

Database design document

Group – 5

Topic : Stock market Portfolio Management System

Description of business problems being addressed by our data base:

Database Management: The primary problem is to build and maintain a database for an online stock trading platform. This includes managing and organizing data related to companies, their listed stocks, trader information, orders, and trade history.

Stock Tracking: The platform aims to provide a solution for traders to easily keep track of companies and the stocks they have available for trading. This involves real-time monitoring of stock prices, trading volumes, and other relevant data.

User Registration and Trading: Enabling traders to register on the platform and perform stock trading activities, including buying and selling stocks. This involves user authentication, order placement, and trade execution.

Stock Listing for Companies: Providing a mechanism for companies to list their stocks on the platform. Companies should be able to submit details of their stocks, making it easier for them to reach a wider audience of potential investors and traders.

Portfolio Management: Traders should have the capability to manage their portfolios, which are collections of stocks they own. This involves tracking the stocks they hold, their current value, and overall portfolio performance.

Convenience for Traders: The platform aims to make it easier for traders to buy and sell stocks of the companies they are interested in at their own convenience. This addresses the need for a user-friendly and accessible trading platform.

Below are the Entities of our initial ER-Diagram

Trader
Trader Login
Employee
Employee Login
Portfolio
Portfolio_stock
Company
Order
Order_stock
Stock

Relationships and Key Design Decisions:

One trader can have only one Trader login details and login details can have only one trader associated with it.

So, Trader and Trader login has mandatory one relationship between them.

One Employee can have only one Employee login details and login details can have only one Employee associated with it.

So Employee and Employee login has mandatory one relationship between them.

Trader can have only one portfolio and portfolio can have only one trader associated with it.

So, Trader and Portfolio share an mandatory one relationship.

Each Portfolio can have many stocks (portfolio_stocks_entity) in it. Each stock can be in only one portfolio of one user.

So, Portfolio has at least one to many relationship with portfolio_stock and portfolio_stock has mandatory one relationship with portfolio.

Stock entity has details of stocks listed by companies. Each portfolio_stock has at least one stock and each stock may be present in many portfolio_stocks.

So, Portfolio_stock has mandatory one with stock entity and stock entity has optional many with portfolio_stock entity.

Each company can list their stocks. If the company exists in the platform then it has to list stocks.

So, the relationship between them is mandatory one.

One trader can place one order at a time consisting many stocks with the attributes being the details of trader and the order the trader has placed. The order placing depends on the trader whether to place or not. Therefore, the relationship between trader and order is one to optional many.

One order can consist of numerous stocks, but when it has to focus on one particular stock, it shares the details of that stock including the details of stock. Therefore, one order and order_stock share mandatory to many relationship.

One stock corresponds the order of the particular stock that has been placed by the trader. Therefore, the relationship between stock and order_stock is mandatory on to optional many.

INITIAL ER – DIAGRAM

