

CPSC 304 Project Cover Page

Milestone #: 1

Date: July 15, 2025

Group Number: 28

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Tom Le	11698529	tomcspan	Letung40@icloud.com
Trung Quan Nguyen	94067857	tnguye63	Quanthangthien1000@gmail.com
Ziyan He	53796967	yh0111	Oliviahe0111@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

2. EigenSim is a web-based algorithmic trading simulator that allow users to backtest custom or predefined trading strategies on historical market data. Users can configure strategies using a set of parameters and evaluate their performance through generated backtest reports

3. Attach later

4.

- Users(user_id: VARCHAR2(12), first_name: VARCHAR2(20), last_name: VARCHAR2(20), password_hash: VARCHAR2(100))
 - PK: user_id
 - Constraints: password_hash NOT NULL
- Strategy(strategy_id: VARCHAR2(12), strategy_type: VARCHAR2(20))
 - PK: strategy_id
- CustomStrategy(strategy_id: CHAR(12), name: VARCHAR2(50), description: CLOB, created_date: DATE, custom_prompt: CLOB)
 - PK: strategy_id
 - FK: strategy_id -> Strategy(strategy_id)
- PredefinedStrategy(strategy_id: CHAR(12), name: VARCHAR2(50), description: CLOB, created_date: DATE, logic: CLOB)
 - PK: strategy_id
 - FK: strategy_id -> Strategy(strategy_id)
- Parameter(parameter_id: CHAR(12), name: VARCHAR2(30), type: VARCHAR2(10))
 - PK: parameter_id
- ParameterValue(strategy_id: CHAR(12), parameter_id: CHAR(12), assigned_value: VARCHAR2(100))
 - PK: (strategy_id, parameter_id)
 - FK: strategy_id → Strategy(strategy_id), parameter_id → Parameter(parameter_id)
- Ticker(ticker_symbol: VARCHAR2(10), company_name: VARCHAR2(100), exchange: VARCHAR2(20))
 - PK: ticker_symbol
- Backtest(backtest_id: CHAR(12), strategy_id: CHAR(12), user_id: VARCHAR2(12), ticker_symbol: VARCHAR2(10), start_date: DATE, end_date: DATE, result_summary: CLOB)
 - PK: backtest_id
 - FK: strategy_id -> Strategy(strategy_id), user_id -> Users(user_id), ticker_symbol -> Ticker(ticker_symbol)
- Report(report_id: VARCHAR2(20), backtest_id: VARCHAR2(12), generated_at: TIMESTAMP, total_return: NUMBER, annualized_return: NUMBER, sharpe_ratio: NUMBER, max_drawdown: NUMBER, win_rate: NUMBER, trade_count: NUMBER, t_stat: NUMBER, p_value: NUMBER, confidence_95_low: NUMBER, confidence_95_high: NUMBER)
 - PK: report_id
 - FK: backtest_id -> Backtest(backtest_id)
 - Constraint: backtest_id is UNIQUE

- Trade(trade_id: VARCHAR2(12), **backtest_id**: VARCHAR2(12), trade_time: TIMESTAMP, price: NUMBER, quantity: NUMBER, price: side VARCHAR2(4))
 - PK: trade_id
 - FK: backtest_id -> Backtest(backtest_id)

5.

- Users
 - user_id → first_name, last_name, password_hash (PK FD)
- Strategy
 - strategy_id → strategy_type (PK FD)
- CustomStrategy
 - strategy_id → name, description, created_date, custom_prompt (PK FD)
- PredefinedStrategy
 - strategy_id → name, description, created_date, logic (PK FD)
- Parameter
 - parameter_id → name, type (PK FD)
 - name → type
- ParameterValue
 - (strategy_id, parameter_id) → assigned_value (PK FD)
- Ticker
 - ticker_symbol → company_name, exchange (PK FD)
- Backtest
 - backtest_id → strategy_id, user_id, ticker_symbol, start_date, end_date, result_summary (PK FD)
- Report
 - report_id → backtest_id, generated_at, total_return, annualized_return, sharpe_ratio, max_drawdown, win_rate, trade_count, t_stat, p_value, confidence_95_low, confidence_95_high (PK FD)
 - backtest_id → report_id (Non-key FD — 1:1 enforced by UNIQUE constraint)
- Trade
 - trade_id → backtest_id, trade_time, price, quantity, side (PK FD)

6.

Everything is already in BCNF except for Parameters and Report

Parameter: parameter_id → name, type, name → type

Normalize into 2 table: Parameter(parameter_id, name), ParameterType(name, type)

Report: report_id → all, backtest_id → report_id

Normalize into 2 table: ReportMetadata(report_id, backtest_id), ReportStats (report_id, generated_at, total_return, annualized_return, sharpe_ratio, max_drawdown, win_rate, trade_count, t_stat, p_value, confidence_95_low, confidence_95_high)

Table Name	PK	FKs
Users	user_id	
Strategy	strategy_id	
CustomStrategy	strategy_id	strategy_id -> Strategy

PredefinedStrategy	strategy_id	strategy_id -> Strategy
Parameter	parameter_id	
ParameterType	name	
ParameterValue	(strategy_id, parameter_id)	strategy_id -> Strategy, parameter_id -> Parameter
Ticker	ticker_symbol	
Backtest	backtest_id	strategy_id, user_id, ticker_symbol
ReportMetadata	report_id	backtest_id
ReportStats	report_id	report_id → ReportMetadata
Trade	trade_id	backtest_id

7. From vscode for better format

```
-- USERS TABLE
CREATE TABLE Users (
    user_id VARCHAR2(12) PRIMARY KEY,
    first_name VARCHAR2(20),
    last_name VARCHAR2(20),
    password_hash VARCHAR2(100) NOT NULL
);

-- STRATEGY TABLE
CREATE TABLE Strategy (
    strategy_id VARCHAR2(12) PRIMARY KEY,
    strategy_type VARCHAR2(20) CHECK (strategy_type IN ('custom', 'predefined'))
);

-- CUSTOM STRATEGY TABLE
CREATE TABLE CustomStrategy (
    strategy_id VARCHAR2(12) PRIMARY KEY,
    name VARCHAR2(50),
    description CLOB,
    created_date DATE,
    custom_prompt CLOB,
    FOREIGN KEY (strategy_id) REFERENCES Strategy(strategy_id) ON DELETE CASCADE
);

-- PREDEFINED STRATEGY TABLE
CREATE TABLE PredefinedStrategy (
    strategy_id VARCHAR2(12) PRIMARY KEY,
    name VARCHAR2(50),
```

```

        description CLOB,
        created_date DATE,
        logic CLOB,
        FOREIGN KEY (strategy_id) REFERENCES Strategy(strategy_id) ON DELETE CASCADE
    );

-- PARAMETER TABLE
CREATE TABLE Parameter (
    parameter_id VARCHAR2(12) PRIMARY KEY,
    name VARCHAR2(30) UNIQUE
);

-- PARAMETER TYPE TABLE (from normalization of Parameter)
CREATE TABLE ParameterType (
    name VARCHAR2(30) PRIMARY KEY,
    type VARCHAR2(10) CHECK (type IN ('int', 'float', 'bool', 'string'))
);

-- PARAMETER VALUE TABLE
CREATE TABLE ParameterValue (
    strategy_id VARCHAR2(12),
    parameter_id VARCHAR2(12),
    assigned_value VARCHAR2(100),
    PRIMARY KEY (strategy_id, parameter_id),
    FOREIGN KEY (strategy_id) REFERENCES Strategy(strategy_id) ON DELETE CASCADE,
    FOREIGN KEY (parameter_id) REFERENCES Parameter(parameter_id) ON DELETE CASCADE
);

-- TICKER TABLE
CREATE TABLE Ticker (
    ticker_symbol VARCHAR2(10) PRIMARY KEY,
    company_name VARCHAR2(100),
    exchange VARCHAR2(20)
);

-- BACKTEST TABLE
CREATE TABLE Backtest (
    backtest_id VARCHAR2(12) PRIMARY KEY,
    strategy_id VARCHAR2(12),
    user_id VARCHAR2(12),
    ticker_symbol VARCHAR2(10),
    start_date DATE,
    end_date DATE,
    result_summary CLOB,
    FOREIGN KEY (strategy_id) REFERENCES Strategy(strategy_id),
    FOREIGN KEY (user_id) REFERENCES Users(user_id),
    FOREIGN KEY (ticker_symbol) REFERENCES Ticker(ticker_symbol)
);

```

```

-- REPORT METADATA TABLE (from normalization of Report)
CREATE TABLE ReportMetadata (
    report_id VARCHAR2(20) PRIMARY KEY,
    backtest_id VARCHAR2(12) UNIQUE,
    FOREIGN KEY (backtest_id) REFERENCES Backtest(backtest_id) ON DELETE CASCADE
);

-- REPORT STATS TABLE
CREATE TABLE ReportStats (
    report_id VARCHAR2(20) PRIMARY KEY,
    generated_at TIMESTAMP,
    total_return NUMBER,
    annualized_return NUMBER,
    sharpe_ratio NUMBER,
    max_drawdown NUMBER,
    win_rate NUMBER,
    trade_count NUMBER,
    t_stat NUMBER,
    p_value NUMBER,
    confidence_95_low NUMBER,
    confidence_95_high NUMBER,
    FOREIGN KEY (report_id) REFERENCES ReportMetadata(report_id) ON DELETE CASCADE
);

-- TRADE TABLE
CREATE TABLE Trade (
    trade_id VARCHAR2(12) PRIMARY KEY,
    backtest_id VARCHAR2(12),
    trade_time TIMESTAMP,
    price NUMBER,
    quantity NUMBER,
    side VARCHAR2(4) CHECK (side IN ('BUY', 'SELL')),
    FOREIGN KEY (backtest_id) REFERENCES Backtest(backtest_id) ON DELETE CASCADE
);

```

8. Again from vscode for better format

```

-- USERS
INSERT INTO Users VALUES ('U001', 'Alice', 'Wong', 'hash123');
INSERT INTO Users VALUES ('U002', 'Bob', 'Smith', 'hash456');
INSERT INTO Users VALUES ('U003', 'Carol', 'Lee', 'hash789');
INSERT INTO Users VALUES ('U004', 'David', 'Kim', 'hash321');
INSERT INTO Users VALUES ('U005', 'Eve', 'Nguyen', 'hash654');

```

```

-- STRATEGY
INSERT INTO Strategy VALUES ('S001', 'custom');
INSERT INTO Strategy VALUES ('S002', 'custom');
INSERT INTO Strategy VALUES ('S003', 'predefined');
INSERT INTO Strategy VALUES ('S004', 'predefined');
INSERT INTO Strategy VALUES ('S005', 'custom');

-- CUSTOM STRATEGY
INSERT INTO CustomStrategy VALUES ('S001', 'RSI Buy', 'Buy when RSI < 30',
TO_DATE('2023-06-01', 'YYYY-MM-DD'), 'Prompt for RSI strategy');
INSERT INTO CustomStrategy VALUES ('S002', 'MA Crossover', 'MA50 crosses MA200',
TO_DATE('2023-06-15', 'YYYY-MM-DD'), 'Prompt for MA strategy');
INSERT INTO CustomStrategy VALUES ('S005', 'Volume Spike', 'Detect volume spikes',
TO_DATE('2023-07-10', 'YYYY-MM-DD'), 'Prompt for volume');

-- PREDEFINED STRATEGY
INSERT INTO PredefinedStrategy VALUES ('S003', 'Mean Reversion', 'Revert to mean',
TO_DATE('2023-05-20', 'YYYY-MM-DD'), 'Built-in mean logic');
INSERT INTO PredefinedStrategy VALUES ('S004', 'Breakout', 'Break resistance',
TO_DATE('2023-07-01', 'YYYY-MM-DD'), 'Built-in breakout logic');

-- PARAMETER
INSERT INTO Parameter VALUES ('P001', 'rsi_threshold', 'int');
INSERT INTO Parameter VALUES ('P002', 'ma_short', 'int');
INSERT INTO Parameter VALUES ('P003', 'ma_long', 'int');
INSERT INTO Parameter VALUES ('P004', 'vol_threshold', 'float');
INSERT INTO Parameter VALUES ('P005', 'lookback_days', 'int');

-- PARAMETER TYPE
INSERT INTO ParameterType VALUES ('rsi_threshold', 'int');
INSERT INTO ParameterType VALUES ('ma_short', 'int');
INSERT INTO ParameterType VALUES ('ma_long', 'int');
INSERT INTO ParameterType VALUES ('vol_threshold', 'float');
INSERT INTO ParameterType VALUES ('lookback_days', 'int');

-- PARAMETER VALUE
INSERT INTO ParameterValue VALUES ('S001', 'P001', '30');
INSERT INTO ParameterValue VALUES ('S002', 'P002', '50');
INSERT INTO ParameterValue VALUES ('S002', 'P003', '200');
INSERT INTO ParameterValue VALUES ('S005', 'P004', '1.5');
INSERT INTO ParameterValue VALUES ('S005', 'P005', '14');

-- TICKER
INSERT INTO Ticker VALUES ('AAPL', 'Apple Inc.', 'NASDAQ');
INSERT INTO Ticker VALUES ('GOOG', 'Alphabet Inc.', 'NASDAQ');
INSERT INTO Ticker VALUES ('TSLA', 'Tesla Inc.', 'NASDAQ');
INSERT INTO Ticker VALUES ('AMZN', 'Amazon.com Inc.', 'NASDAQ');
INSERT INTO Ticker VALUES ('MSFT', 'Microsoft Corp.', 'NASDAQ');

```

```

-- BACKTEST
INSERT INTO Backtest VALUES ('B001', 'S001', 'U001', 'AAPL', TO_DATE('2022-01-01',
'YYYY-MM-DD'), TO_DATE('2022-06-01', 'YYYY-MM-DD'), 'Good return');
INSERT INTO Backtest VALUES ('B002', 'S002', 'U002', 'GOOG', TO_DATE('2022-02-01',
'YYYY-MM-DD'), TO_DATE('2022-07-01', 'YYYY-MM-DD'), 'Mediocre');
INSERT INTO Backtest VALUES ('B003', 'S003', 'U003', 'TSLA', TO_DATE('2022-03-01',
'YYYY-MM-DD'), TO_DATE('2022-08-01', 'YYYY-MM-DD'), 'Poor performance');
INSERT INTO Backtest VALUES ('B004', 'S004', 'U004', 'AMZN', TO_DATE('2022-04-01',
'YYYY-MM-DD'), TO_DATE('2022-09-01', 'YYYY-MM-DD'), 'High drawdown');
INSERT INTO Backtest VALUES ('B005', 'S005', 'U005', 'MSFT', TO_DATE('2022-05-01',
'YYYY-MM-DD'), TO_DATE('2022-10-01', 'YYYY-MM-DD'), 'Stable gains');

-- REPORT METADATA
INSERT INTO ReportMetadata VALUES ('R001', 'B001');
INSERT INTO ReportMetadata VALUES ('R002', 'B002');
INSERT INTO ReportMetadata VALUES ('R003', 'B003');
INSERT INTO ReportMetadata VALUES ('R004', 'B004');
INSERT INTO ReportMetadata VALUES ('R005', 'B005');

-- REPORT STATS
INSERT INTO ReportStats VALUES ('R001', SYSTIMESTAMP, 0.12, 0.24, 1.2, -0.05, 0.65,
100, 2.3, 0.02, 0.10, 0.34);
INSERT INTO ReportStats VALUES ('R002', SYSTIMESTAMP, 0.05, 0.10, 0.5, -0.15, 0.52,
85, 1.5, 0.10, -0.05, 0.25);
INSERT INTO ReportStats VALUES ('R003', SYSTIMESTAMP, -0.03, -0.06, -0.3, -0.25, 0.40,
120, -0.9, 0.35, -0.25, 0.10);
INSERT INTO ReportStats VALUES ('R004', SYSTIMESTAMP, 0.02, 0.04, 0.2, -0.10, 0.47,
95, 0.8, 0.15, -0.08, 0.22);
INSERT INTO ReportStats VALUES ('R005', SYSTIMESTAMP, 0.09, 0.18, 0.9, -0.06, 0.61,
110, 1.9, 0.05, 0.02, 0.30);

-- TRADE
INSERT INTO Trade VALUES ('T001', 'B001', SYSTIMESTAMP, 150.00, 10, 'BUY');
INSERT INTO Trade VALUES ('T002', 'B001', SYSTIMESTAMP, 155.00, 10, 'SELL');
INSERT INTO Trade VALUES ('T003', 'B002', SYSTIMESTAMP, 2800.00, 5, 'BUY');
INSERT INTO Trade VALUES ('T004', 'B003', SYSTIMESTAMP, 700.00, 8, 'SELL');
INSERT INTO Trade VALUES ('T005', 'B004', SYSTIMESTAMP, 3300.00, 3, 'BUY');

```