SQL Queries Documentation for Real Estate Analytics

Introduction

This document provides comprehensive documentation for SQL queries designed to analyze key metrics and KPIs in a real estate business. The queries are organized into six main categories: Property Analytics, Sales Performance, Agent Performance, Client Engagement, Location-Based Insights, and Additional Business Insights. Each query is explained in detail, covering both its technical implementation and business significance.

Table of Contents

Property Analytics

Number of Listed Properties by Type and Location

This query provides a comprehensive breakdown of the property inventory by categorizing properties according to their type and geographical location. By grouping properties in this manner, the business can gain valuable insights into the distribution of their portfolio across different market segments and regions.

```
SELECT
    PropertyType,
    Location,
    COUNT(*) AS NumberOfProperties
From
    Properties
Group By
    PropertyType, Location
ORDER BY
    PropertyType, Location;
```

The query first selects the property type and location from the Properties table, then counts the number of properties that fall into each unique combination of these attributes. The results are grouped by both property type and location, providing a matrix view of the inventory distribution. The ordering by PropertyType and Location ensures that the results are presented in a logical and easily readable format, allowing for quick identification of patterns or anomalies in the property distribution.

This information is crucial for strategic planning, as it helps identify which property types are predominant in specific locations, potentially revealing market preferences or investment opportunities. It also assists in

inventory management by highlighting areas where certain property types might be over or underrepresented, guiding future acquisition or development decisions.

Average Price per Square Meter per City

This query calculates the average property price per square meter for each city, providing a standardized metric for comparing property values across different locations regardless of property size.

```
SELECT
    Location AS City,
    Round(AVG(PriceUSD / NULLIF(Size_sqm, 0)),2) AS AvgPricePerSqm
FROM
    Properties
GROUP BY
    Location
ORDER BY
    AvgPricePerSqm DESC;
```

The query divides the price in USD by the size in square meters for each property, then calculates the average of these values for each location. The NULLIF function is used to prevent division by zero errors in case any properties have a size of 0 square meters. The result is rounded to two decimal places for clarity and readability.

This metric is essential for real estate valuation and market analysis. It provides a normalized view of property values that allows for direct comparison between different cities or regions. The descending order arrangement highlights the most expensive areas first, which is particularly useful for identifying premium markets or potential investment hotspots. This information can guide pricing strategies, help in setting realistic expectations for buyers and sellers, and inform investment decisions by highlighting areas with higher or lower property values relative to their size.

Distribution of Property Types

This query analyzes the composition of the property portfolio by categorizing and counting properties according to their type, such as apartments, villas, or commercial spaces.

```
SELECT
    PropertyType,
    COUNT(*) AS NumberOfProperties
FROM
    Properties
GROUP BY
    PropertyType
ORDER BY
    NumberOfProperties DESC;
```

The query selects the property type from the Properties table and counts how many properties fall into each category. The results are grouped by property type and ordered in descending order of count, placing the

most common property types at the top of the results.

Understanding the distribution of property types is fundamental for portfolio management and market strategy. It reveals which property categories form the core of the business and which might represent niche or emerging markets. This information can guide marketing efforts, resource allocation, and business development strategies. For instance, if apartments constitute a significant portion of the portfolio, the business might want to develop specialized expertise in this area or, conversely, might identify an opportunity to diversify into underrepresented property types. The descending order presentation makes it immediately apparent which property types dominate the portfolio, facilitating quick strategic insights.

Top 10 Most Expensive Properties

This query identifies the ten properties with the highest price tags in the portfolio, providing insights into the premium segment of the market.

```
SELECT
    TOP 10
    PropertyID,
    PropertyType,
    PriceUSD
FROM
    Properties
ORDER BY
    PriceUSD DESC;
```

The query selects the property identifier, type, and price in USD from the Properties table, then orders the results by price in descending order and limits the output to the top 10 entries.

Analyzing the most expensive properties in the portfolio is valuable for understanding the high-end market segment. It helps identify which property types command premium prices and may reveal patterns or characteristics that contribute to higher valuations. This information can be used to refine pricing strategies for luxury properties, guide marketing efforts targeting affluent clients, and inform investment decisions in the premium market segment. Additionally, tracking these top-tier properties over time can provide insights into trends and shifts in the luxury real estate market.

Top 10 Most Visited Properties

This query identifies the properties that have received the highest number of client visits, indicating high interest or market appeal.

```
SELECT
    TOP 10
    v.PropertyID,
    p.PropertyType,
    COUNT(*) AS NumberOfVisits
FROM
    Visits AS v
Inner JOIN
```

```
    Properties AS p
    ON v.PropertyID = p.PropertyID
GROUP BY
    v.PropertyID, p.PropertyType, p.Location
ORDER BY
    NumberOfVisits DESC;
```

The query joins the Visits table with the Properties table to connect visit records with property information. It then counts the number of visits for each property, groups the results by property ID, type, and location, and orders them by the visit count in descending order, limiting the output to the top 10 most visited properties.

Identifying the most visited properties provides valuable insights into market demand and client preferences. High visit counts may indicate properties with attractive features, competitive pricing, or desirable locations. This information can guide marketing strategies by highlighting which properties generate the most interest and potentially inform pricing decisions based on demand levels. It can also help identify successful property presentation techniques or features that attract potential buyers, which could be replicated across other listings. Additionally, properties with high visit counts but no sales might warrant further investigation to understand why interest isn't converting to purchases.

Sales Performance

Total Sales Value Over Time (Monthly, Quarterly, Yearly)

These queries analyze the total sales value across different time periods, providing a comprehensive view of sales performance trends and patterns over time. Breaking down sales by different time intervals allows for both granular and high-level analysis of business performance.

Monthly Sales Analysis

```
SELECT
    MONTH(SaleDate) AS MonthNum,
    CAST(SUM(SalePrice) AS INT) AS TotalSales
FROM
    Sales
GROUP BY
    MONTH(SaleDate)
ORDER BY
    MonthNum;
```

This query extracts the month from each sale date, calculates the total sales amount for each month across all years, and presents the results in chronological order. By casting the sum as an integer, the query removes decimal places for cleaner presentation. This monthly breakdown helps identify seasonal patterns in sales performance, allowing the business to recognize high and low periods throughout the year. Such insights can inform staffing decisions, marketing campaigns, and inventory management to capitalize on peak seasons and mitigate slowdowns during traditionally weaker months.

Quarterly Sales Analysis

```
SELECT
    DATEPART(QUARTER, SaleDate) AS QrtNum,
    CAST(SUM(SalePrice) AS INT) AS TotalSales
FROM
    Sales
GROUP BY
     DATEPART(QUARTER, SaleDate)
ORDER BY
     QrtNum;
```

This query groups sales by quarter, providing a medium-term view of sales performance that smooths out monthly fluctuations while still capturing seasonal trends. The DATEPART function extracts the quarter number from each sale date, and the results are ordered chronologically. Quarterly analysis is particularly valuable for business planning and reporting, as it aligns with common financial reporting periods. It helps management assess performance against quarterly targets, make mid-year adjustments to strategies, and identify broader seasonal patterns that might be less apparent in monthly data.

Yearly Sales Analysis

```
SELECT
    YEAR(SaleDate) AS Year,
    CAST(SUM(SalePrice) AS INT) AS TotalSales
FROM
    Sales
GROUP BY
    YEAR(SaleDate)
ORDER BY
    Year;
```

This query aggregates sales by year, offering a long-term perspective on business growth or contraction. By extracting the year from each sale date and summing the sales prices, it provides a clear picture of annual performance. Yearly analysis is essential for strategic planning, assessing long-term business health, and identifying multi-year trends or cycles in the real estate market. It helps stakeholders understand whether the business is growing year over year and at what rate, informing major decisions about expansion, investment, or restructuring.

Average Sale Value per Property Type

This query calculates the average sale price for each type of property, providing insights into the relative value and market positioning of different property categories.

```
SELECT
    p.PropertyType,
    CAST(ROUND(AVG(s.SalePrice), 2) AS DECIMAL(18,2)) AS AvgSaleValue
FROM
    Sales s
```

```
Inner JOIN
    Properties p
    ON s.PropertyID = p.PropertyID
GROUP BY
    p.PropertyType
ORDER BY
    AvgSaleValue DESC;
```

The query joins the Sales and Properties tables to connect sale transactions with property information, then calculates the average sale price for each property type. The results are rounded to two decimal places, cast as a decimal for consistent formatting, and ordered by average sale value in descending order.

Understanding the average sale value by property type is crucial for pricing strategy, inventory management, and market positioning. It reveals which property categories generate the highest revenue per transaction, helping the business focus resources on high-value segments. This information can guide acquisition decisions, pricing strategies, and marketing efforts to optimize the property portfolio for profitability. The descending order presentation immediately highlights the most valuable property types, facilitating strategic decision-making about which segments to prioritize or develop further.

Conversion Rate (Sales / Visits) per Property

This complex query calculates the conversion rate for each property by determining what percentage of visits resulted in a sale, providing insights into property appeal and sales effectiveness.

```
-- Step 1: Get visit count and sale status per property
WITH PropertyStats AS (
    SELECT
        p.PropertyID,
        p.PropertyType,
        COUNT(DISTINCT v.VisitID) AS VisitCount,
        CASE WHEN
                COUNT(DISTINCT s.SaleID) > 0 THEN 1
                ELSE 0
        END AS WasSold
    FROM
        Properties AS p
        LEFT JOIN Visits AS v
        ON p.PropertyID = v.PropertyID
        LEFT JOIN Sales AS s
        ON p.PropertyID = s.PropertyID
    GROUP BY
        p.PropertyID, p.PropertyType
)
-- Step 2: Compute conversion rate as 1 sale per number of visits
SELECT
    PropertyID,
    PropertyType,
    VisitCount,
    WasSold,
    CASE WHEN
```

```
            VisitCount = 0 THEN 0
            WHEN WasSold = 1 THEN CAST(ROUND(100.0 / VisitCount, 2) AS
DECIMAL(5,2))
            ELSE 0
    END AS ConversionRatePercent
FROM
    PropertyStats
ORDER BY
    ConversionRatePercent DESC;
```

This query uses a Common Table Expression (CTE) to first gather statistics about each property, including the number of visits and whether it was sold. It then calculates the conversion rate as the percentage of visits that resulted in a sale. For properties that were sold, the conversion rate is calculated as 100% divided by the number of visits, representing the percentage of visits that led to a sale. Properties with zero visits or no sales receive a conversion rate of 0%.

Analyzing conversion rates at the property level provides valuable insights into which properties are most effective at converting interest into sales. High conversion rates may indicate properties with compelling features, competitive pricing, or effective presentation. Conversely, properties with many visits but low conversion rates might have issues with pricing, condition, or other factors that prevent interested visitors from making purchases. This information can guide targeted improvements to increase conversion rates across the portfolio and help identify best practices from high-converting properties that could be applied elsewhere.

Conversion Rate per Agent

This query calculates the conversion rate for each agent by determining what percentage of their client visits resulted in sales, providing insights into agent effectiveness and performance.

```
SELECT
    a.AgentID,
    CONCAT(a.FirstName, ' ', a.LastName) AS AgentName,
    COUNT(DISTINCT s.SaleID) AS NumberOfSales,
    COUNT(DISTINCT v.VisitID) AS NumberOfVisits,
    CAST(ROUND((COUNT(DISTINCT s.SaleID) * 100.0) / NULLIF(COUNT(DISTINCT
v.VisitID), 0),2) AS DECIMAL(5,2)) AS ConversionRatePercentage

FROM
    Agents AS a
LEFT JOIN
    Sales AS s
    ON a.AgentID = s.AgentID
LEFT JOIN
    Visits AS v
    ON a.AgentID = v.AgentID
GROUP BY
    a.AgentID, a.FirstName, a.LastName
ORDER BY
    ConversionRatePercentage DESC;
```

The query joins the Agents table with both the Sales and Visits tables to connect agents with their sales and visit activities. It counts the distinct number of sales and visits for each agent, then calculates the conversion rate as the percentage of visits that resulted in sales. The NULLIF function prevents division by zero for agents with no visits, and the result is rounded to two decimal places for clarity.

Analyzing conversion rates by agent is essential for performance management, training, and resource allocation. It identifies which agents are most effective at converting property visits into actual sales, potentially highlighting different sales techniques or approaches that could be shared across the team. High-performing agents might be recognized and rewarded, while those with lower conversion rates might benefit from additional training or mentoring. This metric provides an objective measure of sales effectiveness beyond raw sales numbers, as it accounts for the number of opportunities (visits) each agent had to make a sale.

Time on Market Before Sale

This query calculates how long each property remained on the market before being sold, measured as the number of days between the first recorded visit and the sale date.

```
SELECT
    s.PropertyID,
    p.PropertyType,
    MIN(v.VisitDate) AS FirstVisitDate,
    s.SaleDate,
    DATEDIFF(DAY, MIN(v.VisitDate), s.SaleDate) AS DaysOnMarket
FROM
    Sales AS s
Inner JOIN
    Properties AS p
    ON s.PropertyID = p.PropertyID
Inner JOIN
    Visits AS v
    ON s.PropertyID = v.PropertyID
GROUP BY
    s.PropertyID, p.PropertyType, s.SaleDate
ORDER BY
    DaysOnMarket DESC;
```

The query joins the Sales, Properties, and Visits tables to connect sale transactions with property information and visit history. It identifies the earliest visit date for each property that was sold, calculates the difference in days between that first visit and the sale date, and orders the results by the number of days on market in descending order.

Time on market is a critical metric for understanding property liquidity and market efficiency. Properties that sell quickly may indicate strong market demand, effective pricing, or particularly desirable features. Conversely, properties that remain on the market for extended periods might suggest pricing issues, condition problems, or location challenges. This information can guide pricing strategies, help set realistic expectations for sellers, and identify properties that might benefit from price adjustments or marketing enhancements. The descending order presentation highlights the slowest-moving properties first, drawing attention to potential problem areas that might require intervention.

Agent Performance

Number of Sales per Agent

This query measures the sales productivity of each agent by counting the total number of property sales they have successfully closed, providing a direct measure of their contribution to the company's revenue generation.

```
SELECT
    a.AgentID,
    CONCAT(a.FirstName, ' ', a.LastName) AS AgentName,
    COUNT(s.SaleID) AS NumberOfSales
FROM
    Agents AS a
LEFT JOIN
    Sales AS s
    ON a.AgentID = s.AgentID
GROUP BY
    a.AgentID, a.FirstName, a.LastName
ORDER BY
    NumberOfSales DESC;
```

The query joins the Agents table with the Sales table to connect agents with their sales transactions. It counts the number of sales associated with each agent and presents the results in descending order of sales volume. The LEFT JOIN ensures that all agents are included in the results, even those who haven't made any sales, providing a complete view of the sales team's performance.

This metric is fundamental for evaluating agent productivity and effectiveness. It clearly identifies top performers who drive the most business and may highlight agents who require additional support or training. The descending order presentation immediately draws attention to the highest-performing agents, facilitating recognition of success and identification of best practices that could be shared across the team. This information is crucial for performance reviews, commission calculations, and resource allocation decisions. It also provides a baseline for setting realistic sales targets based on historical performance and can help identify trends or patterns in individual agent performance over time.

Number of Client Visits per Agent

This query measures the activity level of each agent by counting the number of property visits they have conducted with clients, providing insights into their engagement and prospecting efforts.

```
SELECT
    a.AgentID,
    CONCAT(a.FirstName, ' ', a.LastName) AS AgentName,
    COUNT(v.VisitID) AS NumberOfClientVisits
FROM
    Agents AS a
LEFT JOIN
    Visits AS v
```

```
    ON a.AgentID = v.AgentID
GROUP BY
    a.AgentID, a.FirstName, a.LastName
ORDER BY
    NumberOfClientVisits DESC;
```

The query joins the Agents table with the Visits table to connect agents with their client visit activities. It counts the number of visits conducted by each agent and presents the results in descending order of visit volume. The LEFT JOIN ensures that all agents are included in the results, even those who haven't conducted any visits.

Tracking client visits per agent provides valuable insights into agent activity levels and client engagement efforts. High visit numbers may indicate proactive agents who are actively prospecting and showing properties to potential buyers. This metric complements the sales count by revealing the effort behind the results, as some properties may require more showings before a sale is secured. The information can be used to evaluate agent diligence, identify those who might need support in converting visits to sales, and recognize agents who are particularly active in the field. It also helps management understand workload distribution across the team and may inform decisions about territory assignments or specialization based on activity levels.

Conversion Rate per Agent (Visits → Sales)

This query calculates the efficiency of each agent by determining what percentage of their client visits result in successful sales, providing a measure of their sales effectiveness and closing ability.

```
SELECT
    a.AgentID,
    CONCAT(a.FirstName, ' ', a.LastName) AS AgentName,
    COUNT(DISTINCT s.SaleID) AS NumberOfSales,
    COUNT(DISTINCT v.VisitID) AS NumberOfVisits,
    CAST(ROUND((COUNT(DISTINCT s.SaleID) * 100.0) / NULLIF(COUNT(DISTINCT
v.VisitID), 0),2) AS DECIMAL(5,2)) AS ConversionRatePercentage

FROM
    Agents AS a
LEFT JOIN
    Sales AS s
    ON a.AgentID = s.AgentID
LEFT JOIN
    Visits AS v
    ON a.AgentID = v.AgentID
GROUP BY
    a.AgentID, a.FirstName, a.LastName
ORDER BY
    ConversionRatePercentage DESC;
```

The query joins the Agents table with both the Sales and Visits tables to connect agents with their sales and visit activities. It counts the distinct number of sales and visits for each agent, then calculates the conversion

rate as the percentage of visits that resulted in sales. The NULLIF function prevents division by zero for agents with no visits, and the result is rounded to two decimal places for clarity.

The conversion rate is a critical metric for evaluating agent effectiveness beyond raw numbers. It reveals which agents are most efficient at converting property showings into closed deals, highlighting different levels of sales skill or technique. High conversion rates may indicate strong closing abilities, effective client qualification, or strategic property matching. Conversely, low conversion rates despite high visit numbers might suggest areas for improvement in sales technique, negotiation skills, or client needs assessment. This information is invaluable for targeted training and development, performance management, and identifying best practices that could be shared across the team. The descending order presentation immediately highlights the most efficient agents, facilitating recognition of effective sales approaches.

Average Sale Value Handled by Each Agent

This query calculates the average monetary value of sales transactions handled by each agent, providing insights into their contribution to revenue and their positioning within different market segments.

```
SELECT
    a.AgentID,
    CONCAT(a.FirstName, ' ', a.LastName) AS AgentName,
    CAST(ROUND(AVG(s.SalePrice), 2) AS DECIMAL(15,2)) AS AverageSaleValue
FROM
    Agents AS a
LEFT JOIN
    Sales AS s
    ON a.AgentID = s.AgentID
GROUP BY
    a.AgentID, a.FirstName, a.LastName
ORDER BY
    AverageSaleValue DESC;
```

The query joins the Agents table with the Sales table to connect agents with their sales transactions. It calculates the average sale price for each agent's transactions, rounds the result to two decimal places for clarity, and presents the results in descending order of average sale value.

Understanding the average sale value per agent provides insights into market specialization and revenue contribution beyond simple sales counts. Agents with high average sale values may be focusing on premium market segments or luxury properties, while those with lower averages might specialize in entry-level or mid-market properties. This information can guide strategic decisions about agent specialization, market positioning, and commission structures. It also helps identify agents who might benefit from training or support to move into higher-value market segments. When combined with sales count data, this metric provides a more complete picture of each agent's contribution to the company's revenue and profitability. The descending order presentation immediately highlights agents handling the highest-value transactions, facilitating recognition of expertise in premium market segments.

Client Engagement

Number of Properties Visited per Client

This query analyzes client engagement by measuring how many different properties each client has visited, providing insights into their level of interest and the breadth of their property search.

```
SELECT
    c.ClientID,
    CONCAT(c.FirstName, ' ', c.LastName) AS ClientName,
    COUNT(DISTINCT v.PropertyID) AS NumberOfPropertiesVisited
FROM
    Clients AS c
LEFT JOIN
    Visits AS v
    ON c.ClientID = v.ClientID
GROUP BY
    c.ClientID, c.FirstName, c.LastName
ORDER BY
    NumberOfPropertiesVisited DESC;
```

The query joins the Clients table with the Visits table to connect clients with their property viewing history. It counts the distinct number of properties each client has visited, ensuring that multiple visits to the same property are counted only once. The results are ordered by the number of properties visited in descending order, highlighting the most active clients first.

This metric provides valuable insights into client engagement and search behavior. Clients who have visited many properties may be actively searching but having difficulty finding the right match, potentially indicating an opportunity for more targeted property recommendations or needs assessment. Conversely, clients who purchased after visiting few properties might represent efficient transactions or particularly effective property matching. The information can guide client relationship management strategies, help identify serious buyers who warrant additional attention, and inform marketing efforts by revealing patterns in client search behavior. The descending order presentation immediately highlights the most engaged clients, facilitating prioritization of follow-up activities and resource allocation.

Top Clients by Sale Value

This query identifies the clients who have made the highest-value property purchases, highlighting the most financially significant customers for the business.

```
SELECT TOP 10
    c.ClientID,
    CONCAT(c.FirstName, ' ', c.LastName) AS ClientName,
    CAST(SUM(s.SalePrice) AS FLOAT) AS SaleValue
FROM
    Clients c
Inner JOIN
    Sales s ON c.ClientID = s.ClientID
GROUP BY
    c.ClientID, c.FirstName, c.LastName
ORDER BY
    SaleValue DESC;
```

The query joins the Clients table with the Sales table to connect clients with their purchase transactions. It calculates the total value of all purchases made by each client, casts the result as a float for consistent formatting, and presents the top 10 clients in descending order of total purchase value.

Identifying top clients by sale value is crucial for customer relationship management and business development. These high-value clients represent significant revenue contributors and potential sources of referrals or repeat business. Understanding who these clients are allows the business to develop targeted retention strategies, personalized service approaches, and loyalty programs designed to maintain these valuable relationships. This information can also guide marketing efforts by revealing characteristics or preferences of high-value clients that might inform customer segmentation and acquisition strategies. The limited selection of the top 10 clients focuses attention on the most financially significant relationships, facilitating strategic resource allocation to maximize retention and potential future revenue.

First-time vs Repeat Buyers

This query categorizes clients based on their purchase history, distinguishing between those who have made a single purchase (first-time buyers) and those who have made multiple purchases (repeat buyers).

```
WITH ClientSaleCounts AS (
    SELECT
        ClientID,
        COUNT(SaleID) AS NumberOfSales
    FROM
        Sales
    GROUP BY
        ClientID
)
SELECT
    c.ClientID,
    CONCAT(c.FirstName, ' ', c.LastName) AS ClientName,
    ISNULL(csc.NumberOfSales, 0) AS TotalSalesMade,
    CASE WHEN
        csc.NumberOfSales = 1 THEN 'First-Time Buyer'
        WHEN csc.NumberOfSales > 1 THEN 'Repeat Buyer'
        ELSE 'Not a Buyer'
    END AS BuyerType
FROM
    Clients AS c
LEFT JOIN
    ClientSaleCounts AS csc
    ON c.ClientID = csc.ClientID
ORDER BY
    c.LastName, c.FirstName;
```

This query uses a Common Table Expression (CTE) to first count the number of sales associated with each client. It then joins this information with the Clients table and categorizes each client based on their purchase count: those with one purchase are labeled as "First-Time Buyers," those with multiple purchases as "Repeat

Buyers," and those with no purchases as "Not a Buyer." The ISNULL function ensures that clients with no sales are assigned a count of zero rather than NULL.

Distinguishing between first-time and repeat buyers provides valuable insights for customer relationship management and marketing strategies. Repeat buyers demonstrate loyalty and satisfaction with previous purchases, representing a valuable segment for targeted marketing and retention efforts. First-time buyers might benefit from different types of support or follow-up to encourage future purchases. Understanding the distribution of clients across these categories helps the business assess customer loyalty and the effectiveness of retention strategies. It can also inform marketing resource allocation, with different approaches potentially needed for acquiring new customers versus encouraging repeat business from existing ones. This segmentation is fundamental for developing tailored communication strategies and service approaches that address the specific needs and behaviors of different client groups.

Region-based Client Interest (Visits by City)

This query analyzes client preferences for different geographical areas by tracking the number of property visits each client has made in various cities or regions.

```
SELECT
    c.ClientID,
    CONCAT(c.FirstName, ' ', c.LastName) AS ClientName,
    p.Location AS City,
    COUNT(v.VisitID) AS NumberOfVisitsInCity
FROM
    Clients AS c
Inner JOIN
    Visits AS v ON
    c.ClientID = v.ClientID
Inner JOIN
    Properties AS p
    ON v.PropertyID = p.PropertyID
GROUP BY
    c.ClientID, c.FirstName, c.LastName, p.Location
ORDER BY
    NumberOfVisitsInCity DESC;
```

The query joins the Clients, Visits, and Properties tables to connect clients with their property viewing history and the locations of those properties. It counts the number of visits each client has made in each city or region and presents the results in descending order of visit count.

Understanding region-based client interest provides valuable insights for property recommendations, inventory management, and marketing strategies. It reveals which locations are most attractive to specific clients, potentially indicating preferences for certain neighborhoods, school districts, or amenities. This information can guide personalized property recommendations, helping agents suggest listings in areas where clients have demonstrated interest. It can also inform inventory acquisition decisions by highlighting locations with high client demand. From a marketing perspective, this data can support targeted campaigns that promote properties in regions aligned with client preferences. The descending order presentation

immediately highlights the strongest client-location affinities, facilitating prioritization of property recommendations and marketing efforts.

Location-Based Insights

Sales Heatmap by City or Region

These queries analyze the distribution of sales across different geographical locations, providing insights into market activity and performance by region.

Number of Sales per Location (Volume)

```
SELECT
    p.Location,
    COUNT(s.SaleID) AS SalesNum
FROM
    Properties AS p
INNER JOIN
    Sales AS s
    ON p.PropertyID = s.PropertyID
GROUP BY
    p.Location
ORDER BY
    SalesNum DESC;
```

This query joins the Properties and Sales tables to connect property locations with sales transactions. It counts the number of sales in each location and presents the results in descending order of sales volume. This analysis provides a clear picture of which geographical areas are experiencing the highest transaction activity, regardless of monetary value.

Understanding sales volume by location is essential for strategic planning and resource allocation. Areas with high sales volumes may indicate strong market demand, effective local marketing, or particularly desirable neighborhoods. This information can guide decisions about where to focus acquisition efforts, allocate marketing resources, or expand agent presence. It also helps identify emerging hot spots or declining areas by tracking changes in sales volume over time. The descending order presentation immediately highlights the most active markets, facilitating strategic decision-making about geographical focus and expansion opportunities.

Total Sales Value per Location (Value)

```
SELECT
    p.Location,
    CAST(SUM(s.SalePrice) AS FLOAT) AS SaleValue
FROM
    Properties p
Inner JOIN
    Sales AS s
    ON p.PropertyID = s.PropertyID
```

```
GROUP BY
    p.Location
ORDER BY
    SaleValue DESC;
```

This query joins the Properties and Sales tables to connect property locations with sales transactions. It calculates the total monetary value of all sales in each location and presents the results in descending order of total value. The CAST function ensures consistent formatting of the monetary values.

Analyzing total sales value by location provides insights into the financial significance of different geographical markets. Areas with high total sales values may represent premium markets with higher property prices or locations with a high volume of transactions. This information complements the sales volume analysis by adding a financial dimension to the geographical assessment. It can guide investment decisions, pricing strategies, and market positioning by identifying areas where the business generates the most revenue. The descending order presentation immediately highlights the most financially significant markets, facilitating strategic decision-making about resource allocation and market development.

High-performing Areas (Most Sold or Highest Priced)

This query identifies locations with the highest number of property sales, highlighting geographical areas with strong market activity and potential demand.

```
SELECT
    p.Location,
    COUNT(s.SaleID) AS SalesNum
FROM
    Properties AS p
INNER JOIN
    Sales AS s
    ON p.PropertyID = s.PropertyID
GROUP BY
    p.Location
ORDER BY
    SalesNum DESC;
```

The query joins the Properties and Sales tables to connect property locations with sales transactions. It counts the number of sales in each location and presents the results in descending order of sales count.

Identifying high-performing areas based on sales volume is crucial for strategic planning and market analysis. These areas represent locations where properties are moving quickly, potentially indicating strong demand, desirable neighborhood characteristics, or effective local marketing. Understanding which areas generate the most transactions helps the business focus resources where they are most likely to yield results. This information can guide inventory acquisition, agent allocation, and marketing campaigns by highlighting geographical hotspots. It also provides valuable context for clients seeking investment properties or trying to understand market liquidity in different areas. The descending order presentation immediately draws attention to the most active markets, facilitating quick identification of geographical strengths and opportunities.

Average Visit-to-Sale Ratio per Location

This complex query calculates how many property visits, on average, are required to achieve a sale in each location, providing insights into regional differences in conversion efficiency and buyer behavior.

```
-- 1. For every sold property, calculate its number of visits.
WITH SoldProperties AS (
    SELECT
        p.PropertyID,
        p.Location
    FROM
        Properties p
        INNER JOIN Sales AS s
                   ON p.PropertyID = s.PropertyID
),
VisitsPerSoldProperty AS (
    SELECT
        sp.PropertyID,
        sp.Location,
        COUNT(v.VisitID) AS VisitsNum
    FROM
        SoldProperties AS sp
        LEFT JOIN Visits AS v
        ON sp.PropertyID = v.PropertyID
    GROUP BY
        sp.PropertyID, sp.Location
)
-- 2. Now calculate the average number of visits for sold properties by location.
SELECT
    Location,
    AVG(VisitsNum * 1.0) AS AvgVisitToSaleRatio
FROM
    VisitsPerSoldProperty
GROUP BY
    Location
ORDER BY
    AvgVisitToSaleRatio DESC;
```

This query uses multiple Common Table Expressions (CTEs) to first identify all sold properties and their locations, then calculate how many visits each sold property received. Finally, it computes the average number of visits required for a sale in each location. The multiplication by 1.0 ensures that the average is calculated as a decimal rather than an integer, providing more precise results.

Understanding the average visit-to-sale ratio by location provides valuable insights into regional market efficiency and buyer behavior. Areas with low ratios (fewer visits needed per sale) may indicate highly desirable locations where buyers make decisions quickly, effective property presentation, or competitive pricing strategies. Conversely, areas with high ratios might suggest locations where buyers are more hesitant, properties that require more consideration, or pricing issues that delay purchase decisions. This information can guide location-specific marketing strategies, pricing approaches, and agent training to optimize conversion rates in different areas. It also helps set realistic expectations for how many showings might be

needed before securing a sale in various locations. The descending order presentation highlights areas with the highest visit requirements first, drawing attention to locations that might benefit from strategic interventions to improve conversion efficiency.

Additional Business Insights

Potential Customer Email List

This query identifies clients who are either first-time buyers or have not yet made a purchase, creating a targeted list for email marketing campaigns and customer outreach.

```
WITH ClientSaleCounts AS (
    SELECT
        ClientID,
        COUNT(SaleID) AS NumberOfSales
    FROM
        Sales
    GROUP BY
        ClientID
)
SELECT
    c.ClientID,
    CONCAT(c.FirstName, ' ', c.LastName) AS ClientName,
    c.Email,
    ISNULL(csc.NumberOfSales, 0) AS TotalSalesMade
FROM
    Clients AS c
LEFT JOIN
    ClientSaleCounts AS csc
    ON c.ClientID = csc.ClientID
WHERE
    csc.NumberOfSales = 1
    OR
    csc.NumberOfSales IS NULL
ORDER BY
    c.LastName, c.FirstName;
```

This query uses a Common Table Expression (CTE) to first count the number of sales associated with each client. It then joins this information with the Clients table to retrieve contact details, focusing specifically on clients who have made exactly one purchase (first-time buyers) or have not made any purchases yet (potential buyers). The ISNULL function ensures that clients with no sales are assigned a count of zero rather than NULL.

This targeted email list is a valuable asset for marketing campaigns and customer relationship management. First-time buyers represent an opportunity for follow-up communications to encourage repeat business, gather feedback on their purchase experience, or introduce them to additional services. Clients who have not yet made a purchase but have engaged with the company (through property visits or inquiries) represent potential conversion opportunities that could be nurtured through targeted marketing. By focusing on these specific client segments, the business can allocate marketing resources efficiently and develop tailored messaging that addresses the specific needs and concerns of each group. The alphabetical ordering by last

name and first name facilitates organized outreach efforts and ensures no potential customers are overlooked.

Not Visited Properties to Focus On

This query identifies properties that have not received any client visits, highlighting potential inventory issues or marketing opportunities that require attention.

```
SELECT
    p.PropertyID,
    p.PropertyType,
    p.Location
FROM
    Properties AS p
LEFT JOIN
    Visits AS v
    ON p.PropertyID= v.PropertyID
WHERE
    v.VisitID IS NULL
```

The query joins the Properties table with the Visits table using a LEFT JOIN to include all properties regardless of whether they have associated visits. It then filters for properties where no visit records exist, indicating that these properties have not been shown to any clients.

Identifying properties with no visits is crucial for inventory management and marketing strategy. These properties may be overlooked due to inadequate promotion, uncompetitive pricing, poor presentation, or location issues. Alternatively, they might be recent additions to the inventory that haven't had sufficient exposure yet. By focusing attention on these unvisited properties, the business can take proactive steps to increase their visibility and appeal. This might involve adjusting pricing, enhancing property descriptions or photographs, featuring them in marketing materials, or recommending them to agents for targeted client showings. Addressing these overlooked properties can help reduce inventory holding costs, improve portfolio turnover, and potentially uncover hidden gems that could meet specific client needs once properly promoted.

Month-over-Month Percentage Change in Sales

This query calculates the percentage change in total sales value from one month to the next, providing insights into short-term sales trends and performance fluctuations.

```
SELECT
    YEAR(SaleDate) As Year,
    MONTH(SaleDate) AS Month,
    SUM(SalePrice) AS Total_Sales,
    ROUND(
         (SUM(SalePrice) -
                          LAG(SUM(SalePrice)) OVER (ORDER BY YEAR(SaleDate),
MONTH(SaleDate)) ) * 100.0 /
                          NULLIF(LAG(SUM(SalePrice)) OVER (ORDER BY
YEAR(SaleDate), MONTH(SaleDate)), 0)
```

```
          , 2) AS MOM_per
FROM
    Sales
GROUP BY
    YEAR(SaleDate),
    MONTH(SaleDate)
ORDER BY
    YEAR(SaleDate),
    MONTH(SaleDate);
```

This query groups sales by year and month, calculates the total sales value for each month, and then computes the percentage change from the previous month. It uses the LAG window function to access the previous month's sales value, calculates the difference, and expresses it as a percentage of the previous month's value. The NULLIF function prevents division by zero in case there were no sales in the previous month, and the result is rounded to two decimal places for clarity.

Month-over-Month (MoM) percentage change is a critical metric for tracking short-term business performance and identifying trends or anomalies that might require attention. Positive percentages indicate growth compared to the previous month, while negative values signal contraction. This information helps management understand seasonal patterns, assess the impact of marketing campaigns or pricing changes, and make timely adjustments to strategies or tactics. The chronological ordering by year and month provides a clear timeline of performance evolution, making it easy to spot trends, cycles, or unusual fluctuations. This metric is particularly valuable for regular business reviews, allowing management to celebrate successes, address concerns, and set realistic expectations based on historical patterns of monthly performance variation.

Year-over-Year Percentage Change in Sales

This query calculates the percentage change in total sales value from one year to the next, providing insights into long-term business growth or contraction.

```
SELECT
    YEAR(SaleDate) AS Year,
    SUM(SalePrice) AS Total_Sales,
    ROUND(
         (SUM(SalePrice) -
                         LAG(SUM(SalePrice)) OVER (ORDER BY YEAR(SaleDate))) *
100.0 /
                         NULLIF(LAG(SUM(SalePrice)) OVER (ORDER BY
YEAR(SaleDate)), 0)
        , 2) AS YOY_per
FROM
    Sales
GROUP BY
    YEAR(SaleDate)
ORDER BY
    YEAR(SaleDate)
```

This query groups sales by year, calculates the total sales value for each year, and then computes the percentage change from the previous year. It uses the LAG window function to access the previous year's sales value, calculates the difference, and expresses it as a percentage of the previous year's value. The NULLIF function prevents division by zero in case there were no sales in the previous year, and the result is rounded to two decimal places for clarity.

Year-over-Year (YoY) percentage change is a fundamental metric for assessing long-term business health and growth trajectory. It smooths out seasonal variations that might distort monthly comparisons and provides a more strategic view of business performance. Positive percentages indicate annual growth, while negative values signal contraction compared to the previous year. This information is crucial for strategic planning, investor communications, and performance evaluation against long-term goals. The chronological ordering by year creates a clear timeline of business evolution, making it easy to identify multi-year trends or cycles. This metric helps management understand whether the business is consistently growing, plateauing, or declining over time, informing major decisions about expansion, investment, or restructuring. It also provides context for setting realistic annual targets based on historical growth patterns and market conditions.