# DATABASE MANAGEMENT SYSTEM (UCS310)

# PROJECT REPORT

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

## Real Estate Investment & Rental Profitability Tracker

**Submitted To**:

Ms. Reaya Grewal

**Submitted by**:

Arush Singhal – 102303889 (2C63)
Lakshay Sawhney – 102303872 (2C63)
Priyansh Chowhan – 102303880 (2C63)

# Index

# Introduction

Real estate investors face several critical challenges in managing their portfolios, including:

- **Data Fragmentation:** Managing multiple properties, rental agreements, and expenses without a centralized system leads to scattered and inconsistent financial records.
- **Inefficient Financial Calculations:** Computing ROI, cash flow manually is prone to errors and lacks real-time automation.
- **Market Volatility:** Investors struggle to track location-specific rental trends and property appreciation without proper insights.
- **Comparative Analysis Challenges:** Evaluating and comparing properties across different neighborhoods is difficult without structured data retrieval.

To address these issues, this project implements a relational database that utilizes SQL JOIN queries, for fast, accurate, and data-driven decision-making.

# Objectives

- Develop a relational database system to store and manage real estate investment data efficiently.
- Leverage SQL JOIN queries to extract, aggregate, and analyze financial data across multiple entities.
- Automate financial calculations, such as Return on Investment (ROI) and Cash Flow.
- Enable comparative analysis of properties across various locations based on historical and real-time market trends.
- Incorporate and reference the ER Diagram to demonstrate database structure, entity relationships, and system workflow.
- Developing a functional website to display data effectively.

# Methodology

The development of the Real Estate Investment & Rental Profitability Tracker followed a systematic approach to ensure data integrity, functionality, and user interactivity. The major stages of the project are outlined below:

- Data Modeling and Design
    - An Entity-Relationship (ER) Diagram was created using Lucidchart to visualize the database schema and define relationships between various entities such as Properties, Investors, Transactions, and Rentals.
    - The diagram served as the blueprint for the database structure, enabling precise schema creation.

- Database Construction
    - Using the ER diagram as a reference, relational tables were created in Oracle SQL, with appropriate data types, primary keys, and foreign key constraints to enforce data integrity.
    - Sample data entries were inserted into the tables to simulate real-world scenarios and enable testing of query logic.

- Query Development
    - A comprehensive set of SQL queries was designed and implemented to:
        - Retrieve investment and rental data.
        - Compute financial metrics such as ROI, profit margins etc.
        - Identify trends and anomalies.

- Backend Development
    - The backend was developed using Django, a high-level Python web framework, to:
        - Connect the application to the Oracle database.
        - Handle API routing, data fetching, and business logic.
        - Serve processed data securely to the frontend.

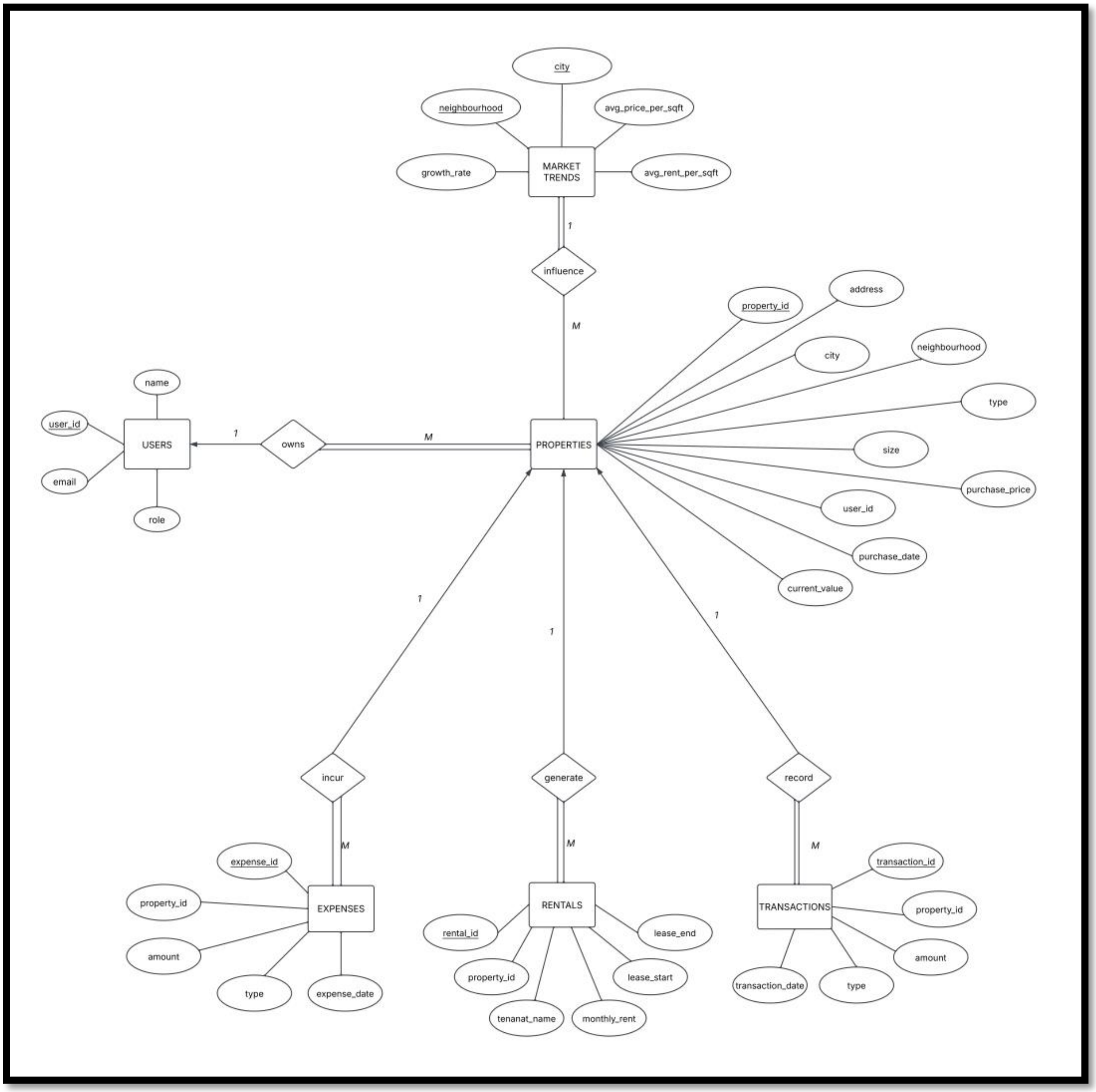
- Frontend Development
    - The user interface was built using React, a modern JavaScript library, to provide a responsive and intuitive experience.
    - The frontend dynamically displays queried data, visualizations, and insights to users for better decision-making.

# Expected Outcome

- Financial Insights & Property Analytics
  - Real-time monitoring of rental income, expenses, and cash flow
  - Automated ROI (Return on Investment) and NOI (Net Operating Income) calculations for investors
- Optimized Query Performance & Data Storage
  - Faster execution of JOIN queries for retrieving rental and expense reports
  - Indexed data structures for efficient storage and quick data retrieval
- Improved Investment Decision-Making
  - Market trend tracking to identify high-growth neighborhoods
  - Comparative property performance analysis across different cities
- Scalability & Future Expansion
  - Capability to integrate predictive analytics and external API data sources in future versions
  - Supports multi-user access, enabling collaboration between property managers and investors

# Entity-Relationship Model

The ER Diagram for the Real Estate Investment & Rental Profitability Tracker shows how system entities interact. It structures data relationships, enforces integrity, and optimizes queries for managing properties, rentals, expenses, transactions, and market trends.



**Standard Notations Used**

- Entities: Represented as rectangles (e.g., Users, Properties).
- Attributes: Listed within entities (e.g., name, price, rent amount).
- Relationships: Depicted as diamonds or lines between entities (e.g., "Owns").
- Primary Keys (PK): Uniquely identify records in each entity.
- Foreign Keys (FK): Establish links between related tables.

**Primary Keys & Foreign Keys Explanation**

- Primary Keys ensure uniqueness:
    - user_id for Users identifies each investor.
    - property_id for Properties differentiates real estate assets.
    - rental_id, expense_id, transaction_id uniquely identify respective records.
- Foreign Keys create relationships:
    - user_id in Properties links properties to owners.
    - property_id in Rentals, Expenses, and Transactions maintains data consistency

**Key Entity Relationships**

Users - Properties (1:M): A user owns multiple properties, but each property belongs to one user.

Properties - Rentals (1:M): A property may have multiple rental agreements.

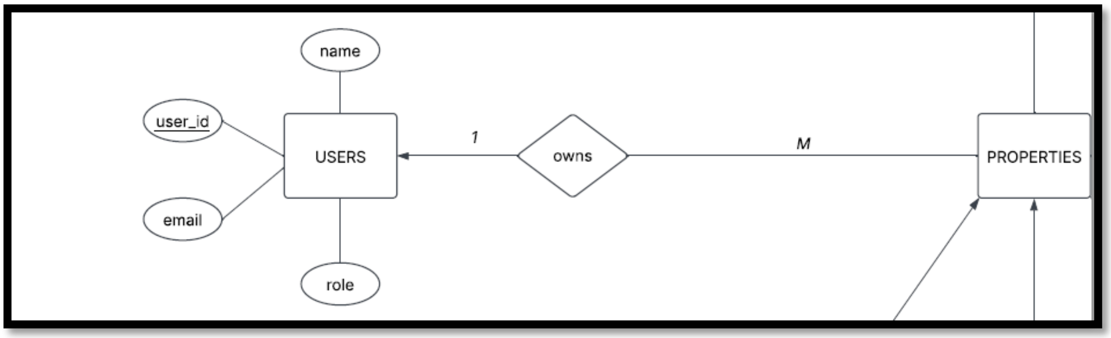
Properties - Expenses (1:M): Each property can incur several expenses.

Properties - Transactions (1:M): Tracks property purchases and sales.

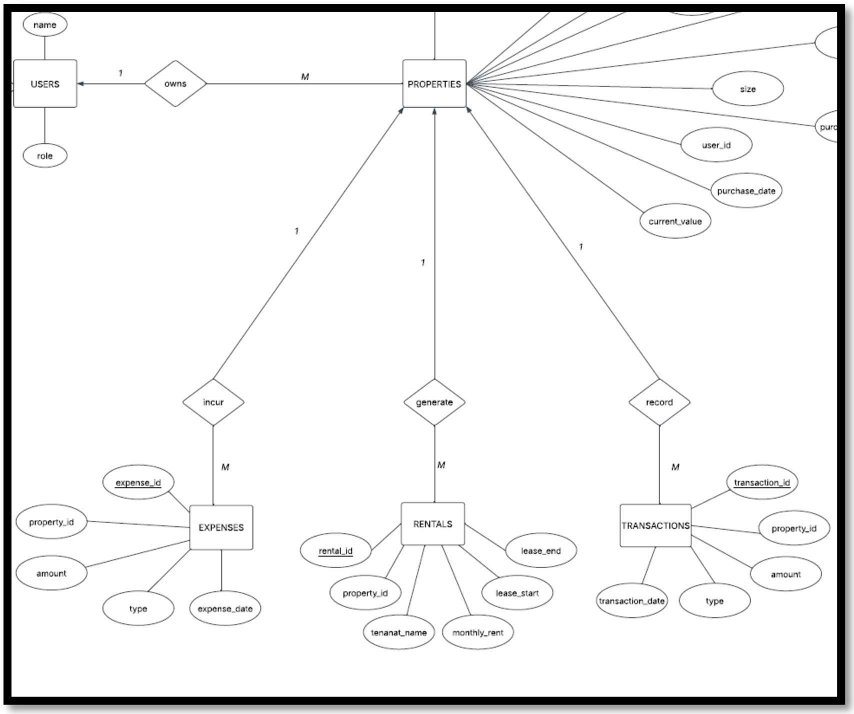
Market Trends - Properties (M:1): Many properties exist within a neighborhood influenced by market conditions.

**Key Relationships in the ER Model**

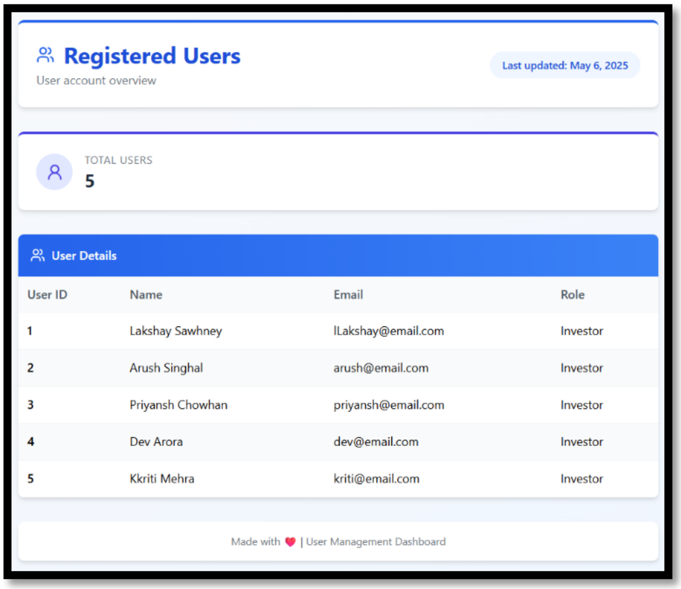
- Users own multiple Properties (1:M).



- Properties generate multiple Rentals (1:M).
- Properties incur multiple Expenses (1:M)
- Properties have multiple Transactions (1:M).
- Market Trends affect Properties (M:1).



# Data Tables Overview



## Registered Users

User account overview

Last updated: May 6, 2025

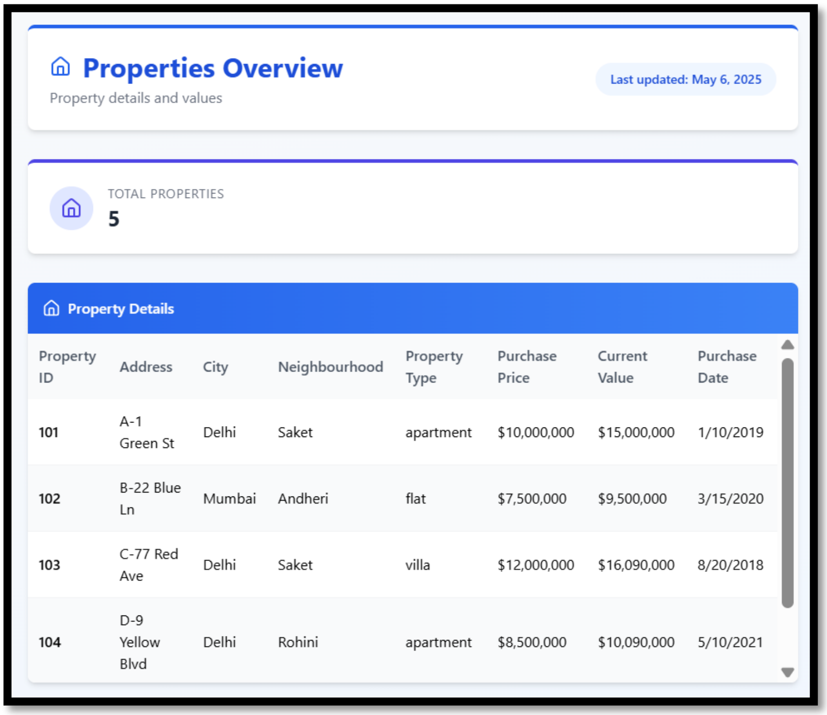
TOTAL USERS

5

User Details

| User ID | Name | Email | Role |
|---|---|---|---|
| 1 | Lakshay Sawhney | lLakshay@email.com | Investor |
| 2 | Arush Singhal | arush@email.com | Investor |
| 3 | Priyansh Chowhan | priyansh@email.com | Investor |
| 4 | Dev Arora | dev@email.com | Investor |
| 5 | Kkriti Mehra | kriti@email.com | Investor |

Made with ❤ | User Management Dashboard



## Properties Overview

Property details and values

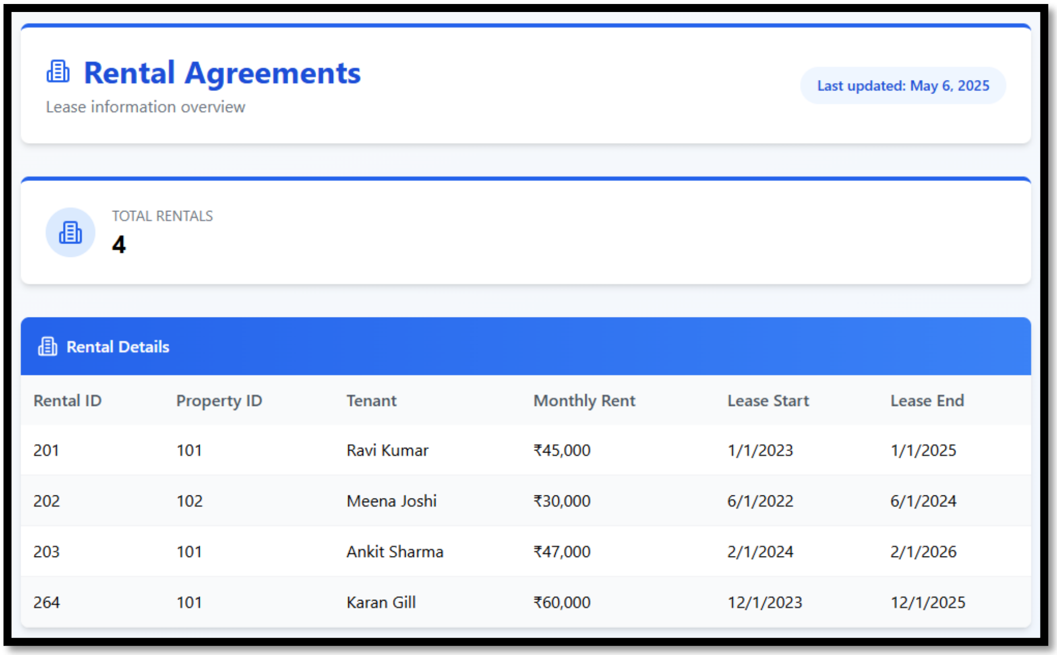
Last updated: May 6, 2025

TOTAL PROPERTIES

5

Property Details

| Property ID | Address | City | Neighbourhood | Property Type | Purchase Price | Current Value | Purchase Date |
|---|---|---|---|---|---|---|---|
| 101 | A-1 Green St | Delhi | Saket | apartment | $10,000,000 | $15,000,000 | 1/10/2019 |
| 102 | B-22 Blue Ln | Mumbai | Andheri | flat | $7,500,000 | $9,500,000 | 3/15/2020 |
| 103 | C-77 Red Ave | Delhi | Saket | villa | $12,000,000 | $16,090,000 | 8/20/2018 |
| 104 | D-9 Yellow Blvd | Delhi | Rohini | apartment | $8,500,000 | $10,090,000 | 5/10/2021 |



## Rental Agreements

Lease information overview

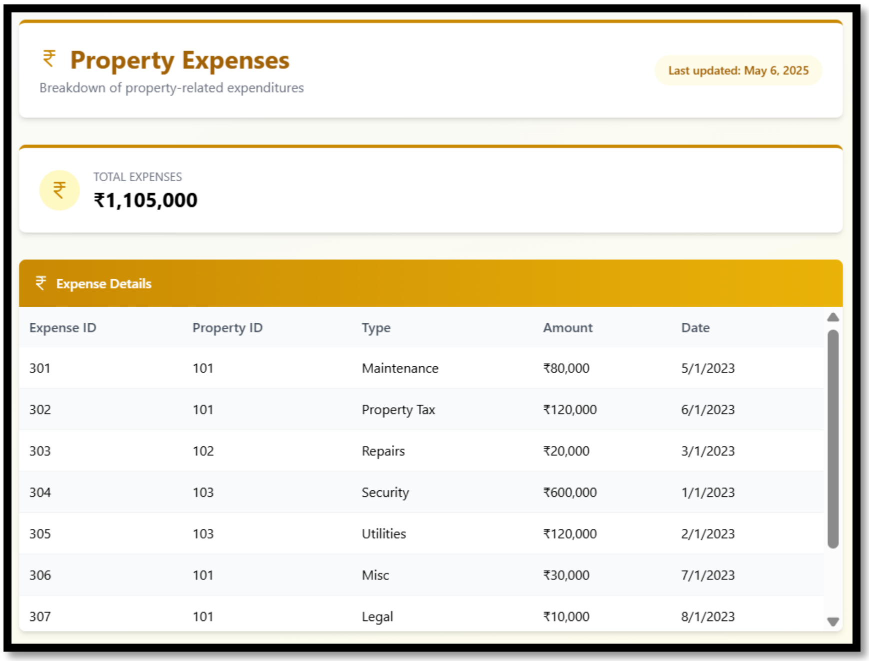
Last updated: May 6, 2025

TOTAL RENTALS

**4**

### Rental Details

| Rental ID | Property ID | Tenant | Monthly Rent | Lease Start | Lease End |
|---|---|---|---|---|---|
| 201 | 101 | Ravi Kumar | ₹45,000 | 1/1/2023 | 1/1/2025 |
| 202 | 102 | Meena Joshi | ₹30,000 | 6/1/2022 | 6/1/2024 |
| 203 | 101 | Ankit Sharma | ₹47,000 | 2/1/2024 | 2/1/2026 |
| 264 | 101 | Karan Gill | ₹60,000 | 12/1/2023 | 12/1/2025 |



## ₹ Property Expenses

Breakdown of property-related expenditures

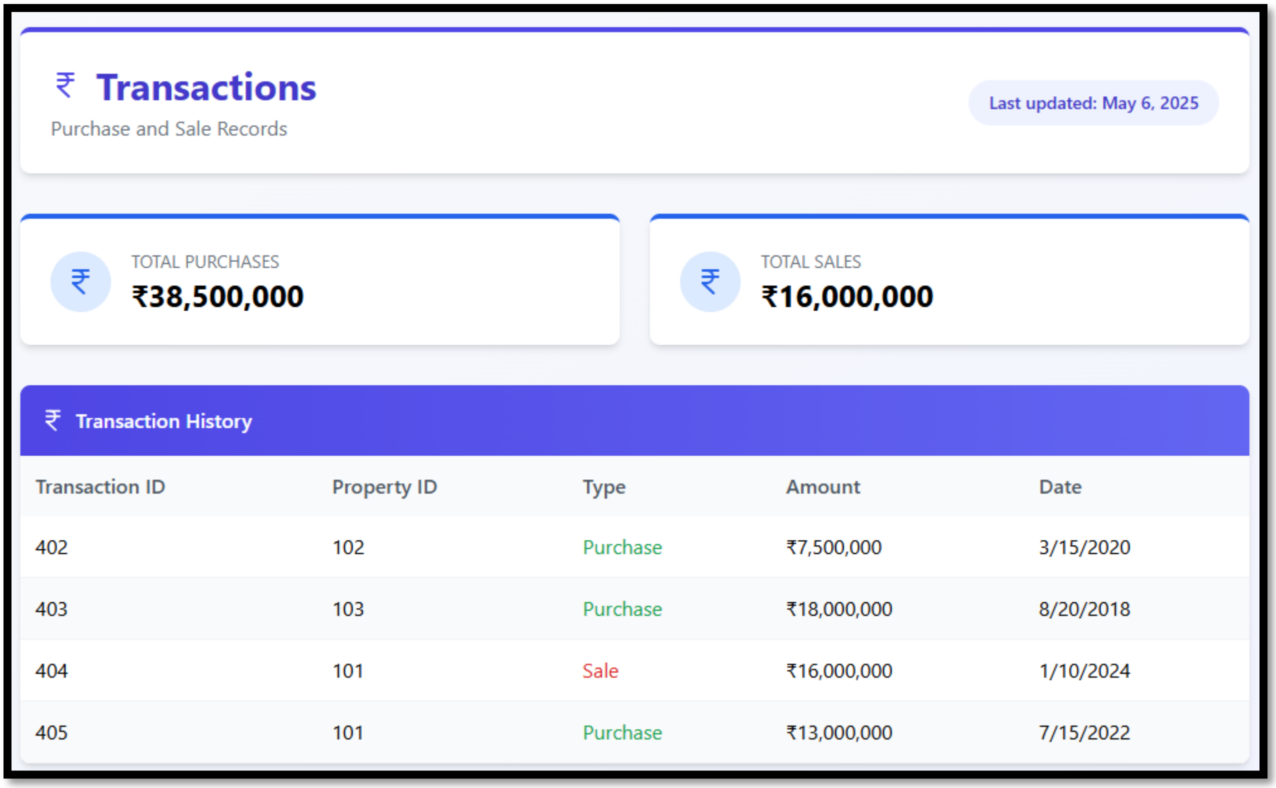
Last updated: May 6, 2025

TOTAL EXPENSES

**₹1,105,000**

### ₹ Expense Details

| Expense ID | Property ID | Type | Amount | Date |
|---|---|---|---|---|
| 301 | 101 | Maintenance | ₹80,000 | 5/1/2023 |
| 302 | 101 | Property Tax | ₹120,000 | 6/1/2023 |
| 303 | 102 | Repairs | ₹20,000 | 3/1/2023 |
| 304 | 103 | Security | ₹600,000 | 1/1/2023 |
| 305 | 103 | Utilities | ₹120,000 | 2/1/2023 |
| 306 | 101 | Misc | ₹30,000 | 7/1/2023 |
| 307 | 101 | Legal | ₹10,000 | 8/1/2023 |



# ₹ Transactions

Purchase and Sale Records

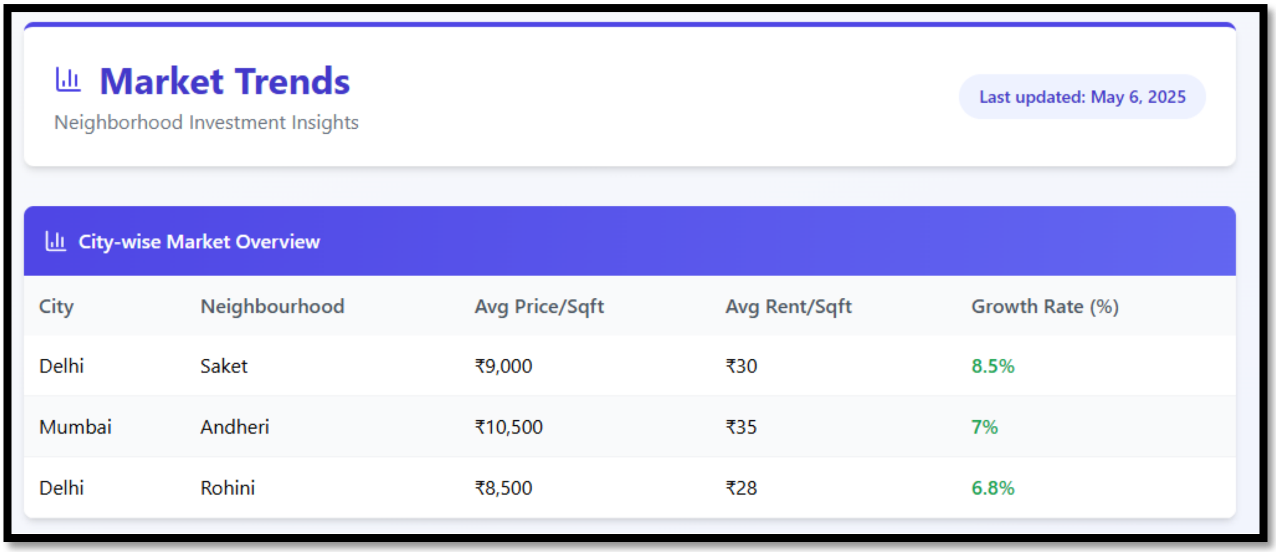
Last updated: May 6, 2025

₹ TOTAL PURCHASES **₹38,500,000**

₹ TOTAL SALES **₹16,000,000**

## ₹ Transaction History

| Transaction ID | Property ID | Type | Amount | Date |
|---|---|---|---|---|
| 402 | 102 | Purchase | ₹7,500,000 | 3/15/2020 |
| 403 | 103 | Purchase | ₹18,000,000 | 8/20/2018 |
| 404 | 101 | Sale | ₹16,000,000 | 1/10/2024 |
| 405 | 101 | Purchase | ₹13,000,000 | 7/15/2022 |



# Market Trends

Neighborhood Investment Insights

Last updated: May 6, 2025

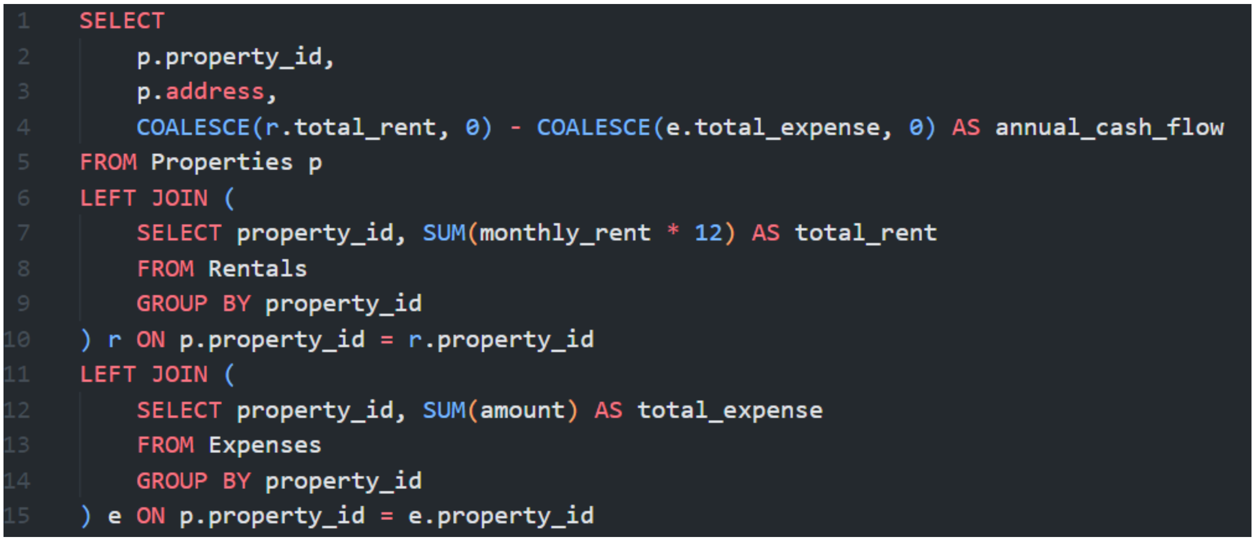
## City-wise Market Overview

| City | Neighbourhood | Avg Price/Sqft | Avg Rent/Sqft | Growth Rate (%) |
|---|---|---|---|---|
| Delhi | Saket | ₹9,000 | ₹30 | 8.5% |
| Mumbai | Andheri | ₹10,500 | ₹35 | 7% |
| Delhi | Rohini | ₹8,500 | ₹28 | 6.8% |

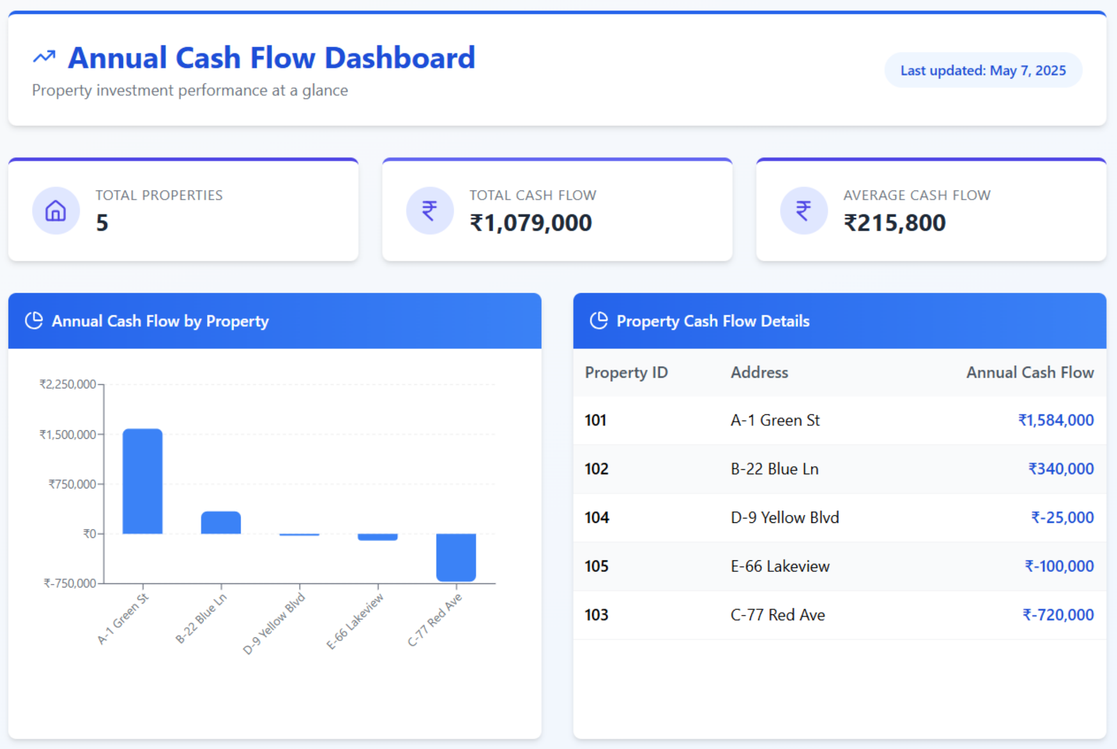
# SQL Queries and Their Results

## Financial Analysis

Cash Flow for each property



```sql
SELECT
    p.property_id,
    p.address,
    COALESCE(r.total_rent, 0) - COALESCE(e.total_expense, 0) AS annual_cash_flow
FROM Properties p
LEFT JOIN (
    SELECT property_id, SUM(monthly_rent * 12) AS total_rent
    FROM Rentals
    GROUP BY property_id
) r ON p.property_id = r.property_id
LEFT JOIN (
    SELECT property_id, SUM(amount) AS total_expense
    FROM Expenses
    GROUP BY property_id
) e ON p.property_id = e.property_id
```



Annual Cash Flow Dashboard

Property investment performance at a glance

Last updated: May 7, 2025

TOTAL PROPERTIES 5

TOTAL CASH FLOW ₹1,079,000

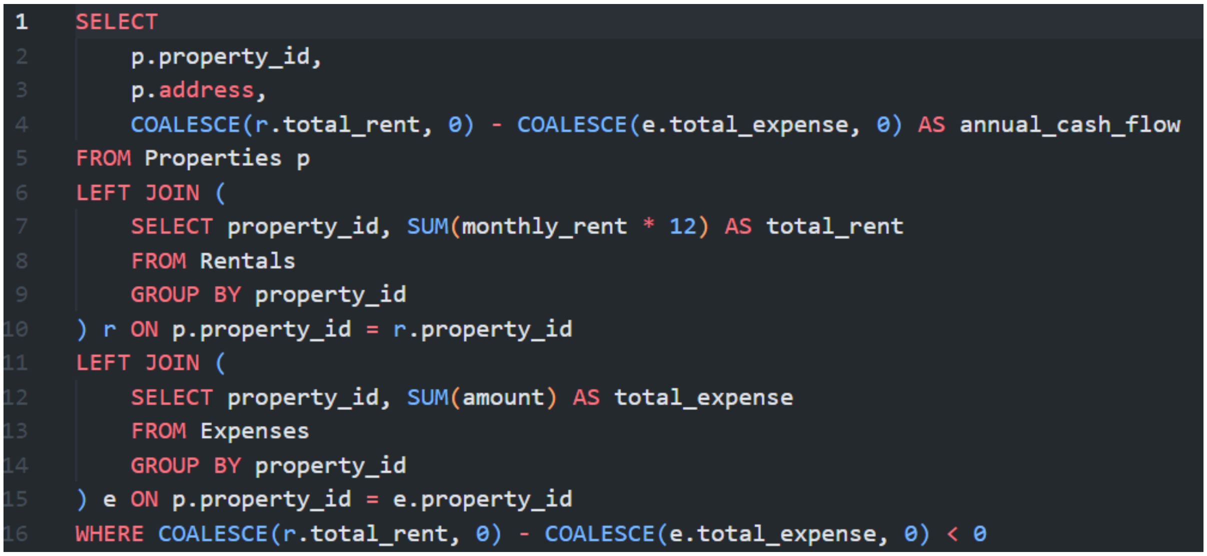
AVERAGE CASH FLOW ₹215,800

Annual Cash Flow by Property

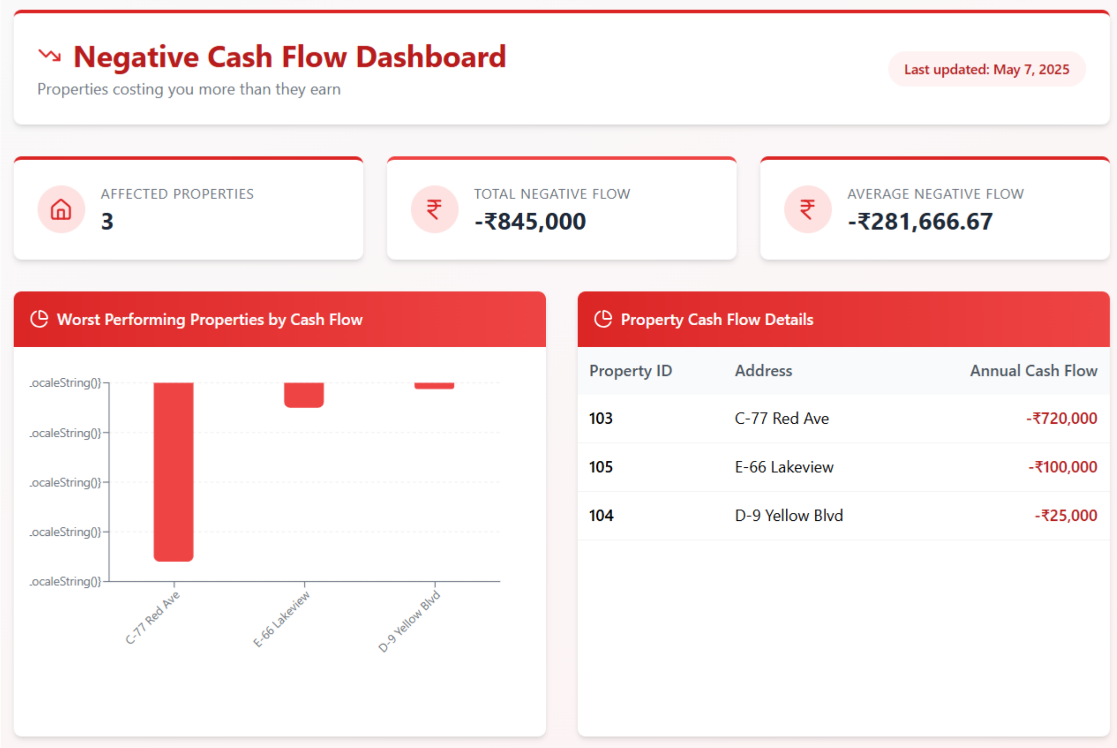
Property Cash Flow Details

| Property ID | Address | Annual Cash Flow |
|---|---|---|
| 101 | A-1 Green St | ₹1,584,000 |
| 102 | B-22 Blue Ln | ₹340,000 |
| 104 | D-9 Yellow Blvd | ₹-25,000 |
| 105 | E-66 Lakeview | ₹-100,000 |
| 103 | C-77 Red Ave | ₹-720,000 |

List properties with negative cash flow



```sql
SELECT
    p.property_id,
    p.address,
    COALESCE(r.total_rent, 0) - COALESCE(e.total_expense, 0) AS annual_cash_flow
FROM Properties p
LEFT JOIN (
    SELECT property_id, SUM(monthly_rent * 12) AS total_rent
    FROM Rentals
    GROUP BY property_id
) r ON p.property_id = r.property_id
LEFT JOIN (
    SELECT property_id, SUM(amount) AS total_expense
    FROM Expenses
    GROUP BY property_id
) e ON p.property_id = e.property_id
WHERE COALESCE(r.total_rent, 0) - COALESCE(e.total_expense, 0) < 0
```



## Negative Cash Flow Dashboard

Properties costing you more than they earn

Last updated: May 7, 2025

AFFECTED PROPERTIES
3

TOTAL NEGATIVE FLOW
-₹845,000

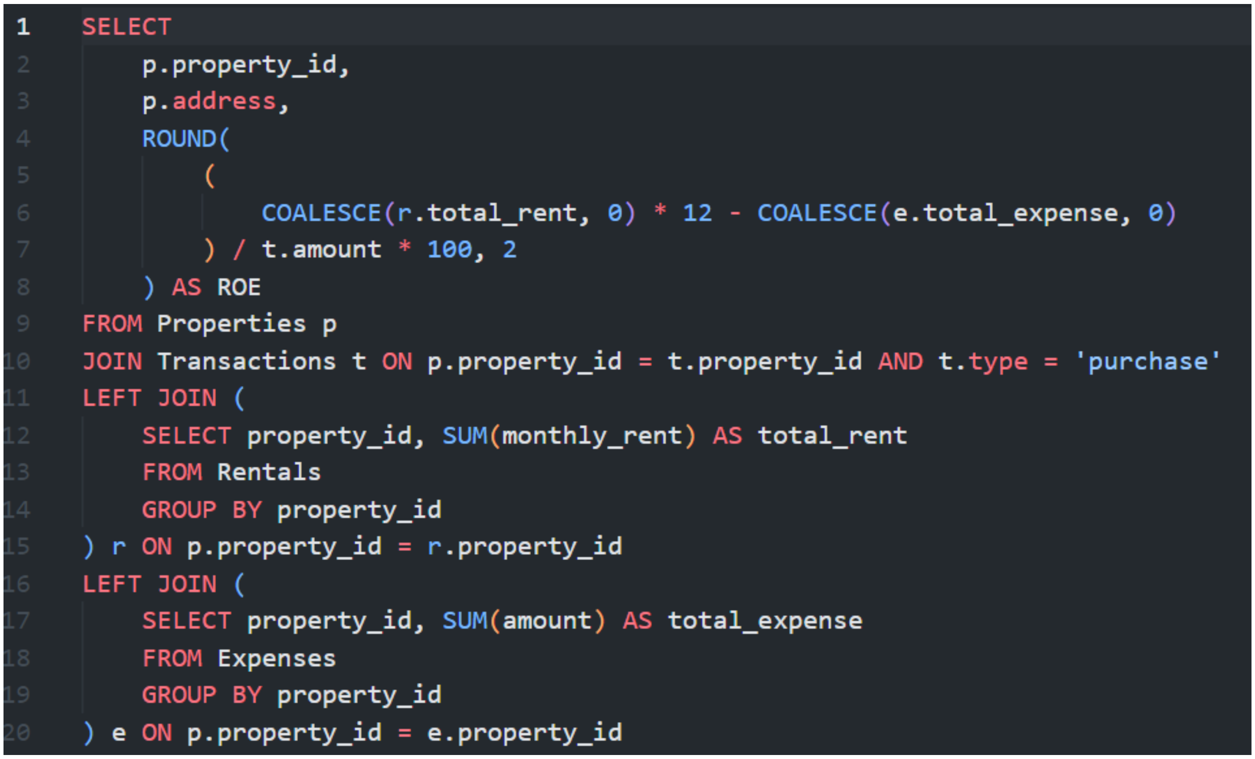
AVERAGE NEGATIVE FLOW
-₹281,666.67

Worst Performing Properties by Cash Flow

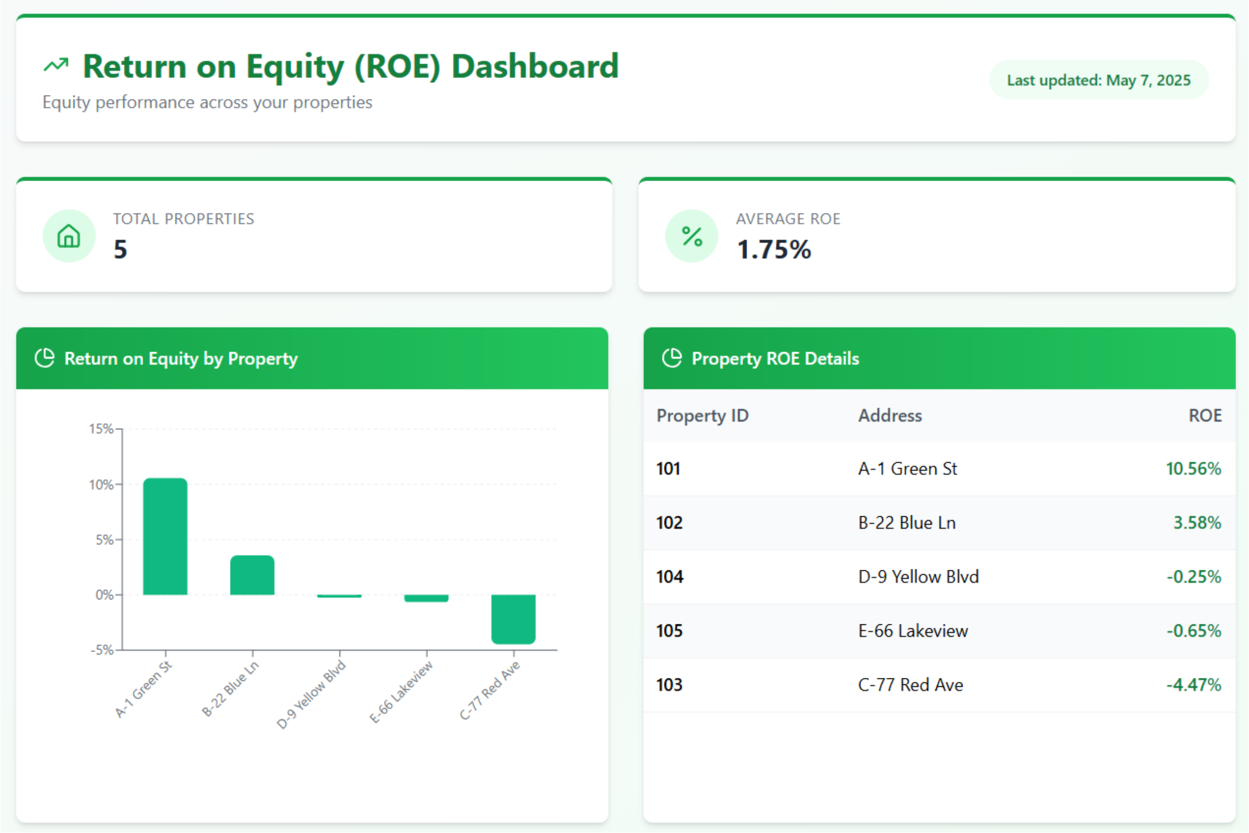
Property Cash Flow Details

| Property ID | Address | Annual Cash Flow |
|---|---|---|
| 103 | C-77 Red Ave | -₹720,000 |
| 105 | E-66 Lakeview | -₹100,000 |
| 104 | D-9 Yellow Blvd | -₹25,000 |

Return on equity per property



```
SELECT
    p.property_id,
    p.address,
    ROUND(
        (
            COALESCE(r.total_rent, 0) * 12 - COALESCE(e.total_expense, 0)
        ) / t.amount * 100, 2
    ) AS ROE
FROM Properties p
JOIN Transactions t ON p.property_id = t.property_id AND t.type = 'purchase'
LEFT JOIN (
    SELECT property_id, SUM(monthly_rent) AS total_rent
    FROM Rentals
    GROUP BY property_id
) r ON p.property_id = r.property_id
LEFT JOIN (
    SELECT property_id, SUM(amount) AS total_expense
    FROM Expenses
    GROUP BY property_id
) e ON p.property_id = e.property_id
```



Return on Equity (ROE) Dashboard

Equity performance across your properties

Last updated: May 7, 2025

TOTAL PROPERTIES

5

AVERAGE ROE

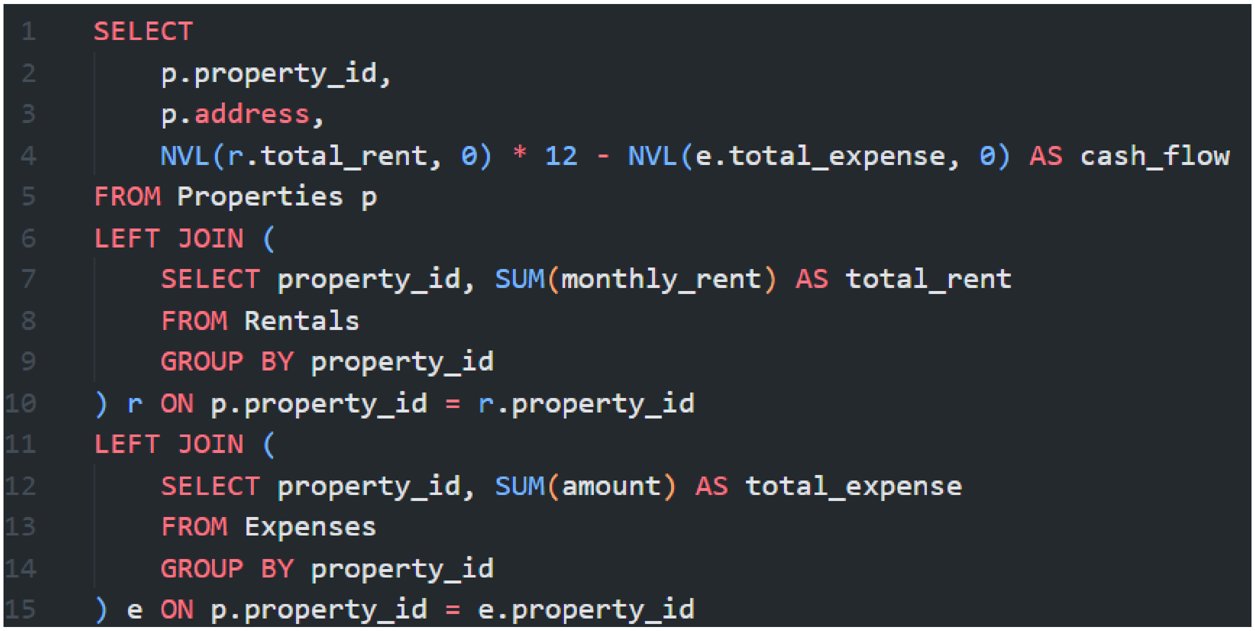
1.75%

Return on Equity by Property

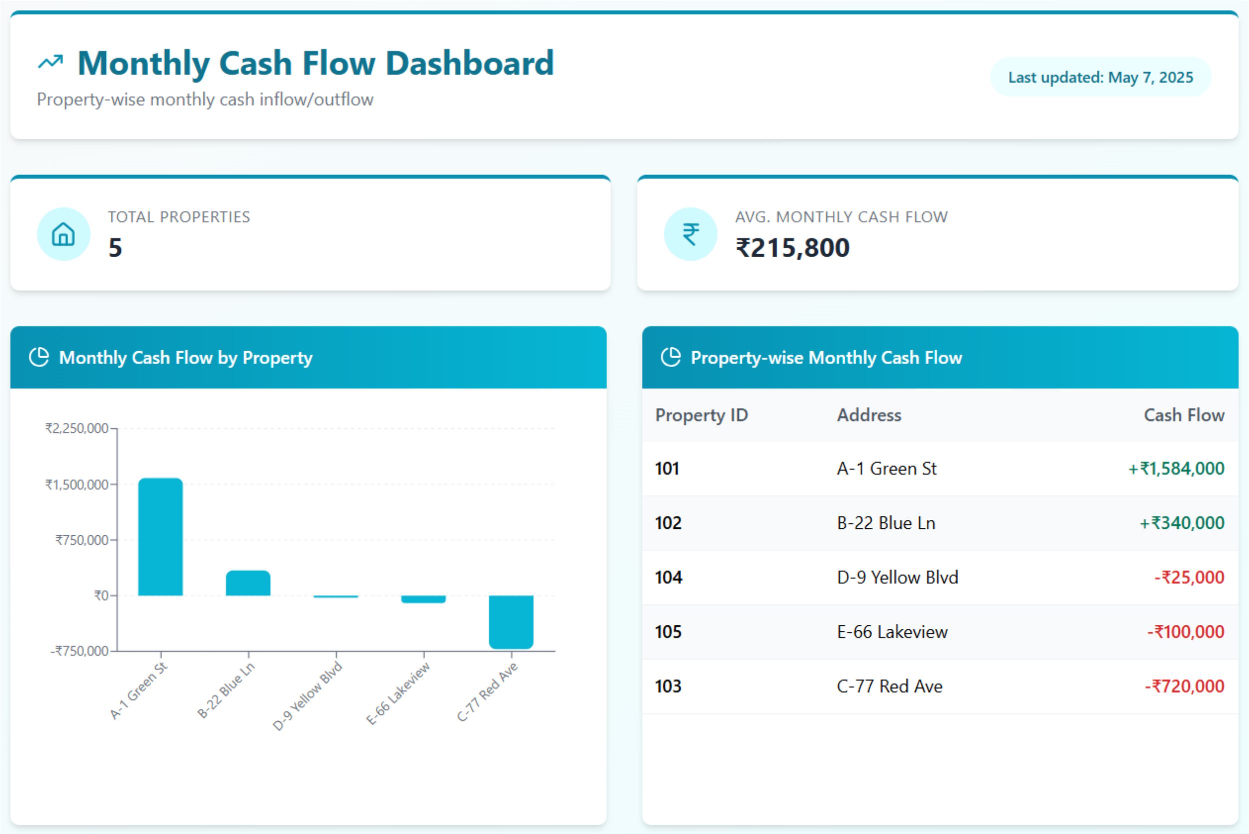
Property ROE Details

| Property ID | Address | ROE |
|---|---|---|
| 101 | A-1 Green St | 10.56% |
| 102 | B-22 Blue Ln | 3.58% |
| 104 | D-9 Yellow Blvd | -0.25% |
| 105 | E-66 Lakeview | -0.65% |
| 103 | C-77 Red Ave | -4.47% |

## Monthly Flow



```
SELECT
    p.property_id,
    p.address,
    NVL(r.total_rent, 0) * 12 - NVL(e.total_expense, 0) AS cash_flow
FROM Properties p
LEFT JOIN (
    SELECT property_id, SUM(monthly_rent) AS total_rent
    FROM Rentals
    GROUP BY property_id
) r ON p.property_id = r.property_id
LEFT JOIN (
    SELECT property_id, SUM(amount) AS total_expense
    FROM Expenses
    GROUP BY property_id
) e ON p.property_id = e.property_id
```



### Monthly Cash Flow Dashboard

Property-wise monthly cash inflow/outflow

Last updated: May 7, 2025

TOTAL PROPERTIES
5

AVG. MONTHLY CASH FLOW
₹215,800

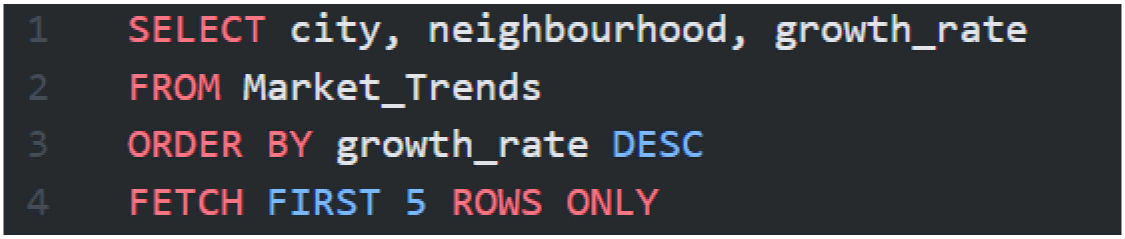
Monthly Cash Flow by Property

Property-wise Monthly Cash Flow

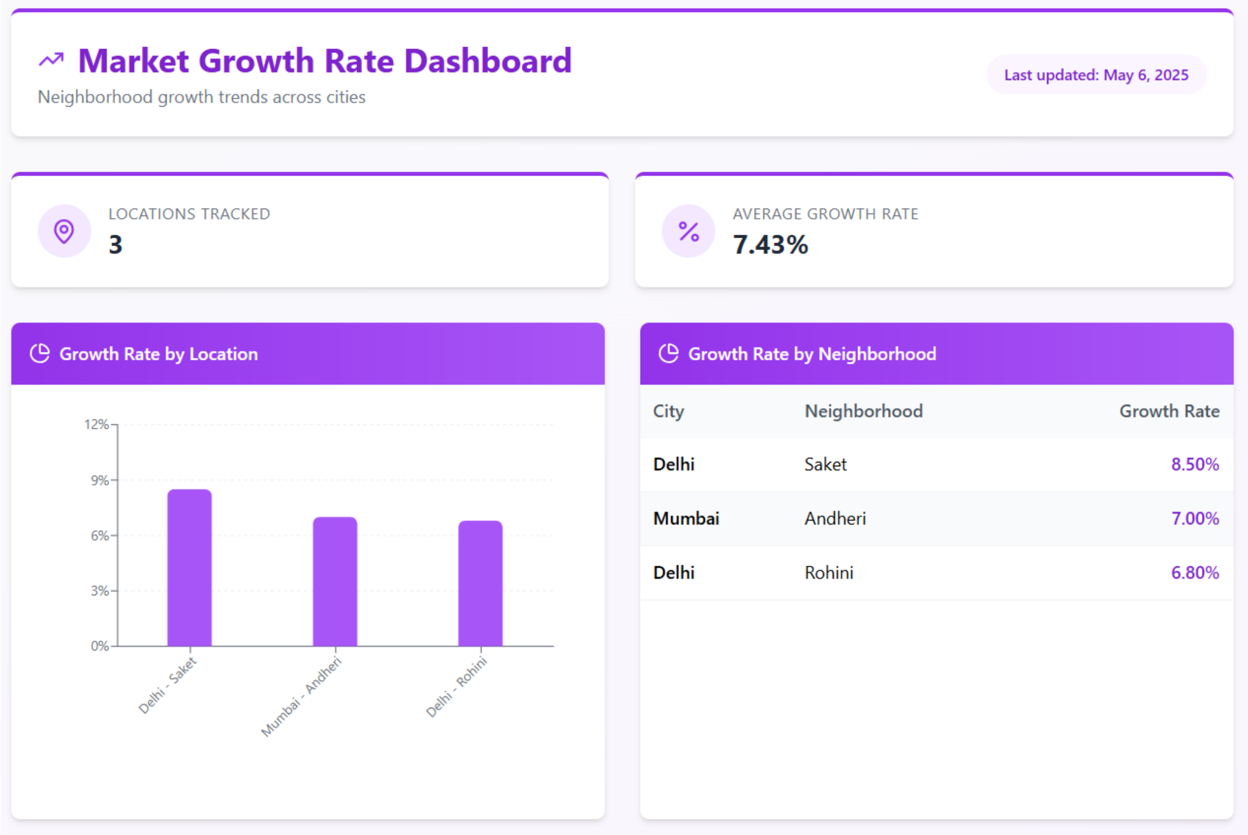
| Property ID | Address | Cash Flow |
|---|---|---|
| 101 | A-1 Green St | +₹1,584,000 |
| 102 | B-22 Blue Ln | +₹340,000 |
| 104 | D-9 Yellow Blvd | -₹25,000 |
| 105 | E-66 Lakeview | -₹100,000 |
| 103 | C-77 Red Ave | -₹720,000 |

## Market Analysis

Fastest growing neighborhoods



```sql
SELECT city, neighbourhood, growth_rate
FROM Market_Trends
ORDER BY growth_rate DESC
FETCH FIRST 5 ROWS ONLY
```



Market Growth Rate Dashboard

Neighborhood growth trends across cities

Last updated: May 6, 2025

LOCATIONS TRACKED
3

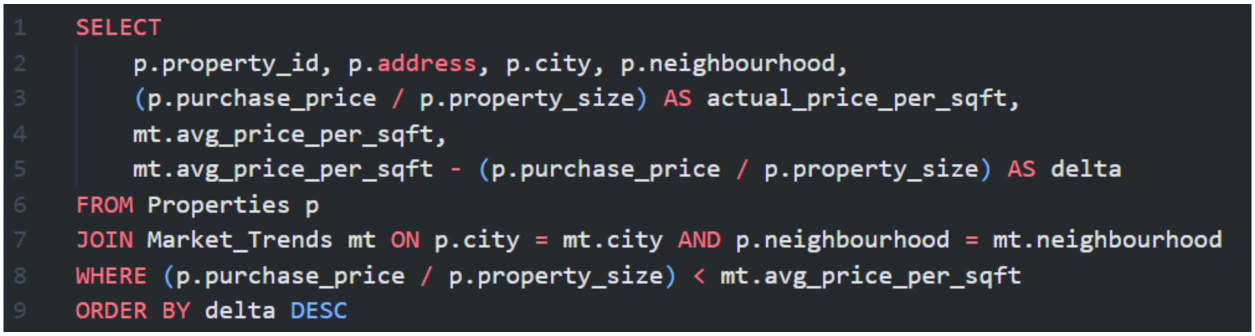
AVERAGE GROWTH RATE
7.43%

Growth Rate by Location

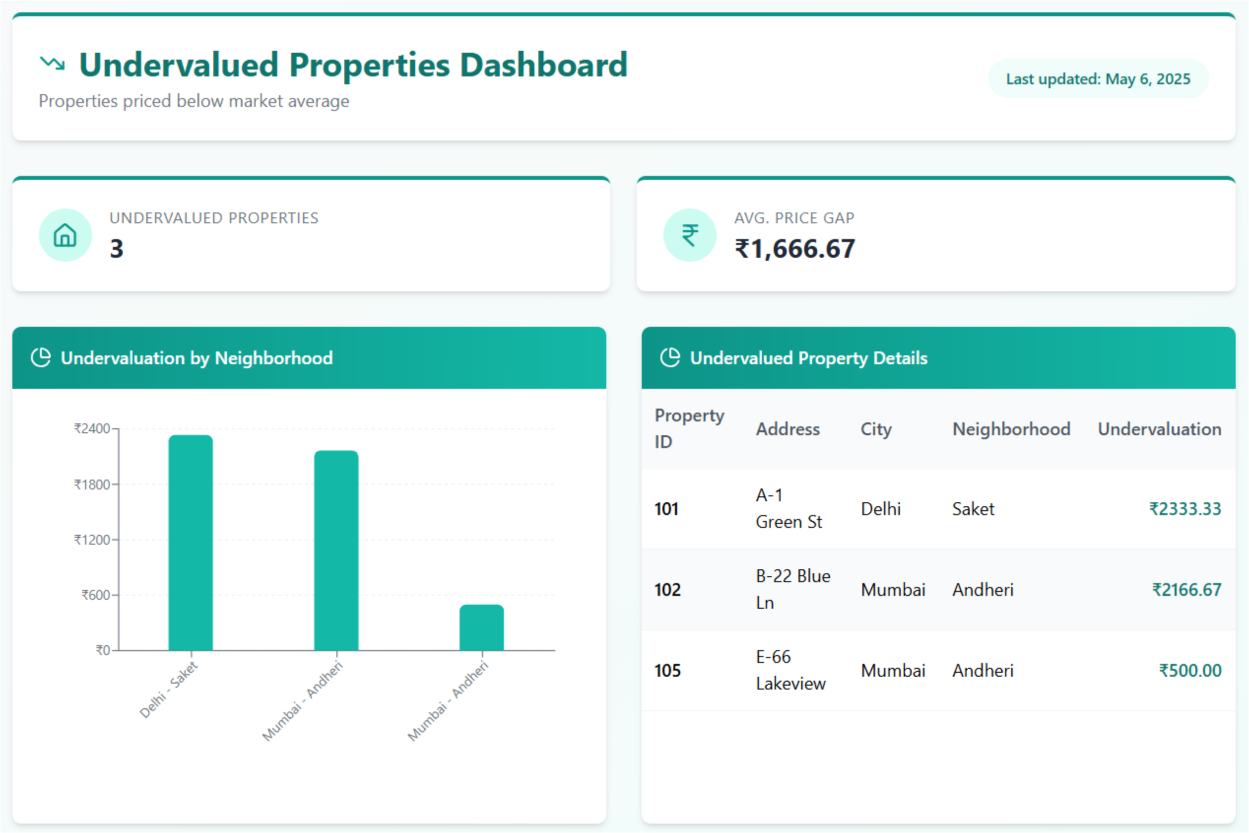
Growth Rate by Neighborhood

| City | Neighborhood | Growth Rate |
|---|---|---|
| **Delhi** | Saket | 8.50% |
| **Mumbai** | Andheri | 7.00% |
| **Delhi** | Rohini | 6.80% |

List undervalued properties



```
SELECT
    p.property_id, p.address, p.city, p.neighbourhood,
    (p.purchase_price / p.property_size) AS actual_price_per_sqft,
    mt.avg_price_per_sqft,
    mt.avg_price_per_sqft - (p.purchase_price / p.property_size) AS delta
FROM Properties p
JOIN Market_Trends mt ON p.city = mt.city AND p.neighbourhood = mt.neighbourhood
WHERE (p.purchase_price / p.property_size) < mt.avg_price_per_sqft
ORDER BY delta DESC
```



## Undervalued Properties Dashboard

Properties priced below market average

Last updated: May 6, 2025

UNDERVALUED PROPERTIES
3

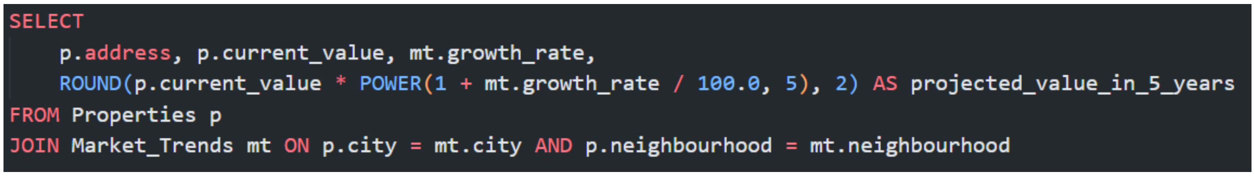
AVG. PRICE GAP
₹1,666.67

### Undervaluation by Neighborhood

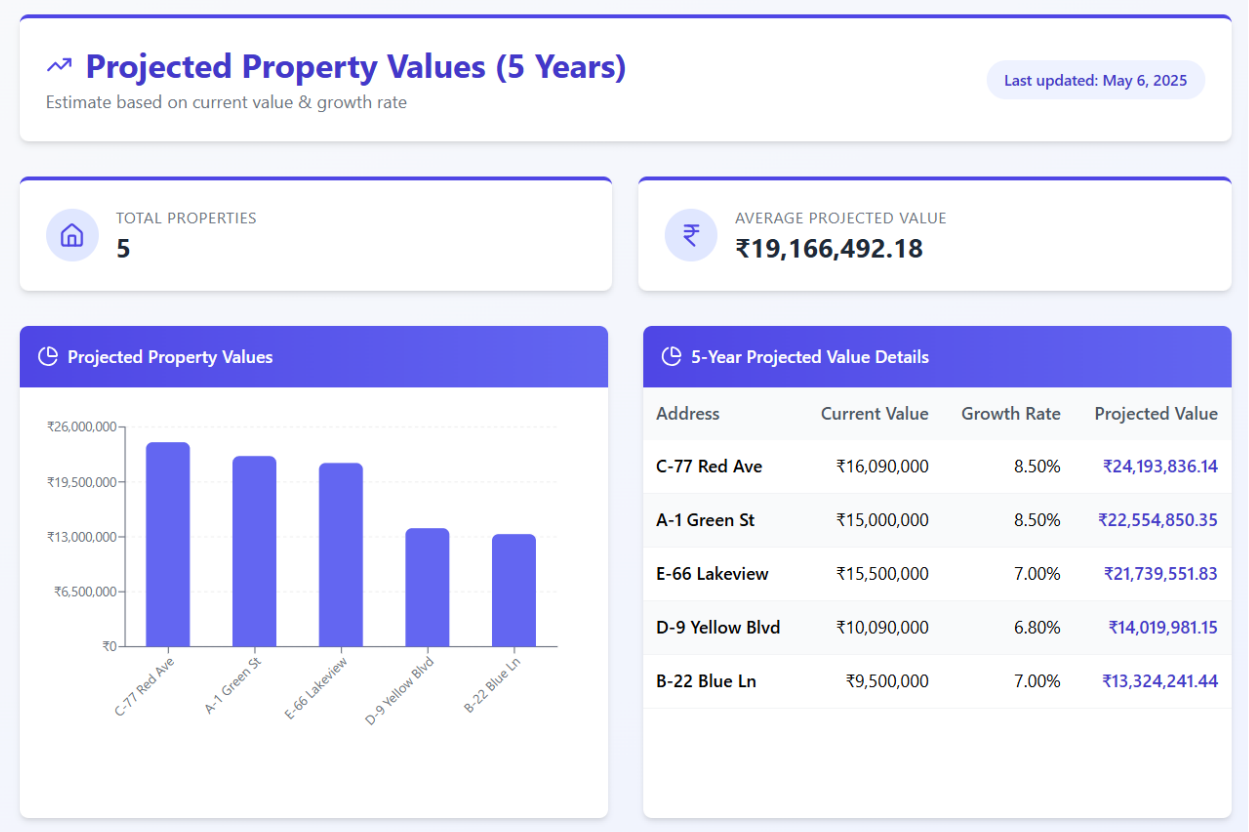
### Undervalued Property Details

| Property ID | Address | City | Neighborhood | Undervaluation |
|---|---|---|---|---|
| 101 | A-1 Green St | Delhi | Saket | ₹2333.33 |
| 102 | B-22 Blue Ln | Mumbai | Andheri | ₹2166.67 |
| 105 | E-66 Lakeview | Mumbai | Andheri | ₹500.00 |

Projected property value in 5 years



```sql
SELECT
    p.address, p.current_value, mt.growth_rate,
    ROUND(p.current_value * POWER(1 + mt.growth_rate / 100.0, 5), 2) AS projected_value_in_5_years
FROM Properties p
JOIN Market_Trends mt ON p.city = mt.city AND p.neighbourhood = mt.neighbourhood
```



## Projected Property Values (5 Years)

Estimate based on current value & growth rate

Last updated: May 6, 2025

TOTAL PROPERTIES

**5**

AVERAGE PROJECTED VALUE

**₹19,166,492.18**

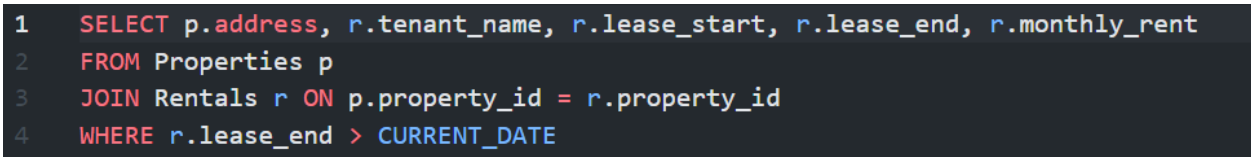
**Projected Property Values**

**5-Year Projected Value Details**

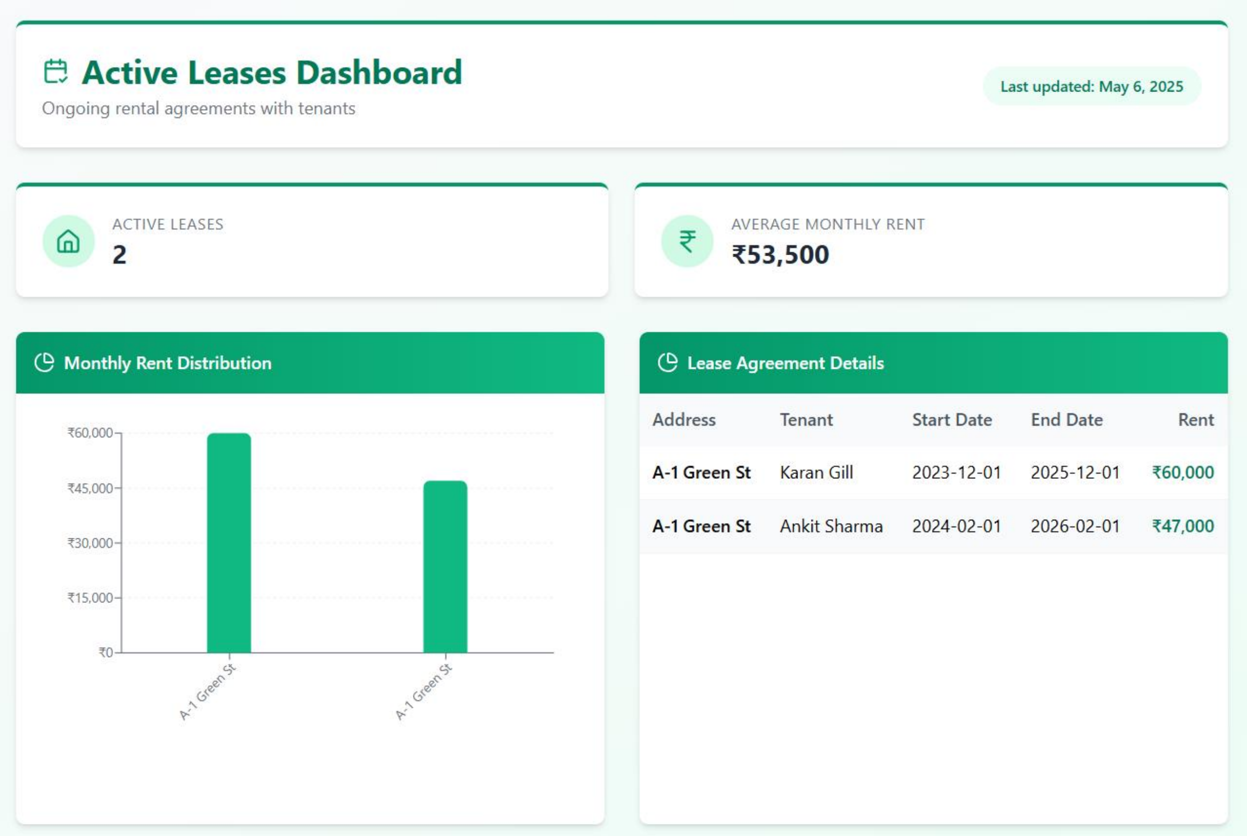
| Address | Current Value | Growth Rate | Projected Value |
|---|---|---|---|
| C-77 Red Ave | ₹16,090,000 | 8.50% | ₹24,193,836.14 |
| A-1 Green St | ₹15,000,000 | 8.50% | ₹22,554,850.35 |
| E-66 Lakeview | ₹15,500,000 | 7.00% | ₹21,739,551.83 |
| D-9 Yellow Blvd | ₹10,090,000 | 6.80% | ₹14,019,981.15 |
| B-22 Blue Ln | ₹9,500,000 | 7.00% | ₹13,324,241.44 |

## Rental Management

Active rentals with lease ending soon



```sql
SELECT p.address, r.tenant_name, r.lease_start, r.lease_end, r.monthly_rent
FROM Properties p
JOIN Rentals r ON p.property_id = r.property_id
WHERE r.lease_end > CURRENT_DATE
```



**Active Leases Dashboard**

Ongoing rental agreements with tenants

Last updated: May 6, 2025

ACTIVE LEASES

**2**

AVERAGE MONTHLY RENT

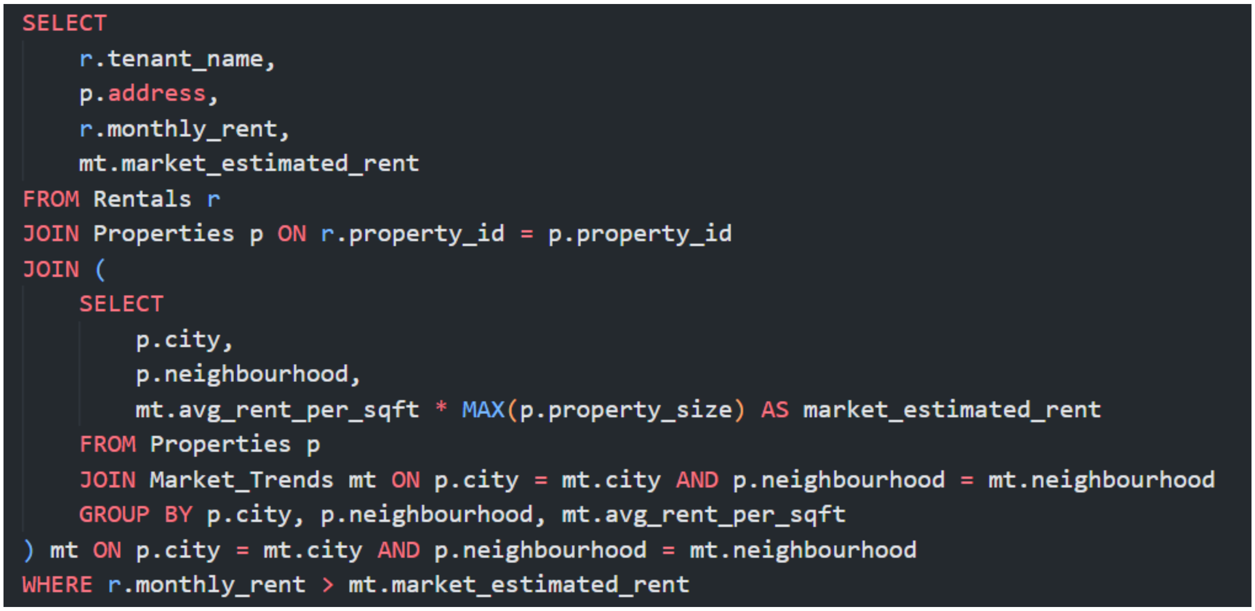
**₹53,500**

**Monthly Rent Distribution**

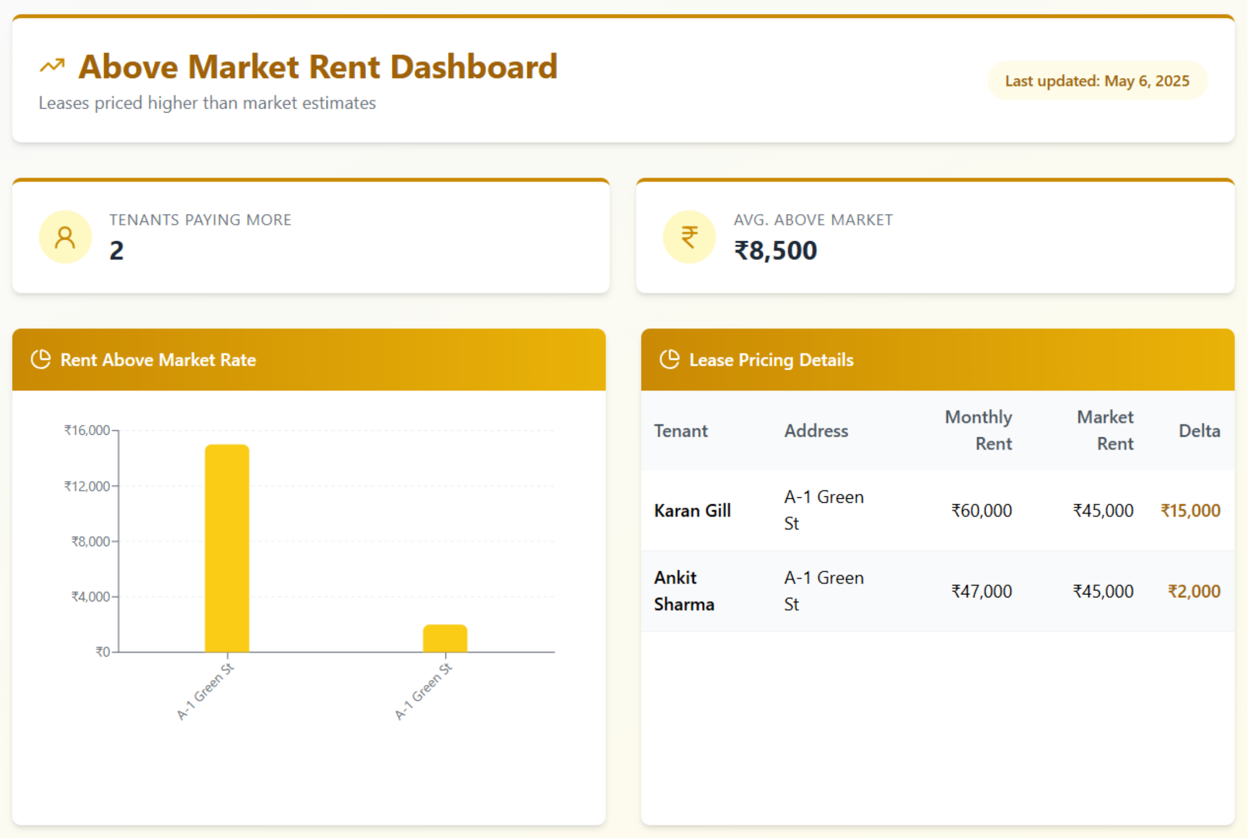
**Lease Agreement Details**

| Address | Tenant | Start Date | End Date | Rent |
|---|---|---|---|---|
| **A-1 Green St** | Karan Gill | 2023-12-01 | 2025-12-01 | **₹60,000** |
| **A-1 Green St** | Ankit Sharma | 2024-02-01 | 2026-02-01 | **₹47,000** |

Tenants paying above market rent



```
SELECT
    r.tenant_name,
    p.address,
    r.monthly_rent,
    mt.market_estimated_rent
FROM Rentals r
JOIN Properties p ON r.property_id = p.property_id
JOIN (
    SELECT
        p.city,
        p.neighbourhood,
        mt.avg_rent_per_sqft * MAX(p.property_size) AS market_estimated_rent
    FROM Properties p
    JOIN Market_Trends mt ON p.city = mt.city AND p.neighbourhood = mt.neighbourhood
    GROUP BY p.city, p.neighbourhood, mt.avg_rent_per_sqft
) mt ON p.city = mt.city AND p.neighbourhood = mt.neighbourhood
WHERE r.monthly_rent > mt.market_estimated_rent
```



## Above Market Rent Dashboard

Leases priced higher than market estimates

Last updated: May 6, 2025

TENANTS PAYING MORE
2

AVG. ABOVE MARKET
₹8,500

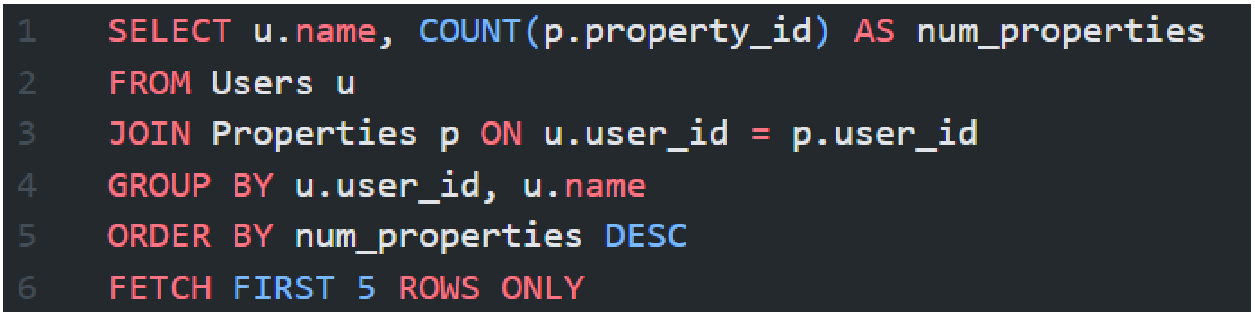
Rent Above Market Rate

Lease Pricing Details

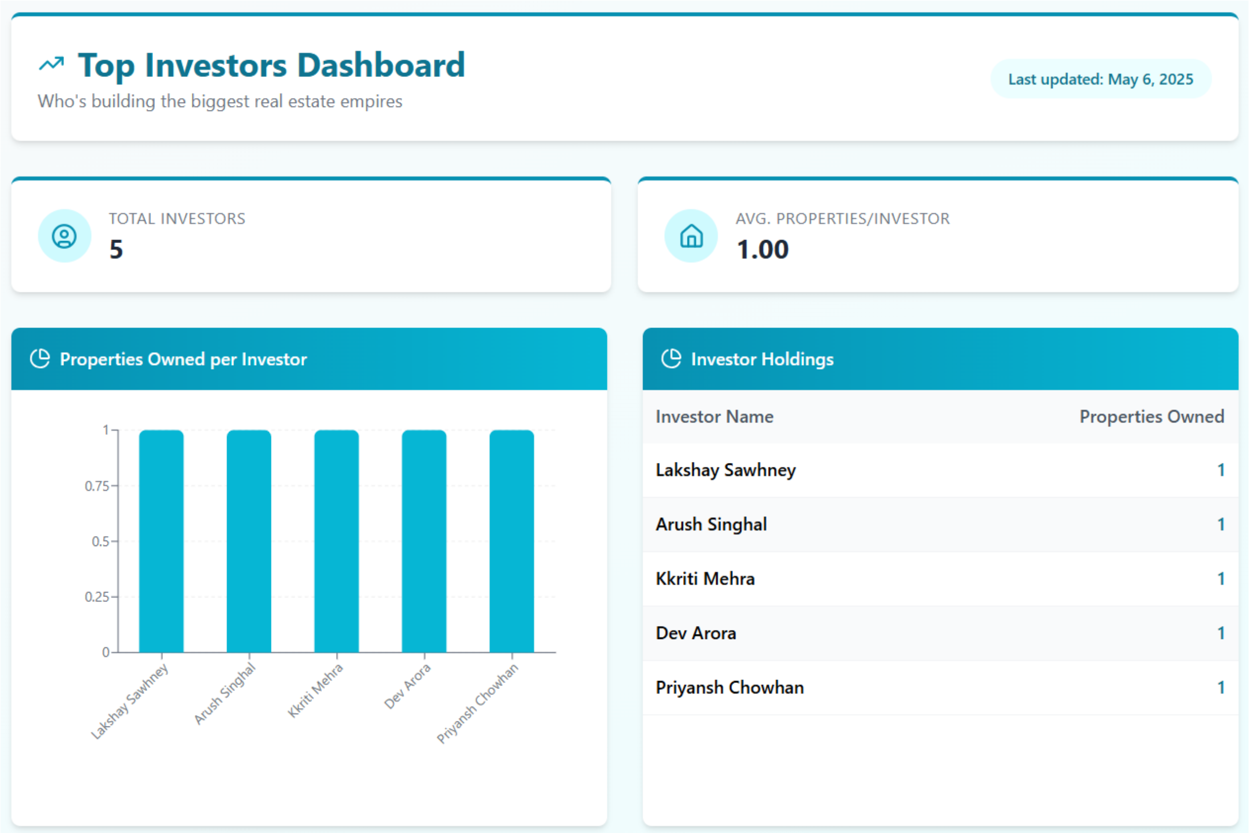
| Tenant | Address | Monthly Rent | Market Rent | Delta |
|---|---|---|---|---|
| Karan Gill | A-1 Green St | ₹60,000 | ₹45,000 | ₹15,000 |
| Ankit Sharma | A-1 Green St | ₹47,000 | ₹45,000 | ₹2,000 |

## Investor Insights

Top 5 investors owning most properties



```sql
SELECT u.name, COUNT(p.property_id) AS num_properties
FROM Users u
JOIN Properties p ON u.user_id = p.user_id
GROUP BY u.user_id, u.name
ORDER BY num_properties DESC
FETCH FIRST 5 ROWS ONLY
```



Top Investors Dashboard

Who's building the biggest real estate empires

Last updated: May 6, 2025

TOTAL INVESTORS
5

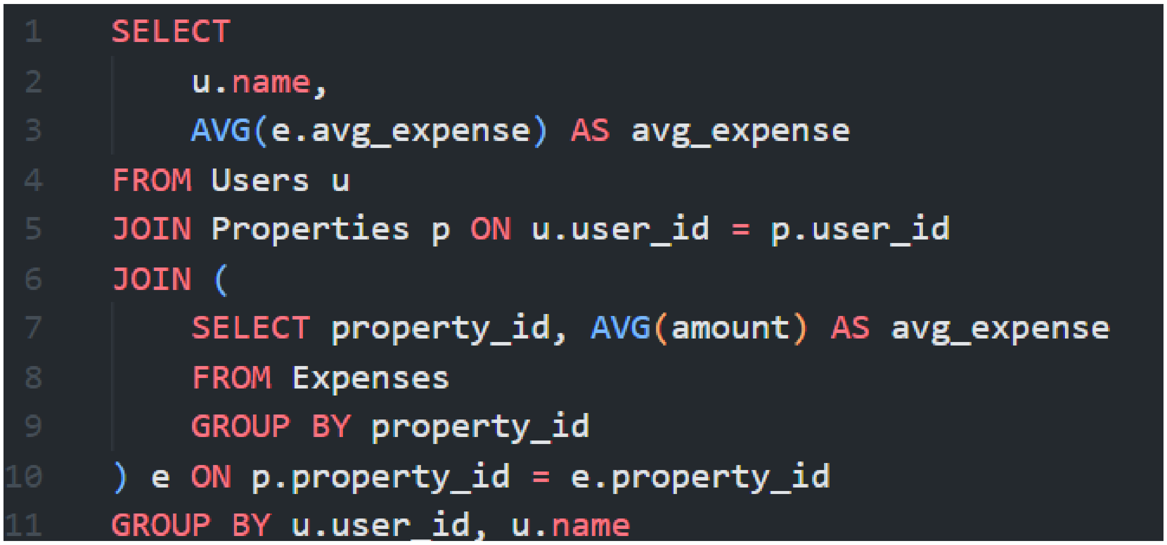
AVG. PROPERTIES/INVESTOR
1.00

Properties Owned per Investor

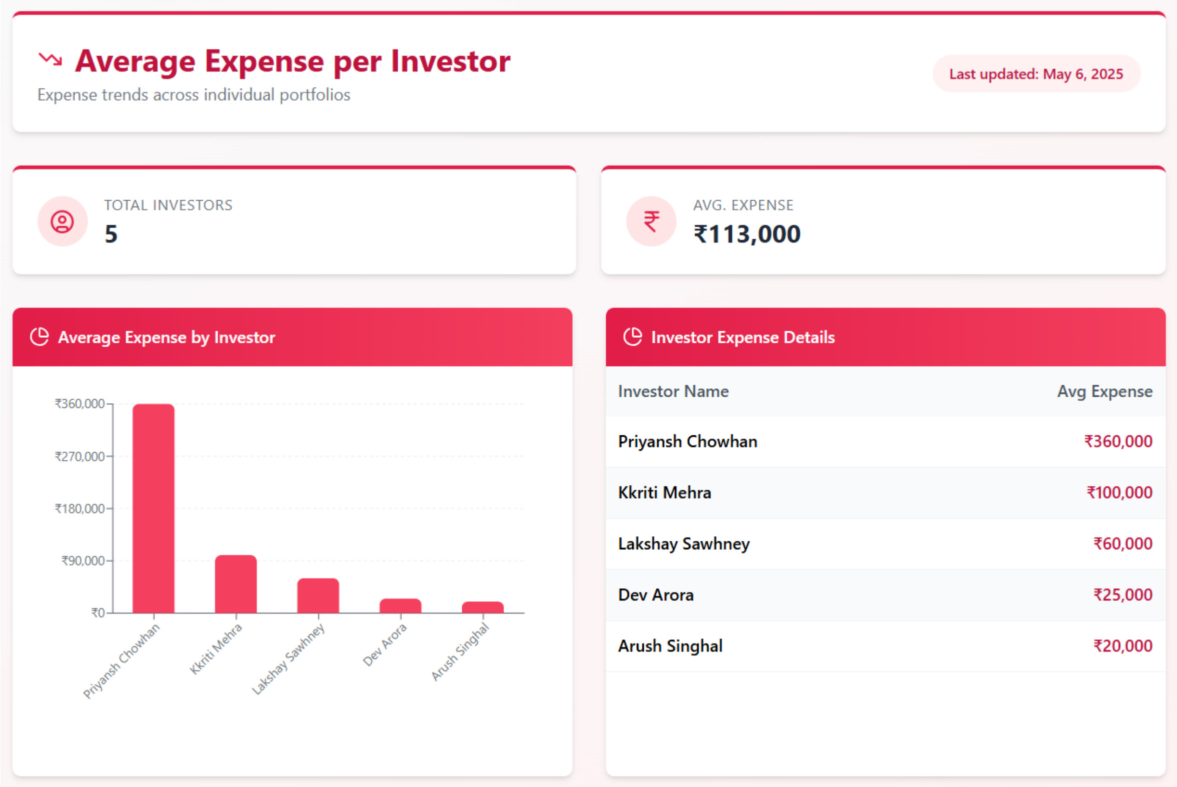
Investor Holdings

| Investor Name | Properties Owned |
|---|---|
| Lakshay Sawhney | 1 |
| Arush Singhal | 1 |
| Kkriti Mehra | 1 |
| Dev Arora | 1 |
| Priyansh Chowhan | 1 |

Average expense per owner



```sql
SELECT
    u.name,
    AVG(e.avg_expense) AS avg_expense
FROM Users u
JOIN Properties p ON u.user_id = p.user_id
JOIN (
    SELECT property_id, AVG(amount) AS avg_expense
    FROM Expenses
    GROUP BY property_id
) e ON p.property_id = e.property_id
GROUP BY u.user_id, u.name
```



**Average Expense per Investor**

Expense trends across individual portfolios

Last updated: May 6, 2025

TOTAL INVESTORS

5

AVG. EXPENSE

₹113,000

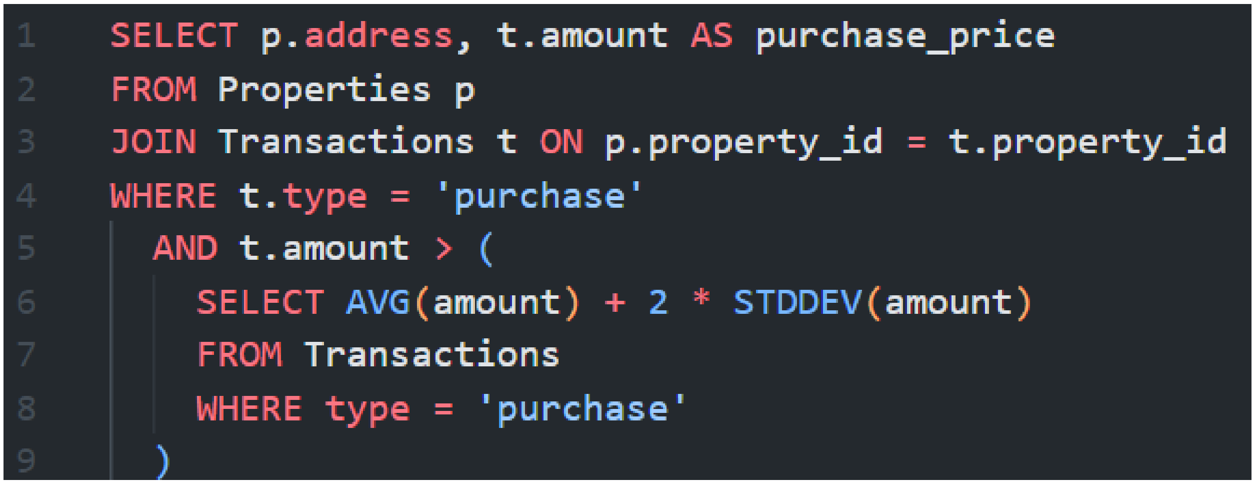
**Average Expense by Investor**

**Investor Expense Details**

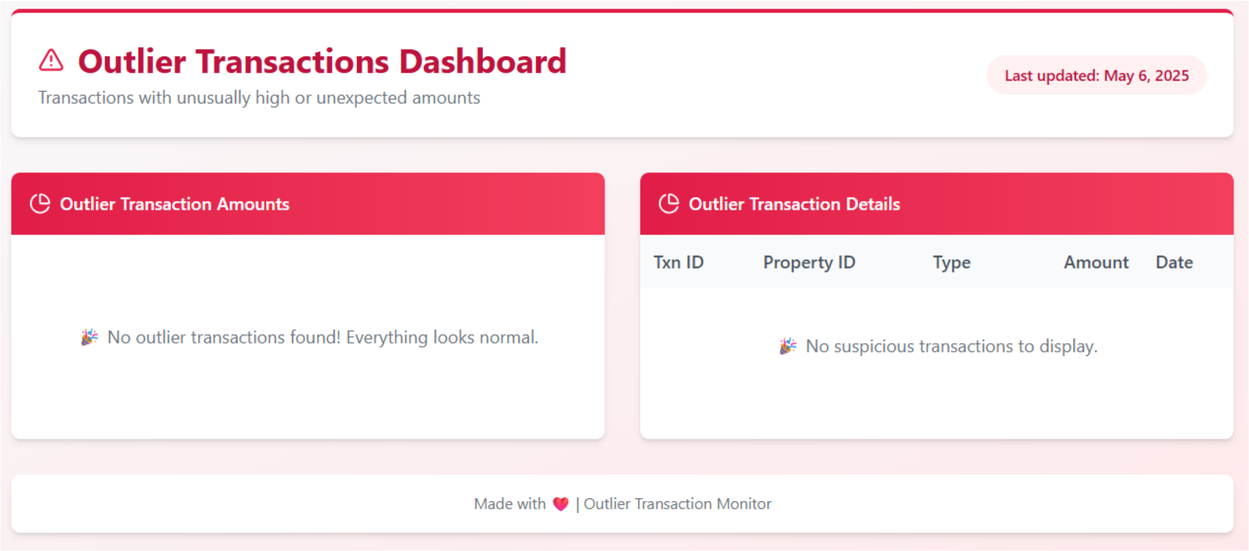
| Investor Name | Avg Expense |
|---|---|
| Priyansh Chowhan | ₹360,000 |
| Kkriti Mehra | ₹100,000 |
| Lakshay Sawhney | ₹60,000 |
| Dev Arora | ₹25,000 |
| Arush Singhal | ₹20,000 |

## Anomaly Detection

Properties with abnormal high purchase price



```sql
SELECT p.address, t.amount AS purchase_price
FROM Properties p
JOIN Transactions t ON p.property_id = t.property_id
WHERE t.type = 'purchase'
  AND t.amount > (
    SELECT AVG(amount) + 2 * STDDEV(amount)
    FROM Transactions
    WHERE type = 'purchase'
  )
```



⚠ Outlier Transactions Dashboard

Transactions with unusually high or unexpected amounts

Last updated: May 6, 2025

Outlier Transaction Amounts

🎉 No outlier transactions found! Everything looks normal.

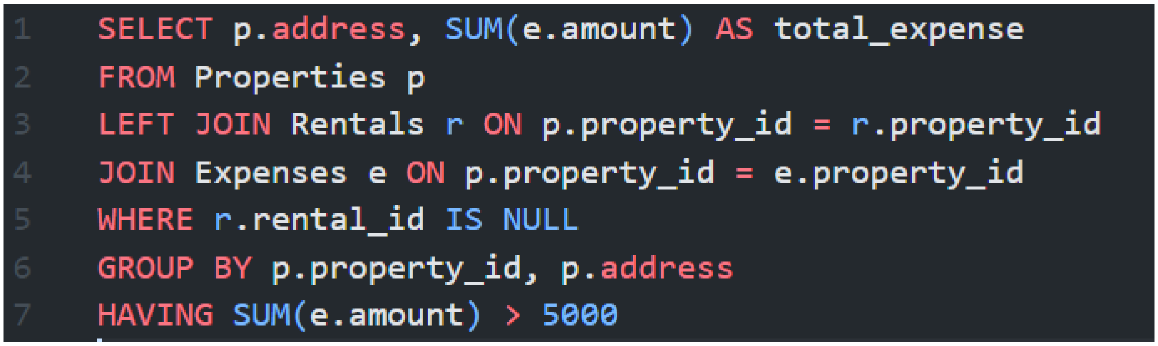
Outlier Transaction Details

| Txn ID | Property ID | Type | Amount | Date |
|---|---|---|---|---|

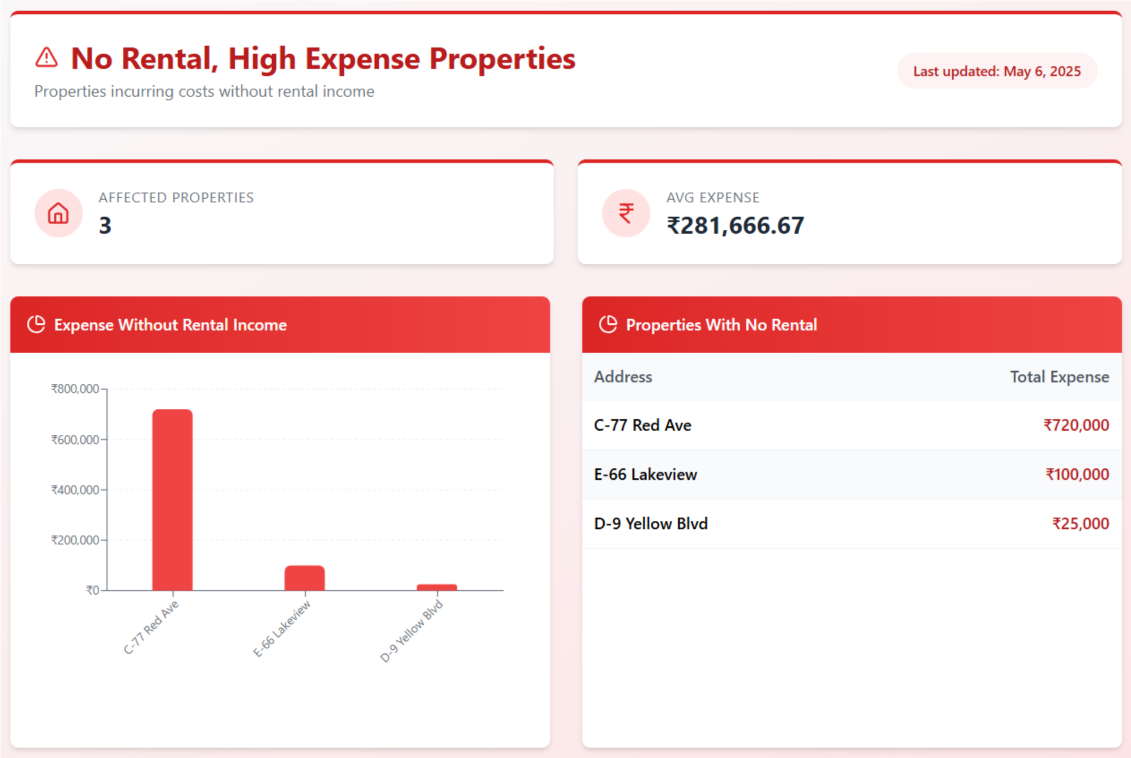
🎉 No suspicious transactions to display.

Made with ❤️ | Outlier Transaction Monitor

Properties without rental but high expenses



```sql
SELECT p.address, SUM(e.amount) AS total_expense
FROM Properties p
LEFT JOIN Rentals r ON p.property_id = r.property_id
JOIN Expenses e ON p.property_id = e.property_id
WHERE r.rental_id IS NULL
GROUP BY p.property_id, p.address
HAVING SUM(e.amount) > 5000
```



⚠ No Rental, High Expense Properties

Properties incurring costs without rental income

Last updated: May 6, 2025

AFFECTED PROPERTIES
3

AVG EXPENSE
₹281,666.67

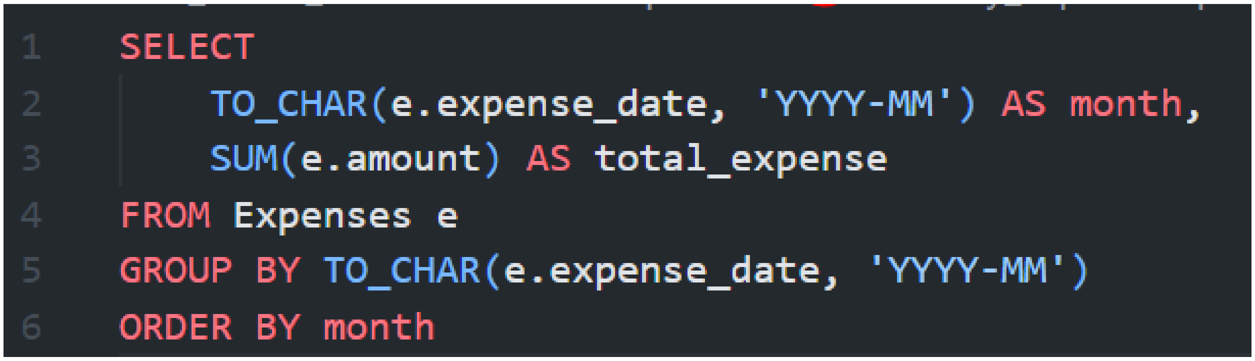
Expense Without Rental Income

Properties With No Rental

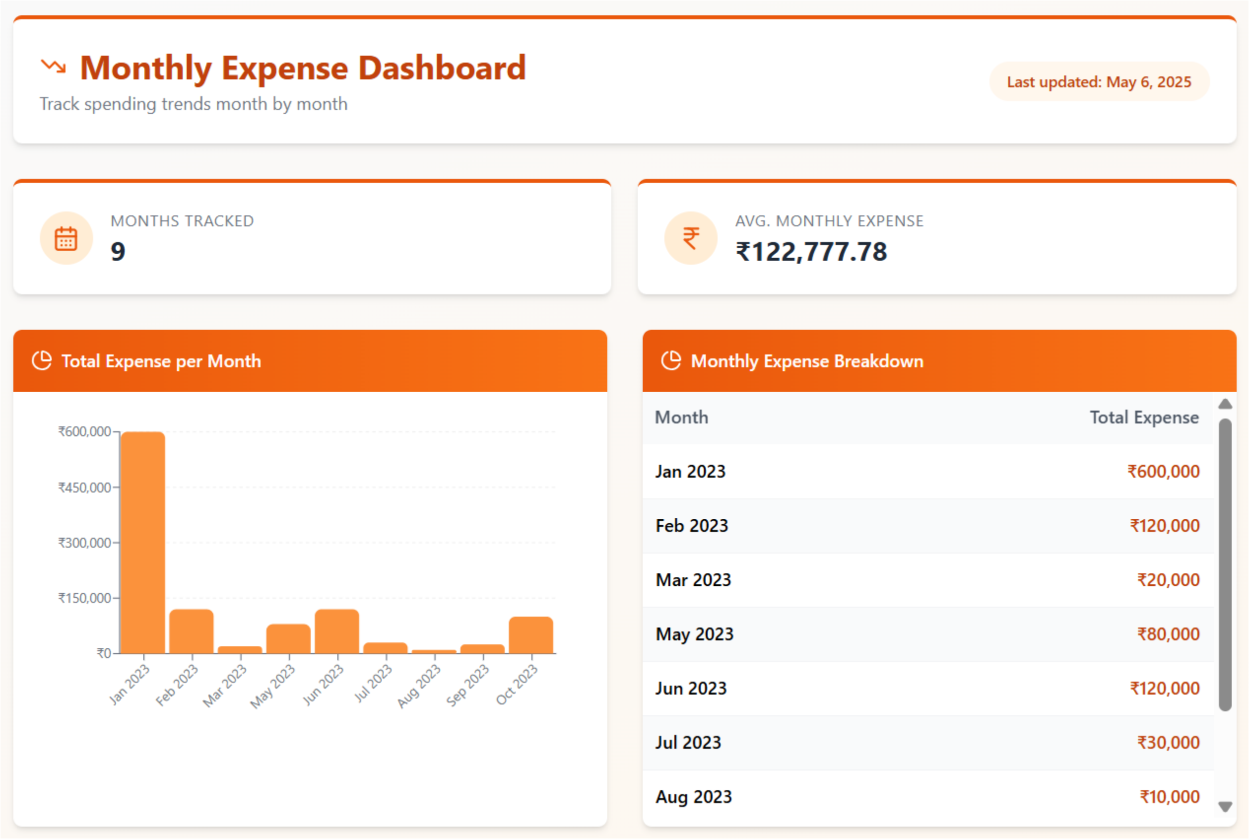
| Address | Total Expense |
|---|---|
| C-77 Red Ave | ₹720,000 |
| E-66 Lakeview | ₹100,000 |
| D-9 Yellow Blvd | ₹25,000 |

## Market Trends

Total monthly expense trend



```sql
SELECT
    TO_CHAR(e.expense_date, 'YYYY-MM') AS month,
    SUM(e.amount) AS total_expense
FROM Expenses e
GROUP BY TO_CHAR(e.expense_date, 'YYYY-MM')
ORDER BY month
```



**Monthly Expense Dashboard**

Track spending trends month by month

Last updated: May 6, 2025

MONTHS TRACKED
**9**

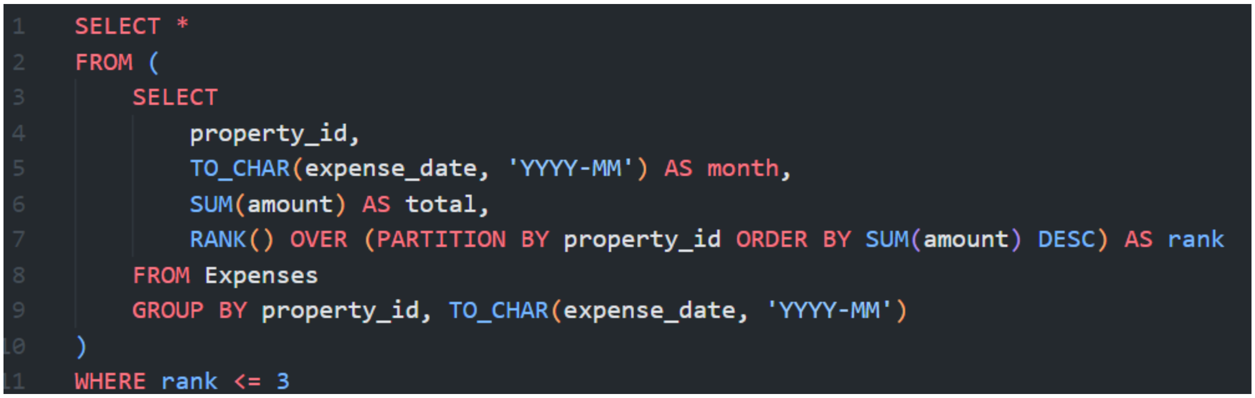
AVG. MONTHLY EXPENSE
**₹122,777.78**

**Total Expense per Month**

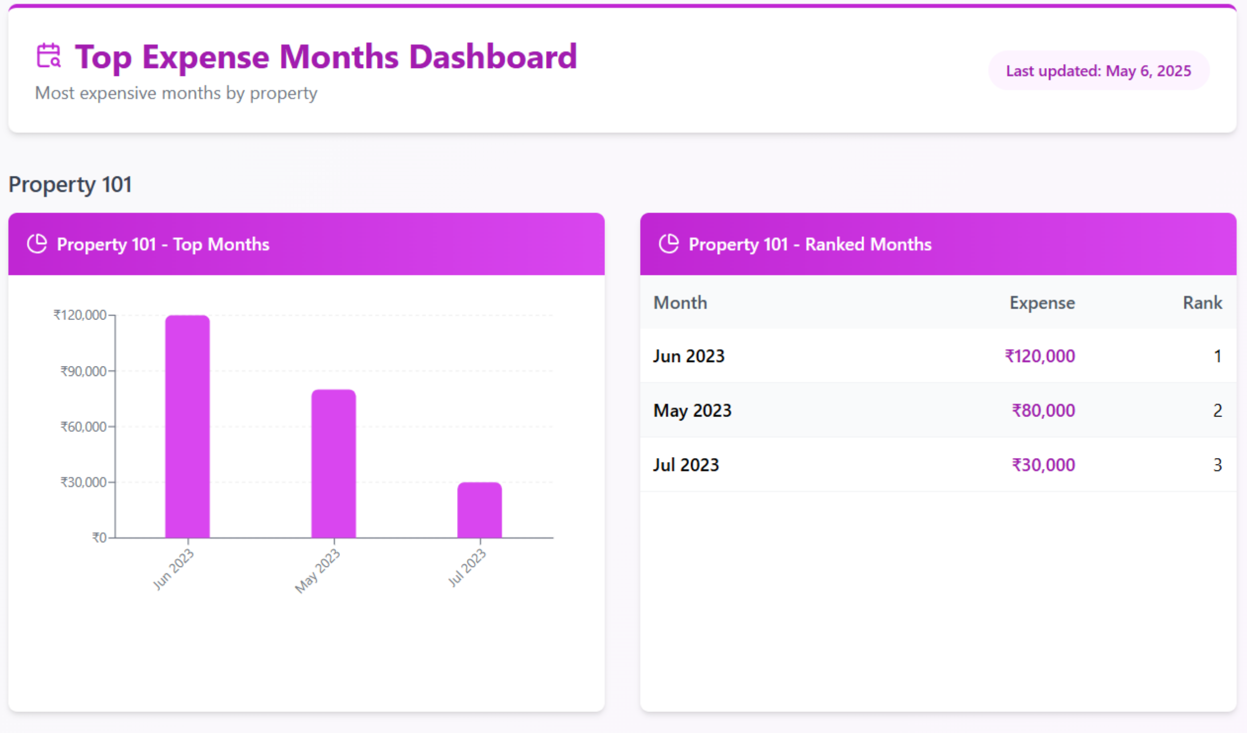
**Monthly Expense Breakdown**

| Month | Total Expense |
|---|---|
| Jan 2023 | ₹600,000 |
| Feb 2023 | ₹120,000 |
| Mar 2023 | ₹20,000 |
| May 2023 | ₹80,000 |
| Jun 2023 | ₹120,000 |
| Jul 2023 | ₹30,000 |
| Aug 2023 | ₹10,000 |

Top 3 most expensive months per property



```sql
SELECT *
FROM (
    SELECT
        property_id,
        TO_CHAR(expense_date, 'YYYY-MM') AS month,
        SUM(amount) AS total,
        RANK() OVER (PARTITION BY property_id ORDER BY SUM(amount) DESC) AS rank
    FROM Expenses
    GROUP BY property_id, TO_CHAR(expense_date, 'YYYY-MM')
)
WHERE rank <= 3
```



## Top Expense Months Dashboard

Most expensive months by property

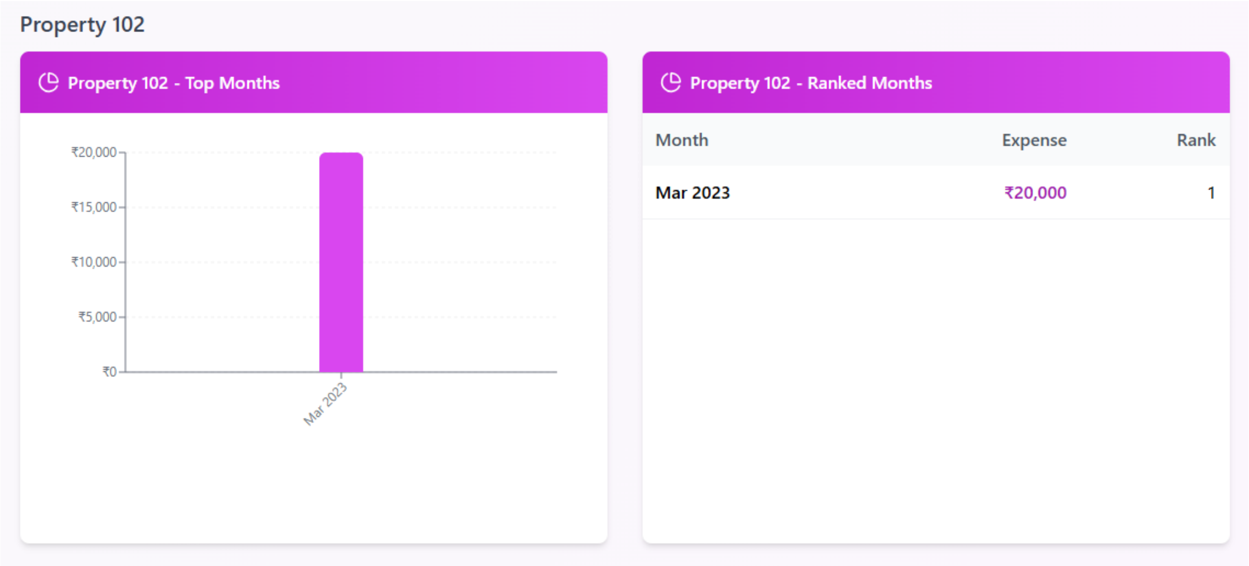
Last updated: May 6, 2025

Property 101

Property 101 - Top Months

Property 101 - Ranked Months

| Month | Expense | Rank |
|---|---|---|
| Jun 2023 | ₹120,000 | 1 |
| May 2023 | ₹80,000 | 2 |
| Jul 2023 | ₹30,000 | 3 |

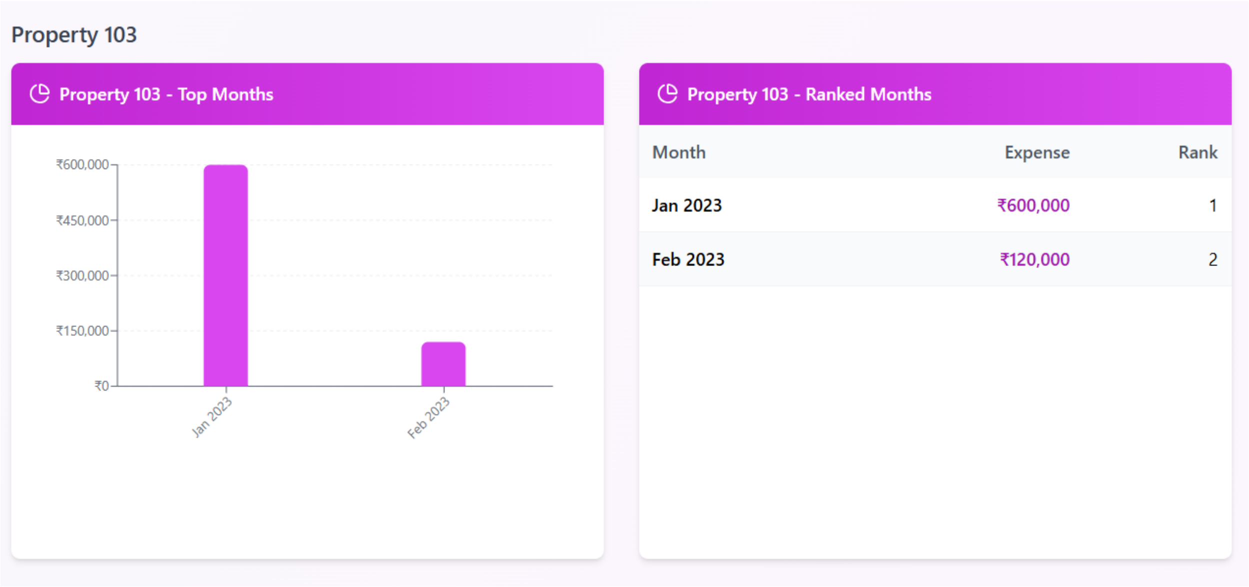


Property 102

Property 102 - Top Months

Property 102 - Ranked Months

| Month | Expense | Rank |
|---|---|---|
| Mar 2023 | ₹20,000 | 1 |

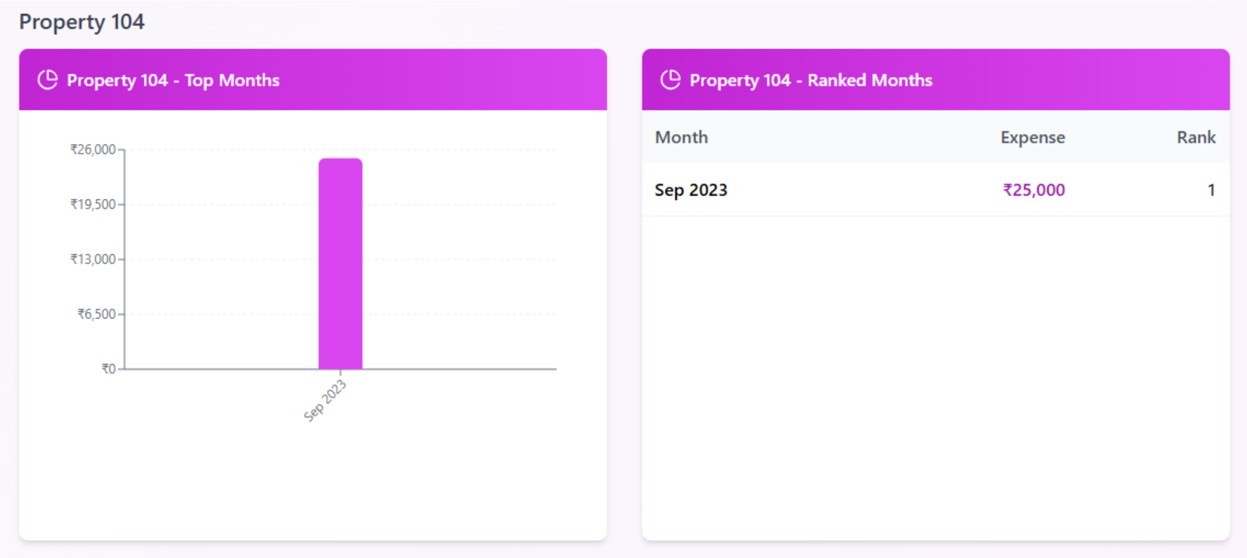


Property 103

Property 103 - Top Months

Property 103 - Ranked Months

| Month | Expense | Rank |
|---|---|---|
| Jan 2023 | ₹600,000 | 1 |
| Feb 2023 | ₹120,000 | 2 |

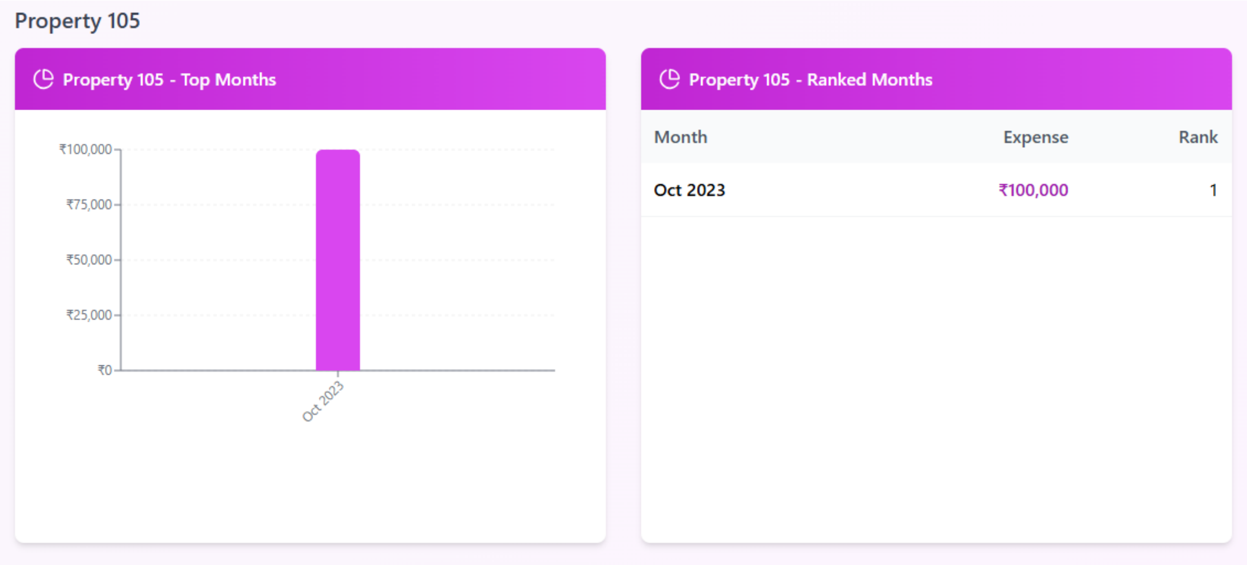


Property 104

Property 104 - Top Months

Property 104 - Ranked Months

| Month | Expense | Rank |
|---|---|---|
| Sep 2023 | ₹25,000 | 1 |



Property 105

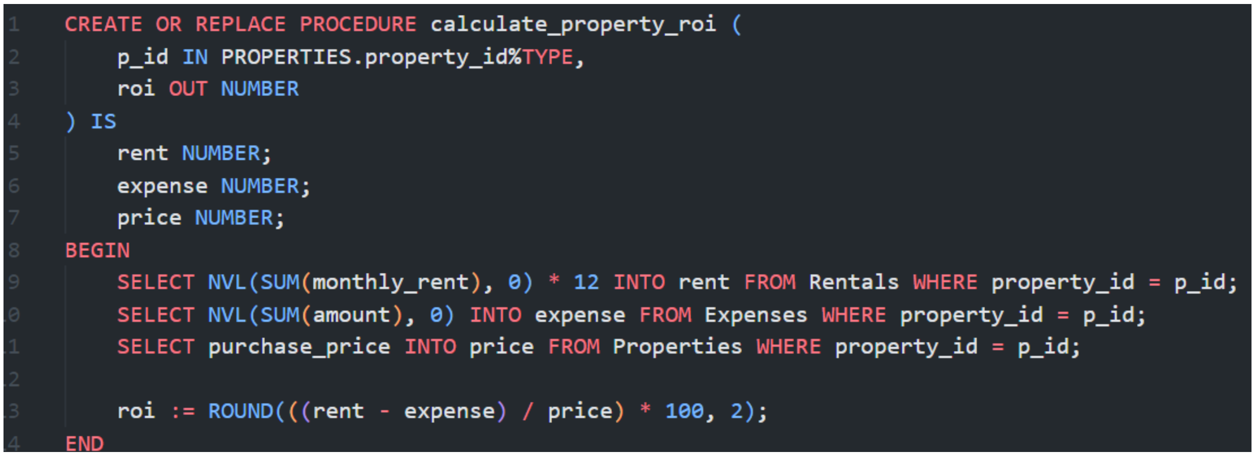
Property 105 - Top Months

Property 105 - Ranked Months

| Month | Expense | Rank |
|---|---|---|
| Oct 2023 | ₹100,000 | 1 |

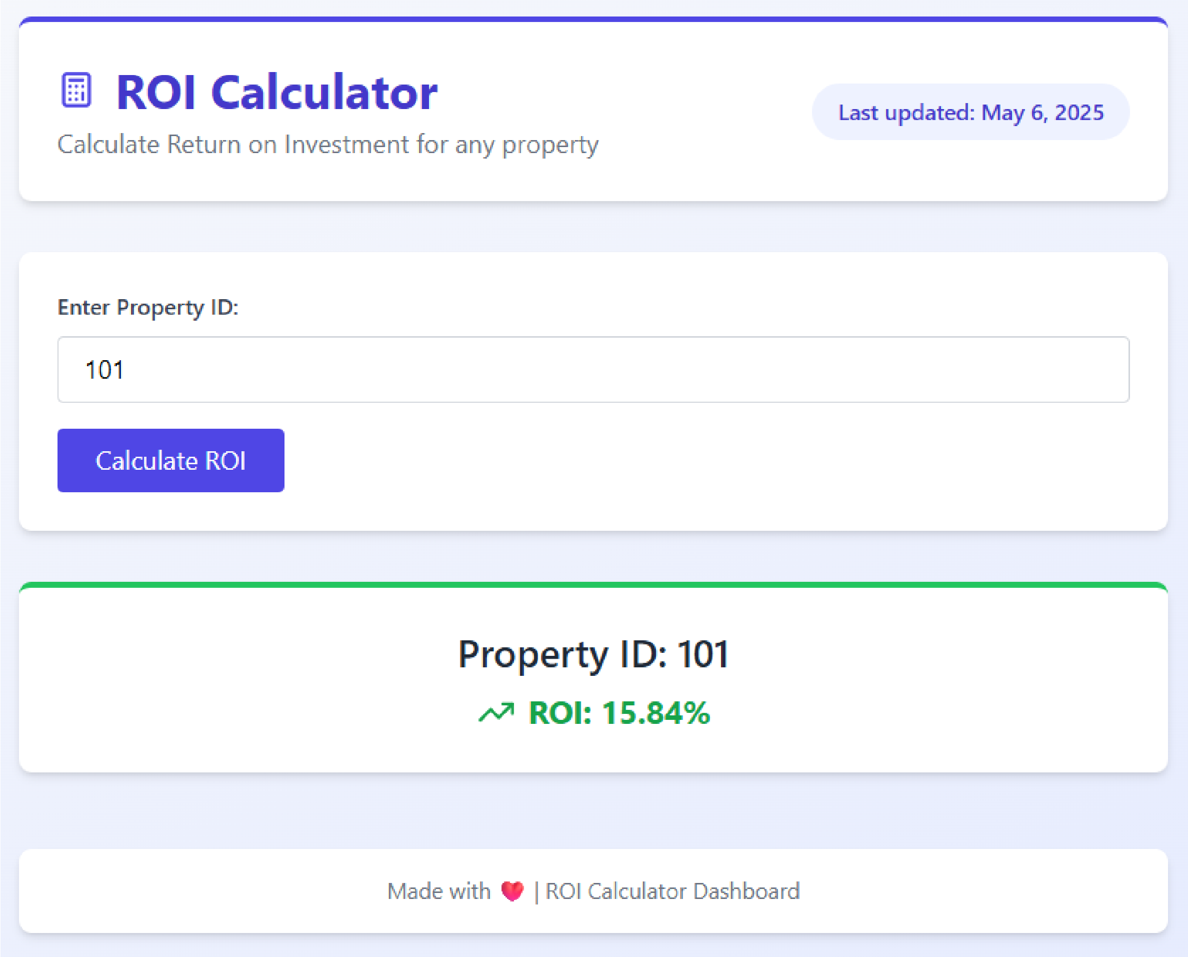
## Calculators

Calculate ROI for given Property



```sql
CREATE OR REPLACE PROCEDURE calculate_property_roi (
    p_id IN PROPERTIES.property_id%TYPE,
    roi OUT NUMBER
) IS
    rent NUMBER;
    expense NUMBER;
    price NUMBER;
BEGIN
    SELECT NVL(SUM(monthly_rent), 0) * 12 INTO rent FROM Rentals WHERE property_id = p_id;
    SELECT NVL(SUM(amount), 0) INTO expense FROM Expenses WHERE property_id = p_id;
    SELECT purchase_price INTO price FROM Properties WHERE property_id = p_id;

    roi := ROUND(((rent - expense) / price) * 100, 2);
END
```



### ROI Calculator

Calculate Return on Investment for any property

Last updated: May 6, 2025

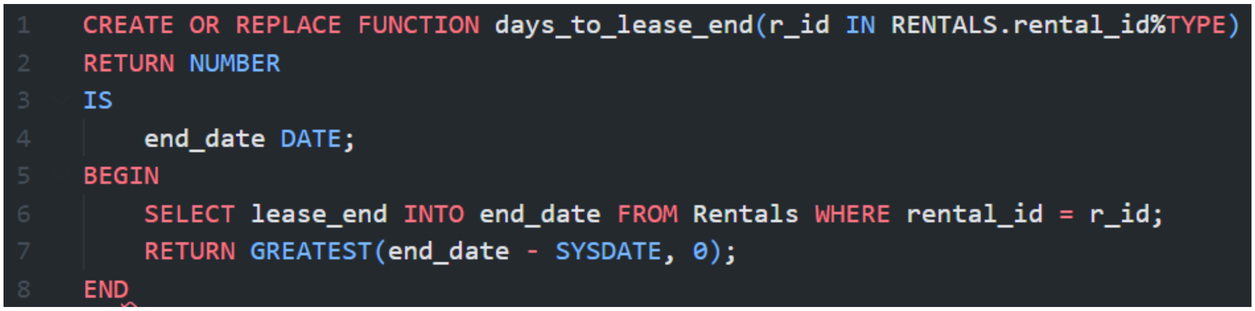
Enter Property ID: 101

Property ID: 101

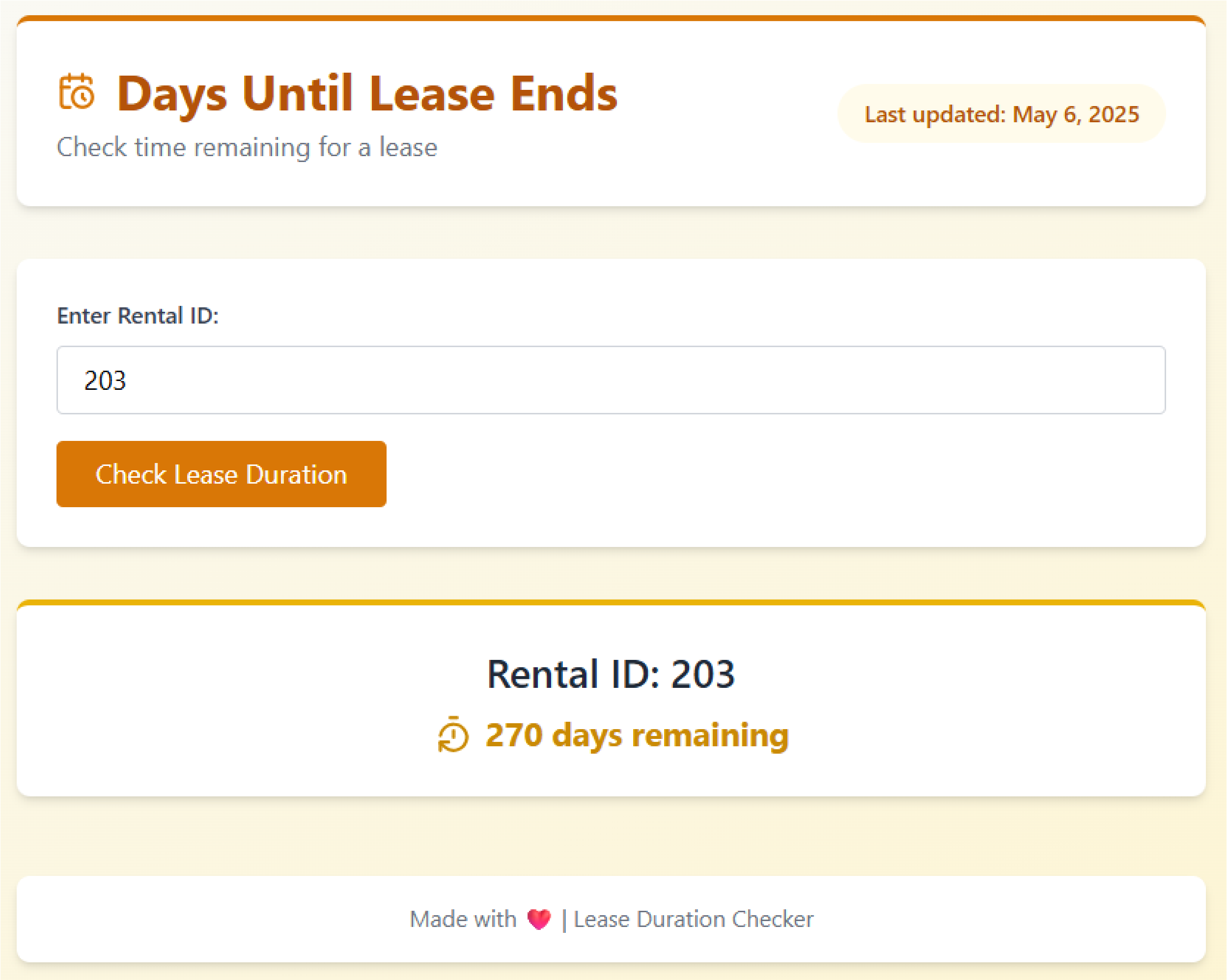
ROI: 15.84%

Made with ❤️ | ROI Calculator Dashboard

Calculate days to lease end



```
CREATE OR REPLACE FUNCTION days_to_lease_end(r_id IN RENTALS.rental_id%TYPE)
RETURN NUMBER
IS
    end_date DATE;
BEGIN
    SELECT lease_end INTO end_date FROM Rentals WHERE rental_id = r_id;
    RETURN GREATEST(end_date - SYSDATE, 0);
END
```



Days Until Lease Ends

Check time remaining for a lease

Last updated: May 6, 2025

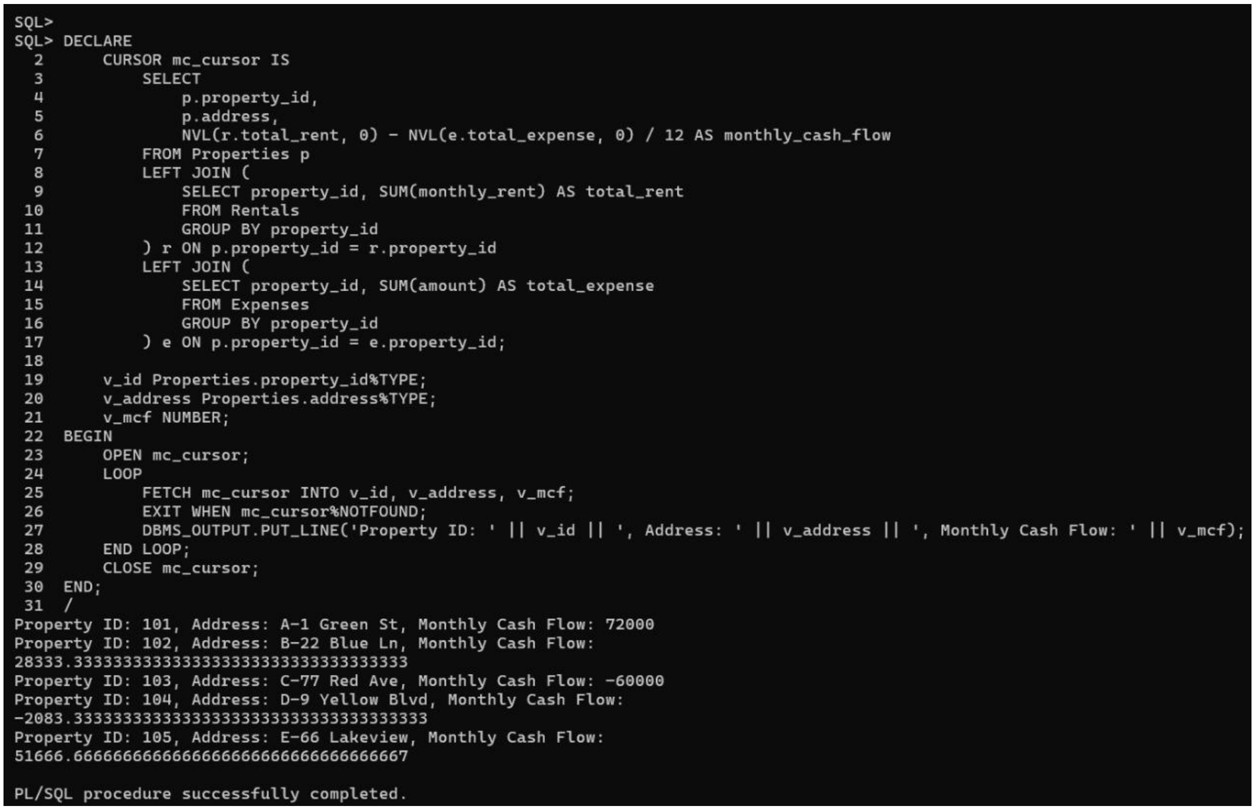
Enter Rental ID: 203

Rental ID: 203

270 days remaining

Made with ❤ | Lease Duration Checker

Monthly cash flow using a cursor

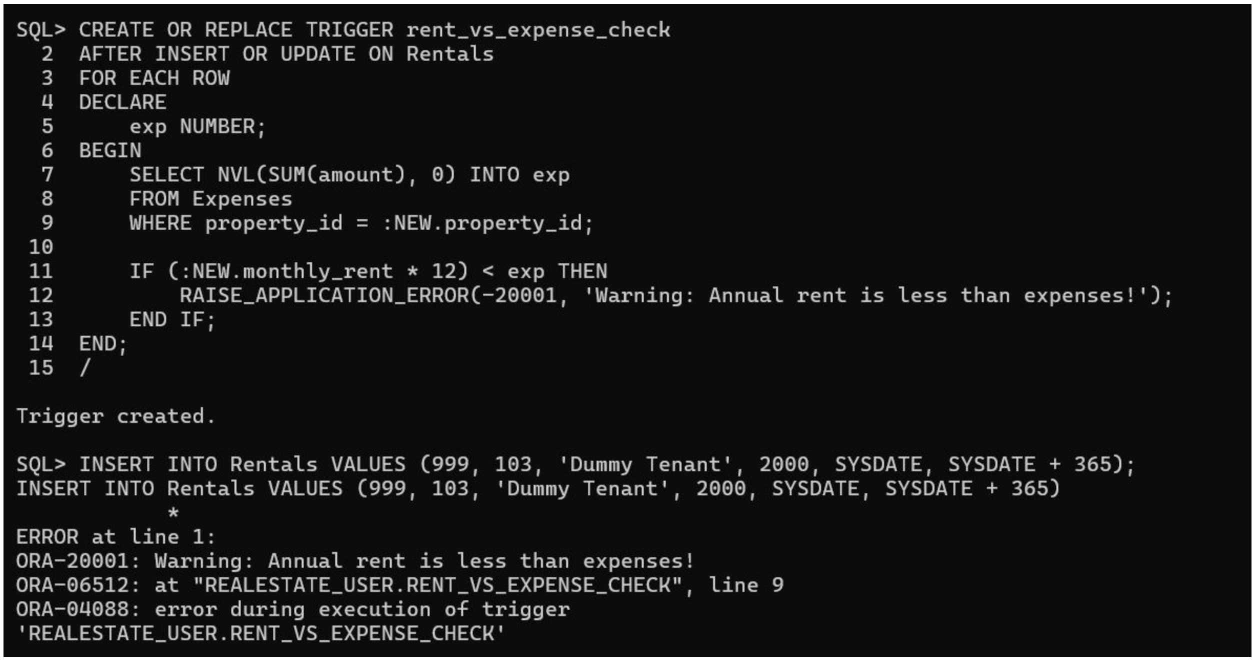


```
SQL>
SQL> DECLARE
         CURSOR mc_cursor IS
             SELECT
                 p.property_id,
                 p.address,
                 NVL(r.total_rent, 0) - NVL(e.total_expense, 0) / 12 AS monthly_cash_flow
             FROM Properties p
             LEFT JOIN (
                 SELECT property_id, SUM(monthly_rent) AS total_rent
                 FROM Rentals
                 GROUP BY property_id
             ) r ON p.property_id = r.property_id
             LEFT JOIN (
                 SELECT property_id, SUM(amount) AS total_expense
                 FROM Expenses
                 GROUP BY property_id
             ) e ON p.property_id = e.property_id;

         v_id Properties.property_id%TYPE;
         v_address Properties.address%TYPE;
         v_mcf NUMBER;
     BEGIN
         OPEN mc_cursor;
         LOOP
             FETCH mc_cursor INTO v_id, v_address, v_mcf;
             EXIT WHEN mc_cursor%NOTFOUND;
             DBMS_OUTPUT.PUT_LINE('Property ID: ' || v_id || ', Address: ' || v_address || ', Monthly Cash Flow: ' || v_mcf);
         END LOOP;
         CLOSE mc_cursor;
     END;
     /
Property ID: 101, Address: A-1 Green St, Monthly Cash Flow: 72000
Property ID: 102, Address: B-22 Blue Ln, Monthly Cash Flow:
28333.3333333333333333333333333333333333
Property ID: 103, Address: C-77 Red Ave, Monthly Cash Flow: -60000
Property ID: 104, Address: D-9 Yellow Blvd, Monthly Cash Flow:
-2083.333333333333333333333333333333333333
Property ID: 105, Address: E-66 Lakeview, Monthly Cash Flow:
51666.6666666666666666666666666666666667

PL/SQL procedure successfully completed.
```

Trigger blocks rentals where annual rent < total expenses for the property.



```
SQL> CREATE OR REPLACE TRIGGER rent_vs_expense_check
  2  AFTER INSERT OR UPDATE ON Rentals
  3  FOR EACH ROW
  4  DECLARE
  5      exp NUMBER;
  6  BEGIN
  7      SELECT NVL(SUM(amount), 0) INTO exp
  8      FROM Expenses
  9      WHERE property_id = :NEW.property_id;
 10
 11      IF (:NEW.monthly_rent * 12) < exp THEN
 12          RAISE_APPLICATION_ERROR(-20001, 'Warning: Annual rent is less than expenses!');
 13      END IF;
 14  END;
 15  /

Trigger created.

SQL> INSERT INTO Rentals VALUES (999, 103, 'Dummy Tenant', 2000, SYSDATE, SYSDATE + 365);
INSERT INTO Rentals VALUES (999, 103, 'Dummy Tenant', 2000, SYSDATE, SYSDATE + 365)
            *
ERROR at line 1:
ORA-20001: Warning: Annual rent is less than expenses!
ORA-06512: at "REALESTATE_USER.RENT_VS_EXPENSE_CHECK", line 9
ORA-04088: error during execution of trigger
'REALESTATE_USER.RENT_VS_EXPENSE_CHECK'
```

# Conclusion

The Real Estate Investment & Rental Profitability Tracker is a robust SQL-based system designed to streamline property management, rental analysis, and financial tracking through a structured relational database. This system efficiently integrates database indexing, JOIN queries, and Lucid chart ER modelling to provide investors with accurate financial insights and comparative property analysis.

By implementing a relational database structure, the system eliminates data redundancy and ensures referential integrity using primary and foreign keys. The Entity-Relationship (ER) Diagram plays a critical role in visualizing how properties, rentals, transactions, expenses, and market trends interact, making it easier for investors to assess profitability and risk factors. The use of SQL JOIN queries optimizes data retrieval, enabling quick access to rental performance, expense tracking, and market trends.

In addition to backend and database efficiency, the system features a user-friendly frontend interface developed using React, which enhances user experience through interactive dashboards and real-time graphs. These graphical visualizations present key metrics such as rental yields, ROI trends, and expense breakdowns in an intuitive format, enabling investors to make quicker, more informed decisions.

The system also emphasizes scalability and efficiency, making it suitable for investors managing a single property or an extensive portfolio, along with indexed queries that enhance database performance.

Furthermore, the ER model provides a clear, structured visualization of data relationships, allowing users to interpret rental trends, property appreciation, and financial transactions efficiently.

Overall, the Real Estate Investment & Rental Profitability Tracker is a data-driven, performance-optimized solution that empowers investors with real-time financial insights, market trend analysis, and automated rental tracking. This system is a scalable and adaptable tool for making informed investment decisions and maximizing profitability in real estate portfolios.