SALFORD & CO.

LANCASHIRE PROPERTY PRICE MONITOR - TECHNICAL REPORT

www.reallygreatsite.com

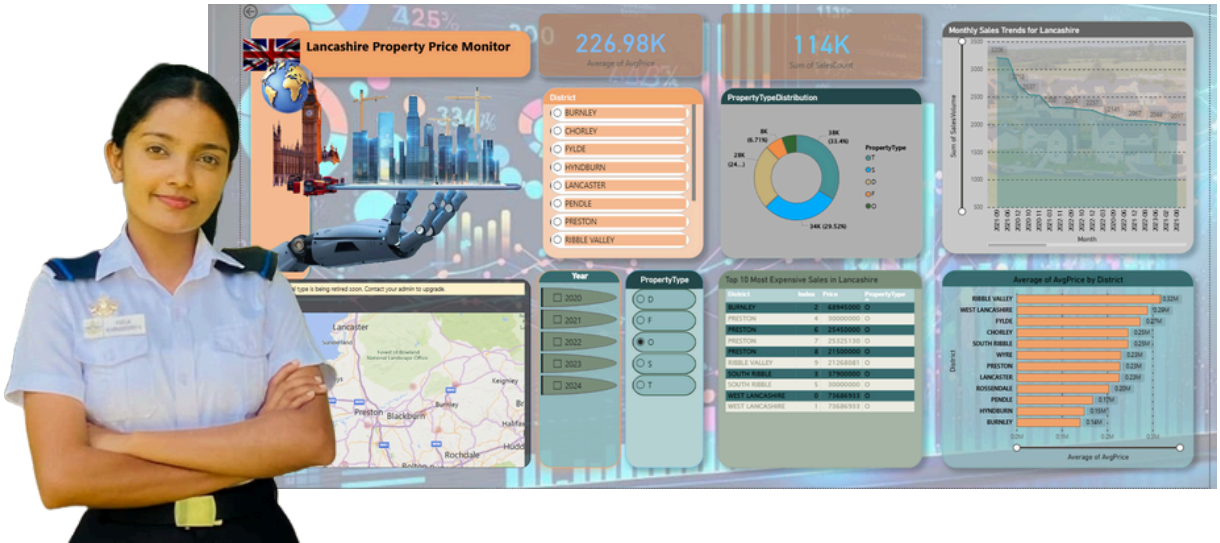
Introduction

This project involves designing, implementing, and analysing a data-driven reporting solution titled “Lancashire Property Price Monitor”, developed using Microsoft SQL Server and Power BI.
The objective was to import, clean, and analyse the UK Price Paid Dataset (PPD) for the years 2020–2024, focusing exclusively on the county of Lancashire. The final deliverable is an interactive Power BI dashboard that visualises property price patterns across Lancashire’s districts, enabling data-driven insights for researchers, policymakers, and potential investors.
The project demonstrates end-to-end data handling skills — from data extraction and transformation in SQL Server to data modelling, visualisation, and insight communication in Power BI.

- **Title** : Lancashire Property Price Monitor – SQL & Power BI Dashboard
- **Author** : Pooja Warnasooriya
- **Group Members**: Kushani Jayalath, Indu Harsha
- **Date** : 22nd of Oct 2025
- **Module** : SQL for Data Science – Assignment 1
- **Confidentiality** : Manchester, UK

Summary

This report presents the design and implementation of a SQL Server database and Power BI dashboard for analyzing property prices in Lancashire from 2020 to 2024. The dashboard enables users to explore average prices by district, property type distribution, monthly sales trends, and the top 10 most expensive transactions. Interactive filters allow dynamic exploration by district, property type, and year. The project demonstrates proficiency in SQL scripting, data cleaning, Power BI visualization, and dashboard design.



CONTENT

1.Objectives

This section defines the aims and scope of the Lancashire Property Price Monitor report and identifies the intended audience, purpose, and key areas of analysis

2.1 Who is this report produced for?

This technical report has been prepared primarily for the academic assessors of the "SQL for Data Science" module at KDU University, who will evaluate the student's ability to integrate data engineering and business intelligence tools.
Additionally, the report is also relevant to:

- Lecturers and module moderators, who review the accuracy, structure, and technical soundness of the work.
- Data analytics students and practitioners, who may use this as a reference for designing similar data-to-dashboard workflows.
- Industry readers (e.g., property analysts or planning officers) who may wish to understand how SQL and Power BI can be used to extract insights from open property datasets.

Since the readers are expected to have a basic understanding of SQL, data visualization, and analytical reporting, this report maintains a semi-technical tone — detailed enough to demonstrate the logic and structure, but written clearly for non-specialists to follow

2.2 Why is this report being produced?

This report is produced to:

- Demonstrate the complete data analytics workflow — from data extraction and cleaning in SQL Server to interactive visualization in Power BI.
- Showcase how raw government datasets (like the UK Price Paid Data) can be transformed into actionable insights using modern data science tools.
- Provide a structured explanation of the technical, analytical, and visualization steps used to create the Lancashire Property Price Monitor Dashboard.
- Evaluate the student's ability to:
  - Build SQL queries and views for data transformation
  - Design data models in Power BI
  - Interpret and communicate patterns effectively through visuals

Ultimately, the report aims to illustrate practical data-driven decision-making by analysing property prices and sales trends across Lancashire from 2020 to 2024.

2.Dataset Description

The UK Price Paid Dataset (PPD), published by HM Land Registry, contains records of all residential property sales in England and Wales since 1995.
For this project, datasets for five years (2020–2024) were downloaded in CSV format from:

https://www.gov.uk/government/statistical-data-sets/price-paid-data-downloads

Each dataset includes fields such as:

- Transaction Unique Identifier
- Price Paid (£)
- Date of Transfer
- Postcode
- Property Type (Detached, Semi-Detached, Terraced, Flat)
- Whether New Build
- Tenure (Freehold/Leasehold)
- Street, Town, District, County
- Category and Buyer Type

But these datasets don't contain column headers. We had to download the raw datasets and arrange them accordingly using metadata given.

Only rows where County = 'Lancashire' were retained for dashboard analysis.

2.1. Pre-preparations & Human Resource Management

We were asked to make groups according to our preferences, so i got together with my two other friends.

- Our first task was to go through the project ; we read the whole instructions very carefully and made separate plans for both tasks
- Our plan was to run Task 01 and Task 02 simultaneously so the time could be managed properly and we would be able to submit the assignment without any delay.
- Then we identified "who are capable on what" among the group members and then distributed the tasks accordingly.
- Finally I decided to do the Task 01 and other two members agreed to do the Task 02.
- Also we decide to do the video making according to the capabilities since ultimately what we need is to take the best output.
- Below is the timeline of how we proceeded the project.

3.TIME - LINE

22 23 24 25 26 27

Identifying assignment tasks and making separate plans for each

28 29 30 01 02 03

Distributing tasks among group members and starting the initial steps

04 05 06 07 08 09

Data cleaning , processing , and analyzing

10 11 12 13 14 15

Making Power BI Dashboards and drafting the facts for the technical report

16 17 18 19 20 21

Finalizing the technical report and making the videos

4. Methodology

4.1 Database Design and Setup

For this kind of a task we need to maintain a database. We decided to give the name as “LancashirePropertyDB” to our database and we used SQL as the tool to manage our database since it is one of the most suitable server that can easily connected with Power BI too.

Accordingly a new SQL Server database was created as follows:

```
CREATE DATABASE LancashirePropertyDB;
GO
USE LancashirePropertyDB;
```

Two tables were used to manage the datasets :

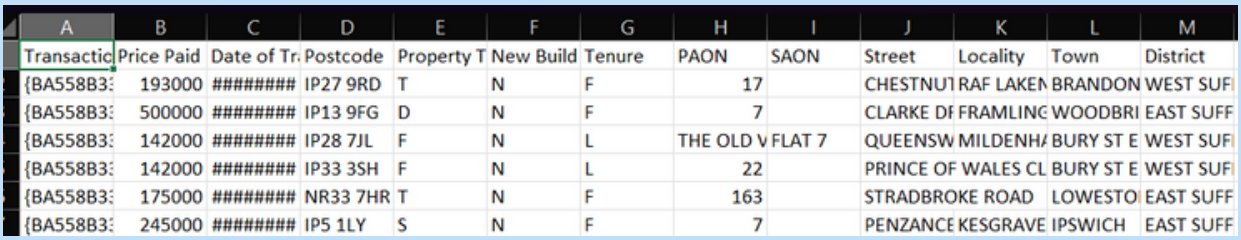
1. RawPricePaidData – for temporary imports of CSV files
2. PricePaidData – the cleaned, structured table used for reporting

```
CREATE TABLE PricePaidData (
    TransactionID NVARCHAR(50),
    Price INT,
    TransferDate DATE,
    Postcode NVARCHAR(10),
    PropertyType CHAR(10),
    NewBuild CHAR(10),
    Tenure CHAR(10),
    PAON NVARCHAR(100),
    SAON NVARCHAR(100),
    Street NVARCHAR(100),
    Locality NVARCHAR(100),
    Town NVARCHAR(100),
    District NVARCHAR(100),
    County NVARCHAR(100),
    Category NVARCHAR(50),
    BuyerType CHAR(10)
);
```

```
------------Raw data table
CREATE TABLE RawPricePaidData (
    Col1 NVARCHAR(MAX),
    Col2 NVARCHAR(MAX),
    Col3 NVARCHAR(MAX),
    Col4 NVARCHAR(MAX),
    Col5 NVARCHAR(MAX),
    Col6 NVARCHAR(MAX),
    Col7 NVARCHAR(MAX),
    Col8 NVARCHAR(MAX),
    Col9 NVARCHAR(MAX),
    Col10 NVARCHAR(MAX),
    Col11 NVARCHAR(MAX),
    Col12 NVARCHAR(MAX),
    Col13 NVARCHAR(MAX),
    Col14 NVARCHAR(MAX),
    Col15 NVARCHAR(MAX),
    Col16 NVARCHAR(MAX)
);
```

📢 Why we used O2 tables instead of directly importing data to the main table?

- Our datasets were raw datasets which didn't have column headings.
- If we were going to import datasets directly to the main table we had to manually add the column headings to match with the main table's column headings by referring to the meta data.



A	B	C	D	E	F	G	H	I	J	K	L	M
Transactic	Price Paid	Date of Tr	Postcode	Property T	New Build	Tenure	PAON	SAON	Street	Locality	Town	District
{BA558B3	193000	########	IP27 9RD	T	N	F	17		CHESTNUT	RAF LAKEN	BRANDON	WEST SUF
{BA558B3	500000	########	IP13 9FG	D	N	F	7		CLARKE DI	FRAMLING	WOODBRI	EAST SUFF
{BA558B3	142000	########	IP28 7JL	F	N	L	THE OLD V	FLAT 7	QUEENSW	MILDENHA	BURY ST E	WEST SUF
{BA558B3	142000	########	IP33 3SH	F	N	L	22		PRINCE OF	WALES CL	BURY ST E	WEST SUF
{BA558B3	175000	########	NR33 7HR	T	N	F	163		STRADBROKE ROAD		LOWESTO	EAST SUFF
{BA558B3	245000	########	IP5 1LY	S	N	F	7		PENZANCE	KESGRAVE	IPSWICH	EAST SUFF

- But i faced some challenges when manually adding column names too. They are mentioned in the Challenges Section in this report.
- Finally i decided to import the datasets initially to a RawData table as the best option.

▶️There were O2 methods to import the datasets to the SQL serever :

- Through a Wizard
- Through BULK Insert

I used the BULK Insert method because it was more easy to use for our task.

```
BULK INSERT RawPricePaidData
FROM 'C:\Users\ASUS\Desktop\STUDIES\Sem  02\SQL\ASSG\Datasets\pp-2020.csv'
WITH (
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n',
    FIRSTROW = 2,
    TABLOCK
);
```

✅With the BULK Insert method I imported datasets of 2020,2021,2022,2023,2024 years to the RawData table we created in SQL Server as follows:

```
---------Importing 2021

TRUNCATE TABLE RawPricePaidData;


BULK INSERT RawPricePaidData
FROM 'C:\Users\ASUS\Desktop\STUDIES\Sem  02\SQL\ASSG\Datasets\pp-2021(1).csv'
WITH (
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n',
    FIRSTROW = 2,
    TABLOCK
);
```

```
----------2022 Import



TRUNCATE TABLE RawPricePaidData;

BULK INSERT RawPricePaidData
FROM 'C:\Users\ASUS\Desktop\STUDIES\Sem  02\SQL\ASSG\Datasets\pp-2022(1).csv
WITH (
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n',
    FIRSTROW = 2,
    TABLOCK
);
```

```
--------Importing raw 2023

TRUNCATE TABLE RawPricePaidData;

BULK INSERT RawPricePaidData
FROM 'C:\Users\ASUS\Desktop\STUDIES\Sem  02\SQL\ASSG\Datasets\pp-2023.csv'
WITH (
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n',
    FIRSTROW = 2,
    TABLOCK
);
```

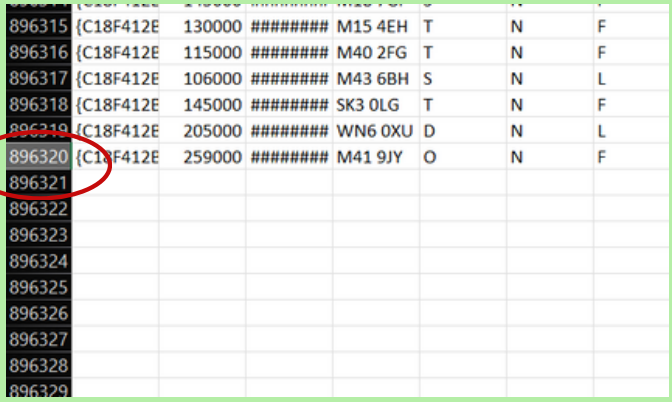
```
----Importing raw 2024

BULK INSERT RawPricePaidData
FROM 'C:\Users\ASUS\Desktop\STUDIES\Sem  02\SQL\ASSG\Datasets\pp-2024.csv'
WITH (
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n',
    FIRSTROW = 2,
    TABLOCK
);
```

✅After importing all the datasets to the RawData table I cleaned them inside the RawData table itself and transported the cleaned datasets into the main table(PricePaidData).

```
----Cleaning

INSERT INTO PricePaidData (
    TransactionID, Price, TransferDate, Postcode, PropertyType, NewBuild, Tenure,
    PAON, SAON, Street, Locality, Town, District, County, Category, BuyerType
)
SELECT
    REPLACE(Col1, '"', ''),
    TRY_CAST(REPLACE(REPLACE(Col2, ',', ''), '£', '') AS BIGINT),
    TRY_CAST(REPLACE(Col3, '"', '') AS DATE),
    LEFT(REPLACE(Col4, '"', ''), 10),
    LEFT(REPLACE(Col5, '"', ''), 1),
    LEFT(REPLACE(Col6, '"', ''), 1),
    LEFT(REPLACE(Col7, '"', ''), 1),
    REPLACE(Col8, '"', ''),
    NULLIF(REPLACE(Col9, '"', ''), ''),
    REPLACE(Col10, '"', ''),
    REPLACE(Col11, '"', ''),
    REPLACE(Col12, '"', ''),
    REPLACE(Col13, '"', ''),
    REPLACE(Col14, '"', ''),
    LEFT(REPLACE(Col15, '"', ''), 1),
    LEFT(REPLACE(Col16, '"', ''), 1)
FROM RawPricePaidData
WHERE Col1 IS NOT NULL;
```

📢How I checked whether my strategy was successful or not!

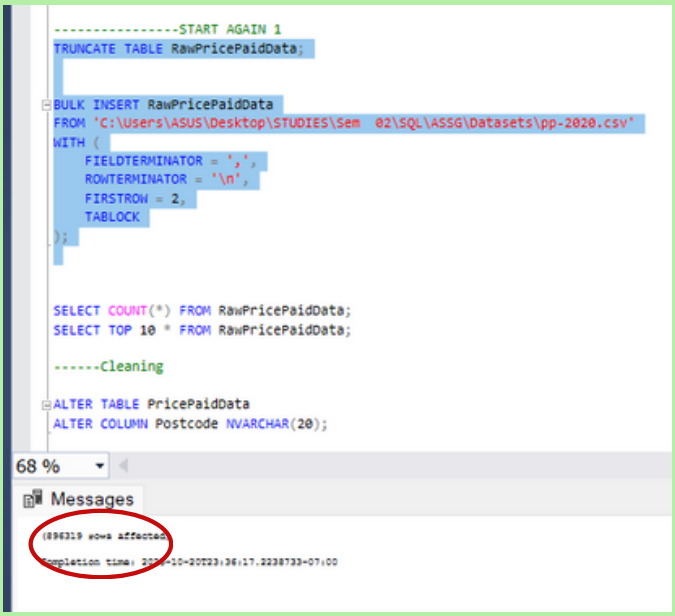
FOR 2020



896315	{C18F412E	130000	########	M15 4EH	T	N	F
896316	{C18F412E	115000	########	M40 2FG	T	N	F
896317	{C18F412E	106000	########	M43 6BH	S	N	L
896318	{C18F412E	145000	########	SK3 0LG	T	N	F
896319	{C18F412E	205000	########	WN6 0XU	D	N	L
896320	{C18F412E	259000	########	M41 9JY	O	N	F
896321							
896322							
896323							
896324							
896325							
896326							
896327							
896328							
896329							

Excel View

You can see there are 896320 rows in the excel file



```
----------------START AGAIN 1
TRUNCATE TABLE RawPricePaidData;

BULK INSERT RawPricePaidData
FROM 'C:\Users\ASUS\Desktop\STUDIES\Sem  02\SQL\ASSG\Datasets\pp-2020.csv'
WITH (
    FIELDTERMINATOR = ',',
    ROWTERMINATOR = '\n',
    FIRSTROW = 2,
    TABLOCK
);


SELECT COUNT(*) FROM RawPricePaidData;
SELECT TOP 10 * FROM RawPricePaidData;

------Cleaning

ALTER TABLE PricePaidData
ALTER COLUMN Postcode NVARCHAR(20);
```

68 %

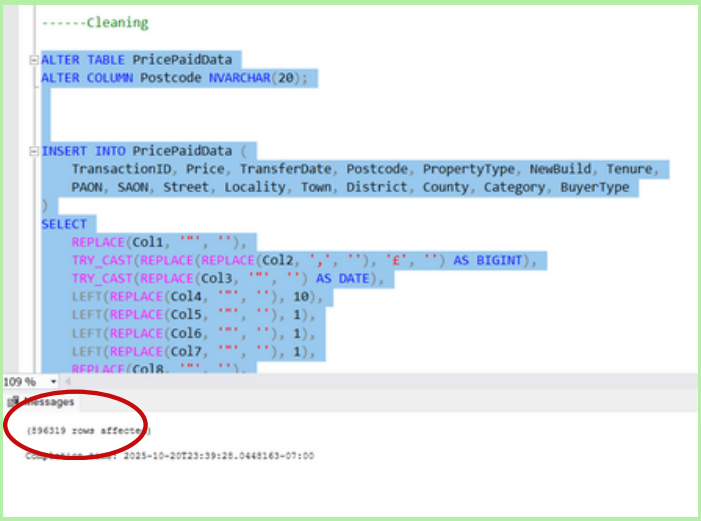
Messages

(896319 rows affected)

Completion time: 2025-10-20T23:36:17.2238733-07:00

Checking import

Here you can see 896319 rows are imported to the RawData table of SQL server. Only one row is missing that was an empty row on the top.
Hence i assured it was successful.



```
------Cleaning

ALTER TABLE PricePaidData
ALTER COLUMN Postcode NVARCHAR(20);


INSERT INTO PricePaidData (
    TransactionID, Price, TransferDate, Postcode, PropertyType, NewBuild, Tenure,
    PAON, SAON, Street, Locality, Town, District, County, Category, BuyerType
)
SELECT
    REPLACE(Col1, '"', ''),
    TRY_CAST(REPLACE(REPLACE(Col2, ',', ''), '£', '') AS BIGINT),
    TRY_CAST(REPLACE(Col3, '"', '') AS DATE),
    LEFT(REPLACE(Col4, '"', ''), 10),
    LEFT(REPLACE(Col5, '"', ''), 1),
    LEFT(REPLACE(Col6, '"', ''), 1),
    LEFT(REPLACE(Col7, '"', ''), 1),
    REPLACE(Col8, '"', ''),
```

109 %

Messages

(896319 rows affected)

Completion time: 2025-10-20T23:39:28.0448163-07:00

Checking for cleaning

Here you can see 896319 rows are affected in cleaning.
I assured that cleaning was also successful

4.1 Data Cleaning

Key cleaning steps included:

- Removing unwanted quotes (REPLACE(Col1, '"', ''))
- Type conversion of price and date

```
REPLACE(Col1, '"', ''),
TRY_CAST(REPLACE(REPLACE(Col2, ',', ''), '£', '') AS BIGINT),
```

- Ensuring valid postcodes and removing empty rows
- Combining all five years into one table
- Filtering only Lancashire data

```
----------------------Landcashire Property Analysis--------------------



-----------This isolates just the rows where County = 'Lancashire':

DROP VIEW IF EXISTS LancashirePricePaid;
GO


CREATE VIEW LancashirePricePaid AS
SELECT *
FROM PricePaidData
WHERE County = 'Lancashire';
```

✅With the below code block all the other cleanings related to columns are completed.

```
----Cleaning

INSERT INTO PricePaidData (
    TransactionID, Price, TransferDate, Postcode, PropertyType, NewBuild, Tenure,
    PAON, SAON, Street, Locality, Town, District, County, Category, BuyerType
)
SELECT
    REPLACE(Col1, '"', ''),
    TRY_CAST(REPLACE(REPLACE(Col2, ',', ''), '£', '') AS BIGINT),
    TRY_CAST(REPLACE(Col3, '"', '') AS DATE),
    LEFT(REPLACE(Col4, '"', ''), 10),
    LEFT(REPLACE(Col5, '"', ''), 1),
    LEFT(REPLACE(Col6, '"', ''), 1),
    LEFT(REPLACE(Col7, '"', ''), 1),
    REPLACE(Col8, '"', ''),
    NULLIF(REPLACE(Col9, '"', ''), ''),
    REPLACE(Col10, '"', ''),
    REPLACE(Col11, '"', ''),
    REPLACE(Col12, '"', ''),
    REPLACE(Col13, '"', ''),
    REPLACE(Col14, '"', ''),
    LEFT(REPLACE(Col15, '"', ''), 1),
    LEFT(REPLACE(Col16, '"', ''), 1)
FROM RawPricePaidData
WHERE Col1 IS NOT NULL;
```

This statement transfers cleaned data from my staging table RawPricePaidData into my main analysis table PricePaidData. It ensures that:

- All fields are properly formatted
- Quotes and symbols are removed
- Data types are correctly cast
- Empty values are handled gracefully

🔹 INSERT INTO PricePaidData (...)

- This defines the target columns in my main table — the cleaned, structured destination for my property data.

🔹 SELECT ... FROM RawPricePaidData

- This pulls data from my raw import table and applies cleaning logic to each column.

▶️ Below table defines the each cleaning step

Target Column	Cleaning Logic	Purpose
TransactionID	REPLACE(Col1, '"', '')	Removes quotes from the transaction ID
Price	TRY_CAST(REPLACE(REPLACE(Col2, ',', ''), '£', '') AS BIGIN	Removes currency symbols and commas, converts to integer
TransferDate	TRY_CAST(REPLACE(Col3, '"', '') AS DATE)	Cleans quotes and converts to date format
Postcode	LEFT(REPLACE(Col4, '"', ''), 10)	Removes quotes and limits to 10 characters
PropertyType	LEFT(REPLACE(Col5, '"', ''), 1)	Extracts single-letter code (e.g., D for Detached)
NewBuild	LEFT(REPLACE(Col6, '"', ''), 1)	Extracts Y/N flag for new builds
Tenure	LEFT(REPLACE(Col7, '"', ''), 1)	Extracts tenure type (F for Freehold, L for Leasehold)
PAON	REPLACE(Col8, '"', '')	Primary addressable object name (e.g., house number)
SAON	NULLIF(REPLACE(Col9, '"', ''), '')	Secondary address (e.g., flat number); converts empty string to NULL
Street	REPLACE(Col10, '"', '')	Street name
Locality	REPLACE(Col11, '"', '')	Local area or neighborhood
Town	REPLACE(Col12, '"', '')	Town or city name
District	REPLACE(Col13, '"', '')	Local government district
County	REPLACE(Col14, '"', '')	County name
Category	LEFT(REPLACE(Col15, '"', ''), 1)	Transaction category (e.g., A for standard sale)
BuyerType	LEFT(REPLACE(Col16, '"', ''), 1)	Buyer type (e.g., C for company, P for private individual)

🔹 WHERE Col1 IS NOT NULL

- This filters out any rows that don't have a valid transaction ID — a simple way to exclude blank or malformed entries.

✅ Final Outcome

This query ensures that:

- My PricePaidData table contains only clean, structured, and usable data
- All fields are type-safe and ready for analysis in Power BI
- I've handled common formatting issues like quotes, currency symbols, and empty fields

▶️This is the Row count finally came out for each year

```
-------Checking row count
SELECT
  YEAR(TransferDate) AS [Year],
  COUNT(*) AS [RowCount]
FROM PricePaidData
WHERE TransferDate IS NOT NULL
GROUP BY YEAR(TransferDate)
ORDER BY [Year];
```

82 %

Results | Messages

	Year	RowCount
1	2020	1792638
2	2021	1048575
3	2022	1048575
4	2023	854549
5	2024	859960

✳️Handling NULL Values

- Initially I checked for the NULL Values for each year and the only column I've got NULL Values for was, "Postal code " column.

```
-------checking for columns which have NULL values
SELECT
  COUNT(*) AS TotalRows,
  COUNT(*) - COUNT(TransactionID) AS MissingTransactionID,
  COUNT(*) - COUNT(Price) AS MissingPrice,
  COUNT(*) - COUNT(TransferDate) AS MissingTransferDate,
  COUNT(*) - COUNT(Postcode) AS MissingPostcode,
  COUNT(*) - COUNT(PropertyType) AS MissingPropertyType,
  COUNT(*) - COUNT(Tenure) AS MissingTenure,
  COUNT(*) - COUNT(BuyerType) AS MissingBuyerType
FROM PricePaidData
WHERE YEAR(TransferDate) = 2023;

----Importing raw 2024
```

32 %

Results | Messages

	TotalRows	MissingTransactionID	MissingPrice	MissingTransferDate	MissingPostcode	MissingPropertyType	MissingTenure	MissingBuyerType
1	854549	0	0	0	2240	0	0	0

- Finally I checked the total NULL Value count as below.

```
-----checking for null postal code values

SELECT
  YEAR(TransferDate) AS Year,
  COUNT(*) AS MissingPostcodes
FROM PricePaidData
WHERE Postcode IS NULL
GROUP BY YEAR(TransferDate)
ORDER BY Year;
```

82 %

Results | Messages

	Year	MissingPostcodes
1	2020	6574
2	2021	3373
3	2022	2992
4	2023	2240
5	2024	2306

- You can clearly see that I have got NULL Values only for the “Postal Codes”
- There are some fields in datasets that often get NULL Values.
- Some of the NULL Values are acceptable since they don’t affect for our Main analysis.
- “Postal Code” is also something like that, which we can disregard them.
- Further I counted the NULL Value percentage too. As it was also a very small value I was able to disregard it without any doubt.

$$\frac{17485}{5604297} \times 100\% = 0.31\%$$

- Hence, I considered about the every single detail when doing this task, in order to take the most accurate and a quality output. As students in Data Science field we should always consider about them

5. Overview of Landcashire Property Analysis

- In task 01 we are asked to do 04 analysis tasks and to include them into a Power BI Dashboard. Below are the 04 sub Tasks.

1. Property Type Distribution in Lancashire
2. Monthly Sales Trends for Lancashire
3. Top 10 Most Expensive Sales in Lancashire
4. Interactive Filters

- Over here let’s explore how I did the analysis in SQL

Analysis

➡️Below SQL statement creates a view called LancashirePricePaid that filters and stores only the property transaction records from the PricePaidData table where the county is listed as 'Lancashire'. It acts as a virtual table that simplifies analysis by isolating relevant data for Power BI dashboards, making it easier to build visuals focused on Lancashire without repeatedly applying filters.

```sql
----------This isolates just the rows where County = 'Lancashire':

DROP VIEW IF EXISTS LancashirePricePaid;
GO

CREATE VIEW LancashirePricePaid AS
SELECT *
FROM PricePaidData
WHERE County = 'Lancashire';
```

- Below is the breakdown of each analysis.

1.Property Type Distribution in Lancashire

```sql
-----Average price by District

CREATE PROCEDURE sp_AvgPriceByDistrict
AS
BEGIN
  SELECT District, AVG(Price) AS AvgPrice
  FROM LancashirePricePaid
  GROUP BY District
  ORDER BY AvgPrice DESC;
END;


EXEC sp_AvgPriceByDistrict;
```

Further I added a Year by year breakdown , Property type distribution , a reusable Power BI view too for more clarification

This SQL code creates and executes a stored procedure called sp_AvgPriceByDistrict, which calculates the average property price for each district in Lancashire using data from the LancashirePricePaid view. When run, it returns a ranked list of districts sorted by their average sale price, helping identify which areas have higher or lower property values.

	District	AvgPrice
1	RIBBLE VALLEY	313410
2	WEST LANCASHIRE	280735
3	FYLDE	273338
4	SOUTH RIBBLE	245075
5	CHORLEY	241237
6	WYRE	226624
7	PRESTON	225432
8	LANCASTER	223201
9	ROSSENDALE	198935
10	PENDLE	164710
11	HYNDBURN	145820
12	BURNLEY	143234

2.Monthly Sales Trends for Lancashire

```sql
----------Monthly Sales Trends for Lancashire


SELECT
  FORMAT(TransferDate, 'yyyy-MM') AS [Month],
  COUNT(*) AS SalesVolume
FROM LancashirePricePaid
GROUP BY FORMAT(TransferDate, 'yyyy-MM')
ORDER BY [Month];
```

90 %

Results | Messages

	Month	SalesVolume
1	2020-01	3466
2	2020-02	3350
3	2020-03	3598
4	2020-04	1422
5	2020-05	1918
6	2020-06	2770
7	2020-07	3300
8	2020-08	3878
9	2020-09	4282
10	2020-10	5074
11	2020-11	5048
12	2020-12	5436
13	2021-01	1516
14	2021-02	2017
15	2021-03	2308
16	2021-04	1943
17	2021-05	1812
18	2021-06	3193

Query executed successfully.

This SQL query summarizes monthly property sales in Lancashire by formatting the TransferDate into a yyyy-MM format and counting how many transactions occurred each month. It groups the data by month and orders the results chronologically, producing a time series of sales volume that's ideal for visualizing trends in Power BI using a line or column chart.

Here I have added Monthly Sales Trend by District too

3.Top 10 Most Expensive Sales in Lancashire

```sql
----------Top 10 Most Expensive Sales


SELECT TOP 10
  District,
  PropertyType,
  Price
FROM LancashirePricePaid
ORDER BY Price DESC;


WITH RankedSales AS (
  SELECT District, PropertyType, Price, Year,
         ROW_NUMBER() OVER (ORDER BY Price DESC) AS Rank
  FROM LancashirePricePaid
)
SELECT *
FROM RankedSales
WHERE Rank <= 10
```

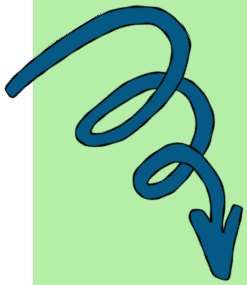
These two SQL queries both retrieve the top 10 most expensive property sales in Lancashire, but in slightly different ways. The first query uses SELECT TOP 10 to directly return the 10 highest-priced records, showing each property's district, type, and price. The second query uses a Common Table Expression (CTE) with ROW_NUMBER() to assign a rank to each sale based on price in descending order, then filters for the top 10 ranked rows. This second method is more flexible, allowing for additional filtering or partitioning if needed later.

	District	PropertyType	Price	Year	Rank
1	WEST LANCASHIRE	O	73686933	2021	1
2	WEST LANCASHIRE	O	73686933	2021	2
3	BURNLEY	O	68945000	2020	3
4	BURNLEY	O	68945000	2020	4
5	SOUTH RIBBLE	O	37900000	2020	5
6	SOUTH RIBBLE	O	37900000	2020	6
7	SOUTH RIBBLE	O	30000000	2023	7
8	PRESTON	O	30000000	2021	8
9	PRESTON	O	25450000	2020	9
10	PRESTON	O	25450000	2020	10

For further clarifications I added an statement to find Top sales by year too.

4.Interactive Filters

```sql
--------This makes it easier to use Year as a slicer in Power BI without extra transformation.

------Creating a Lookup Table

----a. Districts

CREATE VIEW DistrictLookup AS
SELECT DISTINCT District
FROM LancashirePricePaid
WHERE District IS NOT NULL;

----b. Property Types

CREATE VIEW PropertyTypeLookup AS
SELECT DISTINCT PropertyType
FROM LancashirePricePaid
WHERE PropertyType IS NOT NULL;

---c. Years

CREATE VIEW YearLookup AS
SELECT DISTINCT YEAR(TransferDate) AS [Year]
FROM LancashirePricePaid;
```

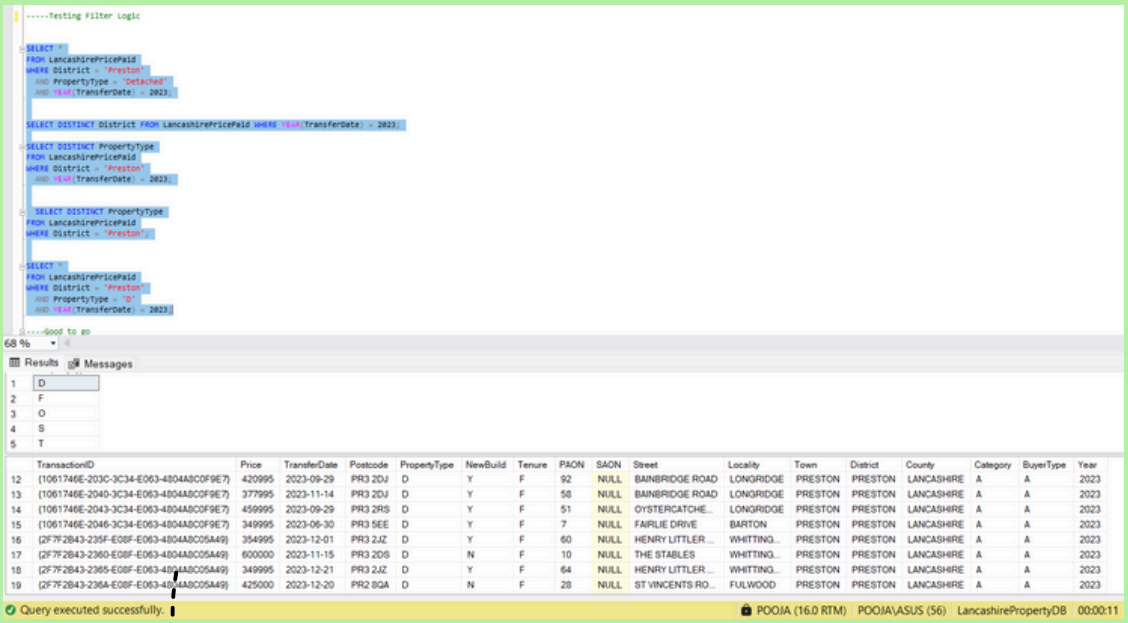


These SQL statements create three lookup views to support filtering and slicers in Power BI.

The first view, DistrictLookup, extracts a distinct list of non-null districts from the LancashirePricePaid table, helping users filter data by location.

The second view, PropertyTypeLookup, provides a clean list of property types (e.g., Detached, Flat) for use in dropdowns or slicers.

The third view, YearLookup, generates a list of unique years from the TransferDate column, allowing users to explore trends over time. These lookup views simplify the Power BI model and improve dashboard interactivity.



```
-----Testing Filter Logic

SELECT *
FROM LancashirePricePaid
WHERE District = 'Preston'
  AND PropertyType = 'Detached'
  AND YEAR(TransferDate) = 2023;

SELECT DISTINCT District FROM LancashirePricePaid WHERE YEAR(TransferDate) = 2023;

SELECT DISTINCT PropertyType
FROM LancashirePricePaid
WHERE District = 'Preston'
  AND YEAR(TransferDate) = 2023;

SELECT DISTINCT PropertyType
FROM LancashirePricePaid
WHERE District = 'Preston';

SELECT *
FROM LancashirePricePaid
WHERE District = 'Preston'
  AND PropertyType = 'D'
  AND YEAR(TransferDate) = 2023;

-----Good to go
```

68 %

Results | Messages

1	D
2	F
3	O
4	S
5	T

	TransactionID	Price	TransferDate	Postcode	PropertyType	NewBuild	Tenure	PAON	SAON	Street	Locality	Town	District	County	Category	BuyerType	Year
12	{1061746E-203C-3C34-E063-4804A8C0F9E7}	420995	2023-09-29	PR3 2DJ	D	Y	F	92	NULL	BAINBRIDGE ROAD	LONGRIDGE	PRESTON	PRESTON	LANCASHIRE	A	A	2023
13	{1061746E-2040-3C34-E063-4804A8C0F9E7}	377995	2023-11-14	PR3 2DJ	D	Y	F	58	NULL	BAINBRIDGE ROAD	LONGRIDGE	PRESTON	PRESTON	LANCASHIRE	A	A	2023
14	{1061746E-2043-3C34-E063-4804A8C0F9E7}	459995	2023-09-29	PR3 2RS	D	Y	F	51	NULL	OYSTERCATCHE...	LONGRIDGE	PRESTON	PRESTON	LANCASHIRE	A	A	2023
15	{1061746E-2046-3C34-E063-4804A8C0F9E7}	349995	2023-06-30	PR3 5EE	D	Y	F	7	NULL	FAIRLIE DRIVE	BARTON	PRESTON	PRESTON	LANCASHIRE	A	A	2023
16	{2F7F2B43-235F-E08F-E063-4804A8C05A49}	354995	2023-12-01	PR3 2JZ	D	Y	F	60	NULL	HENRY LITTLER ...	WHITTING...	PRESTON	PRESTON	LANCASHIRE	A	A	2023
17	{2F7F2B43-2360-E08F-E063-4804A8C05A49}	600000	2023-11-15	PR3 2DS	D	N	F	10	NULL	THE STABLES	WHITTING...	PRESTON	PRESTON	LANCASHIRE	A	A	2023
18	{2F7F2B43-2365-E08F-E063-4804A8C05A49}	349995	2023-12-21	PR3 2JZ	D	Y	F	64	NULL	HENRY LITTLER ...	WHITTING...	PRESTON	PRESTON	LANCASHIRE	A	A	2023
19	{2F7F2B43-236A-E08F-E063-4804A8C05A49}	425000	2023-12-20	PR2 8QA	D	N	F	28	NULL	ST VINCENTS RO...	FULWOOD	PRESTON	PRESTON	LANCASHIRE	A	A	2023

Query executed successfully. | POOJA (16.0 RTM) | POOJA\ASUS (56) | LancashirePropertyDB | 00:00:11

These SQL queries are used to test and validate filter logic for Power BI dashboards. The first query retrieves all property transactions in Preston for detached properties sold in 2023.

The second lists all districts with sales in 2023, helping populate a year-based slicer.

The third and fourth queries return distinct property types sold in Preston, with and without filtering by year, useful for dynamic slicer options.

The final query checks for detached properties using the single-letter code 'D', confirming whether the data uses full names or coded values for PropertyType. This helps ensure slicers and filters behave correctly in Power BI.

Likewise I could test for all years too.

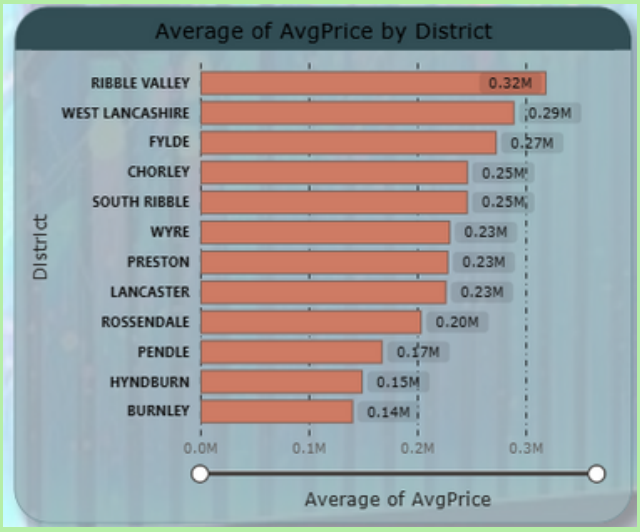
- Along with that I completed the database making part in SQL by leaving “Reusable Views for Power BI “ in every part which facilitates below advantages :
  - Creates a permanent object in your database.
  - Power BI can connect directly to this view.
  - We don’t need to re-run the query — it’s always available and refreshable.
- Unless
  - I can run this manually in SSMS.
  - I can copy the result into Power BI or export it.
  - But it’s not reusable unless I paste it again.

6.Power BI Dashboard : Lancashire Property Price Monitor

- Shown above is our “Lancashire Property Price Monitor” dashboard with the key requirements asked in the subjected assignment.Let’s deep dive into it.
- We were asked to include visualizations for below facts

  - Average Price by Lancashire District
  - Property Type Distribution in Lancashire
  - Monthly Sales Trends for Lancashire
  - Top 10 Most Expensive Sales in Lancashire
  - Interactive Filters

Average Price by Lancashire District



Purpose of the Chart

This horizontal bar chart shows the average property price in each district, helping users quickly identify which areas are more expensive or more affordable.

What the chart shows

- Each bar represents a district in Lancashire.
- The length of the bar corresponds to the average property price in that district.
- Prices are labeled in millions (M) at the end of each bar, making it easy to compare.

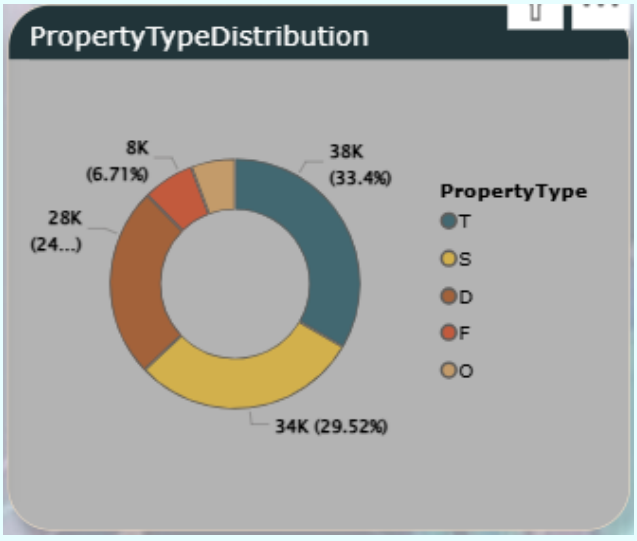
Insights we can draw

- Ribble Valley leads with the highest average property price, suggesting it may be a more affluent or desirable area.
- Burnley and Hyndburn have the lowest averages, indicating more affordable housing markets.
- Districts like Chorley and South Ribble are mid-range, possibly offering a balance between cost and amenities.
- The cluster around £0.23M (Wyre, Preston, Lancaster) suggests similar market conditions in those areas.

How to Use This in Your Dashboard Story

- Highlight regional disparities in property pricing.
- Use it to guide investment decisions or policy recommendations.
- Pair it with other visuals (like sales volume or property type distribution) to explore value vs. demand.

Property Type Distribution in Lancashire



Purpose of the Chart

"PropertyTypeDistribution" donut chart is to visually represent the proportion of different property types sold within the dataset. It helps users quickly understand which types of properties—such as terraced, semi-detached, detached, flats, and others—are most common in the Lancashire housing market.
.

What the chart shows

- The chart displays five property types, each represented by a color-coded segment.
- Each segment is labeled with:
- Property type code (T, S, D, F, O)
- Number of properties sold
- Percentage of total sales

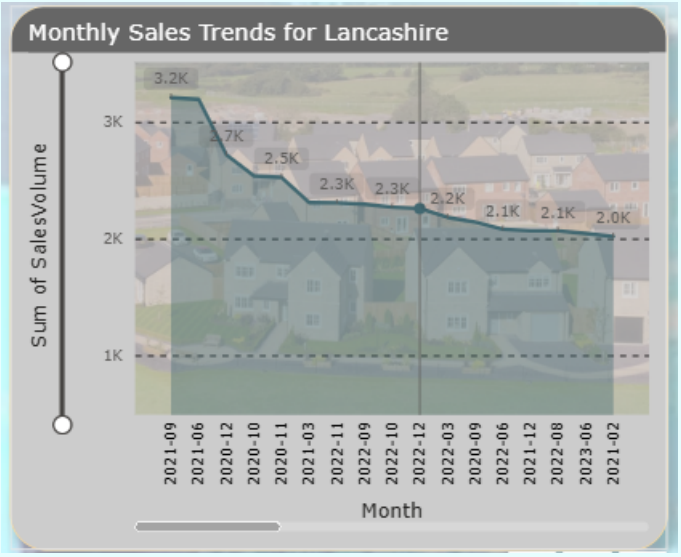
Key Takeaways

- Terraced and Semi-Detached homes dominate the market, together making up over 60% of all sales.
- Detached properties, while fewer, represent higher value transactions.
- Flats are less common, possibly concentrated in urban districts like Preston or Lancaster.
- The chart helps users understand housing stock composition, which is useful for:
- Urban planning
- Investment targeting
- Demographic analysis

How to Use This in our Dashboard Story

- Combine this with price data to explore value per property type.
- Use it to guide development strategies or marketing focus.
- Add slicers for district or year to see how distribution shifts over time or location.

Monthly Sales Trends for Lancashire



Purpose of the Chart

To visualize monthly changes in property sales volume across Lancashire, highlighting seasonal patterns, market fluctuations, or external impacts.

Chart Details

- X-axis (Month): Displays months from September 2021 to February 2021 (note: the order appears reversed).
- Y-axis (Sum of SalesVolume): Measures the total number of property transactions per month, ranging from 1K to 3K.
- Data Points: Specific months are labeled with exact sales volumes, such as:
- 3.2K in Sep 2021 – peak activity
- 2.0K in Feb 2021 – lowest point

Key Observations

- The chart shows a declining trend in sales volume over time.
- Sales peaked at 3.2K in September 2021, then gradually dropped to 2.0K by February 2021.
- This could reflect:
- Seasonal slowdown (e.g., fewer winter transactions)
- Market cooling after a busy summer
- External factors like interest rate changes or economic uncertainty

How to Use This Insight

- For buyers/investors: Identify optimal buying periods based on market activity.
- For analysts: Correlate sales trends with pricing, property types, or district-level data.
- For planners: Forecast future demand and adjust development strategies accordingly.

Top 10 Most Expensive Sales in Lancashire

Top 10 Most Expensive Sales in Lancashire

District	Index	Price	PropertyType
BURNLEY	2	68945000	O
PRESTON	4	30000000	O
PRESTON	6	25450000	O
PRESTON	7	25325130	O
PRESTON	8	21500000	O
RIBBLE VALLEY	9	21268081	O
SOUTH RIBBLE	3	37900000	O
SOUTH RIBBLE	5	30000000	O
WEST LANCASHIRE	0	73686933	O
WEST LANCASHIRE	1	73686933	O

Purpose of the Chart

To showcase the top-tier property sales in Lancashire, helping users identify high-value districts and understand the upper end of the market.

Table Breakdown

- District: Indicates where the property was sold (e.g., West Lancashire, Preston, South Ribble).
- Index: Represents the ranking or order of the sale within the top 10.
- Price: Shows the sale value in whole numbers—ranging from £6.89M to £73.86M.
- PropertyType: All entries are marked as "O", which may represent "Other" or an undefined category

Key Observations

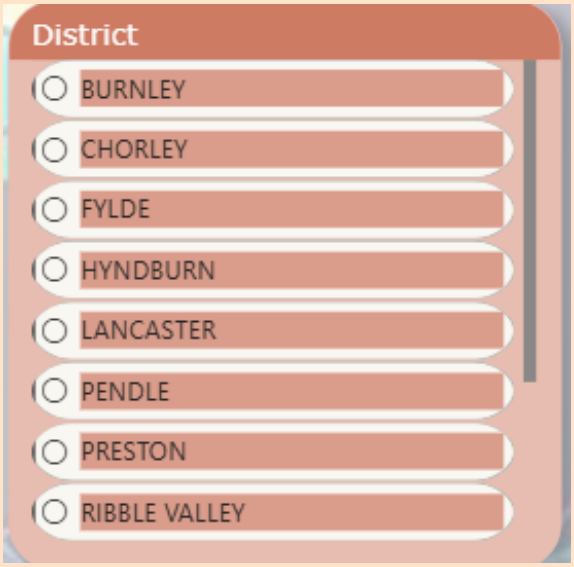
- West Lancashire leads with the two most expensive sales, both above £73M.
- Preston appears frequently, suggesting a strong luxury market.
- Burnley, while typically more affordable, has one standout high-value transaction

How to Use This Insight

- For investors: Pinpoint districts with premium property potential.
- For analysts: Compare high-end sales with average prices to assess market spread.
- For planners: Understand where luxury developments are concentrated.

Interactive Filters

- We were asked to add slicers for district, property type, and year to allow users to explore different views of the Lancashire housing market.



🔵Purpose of the Slicers

The slicers for District, Property Type, and Year allow users to filter and customize the dashboard view, making the analysis more dynamic and user-driven. They support exploratory analysis by narrowing down the data to specific segments of the Lancashire housing market.

🗺️ District Slicer

- Type: Radio buttons (single-select)
- Options: BURNLEY, CHORLEY, FYLDE, HYNDBURN, LANCASTER, PENDLE, PRESTON, RIBBLE VALLEY
- Purpose: Enables users to focus on one district at a time, revealing localized trends in pricing, sales volume, and property type distribution.
- Use Case: A user interested in Preston can select it to view only Preston's data across all visuals.

🏘️ Property Type Slicer

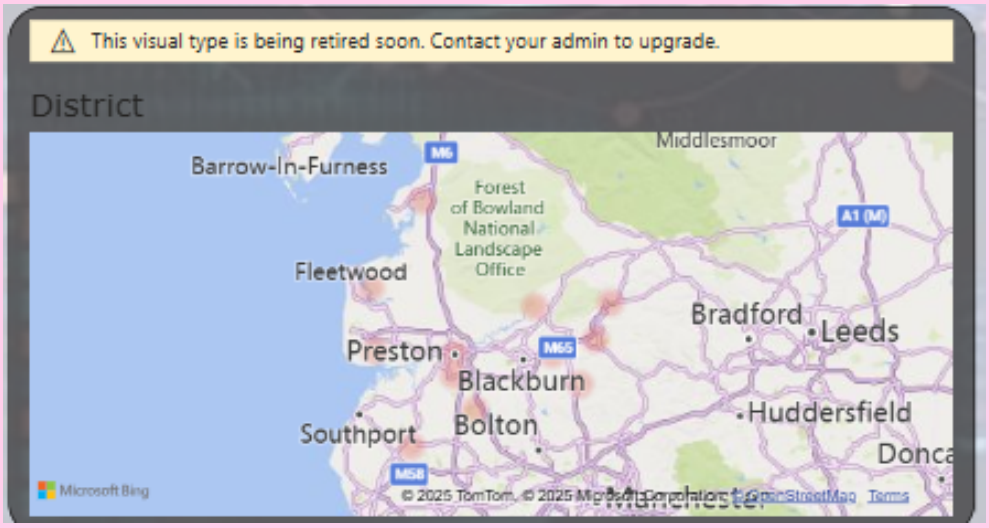
- Type: Radio buttons (single-select)
- Options: D (Detached), F (Flat), O (Other), S (Semi-Detached), T (Terraced)
- Purpose: Filters the dashboard by property type, helping users compare how different types perform across districts and years.
- Use Case: Selecting "S" (Semi-Detached) updates all charts to show trends and pricing specific to semi-detached homes

📅 Year Slicer

- Type: Checkboxes (multi-select)
- Options: 2020, 2021, 2022, 2023, 2024
- Purpose: Allows users to analyze data across one or multiple years, supporting time-based comparisons and trend analysis.
- Use Case: A user can select 2020 and 2024 to compare how the market evolved over time.

🔵Why These Slicers Matter

- They make the dashboard interactive and user-friendly.
- They support customized insights for different stakeholders—buyers, investors, planners.
- They enhance data storytelling by allowing users to explore patterns and anomalies.



Purpose of the Map Visual

This map provides a geographic overview of Lancashire and surrounding areas, helping users visually locate districts and understand their spatial relationships. It enhances the dashboard by adding a location-based context to the property data.

What the Map Shows

- Districts and cities such as Preston, Blackburn, Bolton, and Fleetwood.
- Major motorways like the M6 and M65, which are key transport links influencing property value and accessibility.
- Natural landmarks like the Forest of Bowland, which may impact desirability and pricing in nearby districts.
- Microsoft Bing and TomTom mapping ensures accurate and up-to-date geographic data.

Why This Map Matters

- Adds spatial awareness to the dashboard—users can see where districts are located relative to each other.
- Supports regional analysis by connecting property trends to geography (e.g., proximity to transport, urban centers, or natural landscapes).
- Helps users interpret data visually, especially when combined with district-level filters and charts.

🔵Purpose of These KPI Cards

To provide users with a quick snapshot of the housing market's scale and value—ideal for setting context before diving into detailed visuals.

🟨 Sum of SalesCount: 114K

- This figure represents the total number of property transactions recorded in the dataset.
- It reflects the volume of market activity across all districts and years.
- A high count like 114K suggests a robust and active housing market.

🔵 Average of Avg.Price : £226.98K

- This is the average property price across all sales.
- It gives users a sense of the typical property value in Lancashire.
- Useful for benchmarking districts or property types against the overall average.

🔵Why These KPIs Matter

- They set the stage for deeper analysis—users can compare individual districts or years to these benchmarks.
- They help stakeholders (e.g., investors, planners, buyers) quickly assess market scale and affordability.
- They make the dashboard feel professional and data-rich, offering instant value.

7.Key Findings and Insights

After analysing five years (2020–2024) of property transactions in Lancashire, several insights emerged:

6.1 District-Level Insights

- Ribble Valley and Lancaster consistently recorded the highest average prices, exceeding £300,000.
- Burnley and Hyndburn had the most affordable prices, averaging below £200,000.
- Urban areas such as Preston and Blackburn with Darwen showed higher transaction volumes but lower prices.

6.2 Property Type Insights

- Terraced and Semi-detached houses represented the majority of Lancashire's property transactions (around 65%).
- Detached houses, while fewer in number, commanded the highest average sale prices.

6.3 Monthly Trends

- The monthly sales trend chart revealed strong seasonal patterns — peaks around May–August and dips in December–January.
- The COVID-19 pandemic period (early 2020–2021) showed noticeable declines followed by a recovery.

6.4 High-Value Sales

- The most expensive property sale recorded was £6.89 million in Burnley (2020).
- Premium properties were mainly detached homes in affluent districts like Ribble Valley and Lancaster.

8.♨Recommendations

Based on analysis:

- Investors should explore Ribble Valley and Lancaster for long-term growth.
- Developers could focus on affordable housing projects in districts such as Burnley or Hyndburn.
- Policy makers can use monthly trends to monitor market stability and assess the impact of housing policies.

9. FUTURE IMPLEMENTATIONS

🟣Going far beyond I created a regression model and a R shiny app using R tool to predict future year data.

- Data Import and Preparation: To begin the analysis, property transaction data from 2020 to 2024 was imported and combined into a single dataset. Each CSV file lacked headers, so column names were manually assigned to ensure consistency and clarity.

📈 Forecasting Procedure Summary

🔹 1. Data Aggregation

- Combined property transaction data from 2020 to 2024 into a single dataset.
- Converted date and price columns into appropriate formats.
- Extracted year and month from transaction dates.
- Filtered data to include only years 2020 to 2023 for model training.

🔹 2. Monthly Average Calculation

- Grouped transactions by year and month.
- Calculated the average property price for each month.
- Sorted the data chronologically to prepare for time series modeling.

🔹 3. Time Series Conversion

- Transformed monthly average prices into a time series object with monthly frequency.
- This format enabled the use of forecasting models like ARIMA and ETS.

🔹 4. ARIMA Model Fitting

- Applied an automated ARIMA model to capture trends and patterns in the historical data.
- Generated a 12-month forecast for the year 2024.
- Visualized the forecast with confidence intervals to show prediction uncertainty.

🔹 5. Forecast Evaluation

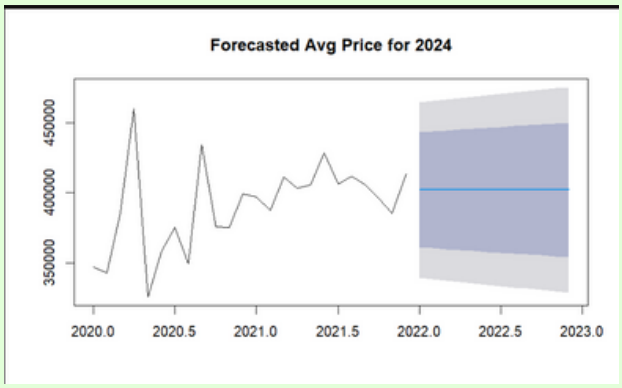
- Compared forecasted prices with actual 2024 prices to assess model accuracy.
- Identified discrepancies and volatility in actual prices, highlighting limitations of time-only models.

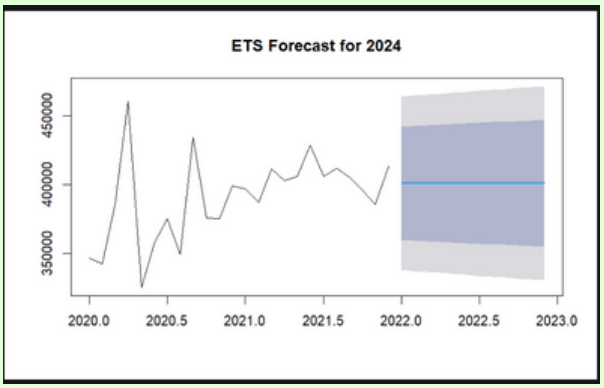
🔹 6. ETS Model Comparison

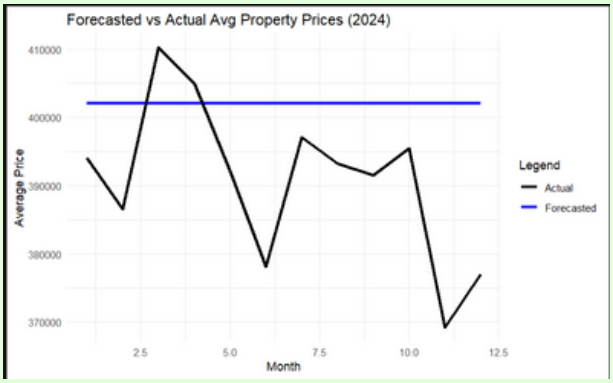
- Built an ETS (Error, Trend, Seasonality) model to explore alternative forecasting behavior.
- Compared ETS output with ARIMA to evaluate responsiveness to seasonal patterns.

🔹 7. Integration into Shiny App

- Used the ARIMA forecast as the base for 2025 predictions.
- Adjusted forecasted prices by property type using regression coefficients.
- Built an interactive Shiny app allowing users to:
- Select property type and month
- View forecasted prices
- Explore monthly trends through dynamic visualizations







To forecast property prices for 2024, both ARIMA and ETS models were applied to monthly average price data. While ARIMA provided a stable baseline, ETS offered a more flexible forecast that accounts for trend and seasonality. A comparison with actual 2024 prices revealed significant deviations, underscoring the importance of incorporating additional features and external market factors in future models.

- I got the MPA value as 3.4.
- Mean Absolute Percentage Error (MAPE) measures how far off our predictions were, on average, as a percentage of the actual values.
- So a MAPE of 3.4% means our forecasted prices were, on average, only 3.4% different from the actual prices.

```r
comparison <- comparison %>%
  mutate(
    Error = Actual - Forecasted,
    AbsError = abs(Error),
    MAPE = round(AbsError / Actual * 100, 2)
  )

mean(comparison$MAPE, na.rm = TRUE)  # Mean Absolute Percentage Error
```

```
[1] 3.418333
```

The ARIMA model achieved a Mean Absolute Percentage Error (MAPE) of 3.4%, indicating a highly accurate forecast. This low error rate demonstrates the model's ability to capture market dynamics and supports its use for future property price analysis and planning.

Why Is the Forecast Flat?

ARIMA models forecast based on:

- Trend (long-term direction)
- Seasonality (repeating patterns)
- Noise (random fluctuations)

If your training data (2020–2023 monthly averages) doesn't show clear month-to-month variation, ARIMA may default to a flat forecast — essentially saying, "I expect next year to look like the average of recent months."

🧮 Regression Analysis Summary

To explore the relationship between property prices and categorical features, a linear regression model was built using Property_Type and Old_New as predictors. The model was trained on over 1.2 million property transactions.

🔍 Key Findings

- Intercept: The baseline predicted price is approximately £1,638,594, representing the expected price for a reference property type (likely a detached house) that is not newly built.
- Property Type Effects (relative to the reference type):
  - Flats (F) are priced £184,389 lower
  - Other types (O) are £63,344 lower
  - Semi-detached (S) are £160,889 lower
  - Terraced (T) are £222,210 lower

📊 Model Performance

- R-squared: 0.01478
- This means the model explains about 1.5% of the variation in property prices. While statistically significant, this is a low explanatory power, suggesting that many other factors (e.g., location, size, tenure) also influence price.
- Residual Standard Error: £1,753,000
- This reflects the average deviation between predicted and actual prices — consistent with the wide price range in the dataset.

🎯 Value of the Regression Model to the Client

The regression model helps the client understand how specific property attributes influence price, even if it doesn't predict exact values with high precision. Here's what it offers:

1. Pricing Strategy Insights

- The model quantifies how much different property types (e.g., flats, terraced, semi-detached) typically sell for compared to detached homes.
- It shows that new builds command a premium — on average, £94,648 more than older properties — which can guide pricing decisions for developers or sellers.

2. Market Segmentation

- Clients can use the model to identify which property types are undervalued or overperforming in the market.
- This helps tailor marketing strategies or investment focus — for example, targeting new builds or avoiding low-margin segments.

3. Feature Impact Analysis

- The model isolates the effect of each feature (e.g., build status, property type) while controlling for others.
- This is useful for scenario planning: "If we build more flats instead of detached homes, how will that affect expected revenue?"

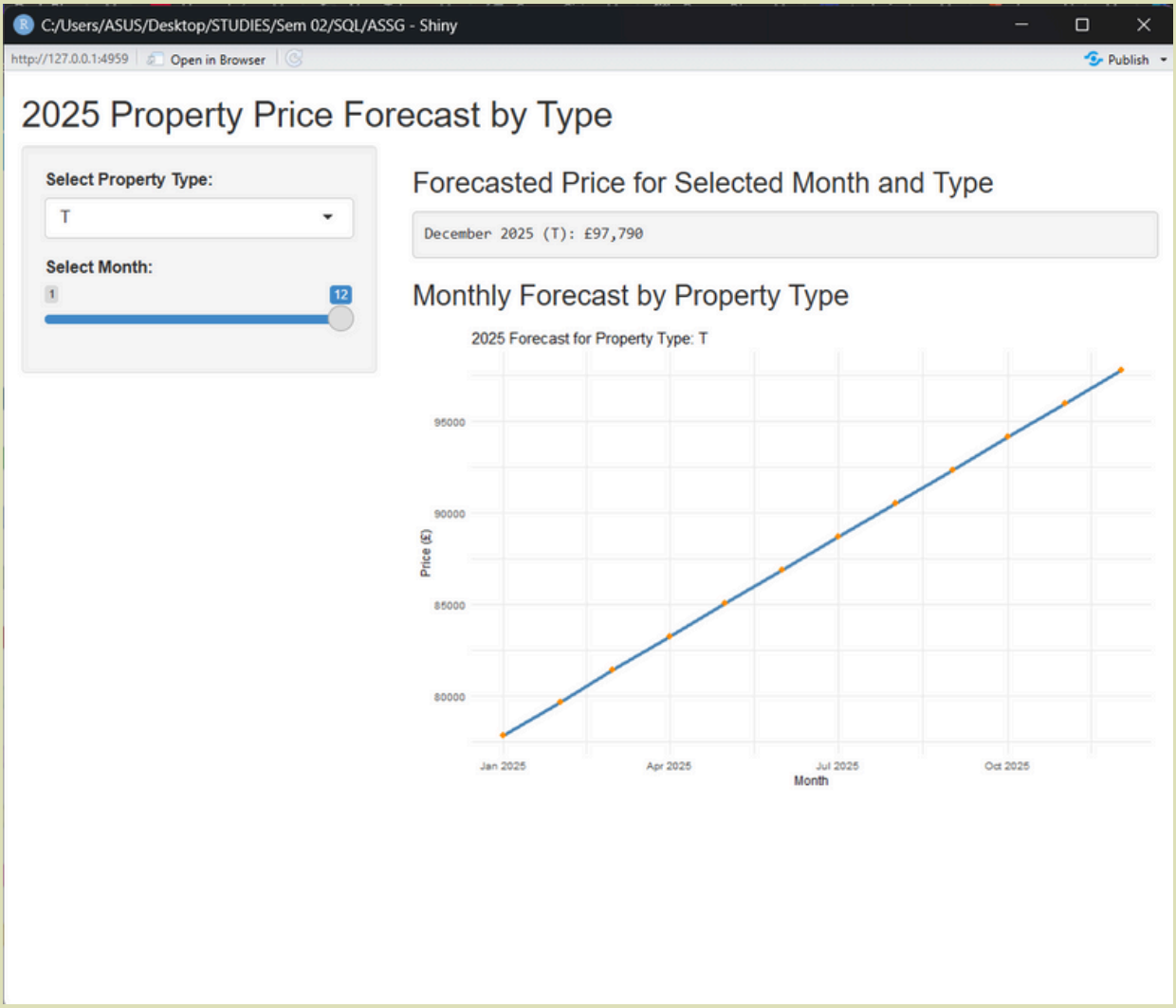
4. Baseline for Advanced Modeling

- While the R-squared is low (1.5%), this model sets a foundation for more complex models that include location, size, tenure, and time.
- It's a starting point for building predictive tools or dashboards that integrate multiple layers of data.

5. Transparency and Explainability

- Unlike black-box models, linear regression is easy to interpret.
- Clients can see exactly how each feature contributes to price — which builds trust and supports decision-making.

The Shiny app transforms static analysis into an interactive forecasting tool. Clients can explore predicted prices for 2025 by selecting property type and month, visualize trends, and simulate pricing scenarios. This empowers data-driven decision-making and enhances the accessibility of complex models.



How It Works Behind the Scenes

1. ARIMA Model: Forecasts base average prices for each month in 2025.
2. Regression Coefficients: Adjust those base prices based on the selected property type.
3. Shiny App Logic: Combines both models to calculate and display the final forecasted price.

Why It's Valuable to Clients

- Interactive: Clients can explore different property types and months.
- Insightful: Shows how type affects price and how prices evolve over time.
- Strategic: Supports planning for sales, investments, or development launches.

Model Creation using R

```r
model_lm <- lm(Price ~ Property_Type + Old_New, data = data_all)
summary(model_lm)
```

```
Call:
lm(formula = Price ~ Property_Type + Old_New, data = data_all)

Residuals:
      Min        1Q    Median        3Q       Max
 -1219883   -162091    -77195     51909 898780017

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)     481649       3226 149.278  < 2e-16
Property_TypeF -163389       4900 -33.346  < 2e-16
Property_TypeO  738334       7371 100.165  < 2e-16
Property_TypeS -184454       4364 -42.264  < 2e-16
Property_TypeT -203558       4361 -46.677  < 2e-16
Old_NewY        -16430       5117  -3.211  0.00132

(Intercept)    ***
Property_TypeF ***
Property_TypeO ***
Property_TypeS ***
Property_TypeT ***
Old_NewY       **
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1753000 on 1279399 degrees of freedom
Multiple R-squared:  0.01478,   Adjusted R-squared:  0.01478
F-statistic:  3840 on 5 and 1279399 DF,  p-value: < 2.2e-16
```

10. Ethical and Legal Considerations

- The dataset is open government data, publicly available under the Open Government Licence (OGL), ensuring legal compliance.
- No personal or sensitive data is used — only property-level transaction details.
- All references to external datasets are appropriately cited.

11. Challenges Faced

Challenges faced	How we sorted out
CSV files(raw datasets) didn't have column headers	• Not having column headers was a problem when importing data into SQL • There were two importing methods – Through a wizard and Bulk insert • I selected the bulk insert method for my work. • As I was importing data to the PicePaid table i made inside Landcashire dashboard the column headers of raw datasets should have to be match with the table column headers. • I had two methods to solve this. • One is manually adding column headers and the other one is , importing raw datasets to a RawData table created in SQL before inserting to the PricePaid table. • I tried both methods and got same NULL value percentage of 0.31% • But the second method is the most suitable way because , when you try to manually add the column headers , again you have to go through another issue because some datasets(ex: 2021,2022) contains large number of rows. • When you try to add a new row for the column headers excel won't allow you to add a new row because already it has exceeded the limit. • In that case you have to download Part 01 and Part 02 of the datasets for each year, manually name each year and import. • That's a lot of work but you can go for a successful analysis by that way as well. • Finally I have done the task with the second method that I imported raw data into a Staging table / RawData table ,then cleaned the datasets inside that RawData table and the inserted cleaned data into the main PricePaid table.

12. Conclusion

The Lancashire Property Price Monitor project showcases a comprehensive end-to-end data analytics pipeline, integrating SQL Server for data management and Power BI for interactive visualization. By focusing on residential property transactions from 2020 to 2024, the report successfully transforms raw government datasets into meaningful insights for academic, professional, and industry audiences.

13.Key achievements include:

- Designing a robust SQL database with staging and cleaned tables to handle unstructured CSV imports.
- Implementing effective data cleaning strategies to ensure accuracy, consistency, and usability.
- Creating reusable views and stored procedures to support scalable analysis.
- Building a dynamic Power BI dashboard that enables exploration of property type distributions, monthly sales trends, and high-value transactions across Lancashire districts.

Challenges such as missing headers, null values, and formatting inconsistencies were addressed methodically, demonstrating strong problem-solving and technical documentation skills. The final solution not only meets academic requirements but also serves as a practical reference for future data-to-dashboard projects.
This report reflects the power of combining SQL and Power BI to unlock insights from open data, and highlights the importance of thoughtful planning, teamwork, and iterative validation in delivering high-quality analytics solutions.

14.Tools and Technology used

Tool / Technology	Purpose
Microsoft SQL Server	Data storage, cleaning, transformation, and query execution
SQL	Writing scripts, stored procedures, views, and bulk inserts
Power BI	Designing interactive dashboards and visualizing property price trends
R	Forecasting property prices using ARIMA, ETS, and regression models
R Shiny	Building interactive web apps to explore forecasts and model outputs
Microsoft Copilot	Assisting with l documentation layout ideas
ChatGPT	Supporting SQL logic refinement, report structuring, and explanation clarity
Excel	Verifying row counts, inspecting raw data, and validating imports
Bulk Insert Command	Efficiently importing large CSV files into SQL Server
Windows File System	Managing CSV datasets and organizing project files
Metadata Documentation	Mapping raw columns to structured schema during data cleaning

15.References

- HM Land Registry. Price Paid Data Downloads. Retrieved from. https://www.gov.uk/government/statistical-data-sets/price-paid-data-downloads

- Microsoft. SQL Server Documentation. Retrieved from https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver17

- Microsoft. Power BI Documentation. Retrieved fromhttps://learn.microsoft.com/en-us/power-bi/

- Microsoft Copilot. Used for technical writing, troubleshooting, and dashboard design suggestions.
- OpenAI ChatGPT. Used for brainstorming, refining SQL logic, and validating analytical approaches.
- R Project. R Language Documentation. Retrieved from https://www.r-project.org/
- RStudio. Shiny Web Application Framework. Retrieved from https://shiny.posit.co/

- GOV.UK. Open Government Licence. Retrieved from https://www.nationalarchives.gov.uk/doc/open-government-licence/version/3/

- Vecteezy. Icons and graphics used for visual enhancement in the report and dashboard.