**Juan Manuel Gonzalez**

**Technical Approach & Design Choices**

**Database and SQL Client Selection**

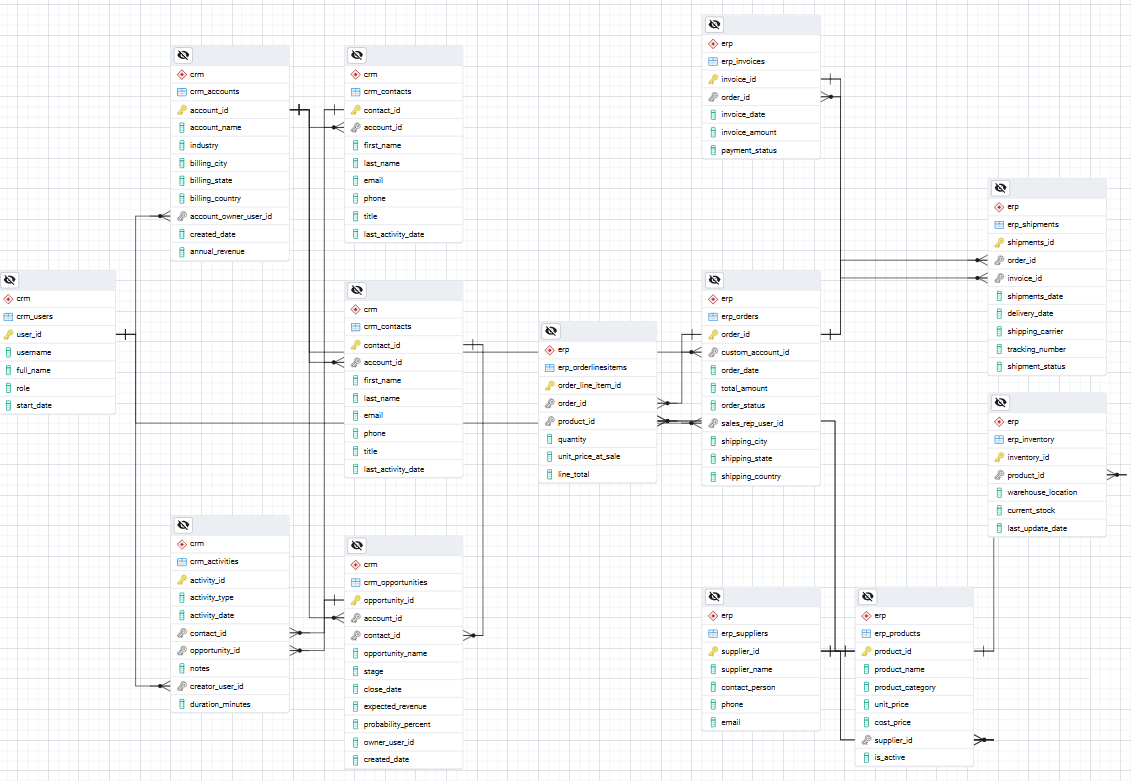
For this project, I chose PostgreSQL as the local database due to its stability, powerful support for complex queries, and solid performance when handling structured data. Its open-source nature and strong presence in the data warehousing ecosystem also made it a practical, scalable choice. I used pgAdmin as the SQL client because of its intuitive interface, built-in support for PostgreSQL, and a capable SQL editor that helped streamline the development and testing of views and stored procedures.

To keep the data structure organized, I worked with two separate schemas: one focused on CRM data (like customer profiles and acquisition channels), and another on ERP-related data (such as products, orders, and shipping). This helped separate business logic clearly and made maintenance easier. I used primary and foreign keys to enforce referential integrity and improve query efficiency. For better performance, I opted for fixed-length VARCHAR where applicable and removed low-value or incomplete records that didn’t affect the overall analysis.

To generate the unified dataset required in Section 1, I created a stored procedure, which allowed for easier automation and future updates. For performance optimization, I used Common Table Expressions (CTEs) combined with DISTINCT to reduce data volume before applying joins, which minimized memory usage and processing load.

During the process, I resolved a couple of data quality issues: one involving case sensitivity in the Category field, which needed explicit handling to ensure consistency, and another related to a boolean field used to identify active suppliers. I made sure only valid, relevant supplier data was included.

The data model followed a star schema design, with fact tables (Orders, OrderDetails) at the center, connected to key dimension tables (Customers, Products, Suppliers, AcquisitionChannels, Dates). This structure simplified aggregation and filtering across business dimensions and provided a solid foundation for both reporting and dashboarding.



**Jaspersoft Report Design**

In Jaspersoft Studio, I structured the report for clarity and modularity. The main body of the report grouped data by Product and Category, using variables for summarized values and applying filtering logic directly at the SQL level through user-defined parameters (e.g., Date, Category). These parameters were passed into the dataset queries to ensure dynamic reporting.

I added a Crosstab component connected to the same database, configured to inherit the filtering parameters. This ensured that all components remained in sync. Additionally, I allocated space for a subreport, which was imported and linked using the same parameters. The subreport ran a specialized query focused on extracting the Top Clients, delivering more granular insights while staying consistent with the main report’s context.

To make key data stand out, I applied conditional formatting in the subreport to highlight the Top 5 clients using a dynamic color expression, making it easier for users to spot the highest performers at a glance.

**Interactive Dashboard Design (Power BI)**

I chose Power BI Desktop for the interactive dashboard because I already had experience with it, and it provides powerful modeling and visualization tools. To communicate insights clearly, I used bar charts to compare performance across customer acquisition channels and line charts to display time-based trends. For more detailed breakdowns—like customer segments or geographic areas—I incorporated tables to better handle large sets of categorical data.

Each page included consistent KPI cards (e.g., Total Revenue, Profit, Customer Count) to keep key metrics front and center. I also added global filters for essential dimensions like Date, Category, Region, and Acquisition Channel to enable dynamic data exploration across the entire dashboard.

On the second page, I included a more focused view showing Top Products and Top Clients, using sortable tables with custom DAX measures like Profit, Total Sales, and Quantity. Finally, I added a detailed operational table for in-depth analysis or auditing, giving users access to raw-level data if needed.

**Challenges Faced & Solutions**

One of the biggest challenges I faced was working with Jaspersoft Studio, a tool I had never used before. Although its interface felt generally intuitive, the way each component manages its own dataset was confusing at first. I ran into issues where filters weren’t behaving as expected or weren’t passed properly between components, which forced me to rebuild certain parts of the report.

A particularly tricky part was ensuring that parameters like Date and Category were correctly passed between the main report, the Crosstab, and the subreport. Aligning all those components took several iterations of testing and adjustment.

To overcome these challenges, I relied on a mix of official documentation, community forums, and hands-on video tutorials. Practical examples helped me understand how to configure datasets and parameter passing correctly. That process not only solved the immediate issues but also gave me the confidence to build a complete, consistent, and professional report from scratch.