Why we should use the relational database?

Once we have defined information on all the necessary structures, it is possible to structure a relational model that normalizes the data, defining tables and relationships between them.
Another type would be Non-Relational Database (NoSQL) but we think that in this case, it won't be the best option since we won't be working with something that will be constantly under development (dynamic) or something that needs horizontal scalability.
That said, SQL database is the best option as it gives us a high degree of restrictions and data security. Let's not forget to mention ACID (Atomicity, Consistency, Isolation and Durability) transactions that allow for fast transactions and data integrity.

Dictionary

Here you can find the most relevant details about our database such as the definition of some attributes and their respective meaning.

Ficha_Tecnica

- N: it's a number type, and the meaning is Nitrogen;
- P2O5: it's a number type, and the meaning is Phosphorus Pentoxide;
- K2O: it's a number type, and the meaning is Potassium Oxide;
- CaO: it's a number type, and the meaning is Calcium Oxide;
- MgO: it's a number type, and the meaning is Magnisium Oxide;
- TOC: it's a number type, and the meaning is Total Organic Carbon;
- pH: it's a number type, and the meaning is the Level of Acidity.

Cliente

- Plafond: it's the maximum limit of credit of a client, meaning that a client can't have a total value of orders with payment pendent higher that his plafond;
- Num_incidentes: it's the number of times that the client didn't it's order by the date of payment;
- Nivel: it's the value of the client to the business, it's measured by the number of incidents in the last year and the amount of money it's orders costed.

We assume that the others tables and respective attributes are well-named and explicit in their meaning.