

**KGSB Clothing**

**Knoelle Grassi & Samantha Bentley**

kgrassi2693 & sbentley4059

kgrassi2693\_kgsb\_clothing\_db\_project

**Revolutionizing KGSB Clothing: Solving Inventory and Data Management Challenges**

**KGSB Clothing** seeks to build a database to manage their data. The *current challenge* lies in the absence of a robust database system to efficiently manage inventory, track customer information, and ensure data integrity. To *address this*, we propose implementing a database solution that streamlines inventory management, maintains detailed customer records, and ensures data integrity through a robust design. This solution will facilitate efficient data access, enabling quick retrieval of information for order processing, reporting, and analytics, ultimately enhancing the overall functionality and customer experience of the platform.

**Optimizing Operations: Solving Inventory and Customer Management Challenges**

Our company’s **problems** include inventory management, managing customer information, data integrity, and efficient data access. We need an efficient way to track the inventory of our store. This includes maintaining accurate records of product quantities, restocking, and handling returns. We also need to manage customer data. We must keep track of customer profiles, purchase history, and other relevant information. We must ensure data consistency and prevent duplicate entries to ensure business operations run smoothly. The store’s database should also allow quick and easy information retrieval for order processing, reporting, and analytics purposes.

Our **objectives** are to create a product catalog, track inventory, create a customer catalog, order processing, and create reports on our store. The product catalog should show details about every product. It requires the name, description, price, category, and availability of each product. The inventory tracking should maintain real-time inventory records. Whenever a product is sold or restocked, the inventory should be updated accordingly. The customer catalog should keep track of customer information. It should maintain personal details like name, contact information, and addresses of every customer along with purchase history with orders and products bought. Through order processing the database should keep track of incoming inventory and who the incoming inventory is from. The database should support various reports such as sales summaries, stock levels, and customer information.

Our **constraints** are budget, scalability, security, performance, and readability. The database needs to be built within budget constraints. It also should accommodate future growth without major overhauls. There should also be proper access controls and encryption to protect Customer data. The database should also be able to perform well even during peak times. The database should be designed where it’s easy to read and maintain by anyone who requires access.

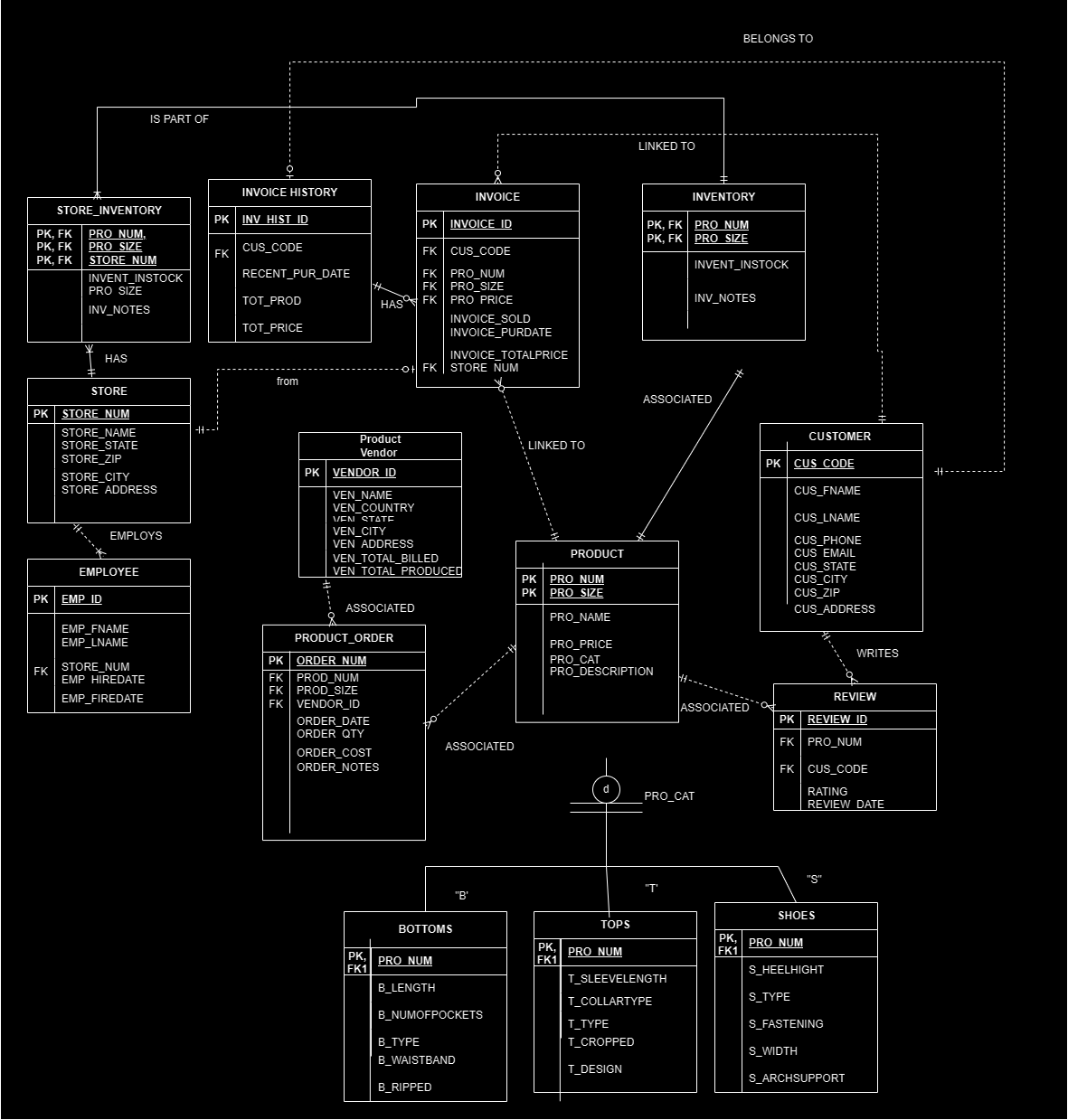
**Scope, Boundaries, and Business Rules:**

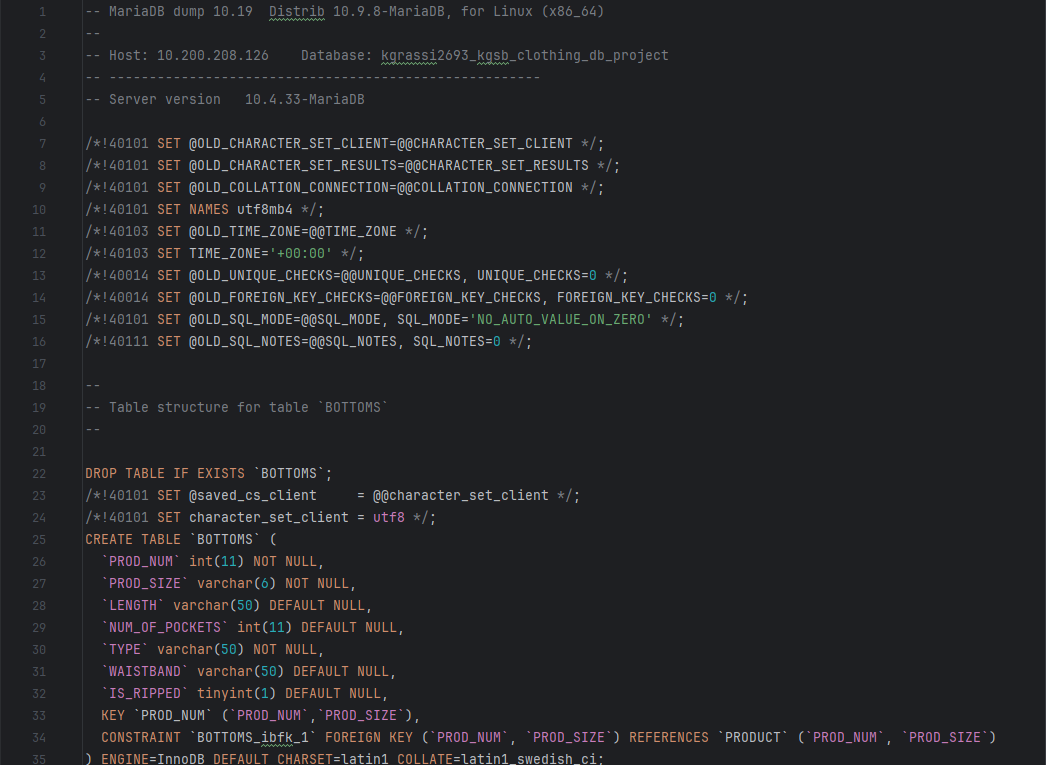
Our **Scope** involves Inventory Management, Customer Information Management, Data Integrity, and Efficient Data Access. Inventory management should track product quantities, restocking, and handling returns. Customer Information Management should maintain customer profiles, purchase history, and other relevant information for the customer. Data Integrity should ensure consistency and prevent duplicate entries. Efficient Data access should have quick and easy access to get information for order processing, reporting, and analytics.

We should have entities for Product, Inventory, Invoice, Customer, Product Vendor, Invoice History, Review, Employee, Stores, Store Inventory, and Product Order.

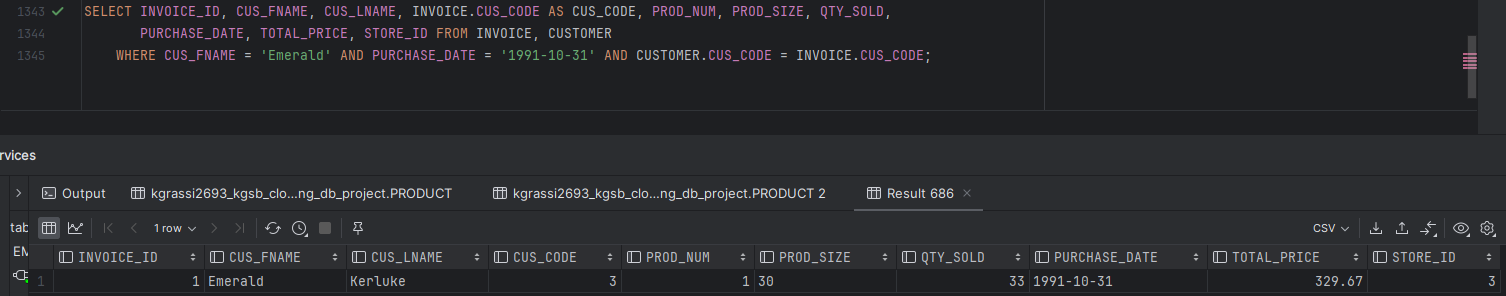
The **Product** entities should have subtypes of *bottoms*, *tops*, and *shoes*. It should have attributes of product number, product name, product price, product category, product size, and product description. The product category should be the subtype discriminator for the subtypes. *Bottoms* should have attributes of length, number of pockets, type, waistband, and if it’s ripped. *Tops* should have attributes of sleeve length, collar type, type, if it’s cropped, and design. *Shoes* should have attributes of heel height, shoe type, fastenings, width, and arch support. The **Inventory** entities should have attributes for product number and product size from product table, quantity in stock, and inventory notes. The **Invoice** entities should have attributes for customer code from customer, product number and product size from product, number of products sold, purchase date, product price from product, total price which is derived from qty sold and product price, invoice number, and store number. The **Invoice History** entities should have attributes customer code from Invoice, total products bought, total price, and invoice history id. The **Customer** entities should have attributes for customer first name, customer last name, customer phone number, customer email, customer state, customer city, customer zip code, customer street number, and customer code. The **product vendor** entities should have attributes for vendor number, vendor name, vendor country, vendor state, vendor city, vendor address, vendor total billed, and vendor total produced. The **Product Order** entities should have attributes for order number, product number and product size from product, vendor number from product vendor, order date, order quantity, order cost, and order notes. The **Review** entities should have attributes for review ID, product number and product size from product, customer code from customer, rating, and review date. The **Employee** entities should have attributes for employee number, employee first name, employee last name, employee hire date, employee fire date, and store number from the store they’re employed at from store. The **Store** entities should have attributes for store number, store name, store state, store ZIP code, store state, store city, and store address. Store Inventory

Each Product can be categorized as Bottoms, Tops, or Shoes. Each Inventory record is associated with one Product. Each Invoice is linked to one Customer and one product. Each Invoice History record is linked to one Customer. Each product is associated with many product orders. Each product order is associated with one product vendor. Each Review is associated with one Product and written by one Customer. Each Employee works at one Store. Each Store can have multiple Store Inventory records, each representing a stocked Product. Each invoice is from a store.

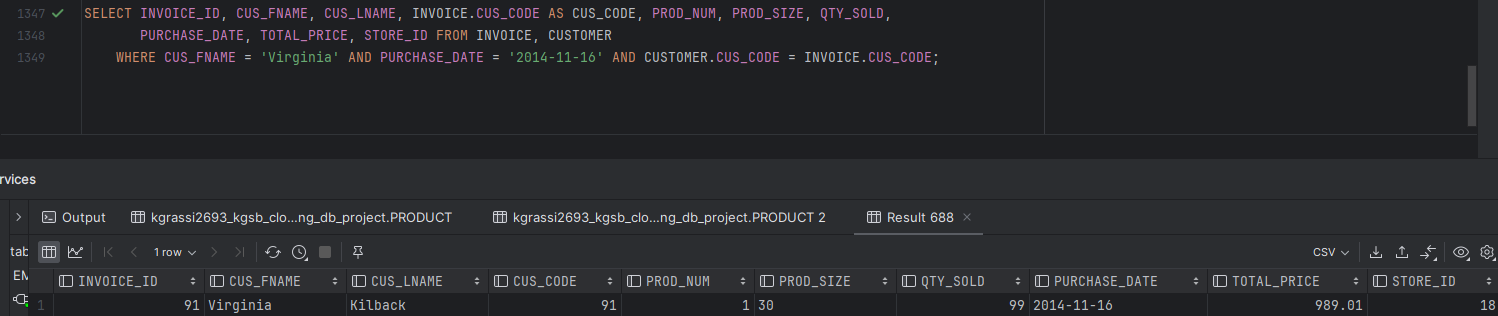




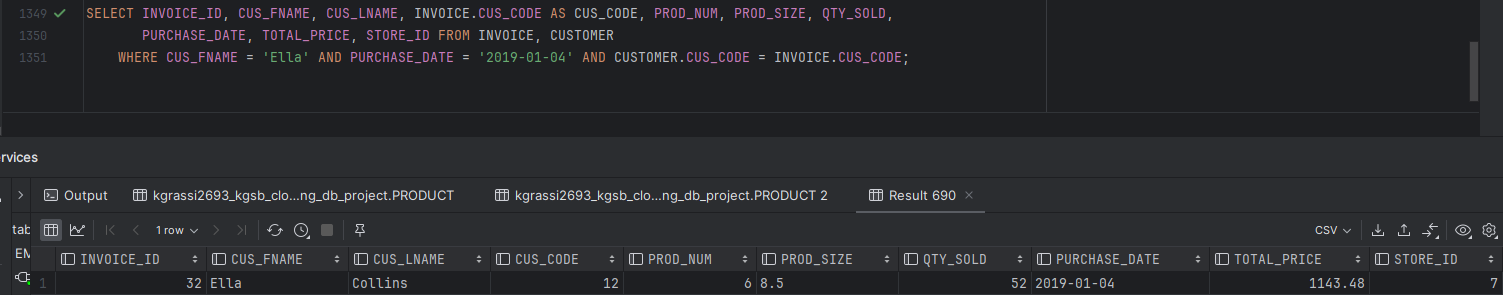
Invoice based on the customer with the first name Emerald and the purchase date 1991-10-31.



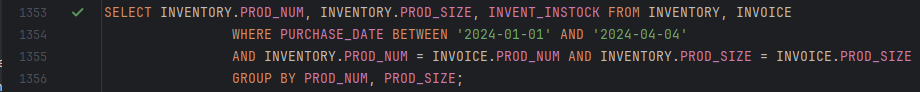
Invoice based on the customer with the first name Virginia and the purchase date 2014-11-16.



Invoice based on the customer with the first name Ella and the purchase date 2019-01-04.

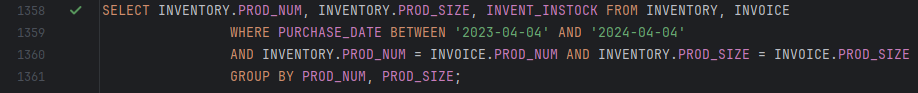


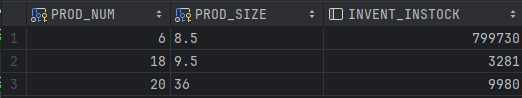
The current inventory of products sold within this year so far, 2024-01-01 to 2024-04-04.



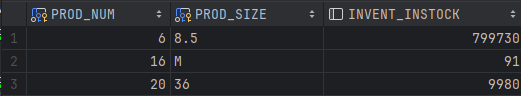
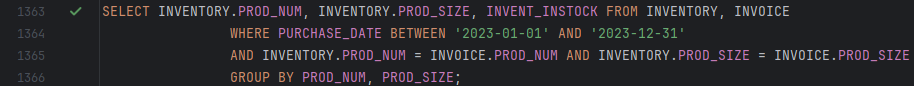


The current inventory of products sold in a year, 2023-04-04 to 2024-04-04.

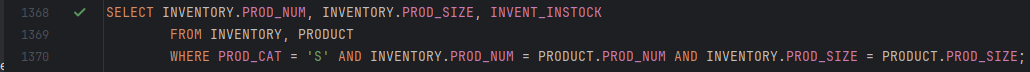


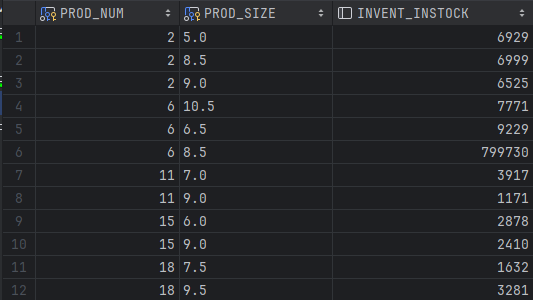


The current inventory of products sold in a calendar year, 2023-01-01 to 2023-12-31.

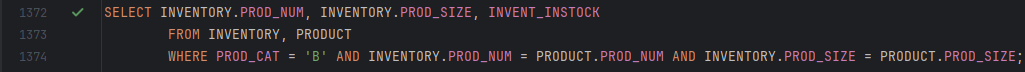


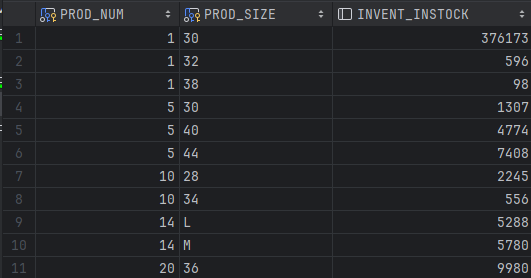
Inventory of all products that are the type Shoes (‘S’).



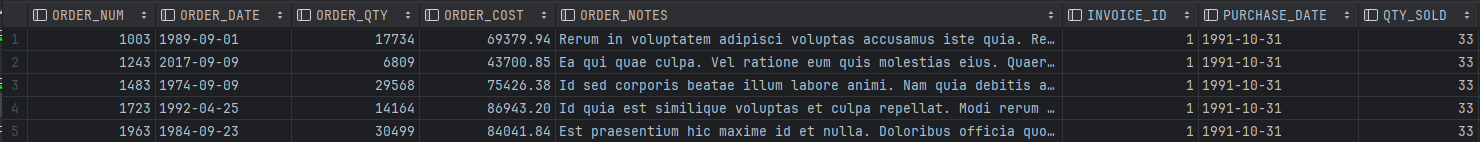
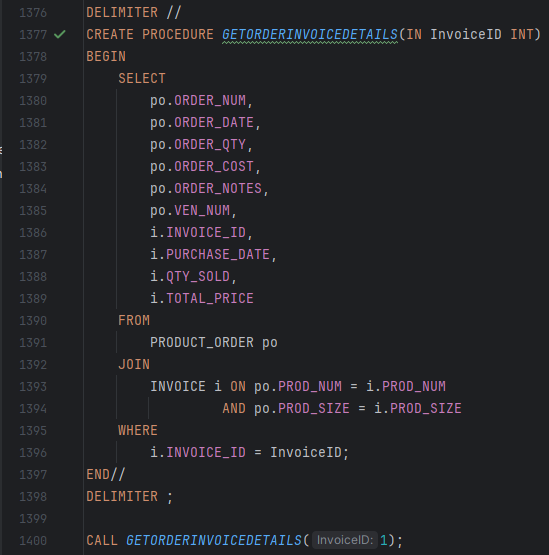


Inventory of all products that are the type Bottoms (‘B').



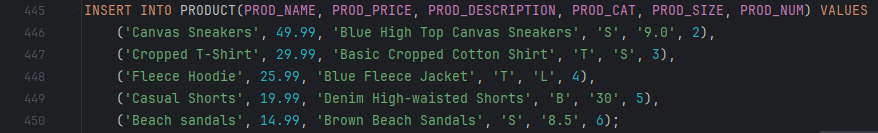


A stored procedure that allows a user to pass in an Invoice ID that shows all product orders associated with the product in the invoice.

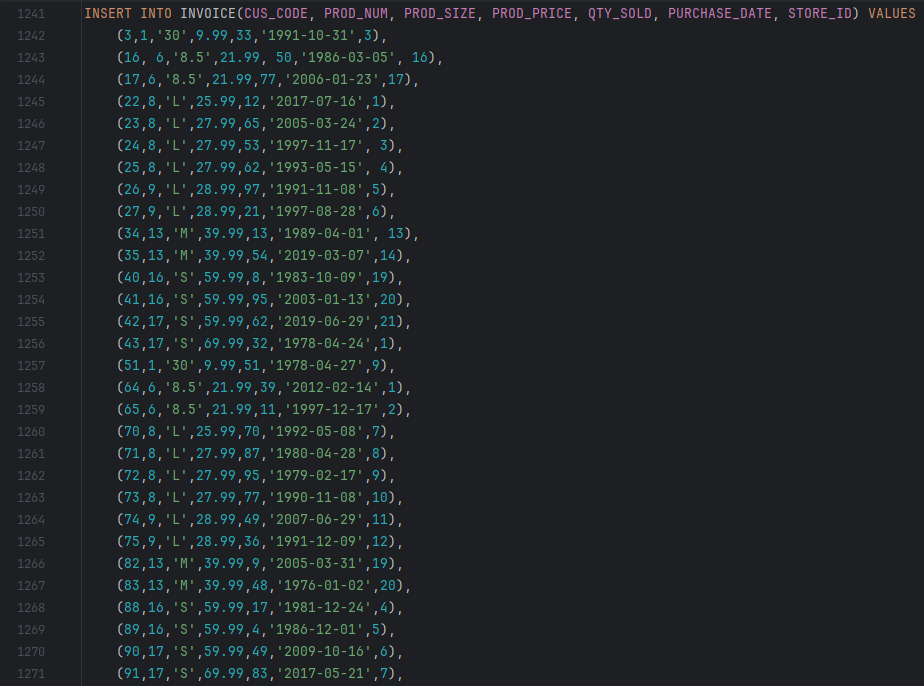


A singular product of Blue Jeans being entered into the product table.

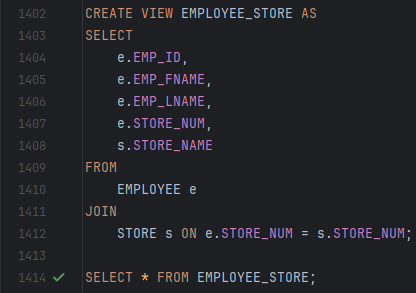
A group of products being entered into the product table.

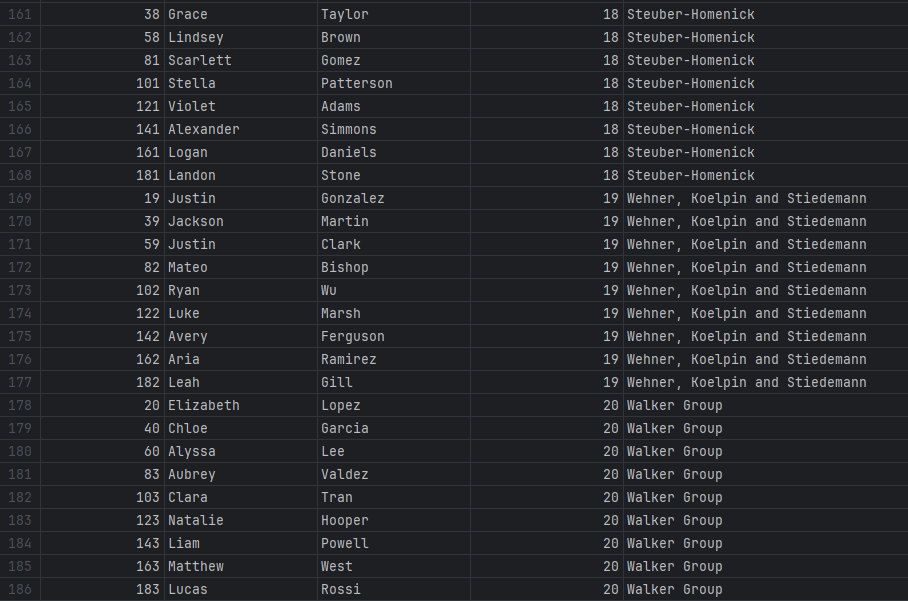
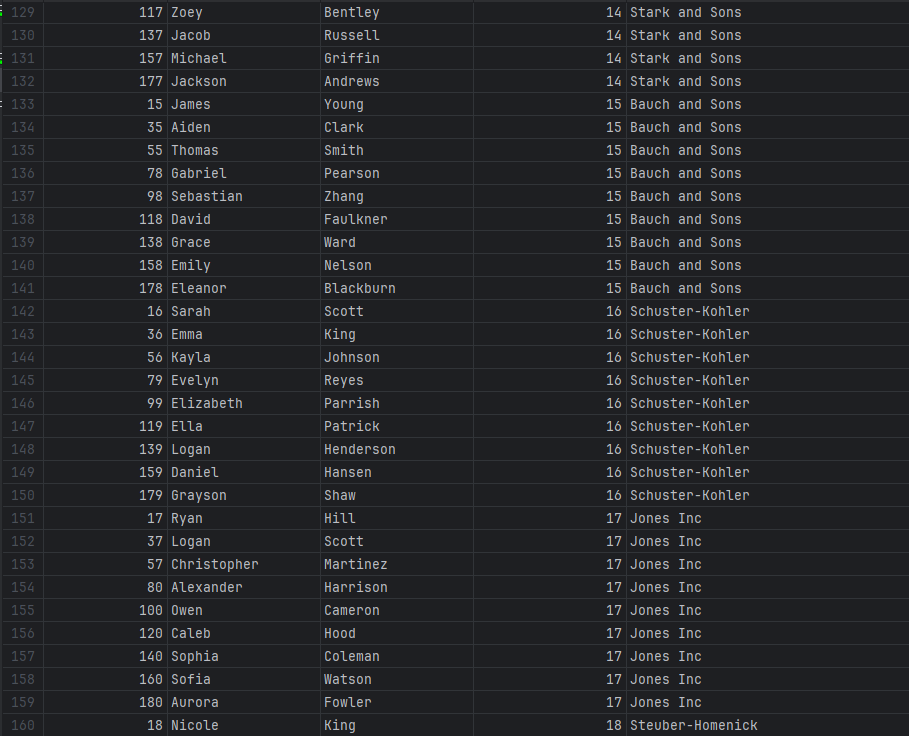
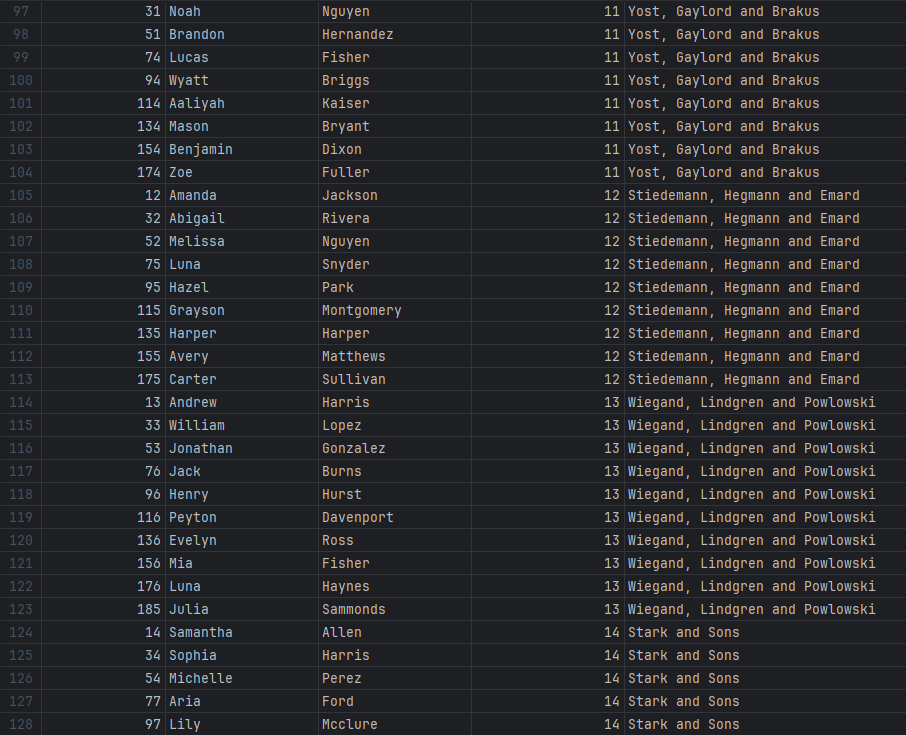
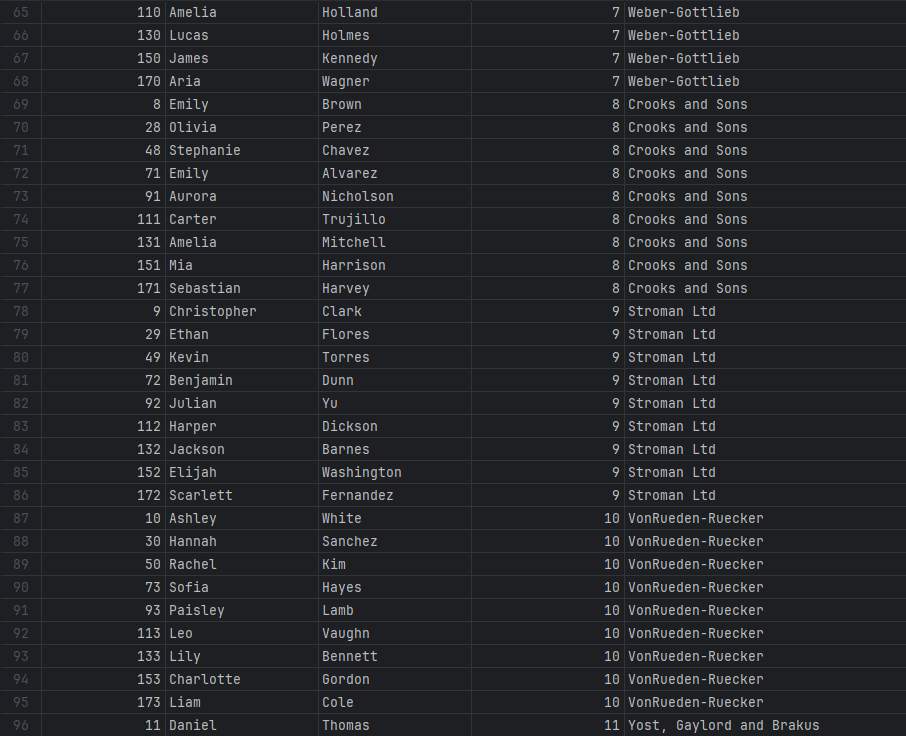
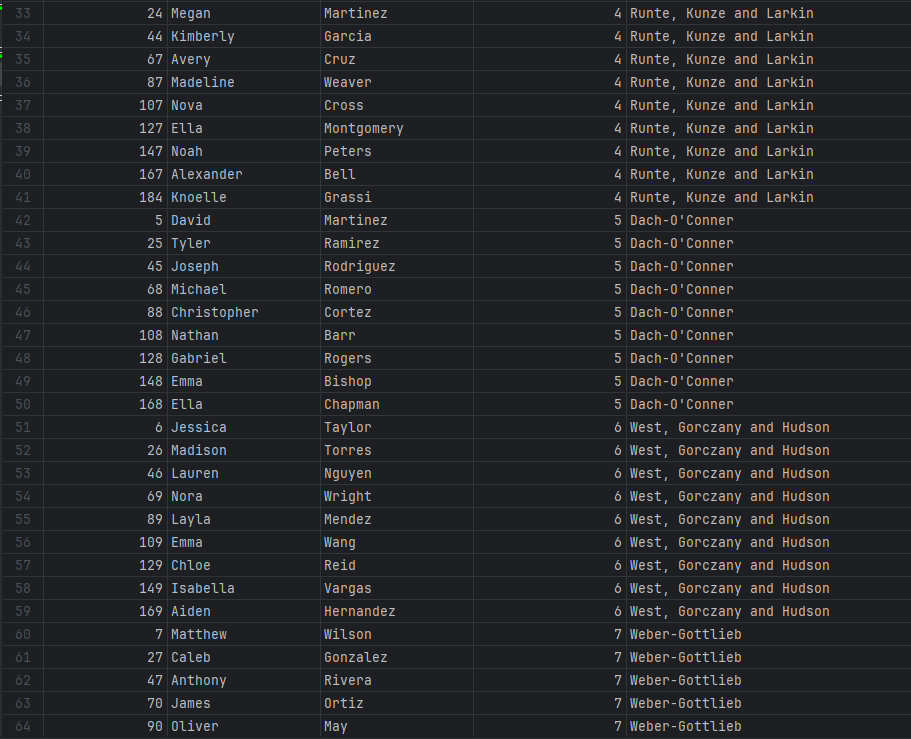
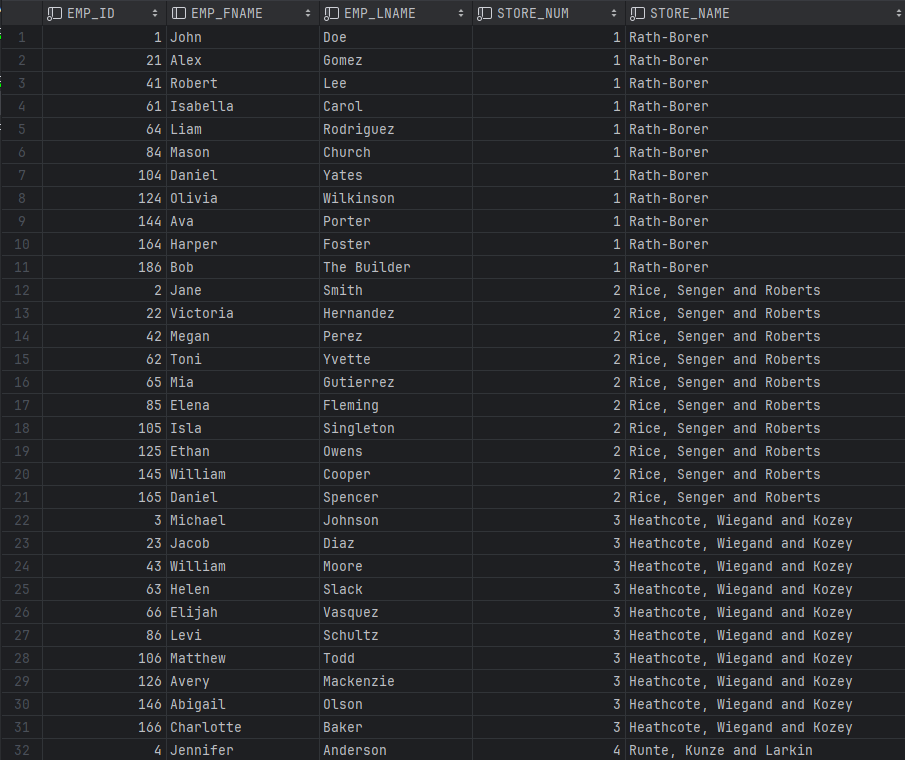


A group of invoices being inserted into the invoice table.

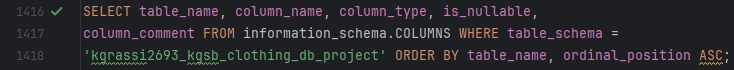


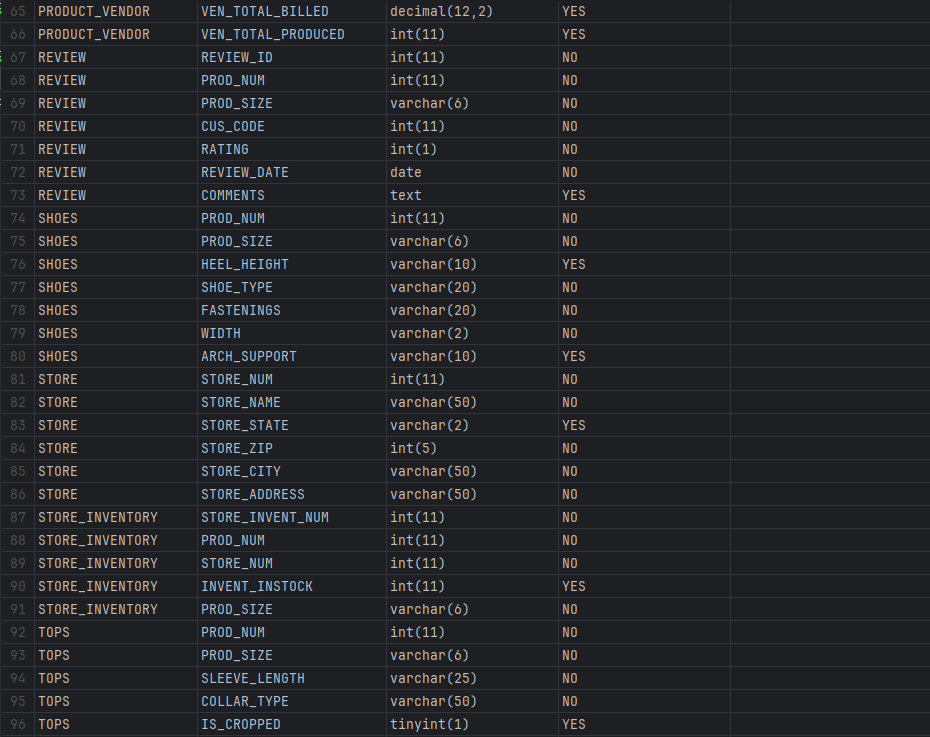
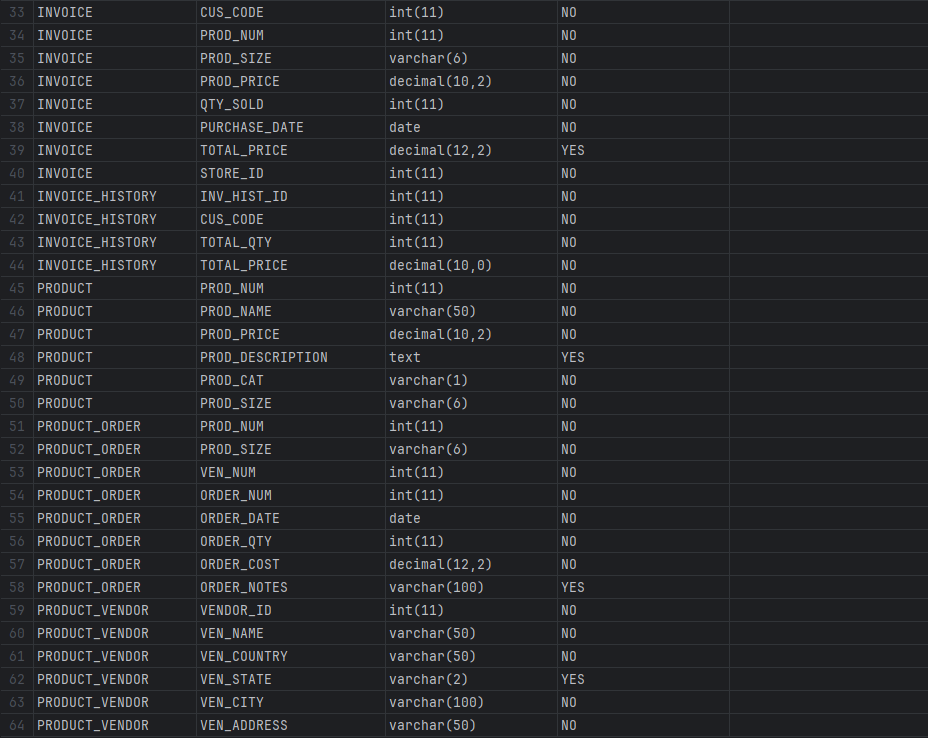
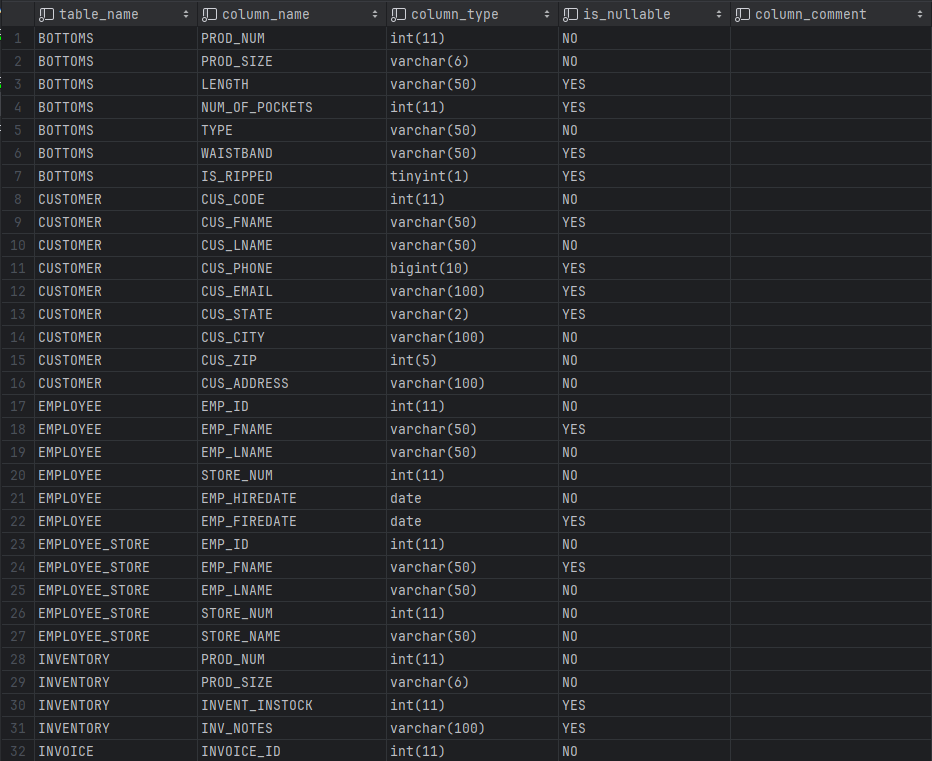
Output of a virtual table showing all employees and their stores.





The data dictionary of all the tables in the database.





For KGSB Clothing, we proposed a database that focused on improving inventory management, customer information tracking, and ensuring data integrity. We implemented a robust database system that adds inventory management processes, accurate customer records, and enhanced data accessibility.

Our database solution has various key components, including a centralized product catalog, real-time inventory tracking, detailed customer profiles, order processing mechanisms, and comprehensive reporting functionalities. The product catalog provides detailed information about each product and facilitates better inventory management and order processing. Real-time inventory tracking will ensure accurate stock levels, enabling proactive restocking and efficient handling of returns. Furthermore, the database solution prioritizes data integrity and security, ensuring that all information remains consistent and accurate. Overall, the proposed database solution will empower KGSB Clothing to optimize their operations, improve customer satisfaction, and drive sustainable growth in the competitive retail market.

In conclusion, the database solution for KGSB Clothing offers a comprehensive approach to addressing their inventory and data management challenges. With our database, KGSB Clothing can streamline operations, enhance data accuracy and unlock new opportunities for business growth and success. With a focus on efficiency, reliability, and data security, the proposed solution aligns with KGSB Clothing’s objectives of optimizing operations and delivering exceptional customer experiences in the ever-evolving retail landscape.

|  |  |  |
| --- | --- | --- |
|  | Knoelle Grassi | Samatha Bentley |
| Editing & Formatting |  | × |
| Typing the Database | × |  |
| Creating the PowerPoint |  | × |
| Creating the EERD | × | × |
| Page 1 | × |  |
| Page 2 |  | × |
| Page 3 | × |  |
| Page 4-13 | × |  |
| Creating the Data for the Database | × | × |