Property Management Application - ER Diagram

Relational Data Model

This document outlines the complete relational database schema for the property management system.

Entity-Relationship Overview

```
┌─────────────────┐
│     AGENTS      │
├─────────────────┤
│ id (PK, UUID)   │
│ firstName       │
│ lastName        │
│ email (UNIQUE)  │
│ mobileNumber    │
│ createdAt       │
│ updatedAt       │
└────────┬────────┘
         │ (1)
         │
         │ (*)
    ┌────▼───────────────┐
    │   PROPERTIES       │
    ├────────────────────┤
    │ id (PK, UUID)      │
    │ agent_id (FK)      │◄──── Foreign Key to Agents
    │ address            │
    │ description        │
    │ createdAt          │
    │ updatedAt          │
    └────┬───────────────┘
         │
      ┌──┴──┐
      │(1)  │(1)
      │     │
      │     └───────────────────┐
      │                         │
      │(*)                      │(*)
   ┌──▼───────────────┐    ┌────▼──────────┐
   │    TENANTS       │    │    NOTES      │
   ├──────────────────┤    ├───────────────┤
   │ id (PK, UUID)    │    │ id (PK)       │
   │ property_id(FK)  │    │ property_id   │
   │ name             │    │ title         │
   │ contact_email    │    │ content       │
   │ contact_phone    │    │ is_reminder   │
   │ lease_start      │    │ reminder_dt   │
   │ lease_end        │    │ createdAt     │
   │ createdAt        │    │ updatedAt     │
   │ updatedAt        │    └───────────────┘
   └──────────────────┘
```

Detailed Table Schemas

1. AGENTS Table

Purpose: Core entity representing property agents managing rental properties.

Column	Type	Constraints	Description
id	UUID	PRIMARY KEY	Unique identifier
firstName	VARCHAR(255)	NOT NULL	Agent's first name
lastName	VARCHAR(255)	NOT NULL	Agent's last name
email	VARCHAR(255)	UNIQUE, NOT NULL	Agent's email (must be unique)
mobileNumber	VARCHAR(20)	NOT NULL	Contact phone number
createdAt	TIMESTAMP	NOT NULL, DEFAULT NOW()	Record creation timestamp
updatedAt	TIMESTAMP	NOT NULL, DEFAULT NOW()	Last update timestamp

Primary Key: `id`

Indexes:

- `email` (UNIQUE)

---

2. PROPERTIES Table

Purpose: Represents rental properties managed by agents.

Column	Type	Constraints	Description
id	UUID	PRIMARY KEY	Unique property identifier
agent_id	UUID	FOREIGN KEY (agents.id), NOT NULL	Managing agent (references Agents)
address	VARCHAR(500)	NOT NULL	Physical property address
description	TEXT	nullable	Detailed property description
createdAt	TIMESTAMP	NOT NULL, DEFAULT NOW()	Record creation timestamp
updatedAt	TIMESTAMP	NOT NULL, DEFAULT NOW()	Last update timestamp

Primary Key: `id`

Foreign Keys:

- `agent_id` → `agents.id` (One Agent manages Many Properties)

Indexes:

- `agent_id` (for filtering by agent)

---

3. TENANTS Table

Purpose: Represents tenants/families occupying rental properties (one or more per property).

Column	Type	Constraints	Description
id	UUID	PRIMARY KEY	Unique tenant identifier
property_id	UUID	FOREIGN KEY (properties.id), NOT NULL	Occupied property (references Properties)
name	VARCHAR(255)	NOT NULL	Tenant/family head name
contact_email	VARCHAR(255)	NOT NULL	Tenant contact email
contact_phone	VARCHAR(20)	NOT NULL	Tenant contact phone
lease_start	DATE	NOT NULL	Lease agreement start date
lease_end	DATE	nullable	Lease agreement end date
createdAt	TIMESTAMP	NOT NULL, DEFAULT NOW()	Record creation timestamp
updatedAt	TIMESTAMP	NOT NULL, DEFAULT NOW()	Last update timestamp

Primary Key: `id`

Foreign Keys:

- `property_id` → `properties.id` (One Property has Many Tenants)

Indexes:

- `property_id` (for filtering by property)

---

4. NOTES Table

Purpose: Stores notes, reminders, and action items for properties (e.g., maintenance, pest control).

Column	Type	Constraints	Description
id	UUID	PRIMARY KEY	Unique note identifier
property_id	UUID	FOREIGN KEY (properties.id), NOT NULL	Associated property (references Properties)
title	VARCHAR(255)	NOT NULL	Brief note title
content	TEXT	NOT NULL	Full note/reminder content
is_reminder	BOOLEAN	DEFAULT FALSE	Flag: is this a reminder?
reminder_date	TIMESTAMP	nullable	Scheduled reminder date
createdAt	TIMESTAMP	NOT NULL, DEFAULT NOW()	Record creation timestamp
updatedAt	TIMESTAMP	nullable	Last update timestamp

Primary Key: `id`

Foreign Keys:

- `property_id` → `properties.id` (One Property has Many Notes)

Indexes:

- `property_id` (for filtering by property)
- `reminder_date` (for scheduling queries)

Relationships Summary

One-to-Many: Agents → Properties

```
Agent (1) ──manages──→ (*) Properties
```

- An agent can manage zero or more properties.
- Each property belongs to exactly one agent.
- **Foreign Key**: `properties.agent_id` references `agents.id`

One-to-Many: Properties → Tenants

```
Property (1) ──houses──→ (*) Tenants
```

- A property can have one or more tenants (per family).
- Each tenant occupies exactly one property.
- **Foreign Key**: `tenants.property_id` references `properties.id`

One-to-Many: Properties → Notes

```
Property (1) ──has──→ (*) Notes
```

- A property can have zero or more notes/reminders.
- Each note is associated with exactly one property.
- **Foreign Key**: `notes.property_id` references `properties.id`

Constraints & Rules

Primary Keys

- Every table has a `id` column (UUID type) as PRIMARY KEY
- Ensures uniqueness and fast lookups

Foreign Keys

- `properties.agent_id` → `agents.id`
- `tenants.property_id` → `properties.id`
- `notes.property_id` → `properties.id`
- Enforce referential integrity (no orphaned records)

Uniqueness Constraints

- `agents.email` must be UNIQUE (no duplicate emails)

Not Null Constraints

- **Agents**: `id`, `firstName`, `lastName`, `email`, `mobileNumber`, `createdAt`, `updatedAt`
- **Properties**: `id`, `agent_id`, `address`, `createdAt`, `updatedAt`
- **Tenants**: `id`, `property_id`, `name`, `contact_email`, `contact_phone`, `lease_start`, `createdAt`, `updatedAt`
- **Notes**: `id`, `property_id`, `title`, `content`, `createdAt`

Nullable Fields

- `properties.description` (optional property details)
- `tenants.lease_end` (may be open-ended or TBD)
- `notes.reminder_date` (only set if is_reminder = true)
- `notes.updatedAt` (nullable for immutable notes)

Indexing Strategy

For Performance:

- Primary Keys: Auto-indexed
- Foreign Keys: `agent_id`, `property_id` (common filter columns)
- Unique Constraints: `agents.email` (auto-indexed)
- Reminder Queries: `notes.reminder_date` (for scheduling queries)

Sample Data Flow Example

```
1. Create Agent
   → INSERT INTO agents (id, firstName, lastName, email, mobileNumber, createdAt, updatedAt)

2. Create Property for Agent
   → INSERT INTO properties (id, agent_id, address, description, createdAt, updatedAt)

3. Add Tenants to Property
   → INSERT INTO tenants (id, property_id, name, contact_email, contact_phone, lease_start, lease_end, createdAt, updatedAt)

4. Create Notes/Reminders for Property
   → INSERT INTO notes (id, property_id, title, content, is_reminder, reminder_date, createdAt)
```

Normalization

This schema follows **Third Normal Form (3NF)**:

- ✓ All attributes depend on the primary key (1NF)
- ✓ No partial dependencies (2NF)
- ✓ No transitive dependencies (3NF)

- ✓ Foreign keys enforce referential integrity
- ✓ No data redundancy across tables

Future Enhancements

Potential extensions without breaking current schema:

1. **Maintenance Logs**: Track completed maintenance work on properties
2. **Payment Records**: Track rent payments from tenants
3. **Documents**: Store lease agreements, ID copies (document storage)
4. **Audit Trail**: Track all changes to agent/property records
5. **Notifications**: Email/SMS alerts for reminders (linked to notes)

ER Diagram generated for Property Management Application Schema Version: 1.0 Last Updated: January 21, 2026