`markets`

One row per market:

- `market_id` (uuid)
- `market_name, city, state`

`properties_core`

Stable identity + location fields:

- `property_id` (pk)
- `market_id` (fk)
- `title, city, state, zipcode`
- `latitude, longitude`
- listing URLs, host URL if needed

`performance`

Time/financial performance fields:

- `property_id` (pk/fk)
- `revenue, revenue_potential, adr, occupancy,` etc.
- `total_reviews, property_rating` (important later for charts)

`property_features`

Amenity + attribute flags:

- `property_id` (pk/fk)
- `bedrooms, accommodates, bathrooms`
- `has_aircon, has_pool`, etc.
- "system_*" amenities

`property_scores`

Computed outputs:

- `property_id` (pk/fk)
- `market_id`
- `investment_score` (0–100)
- `score_breakdown` (JSON explainability)
- optionally `computed_at`

Why this split is good engineering

- performance can change, features change, core rarely changes
- scoring can be recomputed repeatedly without touching raw tables
- API can fetch "just what it needs" per endpoint