

Intro to Databases (Wu) - Project 1 Part 1

Andrew Mazof - alm2263

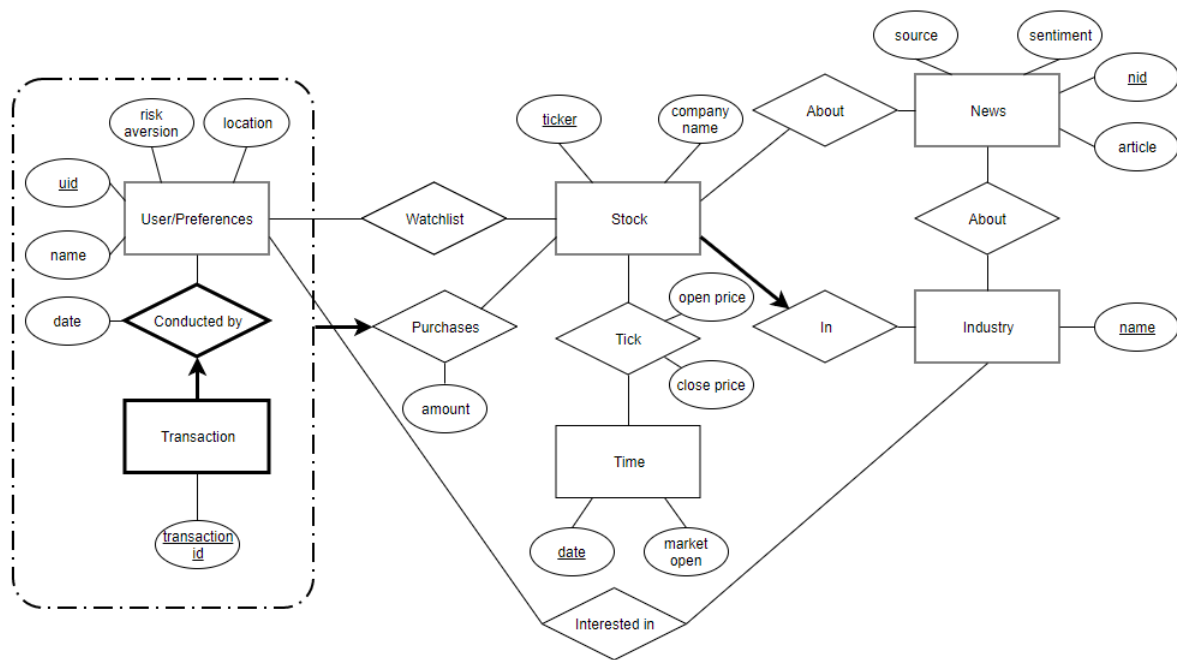
Wesley Tao - wt2271

September 27, 2018

Description

The purpose of the proposed application is to assist in the portfolio decisions of private investors by recommending stock positions based on an investor's personal investment preferences, and combining that with stock performance and publically available information. The database for this application involves data collected about each user, such as risk aversion, as well as financial data collected from third party resources such as Bloomberg. Each stock is uniquely identified by its ticker. The historical performance of each stock would be stored via a 'tick' relationship between the stock and the time entities, providing the open and closing prices for a given stock on a given date. The market open attribute for the time entity is simply a boolean to track whether the market is open on a given date. Other information about a stock includes the relevant industry, which is simply identified by its name, and relevant news. News about stock and companies is highly relevant due to the information it provides about public sentiment towards a stock. Sentiment is an attribute of news stored as an integer on a scale from 1 to 10, with 1 representing a highly negative view of a stock/company by a given news article, and 10 being highly positive. The application connects the information collected from third party sources with information about a user's preferences, which is collected via some sort of survey upon account creation. Specifically, information about a user's risk aversion and stock and industry preferences is stored through several attributes and relationships. Finally, a history of financial transactions conducted by each user is stored. Each transaction is a weak entity of the user, with information about the date and purchase amount for each stock being stored in the database. For the sake of this project, while financial information can be gathered through sources like Bloomberg, information about users will probably be made up.

ER Diagram



SQL Schema

```
CREATE TABLE User(
    uid int,
    name text,
    risk_aversion int,
    location text,
    PRIMARY KEY (uid)
);

CREATE TABLE Stock(
    ticker varchar[5],
    name text,
    PRIMARY KEY (ticker)
);

CREATE TABLE Time(
    date date,
    open boolean,
    PRIMARY KEY (date)
);

CREATE TABLE Tick(
    date date,
    ticker varchar[5],
    open_price real,
```

```

        close_price real,
        PRIMARY KEY (date, ticker),
        FOREIGN KEY (ticker) REFERENCES Stock,
        FOREIGN KEY (date) REFERENCES Time
    );

CREATE TABLE Watchlist(
    uid int,
    ticker varchar[5],
    PRIMARY KEY (uid, ticker),
    FOREIGN KEY (uid) REFERENCES User,
    FOREIGN KEY (ticker) REFERENCES Stock
);

CREATE TABLE Transaction_conducted(
    tid int,
    uid int,
    t_date date,
    PRIMARY KEY (uid, tid),
    FOREIGN KEY (uid) REFERENCES User ON DELETE CASCADE
);

CREATE TABLE Purchases(
    tid int,
    uid int,
    ticker varchar[5] NOT NULL,
    amount real,
    PRIMARY KEY (uid, tid),
    FOREIGN KEY (uid, tid) REFERENCES Transaction_conducted,
    FOREIGN KEY (ticker) REFERENCES Stock
);

CREATE TABLE News(
    nid int,
    article text,
    sentiment int,
    source text,
    PRIMARY KEY (nid)
);

CREATE TABLE Industry(
    name text,
    PRIMARY KEY (name)
);

CREATE TABLE About_stock(
    nid int,
    ticker varchar[5],
    PRIMARY KEY (nid, ticker),
    FOREIGN KEY (nid) REFERENCES News,
    FOREIGN KEY (ticker) REFERENCES Stock

```

```

);

CREATE TABLE About_industry(
    nid int,
    name text,
    PRIMARY KEY (nid, name),
    FOREIGN KEY (nid) REFERENCES News,
    FOREIGN KEY (name) REFERENCES Industry
);

CREATE TABLE In_industry(
    ticker varchar[5],
    name text NOT NULL,
    PRIMARY KEY (ticker),
    FOREIGN KEY (ticker) REFERENCES Stock,
    FOREIGN KEY (name) REFERENCES Industry
);

CREATE TABLE Interested_in(
    name text,
    uid int,
    PRIMARY KEY (name, uid),
    FOREIGN KEY (name) REFERENCES Industry,
    FOREIGN KEY (uid) REFERENCES User
);

```