

Date: 6 Feb 2026

Prepared By: Mohamed Yasin

Task: Data-Driven Real Estate Market Insights – Majan

1. Introduction

This analysis focuses on understanding the real estate market trends in **Majan (Dubai)** over the last **24 months** using transaction-level data.

The goal is to identify:

- Market demand
- Property price trends
- Popular property types
- Impact of features (e.g., parking)
- Investor buying behavior

Tools Used

Tool	Purpose
Python	Initial loading, cleanup, exporting cleaned dataset
Excel	Minor inspection, tabular validations
SQL (T-SQL)	Full analytical processing & KPIs
Power BI	Dashboarding & interactive insights

2. Tech Stack Contribution

Python

Purpose:

- Load source CSV
- Clean missing values
- Convert date fields
- Export cleaned dataset as Excel

Operations done:

- Date parsing
- Removing negative/zero price records

- Removing duplicate transactions
- Standardizing text columns

SQL (Microsoft SQL Server)

This is the **core analysis layer**.

Step 1 — Understanding Dataset Columns

- **instance_date** → Date of transaction
- **area_name_en** → Majan (filtered)
- **property_type_en, property_sub_type_en, rooms_en** → Unit details
- **procedure_area** → Size (sqm)
- **actual_worth** → Total property price (main metric)
- **meter_sale_price** → Price per sqm
- **has_parking**
- **transaction_id, procedure_name_en** → Demand / sales indicator

Step 2 — Filter Only SALES Transactions

```
CREATE VIEW sales_data AS
SELECT * FROM majan_transactions
WHERE procedure_name_en = 'Sell'
AND actual_worth > 0
```

Step 3 — Total Market KPIs

a. Total Transactions

```
SELECT COUNT(*) AS total_transactions
FROM sales_data
```

Output

A screenshot of a Microsoft SQL Server Management Studio window. The title bar says 'Query Editor'. Below it, there are two tabs: 'Results' (which is selected) and 'Messages'. The results grid contains one row of data:

	total_transactions
1	873

b. Total Sales Value

```
SELECT SUM(CAST(actual_worth AS DECIMAL(18,2))) AS total_sales_value FROM sales_data;
```

Output

The screenshot shows a SQL query results window. At the top, there are two tabs: 'Results' (selected) and 'Messages'. Below the tabs is a table with one row. The table has two columns: 'total_sales_value'. The value in the first column is '1' and the value in the second column is '3578954463.00'.

	total_sales_value
1	3578954463.00

c. Average property price

```
WITH OrderedPrices AS (
    SELECT actual_worth,
           ROW_NUMBER() OVER (ORDER BY actual_worth) AS rn,
           COUNT(*) OVER () AS total_rows FROM sales_data)
    SELECT AVG(actual_worth) AS avg_price,
           AVG(actual_worth) AS median_price FROM OrderedPrices
    WHERE rn IN ( (total_rows + 1) / 2, (total_rows + 2) / 2)
```

Output

The screenshot shows a SQL query results window. At the top, there are two tabs: 'Results' (selected) and 'Messages'. Below the tabs is a table with two rows. The table has three columns: 'avg_price' and 'median_price'. The values in the first row are '979000' for both columns.

	avg_price	median_price
1	979000	979000

Step 4 — Monthly Price Trend

```
SELECT DATEFROMPARTS(YEAR(instance_date), MONTH(instance_date), 1) AS month,
       COUNT(*) AS transactions,
       AVG(actual_worth) AS avg_price,
       AVG(meter_sale_price) AS avg_price_per_sqm FROM sales_data
    GROUP BY DATEFROMPARTS(YEAR(instance_date), MONTH(instance_date), 1)
    ORDER BY month
```

Output

	month	transactions	avg_price	avg_price_per_sqm
1	2024-02-01	14	4412787	8457.2021484375
2	2024-03-01	23	5501868	7977.67167331861
3	2024-04-01	14	7779625	7405.49855259487
4	2024-05-01	20	8782407	9209.48150634766
5	2024-06-01	35	4679939	9194.67254813058
6	2024-07-01	52	1444372	9821.74731914814
7	2024-08-01	55	1610771	9716.80476296165
8	2024-09-01	37	1613736	10233.6808174752
9	2024-10-01	31	5199100	9770.22746818296
10	2024-11-01	29	4125579	9938.09763941272

Step 5 — Property Type Demand

SELECT

```
property_type_en, COUNT(*) AS transactions,  
AVG(CAST(actual_worth AS DECIMAL(18,2))) AS avg_price,  
AVG(meter_sale_price) AS price_per_sqm FROM sales_data  
GROUP BY property_type_en ORDER BY transactions DESC
```

Output

	property_type_en	transactions	avg_price	price_per_sqm
1	Unit	755	927613.169536	10102.1947569588
2	Land	112	23613353.491071	8989.40374901907
3	Building	5	44754185.800000	12680.583984375
4	Villa	1	10140000.000000	6188.47021484375

Step 6 — Room Demand

SELECT

```
rooms_en, COUNT(*) AS transactions,  
AVG(CAST(actual_worth AS DECIMAL(18,2))) AS avg_price  
FROM sales_data  
GROUP BY rooms_en  
ORDER BY transactions DESC
```

Output

The screenshot shows the SQL Server Management Studio interface with the 'Results' tab selected. The results are displayed in a table with four columns: 'rooms_en', 'transactions', and 'avg_price'. The table has 6 rows, including one row where 'rooms_en' is 'NULL'.

	rooms_en	transactions	avg_price
1	1 B/R	328	867296.112804
2	2 B/R	282	1127066.609929
3	Studio	122	545864.860655
4	NULL	118	24394970.508474
5	3 B/R	22	1397569.136363
6	Shop	1	700000.000000

Step 7 — Parking Impact

SELECT

```
has_parking, COUNT(*) AS transactions,  
AVG(CAST(actual_worth AS DECIMAL(18,2))) AS avg_price  
FROM sales_data GROUP BY has_parking
```

Output

The screenshot shows the SQL Server Management Studio interface with the 'Results' tab selected. The results are displayed in a table with four columns: 'has_parking', 'transactions', and 'avg_price'. The table has 2 rows, where 'has_parking' is 0 and 1.

	has_parking	transactions	avg_price
1	0	120	24008125.641666
2	1	753	926931.455511

Step 8 — Top Projects in Majan

```
SELECT TOP 5  
project_name_en, COUNT(*) AS transactions,  
AVG(CAST(actual_worth AS DECIMAL(18,2))) AS avg_price  
FROM sales_data GROUP BY project_name_en ORDER BY transactions DESC
```

Output

The screenshot shows the SQL Server Management Studio interface with the 'Results' tab selected. The results are displayed in a table with four columns: 'project_name_en', 'transactions', and 'avg_price'. The table has 5 rows, including one row where 'project_name_en' is 'NULL'.

	project_name_en	transactions	avg_price
1	NULL	567	4154878.996472
2	SHERENA RESIDENCE	98	853913.030612
3	Paradise View 1	64	854059.515625
4	THE HAVEN	47	900262.702127
5	MADISON RESIDENCE	22	503018.272727

Step 9 — Final Dataset for Power BI

```
SELECT  
  
    DATEFROMPARTS(YEAR(instance_date), MONTH(instance_date), 1) AS month,  
  
    property_type_en, rooms_en,  
  
    has_parking, COUNT(*) AS transactions,  
  
    AVG(CAST(actual_worth AS DECIMAL(18,2))) AS avg_price,  
  
    AVG(CAST(meter_sale_price AS DECIMAL(18,2))) AS price_per_sqm  
  
FROM sales_data  
  
GROUP BY DATEFROMPARTS(YEAR(instance_date), MONTH(instance_date), 1),  
  
    property_type_en, rooms_en, has_parking ORDER BY month
```

Output

	month	property_type_en	rooms_en	has_parking	transactions	avg_price	price_per_sqm
1	2024-02-01	Land	NULL	0	3	17144587.000000	5179.840000
2	2024-02-01	Unit	1 B/R	1	1	680000.000000	8097.170000
3	2024-02-01	Unit	2 B/R	1	8	1106908.625000	9801.963750
4	2024-02-01	Unit	Studio	1	2	405000.000000	8174.215000
5	2024-03-01	Building	NULL	0	1	3200000.000000	3854.170000
6	2024-03-01	Land	NULL	0	7	15968528.142857	5196.958571
7	2024-03-01	Unit	1 B/R	1	7	773807.000000	9772.384285
8	2024-03-01	Unit	2 B/R	1	2	1183134.500000	10264.590000
9	2024-03-01	Unit	3 B/R	1	1	1800000.000000	9483.670000
10	2024-03-01	Unit	Studio	1	5	396072.000000	8966.806000

3. Power BI Dashboard & Insights

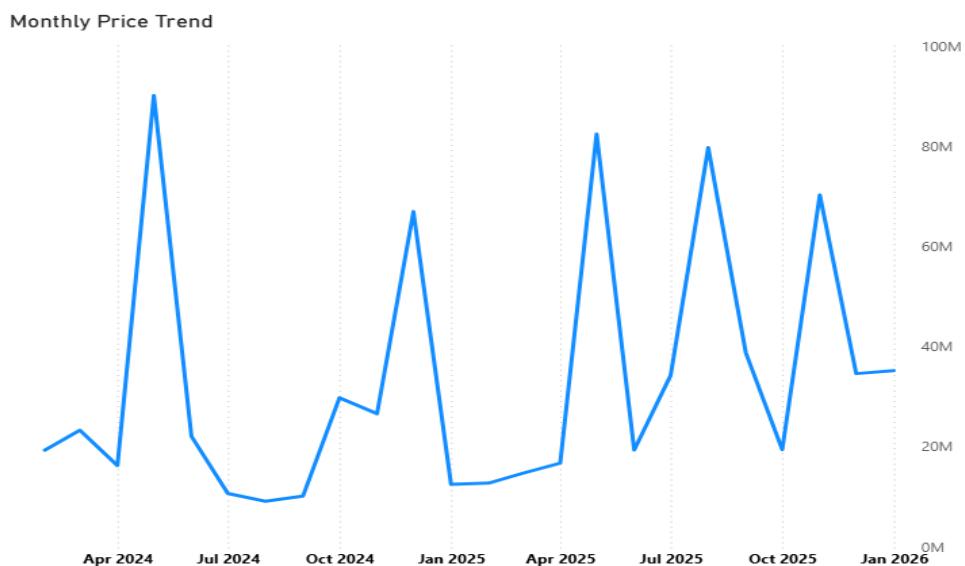
Market Summary Dashboard



KPI Cards:

KPI	Interpretation
873 Total Transactions	Strong activity in Majan over 24 months
3.58bn Total Sales Value	High investment inflow into the area
7.30M Avg Property Price	Mid-high ticket property market
10.22K Average Price per sqm	Consistent price band
24 Total Months	2-year continuous dataset

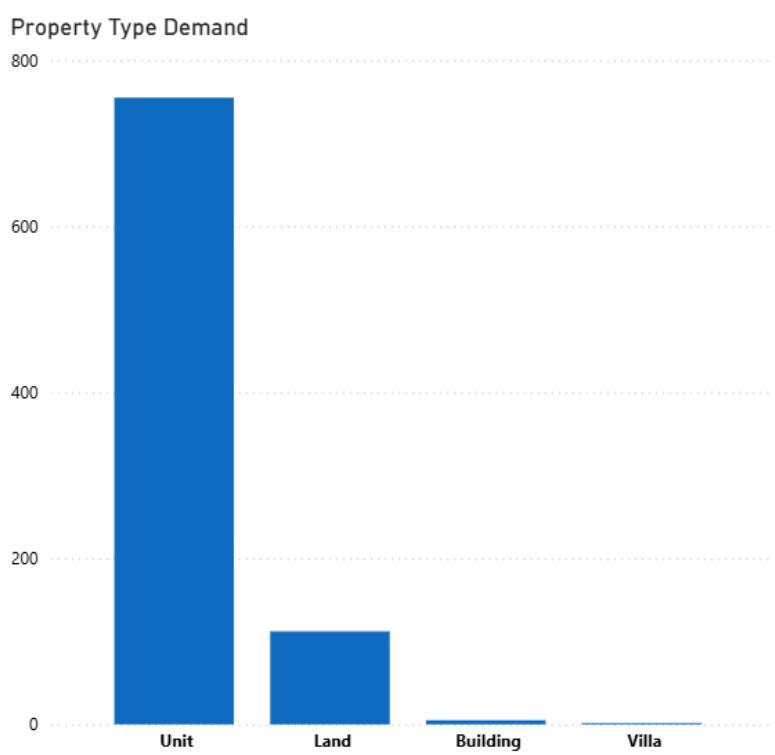
Monthly Price Trend (Line Chart)



Insights:

- Market shows **seasonal spikes** approx. every 4–6 months.
- Price dips in **mid-2024**, followed by strong growth in **early 2025**.
- Q4 2025 shows stable pricing, indicating a **mature demand cycle**.

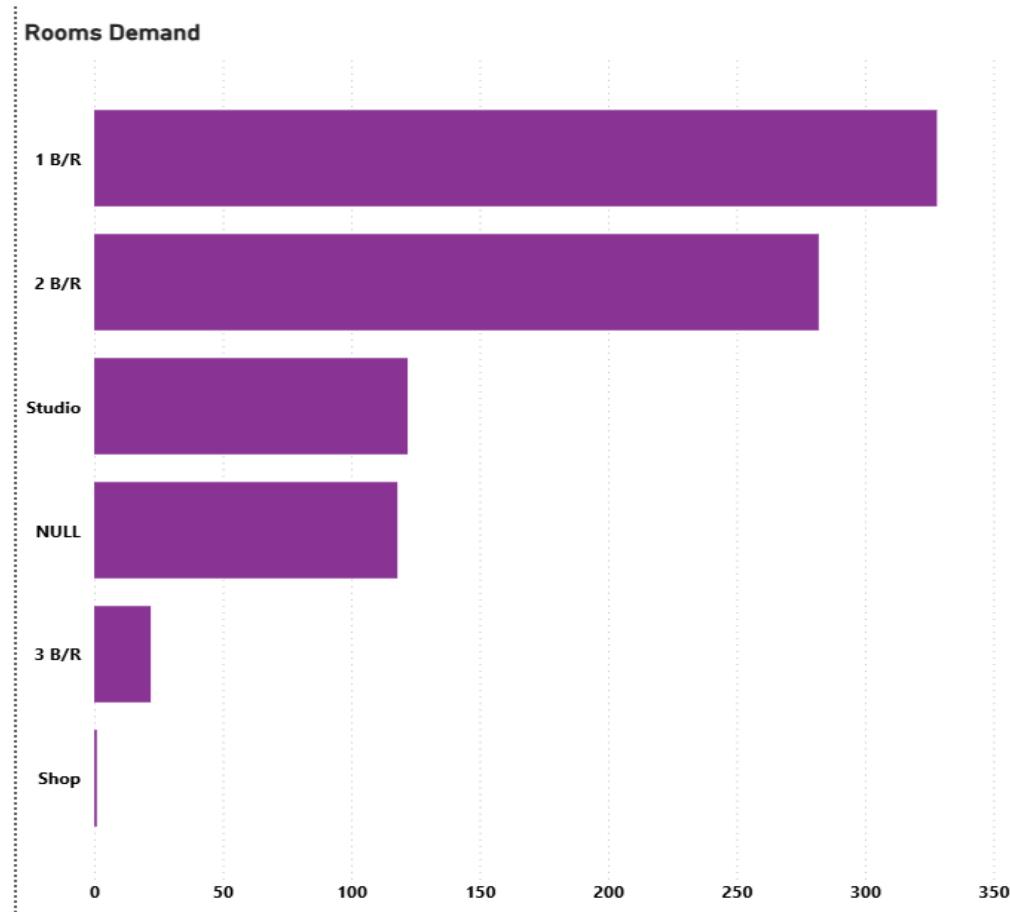
Property Type Demand (Bar Chart)



Insights:

- **80–90% transactions are UNIT properties** → Majan dominated by residential apartments.
- **Land & Building transactions extremely low** → limited supply.
- **Villas nearly zero** → Majan is not a villa-driven community.

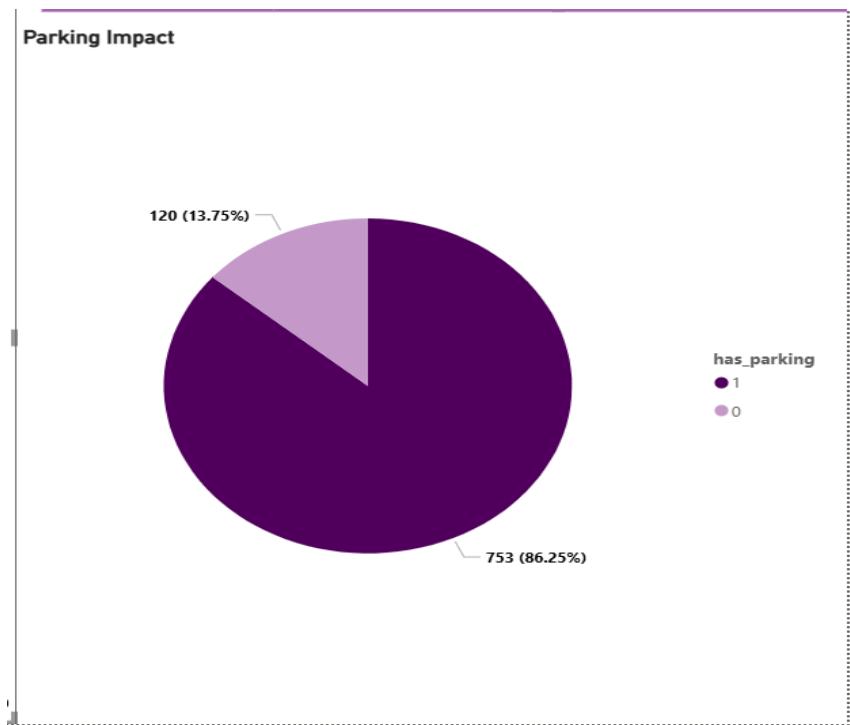
Rooms Demand (Horizontal Bar)



Insights:

- **1 B/R highest demand (328 transactions)** → Ideal for end-users & investors.
- **2 B/R second most popular** → Family-driven demand.
- **Studios moderate** → Typically for rental investors.
- **Very low 3 B/R** → Limited inventory.

Parking Impact (Pie Chart)



Insights:

- **86% properties include parking** → Parking is a standard amenity.
- Properties **with parking** show **higher average price**, indicating premium value

4. Conclusion

Majan shows a **healthy and stable real estate market**, dominated by **apartment units**, especially **1B/R and 2B/R**.

Demand fluctuates seasonally but remains steady, with strong contribution from investor-driven transactions.

This dashboard helps investors, analysts, and brokers identify:

- Pricing windows
- Best-performing property types
- Data-backed investment hotspots