Walkthrough: Wauwatosa Data Analysis Project

In this project, I used Microsoft SQL Server to analyze and assess the housing market in Wauwatosa, WI for homes sold between February 2021 and March 2023. We also review the needs of a hypothetical client, and use queries to find comparable homes based on the client's needs.

For full project and code, please see:

https://github.com/bebenjam/Wauwatosa_Data_Analysis_SQL/blob/main/Wauwatosa_Home_Sales_SQL_Queries.sql

Functions and Skills Used

Aggregate Functions (MAX, MIN, AVG)
WHERE
ORDER BY
COALESCE
CASE Statements
MONTH
PARTITION BY
CTEs
GROUP BY
Window Functions
LAG
YEAR
OVER
Subqueries

The Data

The data consisted of home sales information including price sold, date sold, square footage, number of bedrooms and bathrooms, as well as home location – address, city, and state amongst others.

Address	StreetName	City	State	ZIP_Code	Bedroom:	BathroomsI	Sqft	Date_Sold	Price_Sold
2336	North 84th St	Wauwatosa	WI	53226	3	2	1894	2/28/2023	329900
2525	North 90th St	Wauwatosa	WI	53226	4	3	2825	2/28/2023	650000
2519	North 70th St	Wauwatosa	WI	53213	4	1	1424	2/28/2023	245000
1623	Alta Vista Ave	Wauwatosa	WI	53213	3	3	2780	3/1/2023	647000

With this data, I drilled down into averages per ZIP code, how that varied between home size (Sqft), and number of bedroom and bathrooms. We also broke down the average price by month to see how home prices changed over time.

Business Use Case and Code Example

I used this data to categorize potential home comparisons using the CASE and WHEN statements for a client that desired a home with square footage between 2,000 – 2,500, needed at least 3 bathrooms, and had a budget between $400,000 - $500,000.

Code:

```
SELECT
	Price_Sold,
	BedroomsFormatted,
	BathroomsFormatted,
	Sqft,
	CASE
			WHEN Sqft < 1000 THEN 'Much too small'
			WHEN Sqft <1501 THEN 'Too small'
			WHEN Sqft <2001 THEN 'A little small'
			WHEN Sqft <2501 THEN 'Just right'
			WHEN Sqft <3001 THEN 'Too big'
			ELSE 'Much too big'
			END AS Size_Fit
FROM master.dbo.Wauwatosa_Recent_Home_Sales
WHERE BathroomsFormatted >= 3 AND Price_Sold BETWEEN 40000 AND 500000;
```

Result:

Price_Sold	BedroomsFormatted	BathroomsFormatted	Sqft	Size_Fit
440000	3	3	2599	Too big
450000	4	3	2230	Just right
410000	3	3	1752	A little small
425000	2	3	1702	A little small
420000	4	3	2766	Too big
427500	6	4	2730	Too big
412500	3	3	2166	Just right
470000	3	3	1866	A little small
440000	5	3	2450	Just right

We also found out that when looking at homes sold where the number of bathrooms increases from 2 → 3, the average price jumps from $374,699 to $535,697 – an increase of almost $160,998. See below.

Code:

```
WITH BathAvgPrice AS
(
SELECT BathroomsFormatted, AVG(Price_Sold) as AveragePrice
FROM master.dbo.Wauwatosa_Recent_Home_Sales
WHERE BathroomsFormatted IS NOT NULL
GROUP BY BathroomsFormatted
)

SELECT BathroomsFormatted,
		AveragePrice,
		AveragePrice - COALESCE(LAG(AveragePrice) OVER (ORDER BY AveragePRice),
AveragePrice) AS CostIncreasePerBathroom
FROM BathAvgPrice
GROUP BY BathroomsFormatted, AveragePrice
ORDER BY BathroomsFormatted
```

Result:

Bathrooms...	AveragePrice	CostIncreasePerBathroom
1	278620	0
2	374699	96079
3	535697	160998
4	632477	96780
5	1200000	567523

Since the average price for a 3-bathroom home is greater than $500,000, the client might want to reconsider how many bathrooms they would want.

Project Outcome and Lessons Learned

- Great insights can be gleaned from data with only simple aggregate functions and SQL queries.
- CTEs and subqueries helped dive deeper into the analysis as I could compare more information about averages and the type and features of the homes sold.
- Fine tuning the code to help present my findings was really helpful. For example, using COALESCE to remove a NULL value, or using the CASE/WHEN statements to make information more digestible for a client can go a long way. Always keeping the audience and client in mind is important.
- SQL is a great program to use for this purpose because the code can be used and modified for multiple client needs, so it is highly reusable. SQL allows for manipulation of data for many other business use cases and this project just touched on a couple. For example, I could have compared characteristics of homes across locations (ZIP, city) to help a client find a home with the features they would love, but maybe in a different location.