

RedRealty — Modèle de données "Immobilier neuf" (programmes neufs)

Préfixe de table : **cm_pn_** (Casa Media — Programmes Neufs), en cohérence avec vos tables existantes (**cm_type_demande**, etc.).

Ce document couvre : les tables à créer, les colonnes de chacune, les clés primaires/étrangères, la logique de génération des références (id + reference, comme demandé), et un exemple concret qui retrace chacune des URLs Yakeey que vous avez partagées vers les lignes de tables correspondantes.

1. Principe général

Deux niveaux d'entités, comme sur Yakeey :

- **Programme** (**cm_pn_programmes**) — le projet immobilier dans son ensemble (ex. *Riad Sidi Maarouf*). Une seule fiche, une seule galerie, une seule adresse.
- **Lot** (**cm_pn_lots**) — chaque unité vendable à l'intérieur d'un programme (un appartement, un local commercial, un terrain...). **Un programme peut contenir des lots de types différents** — c'est exactement ce que montrent vos URLs **pc000356** : le programme est affiché comme "local commercial" mais contient un lot **appartement** (**ca601199**) et un lot **local-commercial** (**cb601201**). Le **type_bien_id** vit donc à deux niveaux : un type "dominant" sur le programme (pour le slug/l'affichage), et un type propre à chaque lot.

Chaque programme et chaque lot a un **id** auto-incrémenté (clé technique, jamais exposée) **et** une **reference** lisible (**PC000356**, **CA601199**...) utilisée dans les URLs — jamais l'inverse.

2. Table existante à étendre

`type_bien` (déjà en place dans votre projet — on l'enrichit, on ne la duplique pas)

Colonne	Type	Clé	Notes
<code>parent_id</code>	<code>INT UNSIGNED NULL</code>	FK → <code>type_bien.id</code>	Auto-référence, pour les sous-types (ex. <i>Terrain pour villa</i> est un enfant de <i>Terrain</i> — vu dans <code>tv604093</code>)
<code>prefixe_programme</code>	<code>CHAR(2) NULL</code>	—	Préfixe de référence quand ce type est le type dominant d'un programme (ex. <code>PT</code> pour Terrain)
<code>prefixe_lot</code>	<code>CHAR(2) NULL</code>	—	Préfixe de référence quand un lot est de ce type (ex. <code>CA</code> Appartement, <code>CB</code> Local commercial, <code>TV</code> Terrain pour villa)

C'est la table `type_bien` qui pilote la génération des références — pas de table à part à maintenir en double.

3. Tables de référence (lookups)

`cm_pn_villes`

Colonne	Type	Clé
<code>id</code>	<code>INT UNSIGNED</code>	PK
<code>nom</code>	<code>VARCHAR(100)</code>	

Colonne	Type	Clé
slug	VARCHAR(100)	UNIQUE (casablanca, dar-bouazza...)
etat	TINYINT(1) DEFAULT 1	

cm_pn_quartiers

Colonne	Type	Clé
id	INT UNSIGNED	PK
ville_id	INT UNSIGNED	FK → cm_pn_villes.id
nom	VARCHAR(100)	
slug	VARCHAR(100)	UNIQUE avec ville_id (beausejour, tamaris1...)
etat	TINYINT(1) DEFAULT 1	

cm_pn_promoteurs

Colonne	Type	Clé
id	INT UNSIGNED	PK
nom	VARCHAR(150)	
slug	VARCHAR(150)	UNIQUE
logo	VARCHAR(255) NULL	
description	TEXT NULL	
telephone	VARCHAR(20) NULL	
email	VARCHAR(150) NULL	
etat	TINYINT(1) DEFAULT 1	

cm_pn_conseillers

Colonne	Type	Clé
id	INT UNSIGNED	PK
prenom / nom	VARCHAR(80)	
telephone	VARCHAR(20)	
email	VARCHAR(150) NULL	
photo	VARCHAR(255) NULL	
fonction	VARCHAR(100) NULL	ex. "Conseillère immobilier neuf"
etat	TINYINT(1) DEFAULT 1	

cm_pn_statuts_programme

id PK · code VARCHAR(30) UNIQUE (en_construction, livre, travaux_prevus) · libelle VARCHAR(100) · couleur VARCHAR(20) · ordre TINYINT · etat

cm_pn_statuts_lot

id PK · code VARCHAR(30) UNIQUE (disponible, reserve, vendu) · libelle VARCHAR(100) · couleur VARCHAR(20) · etat

cm_pn_badges

id PK · code VARCHAR(40) UNIQUE (neuf, exclusif, ideal_investisseur, parking_titre, temoin_disponible, standing) · libelle · icone · ordre · etat

Modélisé en table + association (plutôt qu'en colonnes booléennes fixes) pour pouvoir ajouter un nouveau badge depuis l'admin sans migration.

cm_pn_prestations

id PK · libelle VARCHAR(100) (Piscine, Salle de sport, Gardiennage 24h/24...) · icone VARCHAR(60) · ordre · etat

cm_pn_type_poi

id PK · libelle VARCHAR(100) (Autoroute, Commerce, École, Mosquée, Pôle d'affaires) · icone · etat

cm_pn_reference_sequences

Compteur qui alimente la génération des références (voir §6).

Colonne	Type	Clé
prefixe	CHAR(3)	PK (PC, PT, PI, CA, CB, TV...)
dernier_numero	INT UNSIGNED DEFAULT 0	
longueur_numero	TINYINT UNSIGNED DEFAULT 6	pour le zero-padding (000356)

4. Tables cœur

cm_pn_programmes

Colonne	Type	Clé	Notes
id	BIGINT UNSIGNED	PK auto-incrément	jamais exposé en URL
reference	VARCHAR(10)	UNIQUE, NOT NULL	PC000356
reference_prefix	CHAR(3)		PC — conservé séparément pour filtrer/régénérer
reference_number	INT UNSIGNED		356
slug	VARCHAR(220)	UNIQUE, NOT NULL	acheter-local-commercial-casablanca-beausejour-pc000356
titre	VARCHAR(180)	NOT NULL	ex. "Riad Sidi Maarouf"
type_bien_id	INT UNSIGNED	FK → type_bien.id	type dominant / affiché dans le slug

Colonne	Type	Clé	Notes
promoteur_id	INT UNSIGNED NULL	FK → cm_pn_promoteurs. id	
conseiller_id	INT UNSIGNED NULL	FK → cm_pn_conseillers. id	agent affiché en sidebar
quartier_id	INT UNSIGNED	FK → cm_pn_quartiers. id	
ville_id	INT UNSIGNED	FK → cm_pn_villes.id	dénormalisé depuis quartier.ville_id, tenu à jour par l'appli, pour indexer/filtrer sans jointure
adresse	VARCHAR(255) NULL		
latitude / longitude	DECIMAL(10,7) NULL		pour la carte
description	TEXT NULL		
statut_id	INT UNSIGNED	FK → cm_pn_statuts_programme.id	
date_livraison_prevue	DATE NULL		
nombre_lots_total	SMALLINT UNSIGNED DEFAULT 0		caché , recalculé à chaque écriture sur cm_pn_lots
nombre_lotsDisponibles	SMALLINT UNSIGNED DEFAULT 0		caché

Colonne	Type	Clé	Notes
prix_min / prix_max	DECIMAL(12,2) NULL		caché, MIN/MAX(cm_pn_lots.prix) — évite un JOIN+GROUP BY sur chaque page de listing
surface_min / surface_max	DECIMAL(8,2) NULL		caché
chambres_min / chambres_max	TINYINT NULL		caché
brochure_file	VARCHAR(255) NULL		
meta_title / meta_description	VARCHAR(180) / VARCHAR(300) NULL		SEO
etat	TINYINT(1) DEFAULT 1		
date_creation	DATETIME DEFAULT CURRENT_TIMESTAMP		
date_modification	DATETIME NULL		

Les champs "cachés" (prix_min/max, surface_min/max, chambres_min/max, nombre_lots_*) sont dérivés des lots. Recalculez-les côté application juste après chaque INSERT/UPDATE/DELETE sur cm_pn_lots (ou via un trigger MySQL) — c'est ce qui rend la page listing rapide même avec des filtres budget/chambres.

cm_pn_lots

Colonne	Type	Clé	Notes
id	BIGINT UNSIGNED	PK	

Colonne	Type	Clé	Notes
programme_id	BIGINT UNSIGNED	FK → cm_pn_programmes .id, NOT NULL	
reference	VARCHAR(10)	UNIQUE, NOT NULL	CA601199
reference_prefix	CHAR(2)		CA
reference_number	INT UNSIGNED		601199
slug	VARCHAR(120)	UNIQUE avec programme_id	acheter-appartem ent-ca601199 (unique dans le programme, pas globalement — l'URL complète inclut déjà le slug du programme)
type_bien_id	INT UNSIGNED	FK → type_bien.id, NOT NULL	type propre à ce lot — peut différer du type dominant du programme
numero_lot	VARCHAR(20) NULL		code interne, ex. A101
etage_libelle	VARCHAR(40) NULL		"1er étage", "Dernier étage"
etage_ordre	SMALLINT NULL		valeur numérique pour le tri (0=RDC, 1=1er...)
surface	DECIMAL(8,2)	NOT NULL	
surface_exterieu re	DECIMAL(8,2) NULL		balcon/terrasse
nombre_chambres	TINYINT NULL		NULL pour local/terrain

Colonne	Type	Clé	Notes
nombre_salles_bain	TINYINT NULL		
prix	DECIMAL(12,2)	NOT NULL	
orientation	VARCHAR(60) NULL		
statut_id	INT UNSIGNED	FK → cm_pn_statuts_lot.id, NOT NULL	
plan_file	VARCHAR(255) NULL		plan du lot (PDF/image)
description	TEXT NULL		
etat	TINYINT(1) DEFAULT 1		
date_creation / date_modification	DATETIME		

5. Tables d'enrichissement

cm_pn_medias

Galerie du programme (photos, plan de masse, brochure).

id PK · programme_id FK → cm_pn_programmes.id · type ENUM(photo,video,plan_masse,brochure) DEFAULT photo · url VARCHAR(255) · legende VARCHAR(180) NULL · ordre SMALLINT DEFAULT 0 · is_cover TINYINT(1) DEFAULT 0 · etat

cm_pn_programme_badges (association $N \leftrightarrow N$)

programme_id FK · badge_id FK · **PK composite** (programme_id, badge_id)

cm_pn_programme_prestations (association $N \leftrightarrow N$)

programme_id FK · prestation_id FK · **PK composite**

cm_pn_programme_poi

Points d'intérêt à proximité, avec distance propre à **ce** programme (le même "Autoroute A5" a une durée différente selon le programme).

id PK · programme_id FK · type_poi_id FK → cm_pn_type_poi.id · nom
VARCHAR(150) · duree_minutes SMALLINT · mode ENUM(pied,voiture) · ordre
SMALLINT

cm_pn_leads

Demandes "Je me fais rappeler" — le pendant de votre flux [ajax/submit_vite.php](#) pour l'immobilier neuf.

Colonne	Type	Clé
id	BIGINT UNSIGNED	PK
programme_id	BIGINT UNSIGNED NULL	FK → cm_pn_programmes.id
lot_id	BIGINT UNSIGNED NULL	FK → cm_pn_lots.id
conseiller_id	INT UNSIGNED NULL	FK → cm_pn_conseillers.id
prenom / nom	VARCHAR(80)	NOT NULL
telephone	VARCHAR(20)	NOT NULL
email	VARCHAR(150) NULL	
source	VARCHAR(40) DEFAULT 'sidebar'	sidebar / sticky_bar / listing
statut	VARCHAR(30) DEFAULT 'nouveau'	nouveau / contacte / qualifie / perdu
date_creation	DATETIME DEFAULT CURRENT_TIMESTAMP	

6. Génération des références et des slugs

Au niveau programme — préfixe à 2 lettres tiré de `type_bien.prefixe_programme` (avec possibilité de forcer une valeur manuelle sur le programme pour les cas particuliers, cf.

`PI000421` observé chez Yakeey qui ne colle à aucune règle type/ville évidente) :

1. Déterminer le préfixe (ex. `PC` pour Appartement/Local commercial à Casablanca, `PT` pour Terrain).
2. `UPDATE cm_pn_reference_sequences SET dernier_numero = dernier_numero + 1 WHERE prefixe = 'PC'` (dans une transaction), puis lire la nouvelle valeur.
3. `reference = prefixe . LPAD(dernier_numero, longueur_numero, '0')` → `PC000356`.
4. `slug = 'acheter-' . type_bien.slug . '-' . ville.slug . '-' . quartier.slug . '-' . LOWER(reference)`.

Au niveau lot — même mécanique, mais le préfixe vient de `type_bien.prefixe_lot` du lot lui-même (c'est ce qui explique que `CA`, `CB`, `TV` correspondent proprement à Appartement / Local commercial / Terrain pour villa dans vos exemples) :

1. `reference = type_bien.prefixe_lot . LPAD(numero, 6, '0')` → `CA601199`.
2. `slug = 'acheter-' . type_bien.slug . '-' . LOWER(reference)` (le slug du programme parent préfixe déjà l'URL, pas besoin de répéter ville/quartier).

7. Vérification — vos 5 URLs retracées dans le modèle

URL	Résolution dans le modèle
<code>/immobilier-neuf?page=6</code>	Listing paginé sur <code>cm_pn_programmes</code> (<code>WHERE etat=1 ORDER BY ... LIMIT 24 OFFSET 120</code>) — index recommandé sur (<code>etat</code> , <code>statut_id</code> , <code>ville_id</code>)
<code>/programme/acheter-local-commercial-casablanca-beausejour-pc000356</code>	<code>cm_pn_programmes</code> : <code>reference='PC000356'</code> , <code>type_bien.slug='local-commercial'</code> ,

URL	Résolution dans le modèle
	<code>ville.slug='casablanca', quartier.slug='beausejour'</code>
<code>.../pc000356/acheter-appartement-ca601199</code>	<code>cm_pn_lots : programme_id = id du programme ci-dessus, reference='CA601199', type_bien.slug='appartement' — différent du type dominant du programme, ce qui est autorisé</code>
<code>.../pc000356/acheter-local-commercial-cb601201</code>	<code>cm_pn_lots : même programme_id, reference='CB601201', type_bien.slug='local-commercial'</code>
<code>/programme/acheter-terrain-dar-bouazza-tamaris1-pt000349/acheter-terrain-pour-villa-tv604093</code>	<code>Programme : reference='PT000349', ville.slug='dar-bouazza', quartier.slug='tamaris1', type dominant terrain. Lot : reference='TV604093', type_bien.slug='terrain-pour-villa' (enfant de terrain via parent_id)</code>

8. Index recommandés

- `cm_pn_programmes : (etat, statut_id, ville_id), (type_bien_id), (prix_min), UNIQUE(slug), UNIQUE(reference)`
- `cm_pn_lots : (programme_id, statut_id), (type_bien_id), UNIQUE(reference), UNIQUE(programme_id, slug)`
- `cm_pn_leads : (programme_id), (date_creation)`
- `cm_pn_quartiers : UNIQUE(ville_id, slug)`

La pagination (`?page=6`) reste une affaire de `LIMIT/OFFSET` côté application — aucune table dédiée n'est nécessaire pour ça.