

REAL ESTATE MANAGEMENT SYSTEM – PL/SQL PROJECT REQUIREMENT DOCUMENT

PROJECT SUMMARY

A Real Estate company manages properties, agents, customers, bookings, payments, maintenance, and ownership transfers.

This project manages all these using:

- 8 relational tables
 - Views (read-only simplified data)
 - Procedures (business logic for front-end)
 - Triggers (automatic rules/validations)
 - Indexes (performance improvement)
-

1. TABLE STRUCTURE (8 TABLES)

1) PROPERTY_MASTER

Fields:

- property_id (PK)
- property_name
- type (Flat/Villa/Plot)
- location
- price
- status (AVAILABLE / BOOKED / SOLD)

2) CUSTOMER_MASTER

- customer_id (PK)
- customer_name
- contact_no
- email
- address

3) AGENT_MASTER

- agent_id (PK)
- agent_name
- phone
- commission_percent

4) BOOKING_MASTER

- booking_id (PK)
- property_id (FK)
- customer_id (FK)
- agent_id (FK)
- booking_date
- booking_amount
- status (BOOKED / CANCELLED)

5) PAYMENT_MASTER

- payment_id (PK)
- booking_id (FK)
- payment_date
- payment_amount
- mode (UPI/CARD/CASH)

6) MAINTENANCE_MASTER

- maintenance_id (PK)
- property_id (FK)
- issue_description
- status (OPEN/CLOSED)
- updated_on

7) OWNERSHIP_MASTER

- owner_id (PK)
- property_id (FK)
- customer_id (FK)
- ownership_date

8) AUDIT_LOG

- log_id (PK)
- table_name
- operation_type
- description
- log_date

2. INDEXES – WHY THEY ARE NEEDED

1) IDX_PROPERTY_LOCATION

Purpose: Fast property search by location.

2) IDX_PROPERTY_STATUS

Purpose: Retrieve available properties quickly.

3) IDX_CUSTOMER_CONTACT

Purpose: Speed duplicate contact checking.

4) IDX_BOOKING_PID

Purpose: Fast lookup of bookings per property.

5) IDX_PAYMENT_BID

Purpose: Fast payment lookup.

6) IDX_MAINTENANCE_PID

Purpose: Get maintenance issues property-wise.

7) IDX_OWNERSHIP_PID

Purpose: Quickly find ownership details.

8) IDX_AGENT_PHONE

Purpose: Prevent duplicate phone numbers.

3. VIEWS – WHY THEY ARE NEEDED

1) VW_AVAILABLE_PROPERTIES

Shows list of all available properties.

2) VW_BOOKING_DETAILS

Shows booking + customer + agent info.

3) VW_PAYMENT_HISTORY

Shows payment list for customers.

4) VW_AGENT_PERFORMANCE

Shows agent's total sales/bookings.

5) VW_PROPERTY_MAINTENANCE

Shows all open maintenance issues.

6) VW_OWNERSHIP_DETAILS

Shows owner details for each property.

7) VW_CUSTOMER_PROFILE

Shows customer + booking summary.

8) VW_PROPERTY_FULL_INFO

Shows full details of a property.

4. PROCEDURES – WHY THEY ARE NEEDED

1) ADD_PROPERTY

To add new property.

2) UPDATE_PROPERTY_PRICE

To update pricing.

3) REGISTER_CUSTOMER

Customer registration logic.

4) BOOK_PROPERTY

Creates booking + updates status.

5) CANCEL_BOOKING

Cancels booking + reopens property.

6) RECORD_PAYMENT

Payment insert + validation.

7) ADD_MAINTENANCE_REQUEST

Customer raises issue.

8) CLOSE_MAINTENANCE

Admin closes issue.

9) TRANSFER_OWNERSHIP

After full payment, transfer owner.

10) GET_PROPERTY_DETAILS

Frontend fetches complete info.

5. TRIGGERS – WHY THEY ARE NEEDED

1) TRG_BOOKING_VALIDATE

Prevents booking of SOLD/BOOKED property.

2) TRG_PAYMENT_VALIDATE

Prevents over-payment.

3) TRG_PROPERTY_PRICE_HISTORY

Maintains old price logs.

4) TRG_CUSTOMER_CONTACT_DUPLICATE

Blocks duplicate contact numbers.

5) TRG_AUDIT_BOOKING

Logs booking changes.

6) TRG_AUDIT_PAYMENT

Logs payment changes.

7) TRG_MAINTENANCE_AUTO_DATE

Auto-updates timestamp.

8) TRG_BLOCK_UPDATE_AFTER_SOLD

Prevents updating SOLD properties.

9) TRG_BOOKING_DATE_LOCK

Blocks booking on holidays.

10) TRG_PROPERTY_STATUS_CONTROL

Auto-marks property SOLD after ownership.

END OF REQUIREMENT DOCUMENT
