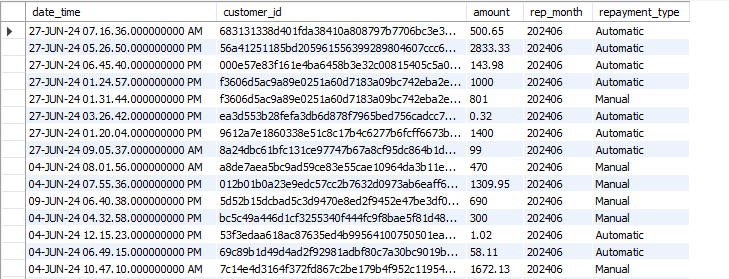
During the initial assessment of the bank loan dataset, it was identified that several columns required cleaning to ensure data quality and consistency. In particular:

* The date\_time column contained timestamp values in inconsistent formats that were not readily suitable for temporal analysis.
* The rep\_month field required normalization to ensure uniformity in monthly reporting.

To address these issues, SQL-based data transformation techniques were applied. The cleaning process involved:

* Parsing and reformatting the date\_time values into a standardized datetime format.
* Extracting and formatting the rep\_month field to ensure accurate grouping and aggregation for time-based analyses.

This cleaning phase ensured the dataset was ready for further exploratory data analysis, trend identification, and dashboard development.

**SQL-Based Date and Time Field Normalization**

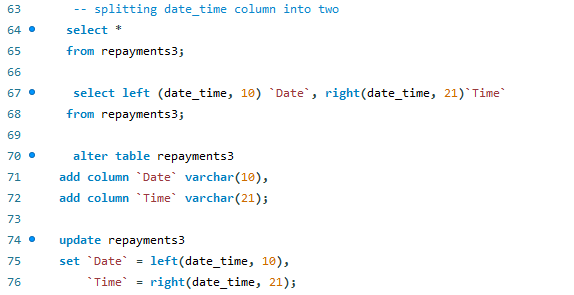
To improve the usability of the date\_time column in the repayments3 table, a transformation was performed to separate it into two distinct fields: Date and Time. The steps included:

1. **Reviewing the Table:**  
   A simple SELECT \* query was used to inspect the current structure and contents of the repayments3 table.
2. **Extracting Date and Time:**  
   A query using the LEFT() and RIGHT() string functions was written to extract:

* The first 10 characters of date\_time as the Date (typically in YYYY-MM-DD format).
* The last 21 characters as the Time (which includes the timestamp portion).

1. **Schema Modification:**  
   Two new columns, Date and Time, were added to the table using an ALTER TABLE statement, with appropriate lengths to store the extracted values.
2. **Populating the New Columns:**  
   An UPDATE statement was executed to fill the new Date and Time columns by applying the LEFT() and RIGHT() functions to the existing date\_time column.

This process effectively normalized the datetime information, making it easier to perform date-based filtering, aggregation, and reporting in subsequent analysis steps.

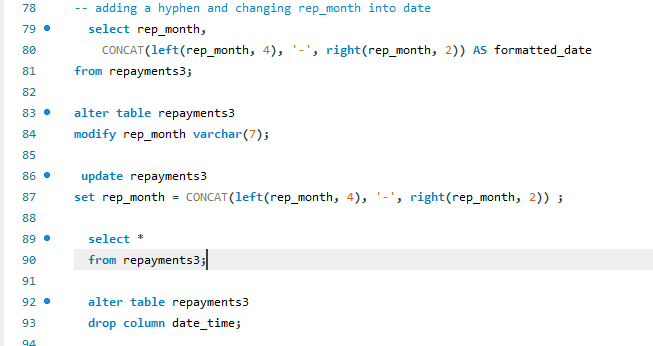


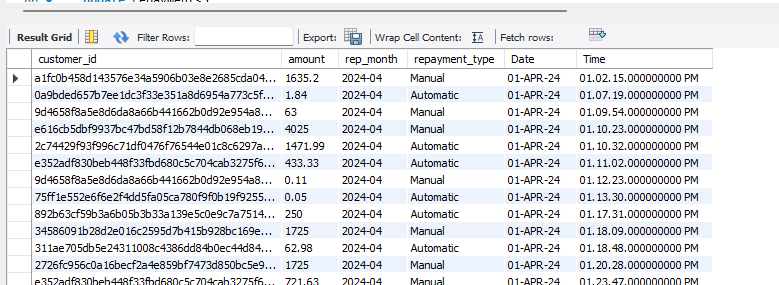
**Standardizing rep\_month Format and Cleaning Redundant Columns**

To enhance consistency and enable proper date-based aggregation, the rep\_month field in the repayments3 table was reformatted and the now-unnecessary date\_time column was removed. The steps taken include:

1. **Reformatting rep\_month:**  
   The rep\_month values were initially stored in a compact YYYYMM format. To improve readability and compatibility with date functions, a query was used to transform this into the standardized YYYY-MM format using the CONCAT(), LEFT(), and RIGHT() functions.
2. **Altering Column Type:**  
   The data type of the rep\_month column was explicitly set to VARCHAR(7) using the ALTER TABLE statement to ensure that the new format (e.g., 2024-04) fits correctly and is treated as a string for formatting purposes.
3. **Updating Values:**  
   An UPDATE statement was executed to overwrite the existing values in rep\_month with the newly formatted YYYY-MM strings.
4. **Dropping Redundant Column:**  
   Once the necessary date components were extracted and standardized, the original date\_time column—which was no longer required—was removed from the table using the ALTER TABLE ... DROP COLUMN command. This helped reduce redundancy and streamline the dataset.

These transformations were part of the overall data cleaning process to prepare the dataset for accurate time-series analysis and dashboard reporting.



**The final product**

The time column was further corrected in PowerBI on my Bank Loan dashboard to enable an easier read on the visualization