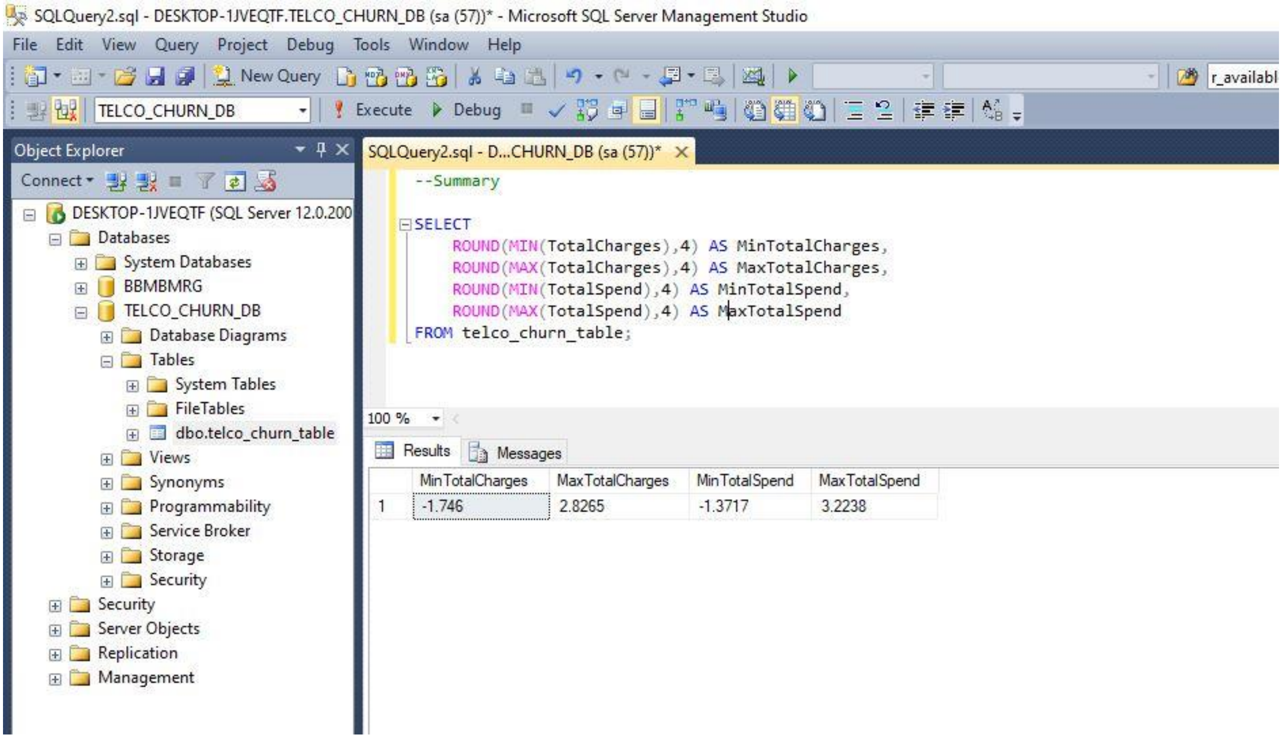
Data Transformation and Storage

We applied transformations to ensure the data is clean, structured, and suitable for machine learning models. The following transformations were applied on the dataset before saving it to database.

- “TotalCharges” is stored as an **object (string)** instead of a numeric value. So, we converted it to a numeric field.
- We are imputing missing values in Numerical columns with mean and missing values in Categorical columns with mode.
- Checked for duplicate rows and deleting if any exists.
- Converted Tenure to Categories since Tenure is numerical, but customer retention is often linked to different periods (short-term, medium-term, long-term customers).
- Created a featue “TotalSpend” to know customer spending behaviour
- Created a binary column “HasInternet” so that we can get useful information
- Created a column “NumServices” which counts for the number of services the user has opted for.
- Standardized the numerical features to bring them into same scale otherwise it might impact the ML model
- Encoded all the categorical columns with label encoder.

Sample queries to retrieve transformed data

1. Summary

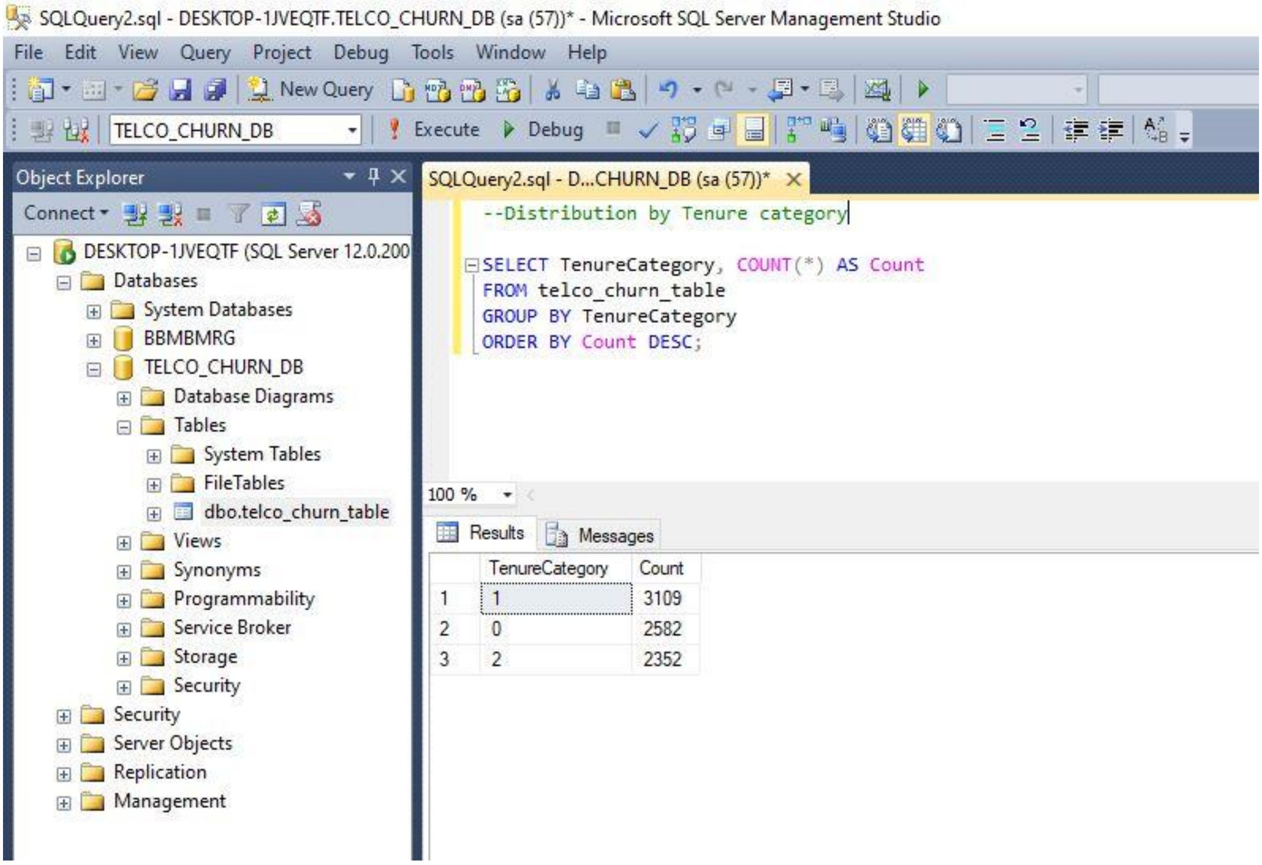


```
--Summary

SELECT
    ROUND(MIN(TotalCharges),4) AS MinTotalCharges,
    ROUND(MAX(TotalCharges),4) AS MaxTotalCharges,
    ROUND(MIN(TotalSpend),4) AS MinTotalSpend,
    ROUND(MAX(TotalSpend),4) AS MaxTotalSpend
FROM telco_churn_table;
```

	MinTotalCharges	MaxTotalCharges	MinTotalSpend	MaxTotalSpend
1	-1.746	2.8265	-1.3717	3.2238

2. Distribution by Tenure Category

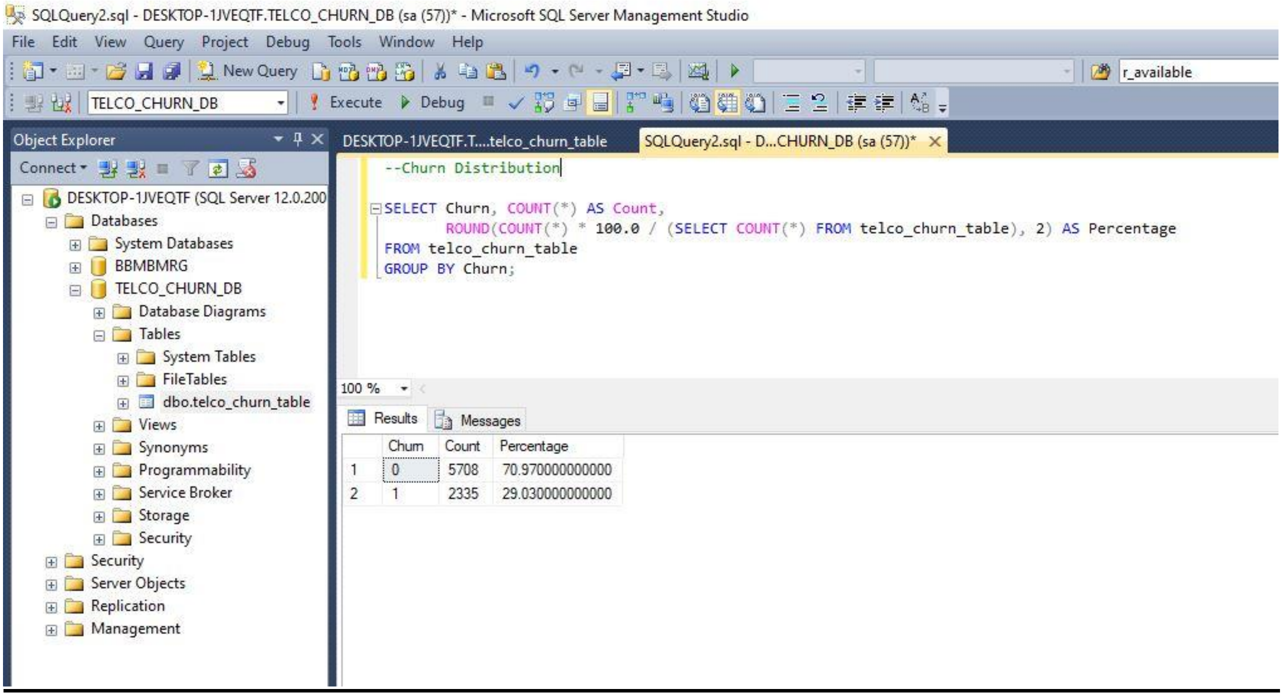


```
--Distribution by Tenure category

SELECT TenureCategory, COUNT(*) AS Count
FROM telco_churn_table
GROUP BY TenureCategory
ORDER BY Count DESC;
```

	TenureCategory	Count
1	1	3109
2	0	2582
3	2	2352

3. Churn Distribution

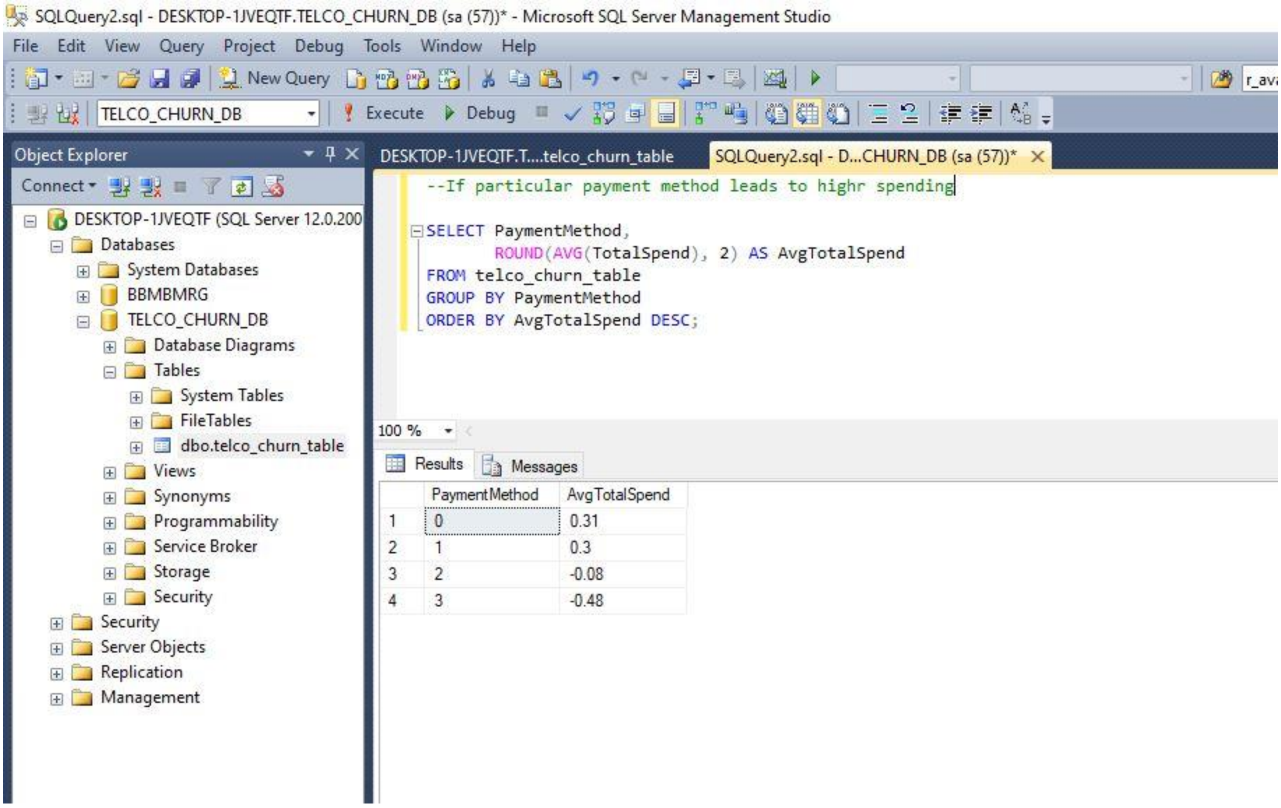


```sql
--Churn Distribution

SELECT Churn, COUNT(*) AS Count,
       ROUND(COUNT(*) * 100.0 / (SELECT COUNT(*) FROM telco_churn_table), 2) AS Percentage
FROM telco_churn_table
GROUP BY Churn;
```

	Churn	Count	Percentage
1	0	5708	70.970000000000
2	1	2335	29.030000000000

4. To check if a particular payment method leads to leads to higher spending



```sql
--If particular payment method leads to highr spending

SELECT PaymentMethod,
       ROUND(AVG(TotalSpend), 2) AS AvgTotalSpend
FROM telco_churn_table
GROUP BY PaymentMethod
ORDER BY AvgTotalSpend DESC;
```

	PaymentMethod	AvgTotalSpend
1	0	0.31
2	1	0.3
3	2	-0.08
4	3	-0.48