ER diagram

+----------+ +------------+ +-----------------+

| Clients | | Quotation | | Quotation\_itemss |

|----------| |------------| |-----------------|

| Client\_id PK|<-----| client\_id FK| | item\_id PK |

| Name | | Quotation\_id PK|<-----| quotation\_id FK |

| Email | | status | | | product\_id FK |----+

| Phone | | created\_at | | | quantity | |

| Address | +------------+ | | discount | |

+-----------+ | +-----------------+ |

| |

| |

| +-------------+

| | Product |

| |-------------|

| | Product\_id PK|

| | Name |

| | Description |

| | Price |

| | Tax\_rate |

+-----------+ | +-------------+

| Invoices | |

|-----------| |

| invoice\_id PK | |

| quotation\_id FK|-----------------------+

| invoice\_number |

| status |

| issued\_at |

+---------------+

**Schema Explanation**

* **Clients:** Stores client/customer information with unique Client\_id as primary key.
* **Product:** Stores product catalog details including pricing and tax information.
* **Quotation:** Represents quotations created for clients. Each quotation links to one client via client\_id. Status tracks the state of the quotation (Draft, Sent, Approved).
* **Quotation\_items:** Contains product line items for each quotation. Connects to both the quotation (quotation\_id) and product (product\_id). It includes quantity and discount details.
* **Invoices:** Generated invoices linked to quotations. Each invoice references a quotation via quotation\_id. Tracks invoice number, status (Unpaid or Paid), and issue date.

**Relationships:**

* One-to-many between Clients and Quotations (one client can have many quotations).
* One-to-many between Quotations and Quotation\_items (one quotation can include multiple products).
* One-to-many between Products and Quotation\_items (one product can appear in many quotation items).
* One-to-many between Quotations and Invoices (one quotation can have multiple invoices

## Instructions for Setting Up and Testing the Solution

### Setup

1. **Create the database and tables**  
   Run the provided SQL scripts in your preferred SQL environment (MySQL, MariaDB, etc.).
2. **Insert sample data**  
   Prepare and run sample INSERT statements to populate the tables with clients, products, quotations, quotation items, and invoices.
3. **Create stored procedures, triggers, or functions (optional)**  
   Implement additional logic needed for your application, like auto-generating invoice numbers or updating quotation status on invoice payment.

**Tables.sq** → **sample\_data.sql** → **procedures.sql** → **triggers.sql** → **queries.sql**

### Testing

1. **Test data integrity**
   * Insert clients and verify they appear correctly.
   * Create quotations linked to clients.
   * Add quotation items linking products to quotations.
   * Generate invoices linked to quotations.
2. **Test business logic**
   * Check status transitions on quotations (Draft → Sent → Approved).
   * Verify discounts and tax calculations on quotation items and invoices.
   * Validate invoice status updates (Unpaid → Paid).
3. **Run example queries**
   * Fetch all quotations for a client.
   * List products in a specific quotation.
   * Retrieve invoices with their quotation details