University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: __2_____

Date: __Oct 21, 2025___

Group Number: __55___

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Windy Huang	68306661	c1y3i	windyuniapp@gmail.com
Kylie Wang	69075752	o8s0u	kyliewang025@gmail.com
Zhengyang Xu	65484644	v0u7d	xuzhengyang2023@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

University of British Columbia, Vancouver
Department of Computer Science

Project summary:

Our project provides recommendations to people on which stocks to sell or buy. People using this application would be able to make more informed decisions when manipulating their stock market portfolio.

ER Diagram:

Our Response to Feedback and Explanation of Changes to the ER Diagram are below.

Dividend:

No change. Dividend is not an attribute but an entity, consistent with the original ER diagram submitted for Milestone 1.

Company, Stock, and Report:

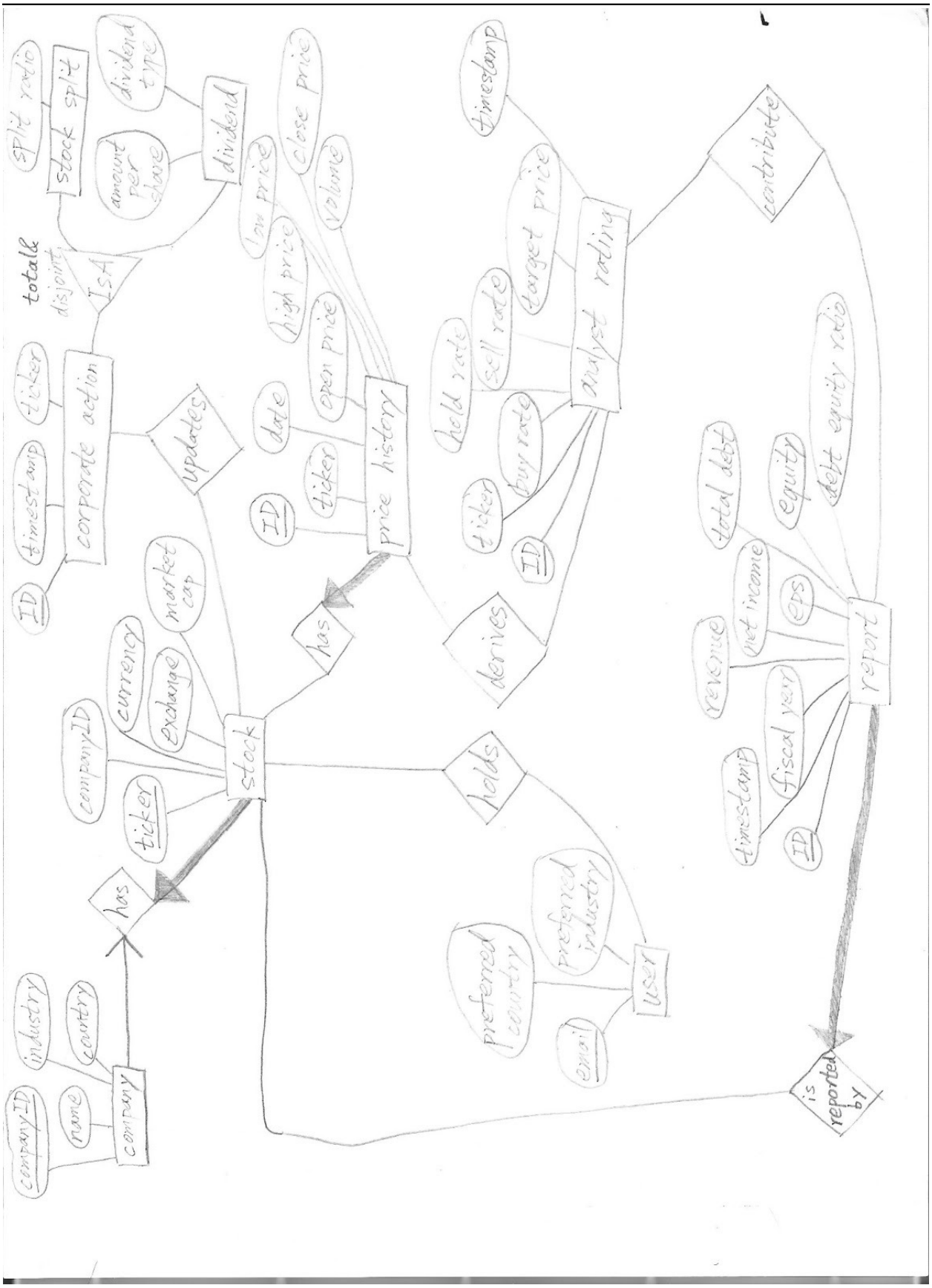
We changed the relationship between Company and Stock to one-to-one, and between Stock and Report to one-to-many.
This results in the implicit relation Report is many to one with Company.

IsA constraint for Corporate Action:

We adjusted the constraint from partial disjoint to total disjoint, as we realized our application does not involve any other types of corporate actions.

Updated ER diagram attached in the following page.

University of British Columbia, Vancouver
Department of Computer Science



University of British Columbia, Vancouver
Department of Computer Science

Relational schema from ER Diagram:

Company (companyID: integer, name: string, industry: string, country: string)
[primary key: companyID]

Stock (ticker: string, companyID: integer, exchange: string, currency: string, market_cap: integer)
[primary key: ticker, foreign key: companyID, unique: companyID, not null: companyID]

Updates (actionID: integer, ticker: string)
[primary key: (actionID, ticker), foreign key: (actionID, ticker)]

Stock split (actionID: integer, timestamp: date, split ratio: float)
[primary key: actionID]

Dividend (actionID: integer, timestamp: date, amount per share: float, dividend type: string)
[primary key: actionID]

Price history (priceHistoryID: integer, ticker: string, date: date, open price: float, high price: float, low price: float, close price: float, volume: integer)
[primary key: priceHistoryID, foreign key: ticker, not null: ticker]

User (email: string, preferred country: string, preferred industry: string)
[primary key: email]

Holds (ticker: string, email: string)
[primary key: (ticker, email), foreign key: (ticker, email)]

Report (reportID: string, timestamp: date, fiscal_year: integer, revenue: integer, net_income: integer, eps: float, total_debt: integer, equity: integer, debt equity ratio: float, ticker: string)
[primary key: reportID, foreign key: ticker, not null: ticker]

Analyst Rating (analystRatingID: integer, ticker: string, buy rate: float, hold rate: float, sell rate: float, target_price: float, timestamp: date)
[primary key: analystRatingID, foreign key: ticker]

Contribute (reportID: string, analystRatingID: integer)
[primary key: (reportID, analystRatingID), foreign key: (reportID, analystRatingID)]

Derives (priceHistoryID: integer, analystRatingID: integer)
[primary key: (priceHistoryID, analystRatingID), foreign key: (priceHistoryID, analystRatingID)]

Functional dependency and normalization:

Table	**FD (PK)**	**Non PK, CK FDs**	**Normalization (BCNF)**
Company	companyID -> Name, Industry, Country	-	Already in BCNF
Stock	ticker -> companyID, exchange, currency, market_cap	1. companyID -> ticker 2. companyID, exchange -> ticker 3. exchange -> currency	CK: ticker, companyID FD 3 violates BNCF since exchange is not superkey • Decompose on FD with PK • R1 = {exchange, currency}, R2 = {ticker, companyID, exchange, market_cap} • In BCNF
Updates	-	-	Already in BCNF
Stock split	actionID -> timestamp, split ratio	-	Already in BCNF
Dividend	actionID -> timestamp, amount per share, dividend type	-	Already in BCNF

Price history	priceHistoryID -> ticker, date, open price, high price, low price, close price, volume	ticker, date -> ticker, date, open price, high price, low price, close price, volume	Already in BCNF
User	ticker, email -> preferred country, preferred industry	-	Already in BCNF
Report	reportID -> timestamp, fiscal_year, revenue, net_income, eps, total_deb, equity, debt equity ratio, ticker	1. ticker, timestamp -> revenue, net_income, eps, total_deb, equity, debt equity ratio 2. Equity, total_debt -> debt equity ratio	CK: reportID, (ticker, timestamp) - FD2 violates BCNF since (equity, total_debt) is not a superkey R1 = {equity, total_debt, debt_equity_ratio} R2 = {reportID, ticker, timestamp, fiscal_year, revenue, net_income, eps, total_debt, equity} Both in BCNF
Analyst Rating	analystRatingID -> ticker, buy rate, hold rate, sell rate, target_price, timestamp	ticker, timestamp -> buy rate, hold rate, sell rate, target_price	Already in BCNF
Contribute	-	-	Already in BCNF
Derives	-	-	Already in BCNF

University of British Columbia, Vancouver
Department of Computer Science

Relational schema post normalization:

Company (companyID: integer, name: string, industry: string, country: string)
[primary key: companyID]

Stock (ticker: string, companyID: integer, exchange: string, market_cap: float)
[primary key: ticker, foreign key: (companyID, exchange), unique: companyID, not null: companyID]

Exchange (exchange: string, currency: string)
[primary key: exchange]

Updates (actionID: integer, ticker: string)
[primary key: (actionID, ticker), foreign key: (actionID, ticker)]

Stock split (actionID: integer, timestamp: date, split ratio: float)
[primary key: actionID]

Dividend (actionID: integer, timestamp: date, amount per share: float, dividend type: string)
[primary key: actionID]

Price history (priceHistoryID: integer, ticker: string, date: date, open price: float, high price: float, low price: float, close price: float, volume: integer)
[primary key: priceHistoryID, foreign key: ticker, not null: ticker]

User (email: string, preferred country: string, preferred industry: string)
[primary key: email]

Holds (ticker: string, email: string)
[primary key: (ticker, email), foreign key: (ticker, email)]

Report (reportID: string, timestamp: date, fiscal_year: integer, revenue: integer, net_income: integer, eps: float, total_debt: integer, equity: integer, ticker: string)
[primary key: reportID, foreign key: (ticker, equity, total_debt), not null: ticker]

DebtEquity(equity: integer, total_debt: integer, debt_equity_ratio: float)
[primary key: equity, total_debt]

Analyst Rating (analystRatingID: integer, ticker: string, buy rate: float, hold rate: float, sell rate: float, target_price: float, timestamp: date)
[primary key: analystRatingID, foreign key: ticker]

Contribute (reportID: string, analystRatingID: integer)
[primary key: (reportID, analystRatingID), foreign key: (reportID, analystRatingID)]

Derives (priceHistoryID: integer, analystRatingID: integer)
[primary key: (priceHistoryID, analystRatingID), foreign key: (priceHistoryID, analystRatingID)]

SQL DDL and INSERT statement:

The SQL script is attached in the following link:
https://drive.google.com/file/d/1FYmEq69M7yBGe6tujBq6SukeQ_HsGBfq/view?usp=sharing

Rationales for ON DELETE and ON UPDATE choice:

All foreign key constraints in our schema use ON DELETE CASCADE. Our application continuously processes and refreshes large volumes of stock related data, and old records are periodically deleted to improve performance. Therefore, when a parent record is removed, the related child records should also be deleted automatically, as they no longer represent the most up to date information.

We did not specify any ON UPDATE actions since those referenced primary keys are designed to be stable identifiers that should not change once created. Since updates are not expected in our application, defining ON UPDATE actions is unnecessary.

A note on naming of attributes:

All attribute names from the ER diagram have been preserved in the relational schema. However, when translating to SQL, some attribute names containing spaces (Ex: high price) were converted to camelCase (Ex: highPrice) for compatibility. Additionally, all ID attributes now include the table name as a prefix (Ex: priceHistoryID) to improve clarity.

AI Declaration:

We confirm that no AI tools were used in completing this assignment.