SQL Propositions in Azure Data Studio

A Step-by-Step Explanation of Query Solutions

Proposition 1

Problem: Find the 10 most recent invoices

Query:

```
SELECT TOP 10 InvoiceID, CustomerID, InvoiceDate

FROM Sales.Invoices

ORDER BY InvoiceDate DESC;
```

- **Explanation:**
  - Retrieves the 10 most recent invoices by sorting them by date.
  - Helps track the latest billing activities.

Proposition 2

Problem: Top 5 customers by credit limit

Query:

```
SELECT TOP 5 CustomerID, CustomerName, CreditLimit

FROM Sales.Customers

ORDER BY CreditLimit DESC;
```

- **Explanation:**
  - Identifies high-value customers by sorting by their spending limits.
  - Focuses on customers with the highest credit

Proposition 3

- **Problem:** Top 5 cities by population

Query:

```
SELECT TOP 5 CityName, LatestRecordedPopulation

FROM Application.Cities

ORDER BY LatestRecordedPopulation DESC;
```

- **Explanation:**
  - Lists the most populated cities.
  - Useful for understanding market size and population trends.

Proposition 4
Problem: Customers with a credit limit greater than 3000

Query:

```
SELECT CustomerName, CreditLimit

FROM Sales.Customers

WHERE CreditLimit > 3000

ORDER BY CreditLimit DESC;
```

- **Explanation:**
  - Filters customers by higher credit limits.
  - Focuses on key accounts with significant purchasing power.

Proposition 5

- **Problem:** Total invoices per customer

Query:

```
SELECT CustomerID, COUNT(InvoiceID) AS TotalInvoices

FROM Sales.Invoices

GROUP BY CustomerID

ORDER BY TotalInvoices DESC;
```

- **Explanation:**
  - Groups invoices by customer and counts them.
  - Shows which customers are most active.

Proposition 6

- **Problem: Find the top 5 stock items with the highest unit prices.**

  -
  - **Query**
  -
  - **SELECT TOP 5 StockItemName, UnitPrice**
  - **FROM Warehouse.StockItems**
  - **ORDER BY UnitPrice DESC;**

  - **We want to retrieve the stock items with the highest unit prices, limited to the top 5.**
  -

Proposition 7
Problem: Top 5 customers with the most invoices

Query:

```
SELECT TOP 5 CustomerID, COUNT(InvoiceID) AS TotalInvoices

FROM Sales.Invoices

GROUP BY CustomerID

ORDER BY TotalInvoices DESC;
```

- **Explanation:**
  - Shows customers with the most transactions.
  - Useful for identifying loyal or frequent buyers.

Proposition 8
Problem: 10 most recent orders

Query:

```
SELECT TOP 10 OrderID, CustomerID, OrderDate
FROM Sales.Orders
ORDER BY OrderDate DESC;
```

Explanation:

Tracks the 10 most recent orders by sorting by date.

Helps analyze the latest sales activities.

Proposition 9

- **Problem:** Average credit limit by buying group

Query:

```
SELECT BuyingGroupID, AVG(CreditLimit) AS AverageCreditLimit
FROM Sales.Customers
GROUP BY BuyingGroupID
ORDER BY AverageCreditLimit DESC;
```

Explanation:

Calculates the average credit limit per buying group.

Provides insights into group purchasing power.

Proposition 10
Problem: Total orders per year

Query:

```
SELECT YEAR(OrderDate) AS OrderYear, COUNT(OrderID) AS TotalOrders
FROM Sales.Orders
GROUP BY YEAR(OrderDate)
ORDER BY OrderYear DESC;
```

Explanation:

Groups and counts orders by year.

Shows yearly trends in order activity.

Conclusion
Key Takeaways:

These 10 SQL queries provide actionable insights into customer behavior, sales, and market analysis.

Techniques like filtering, grouping, and sorting make data analysis efficient.

Queries were tested in Azure Data Studio for accuracy.