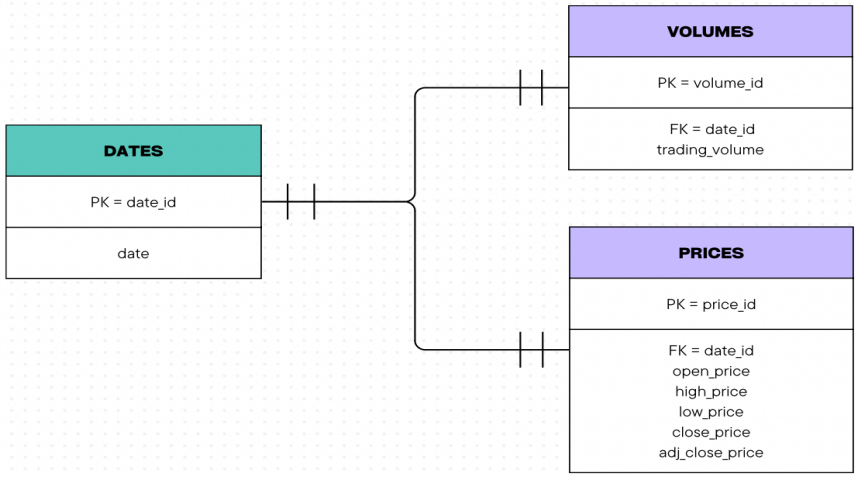
Report

Entity Relationship Diagram



---

1. Entities and Attributes

a. Dates

Primary Key (PK): date_id – It plays the role of the identifier for the different dates uniquely.
Attributes:
date – used to represent the date

b. Volumes

Primary Key (PK): volume_id – Every record here will have a unique Volume ID.
Foreign Key: date_id – Relating every volume record to a single specific date of the Date entity
Attributes:
trading_volume – used to represent volume of trade on any date.

c. Prices

Primary Key (PK): price_id – Each distinct price has its identifying unique price_id.
Foreign Key (FK): date_id – Each price record would relate to one particular date in the dates entity.

Attributes:
open_price – Price at the opening of the market.
high_price – Highest price of the trading session.
low_price – Lowest price of the trading session.
close_price – Price at the close of the market.
adj_close_price – Adjusted closing price, for stock splits, dividends etc.

2. Relationships

Dates to Prices = 1 to N
Dates to Volumes = 1 to N
One-to-Many Relationship:

- Here, the dates entity is the parent that will relate to volumes and prices entities.
- For every date_id in the dates, there can be n number of records in volumes and prices.
- This can be shown using the crow's foot notation.

3. Normalization Review

1NF (First Normal Form) : All the tables are in 1NF as every attribute in the tables is atomic and there aren't any repeating groups within a table.

2NF (Second Normal Form) : All non-key attributes are fully functionally dependant on their respective primaries. Example: In volumes, trading_volume relies on volume_id(the key of the table).

3NF : All transitive dependencies are resolved.
Example :
In prices, attributes such as open_price, high_price, etc. are depending only on price_id - the primary key and not on any other attribute.
If higher forms (e.g., BCNF or 4NF) are not implemented:

The database design achieves 3NF, which is sufficient for ensuring data integrity and reducing redundancy. Further normalization, such as BCNF or 4NF, is not needed for this simple structure of data because no multivalued dependencies are present.

4. Database Tables and Fields

Table Name	Field Name	Data Type	Constraints	Description
dates	date_id	Integer	Primary Key	Unique identifier for each date

	date	Date	NOT NULL	The actual calendar date
volumes	volume_id	Integer	Primary Key	Unique identifier
	date_id	Integer	Foreign Key	Connects the volume into a specific date
	trading_volume	BigInt	NOT NULL	Trading volume for a specific date
prices	price_id	Integer	Primary Key	Unique identifier for each price
	date_id	Integer	Foreign Key	Connecting the price into a specific date
	open_price	Decimal(10,2)	NOT NULL	The opening price of a stock on a specific date
	high_price	Decimal(10,2)	NOT NULL	Highest price reach by a stock on a specific date
	low_price	Decimal(10,2)	NOT NULL	Lowest price reach by a stock on a specific date
	close_price	Decimal(10,2)	NOT NULL	The closing price of a stock on a specific date
	adj_close_price	Decimal(10,2)	NULLABLE	The adjusted closing price of a stock

5. Analysis

a. Purpose and Use Case

This ER diagram is the model of financial trading data. It is used to:

- Keep daily trading volumes.
- Log daily price statistics about financial instruments, stocks, or commodities.

b. Strengths

- Normalization: There is no redundancy in the data as dates, volumes, and prices have been kept separate.
- Scalability: It can handle a good amount of volume of data which refers to dates via foreign keys without duplication.

c. Data Flow

Input: Dates fed into dates table

Volumes: Against every date the volume of trading is updated in the volumes table.
Prices: For each date, every price action -open, high, low, close- is recorded within the price table.

Conclusion
Above ER diagram depicts day-to-day trading data that can be used in performing financial analysis. It follows all the database design standards and is fully structured and normalized so that its data integrity is guaranteed and redundancy is minimum.

6. Relevance and Interest

That's an interesting dataset mainly because, for one whole year, the dataset will consist of daily stock prices for Apple Inc., from which the trend of stock prices can be analyzed, predictions about future prices can be made, and various elements affecting the movement of stock prices can be assessed. It contains Open, High, Low, Close, Adjusted Close, and Trading Volume, so it is very promising in financial and time-series analysis.

Following is some of the questions that one may ask about this dataset, which may also help in some database applications:

1. How are highest and lowest stock prices compared in a certain period?
We can go ahead and compare the highest with the lowest in different time intervals to gauge market volatility.

2. Is there a relationship between trading volume and stock price?
By using trading volume and stock price, we can analyze if the price increase often occurs along with the increase of trading volume or not.

3. How is the trend of the stock price in a month or a week?
Can use queries to find the average price in a certain week or month and identify whether it has a positive or negative trend.