prezzi e valori monetari:** DECIMAL(10,2)
- **Percentuali:** DECIMAL(5,2)
- **Flag boolean:** BOOLEAN (default FALSE dove non specificato)
- **Testi brevi** (titoli, nomi): VARCHAR(255)
- **Testi medi** (descrizioni): TEXT
- **Testi lunghi** (contenuti): LONGTEXT
- **Dati strutturati:** JSON
- **Slug e codici:** VARCHAR(255) con indice UNIQUE
- **Date semplici:** DATE
- **Date con ora:** DATETIME o TIMESTAMP

RUOLI

Il sistema di gestione ruoli e permessi è fondamentale per controllare chi può fare cosa all'interno della piattaforma. Pensate ai ruoli come a delle "etichette professionali" che identificano il tipo di lavoro che una persona svolge, mentre i permessi sono le singole azioni specifiche che quella persona può compiere.

Il sistema funziona in modo gerarchico: ogni ruolo ha un insieme di permessi associati, e un utente può avere uno o più ruoli assegnati. Questo approccio è molto più flessibile rispetto ad assegnare i permessi direttamente agli utenti uno per uno.

MODULO CMS - SPECIFICHE DESCRITTIVE

Panoramica Generale

Il modulo CMS gestisce tutti i contenuti testuali del sito attraverso un sistema unificato. Ispirato a WordPress, utilizza un approccio "single table" dove pagine, articoli e prodotti coesistono nella

stessa tabella, differenziati da un campo tipo. Questo garantisce flessibilità, semplicità di manutenzione e facilità di estensione futura.

Posts - Repository Centrale Contenuti

La tabella posts è il contenitore universale di tutti i contenuti. Ogni record rappresenta una pagina, un articolo o un prodotto, identificato dal campo post_type.

Il contenuto è caratterizzato da un titolo e uno slug URL-friendly univoco per lingua. Il corpo principale è salvato in HTML nel campo content, mentre l'excerpt fornisce un riassunto per anteprime e condivisioni social.

La pubblicazione è gestita da uno stato (draft, published, scheduled, private, trash) e da una data di pubblicazione. Questo permette la preparazione anticipata di contenuti e la pubblicazione automatica programmata. Ogni contenuto è collegato all'autore che lo ha creato.

La presentazione è personalizzabile attraverso l'associazione di un'immagine in evidenza, la scelta di un template specifico e flag per evidenziare contenuti importanti.

La struttura gerarchica è supportata tramite il campo parent_id, permettendo di creare pagine figlie e generare automaticamente URL gerarchici e breadcrumb.

Il sistema è multilingua nativamente: ogni contenuto appartiene a una lingua specifica, permettendo traduzioni gestite come contenuti separati ma concettualmente collegati.

Funzionalità accessorie includono il conteggio automatico delle visualizzazioni, la gestione dei commenti (aperti o chiusi), e l'ordinamento personalizzato per menu.

Il soft delete protegge da eliminazioni accidentali: i contenuti cancellati rimangono recuperabili finché non vengono eliminati definitivamente.

Post_Meta - Sistema di Estensione Flessibile

Post_meta implementa un pattern chiave-valore per estendere i post con dati arbitrari senza modificare la struttura base.

Ogni metadato è composto da una chiave (identificatore del campo custom), un valore (il dato effettivo) e il riferimento al post a cui appartiene. Un singolo post può avere illimitati metadati.

Questo approccio risolve il problema dei "campi specifici": quando diversi tipi di contenuto necessitano dati differenti, invece di aggiungere colonne alla tabella principale (che rimarrebbero vuote per molti record), si salvano come metadati.

Casi d'uso tipici includono: configurazioni specifiche per template, dati aggiuntivi per prodotti specializzati, link a risorse esterne, parametri tecnici, valori per integrazioni con servizi esterni, e qualsiasi informazione che non si applica a tutti i contenuti.

Il campo valore supporta testo lungo e JSON, permettendo di salvare sia valori semplici che strutture dati complesse.

Categories - Tassonomia Gerarchica

Le categorie forniscono un sistema di classificazione ad albero per organizzare i contenuti. A differenza dei tag, sono strutturate gerarchicamente: una categoria può contenere sotto-categorie, che a loro volta possono contenere altre sotto-categorie.

Ogni categoria ha un nome visualizzato pubblicamente, uno slug per gli URL, e una descrizione opzionale che appare nella pagina archivio della categoria stessa.

Il campo type separa nettamente le categorie blog (per articoli editoriali) dalle categorie prodotto (per l'e-commerce), evitando confusione e mantenendo i due sistemi indipendenti.

La gerarchia è costruita tramite parent_id: le categorie radice hanno parent_id nullo, le sotto-categorie puntano al loro genitore. Questa struttura permette navigazioni intuitive e URL descrittivi.

Ogni categoria può avere un'immagine rappresentativa, utile specialmente nell'e-commerce per pagine categoria visivamente ricche. L'ordinamento è controllabile manualmente, e ogni categoria può essere attivata/disattivata senza eliminarla.

Il supporto multilingua permette categorie separate per lingua, mantenendo strutture di classificazione appropriate per ogni mercato.

Tags - Classificazione Trasversale

I tag implementano un sistema di etichettatura piatto e non gerarchico, complementare alle categorie. Sono parole chiave libere che descrivono aspetti trasversali dei contenuti.

A differenza delle categorie (strutturate e limitate), i tag sono informali e illimitati. Un contenuto tipicamente ha 1-2 categorie ma può avere molti tag.

La struttura è minimale: nome, slug e descrizione opzionale. Ogni tag appartiene a una lingua specifica per il multilingua.

I tag emergono organicamente: vengono creati dagli editor durante la scrittura dei contenuti piuttosto che essere pre-pianificati come le categorie. Permettono scopribilità e connessioni inaspettate tra contenuti apparentemente diversi.

Flussi Principali

Creazione Contenuto: Autore crea bozza → compila titolo, contenuto, excerpt → assegna categorie/tag → carica featured image → salva. Contenuto è in stato draft, visibile solo agli editor.

Pubblicazione: Editor revisiona → approva → cambia status a published o scheduled con data futura → contenuto diventa pubblico immediatamente o all'ora programmata.

Organizzazione: Categorie pre-create dall'amministratore formano struttura di navigazione. Tag aggiunti dagli autori durante scrittura emergono organicamente. Gerarchia pagine costruita assegnando parent.

Estensione: Nuovi campi specifici aggiunti come post_meta senza toccare struttura database. Template custom creati per visualizzazioni speciali.

Multilingua: Stesso contenuto tradotto creando post separati per ogni lingua con slug localizzati. Collegamento concettuale gestito a livello applicativo.