

## 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

-----