Project Requirements:

– The stone sales company employs several individuals organized by roles (Admin, Employee, and Customer). Each **user** has a unique user **ID**, a **username**, and a **password** used for system login. We store each user's full name, **phone number**, **address**, **email** (optional), **role**. Each user has exactly one role, and customer/employee details are stored in separate tables linked to the Users table.

– The company maintains a catalog of **stones**, each identified by a unique stone **ID**. For each stone, we store the **name**, **type** (e.g., Marble, Limestone, Jerusalem Stone), **size**, **price per unit**, **quantity in stock**, and optionally an **image**. Stones are managed by the admin, who can add, update, or delete stone records. A single stone can be included in many different orders over time (through Order_Details), and an order can include many different stones.

– Each **order** in the system has a unique order **ID** and is linked to exactly one customer (who places the order). An order may also be linked to one employee (who handles it), but **Employee_ID** can be NULL if the order has not been assigned yet. We store the **order date**, **total amount**, and **order status** ("Pending", "Assigned", "In Progress", "Completed", "Delivered", "Canceled"). A customer can create many orders, but each order belongs to only one customer. Likewise, one employee can handle many orders, while each order is handled by at most one employee.

– Each **Order_Details** record identifies one stone item within an order. It includes a unique **order_detail ID**, the **order ID** (to which it belongs), the **stone ID**, the **quantity ordered**, and **unit price**. This allows the system to manage orders that include multiple stone types and quantities. The relationship is many-to-many between Orders and Stones, resolved through Order_Details (one order has many order details, and one stone can appear in many order details). The quantity ordered must not exceed the stone's available quantity in stock at the time of purchase, and stock is reduced when the order is confirmed.

– Each **employee** has detailed information stored in a separate Employee table linked to the Users table, including **employee ID**, **full name**, **phone number**, **address**, **salary**, and **date hired**. Employees can log in using their assigned credentials, manage assigned orders for customers, update order status.

– **Customers** are stored in a separate Customer table linked to the Users table. Each customer has a unique **customer ID**, and customer-specific information includes **full name**, **phone number**, and **address**. When someone registers through the application, a record is created in the Users table with role = Customer, and a corresponding record is created in the Customers table for additional customer information. A customer can place many orders, and can also create many stone requests.

– The system maintains a **Custom_Orders** table to store stone requests made by customers before creating a formal order. Each request has a unique **ID** and is linked to exactly one customer. For each stone request, the system stores the **stone name**, **stone type**, **size**, requested **quantity**, additional notes (optional), **request status** (such as "Pending", "Approved", "Converted"), and the **date and time when the request was created**. A customer can create

many stone requests, and each request belongs to one customer. Stone requests are reviewed by the admin or employees and may later be converted into regular orders (when converted, the request status becomes “Converted” and an order is created based on the request details).