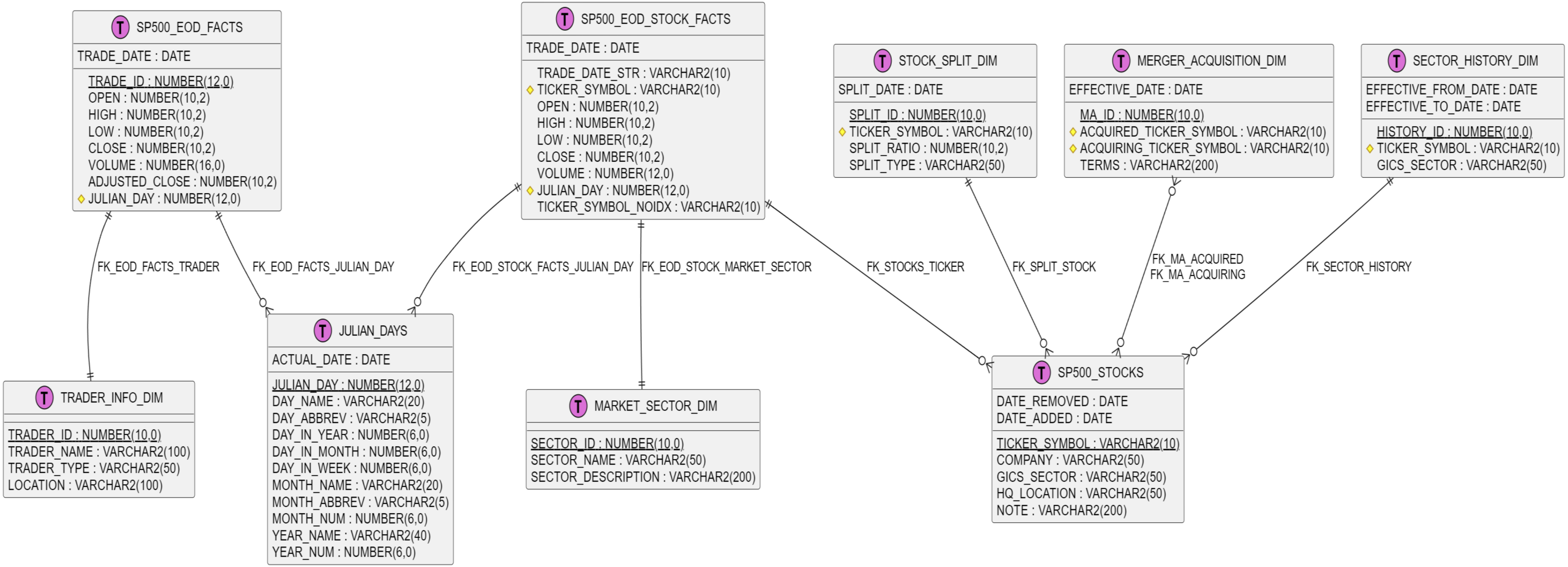
P1 Assignment: Dimension Modelling

By Group 7:
Advith Reddy Kunda
Neshma Kasamneni
Purna Chandra Shekar Reddy Gosala
Raajitha Sai Bondada
Sai Anirudh Mantha
Shyam Rishi Kashyap Sai Kalaparthy

Part A: Design for Stock Market Data

Entity Relationship Diagram



SP500_EOD_FACTS

TRADE_DATE : DATE

<u>TRADE_ID : NUMBER(12,0)</u>
OPEN : NUMBER(10,2)
HIGH : NUMBER(10,2)
LOW : NUMBER(10,2)
CLOSE : NUMBER(10,2)
VOLUME : NUMBER(16,0)
ADJUSTED_CLOSE : NUMBER(10,2)
◇ JULIAN_DAY : NUMBER(12,0)

SP500_EOD_STOCK_FACTS

TRADE_DATE : DATE

TRADE_DATE_STR : VARCHAR2(10)
◇ TICKER_SYMBOL : VARCHAR2(10)
OPEN : NUMBER(10,2)
HIGH : NUMBER(10,2)
LOW : NUMBER(10,2)
CLOSE : NUMBER(10,2)
VOLUME : NUMBER(12,0)
◇ JULIAN_DAY : NUMBER(12,0)
TICKER_SYMBOL_NOIDX : VARCHAR2(10)

STOCK_SPLIT_DIM

SPLIT_DATE : DATE

<u>SPLIT_ID : NUMBER(10,0)</u>
◇ TICKER_SYMBOL : VARCHAR2(10)
SPLIT_RATIO : NUMBER(10,2)
SPLIT_TYPE : VARCHAR2(50)

MERGER_ACQUISITION_DIM

EFFECTIVE_DATE : DATE

<u>MA_ID : NUMBER(10,0)</u>
◇ ACQUIRED_TICKER_SYMBOL : VARCHAR2(10)
◇ ACQUIRING_TICKER_SYMBOL : VARCHAR2(10)
TERMS : VARCHAR2(200)

SECTOR_HISTORY_DIM

EFFECTIVE_FROM_DATE : DATE
EFFECTIVE_TO_DATE : DATE

<u>HISTORY_ID : NUMBER(10,0)</u>
◇ TICKER_SYMBOL : VARCHAR2(10)
GICS_SECTOR : VARCHAR2(50)

FK_EOD_FACTS_TRADER
FK_EOD_FACTS_JULIAN_DAY
FK_EOD_STOCK_FACTS_JULIAN_DAY
FK_EOD_STOCK_MARKET_SECTOR
FK_STOCKS_TICKER
FK_SPLIT_STOCK
FK_MA_ACQUIRED
FK_MA_ACQUIRING
FK_SECTOR_HISTORY

TRADER_INFO_DIM

<u>TRADER_ID : NUMBER(10,0)</u>
TRADER_NAME : VARCHAR2(100)
TRADER_TYPE : VARCHAR2(50)
LOCATION : VARCHAR2(100)

JULIAN_DAYS

ACTUAL_DATE : DATE

<u>JULIAN_DAY : NUMBER(12,0)</u>
DAY_NAME : VARCHAR2(20)
DAY_ABBREV : VARCHAR2(5)
DAY_IN_YEAR : NUMBER(6,0)
DAY_IN_MONTH : NUMBER(6,0)
DAY_IN_WEEK : NUMBER(6,0)
MONTH_NAME : VARCHAR2(20)
MONTH_ABBREV : VARCHAR2(5)
MONTH_NUM : NUMBER(6,0)
YEAR_NAME : VARCHAR2(40)
YEAR_NUM : NUMBER(6,0)

MARKET_SECTOR_DIM

<u>SECTOR_ID : NUMBER(10,0)</u>
SECTOR_NAME : VARCHAR2(50)
SECTOR_DESCRIPTION : VARCHAR2(200)

SP500_STOCKS

DATE_REMOVED : DATE
DATE_ADDED : DATE

<u>TICKER_SYMBOL : VARCHAR2(10)</u>
COMPANY : VARCHAR2(50)
GICS_SECTOR : VARCHAR2(50)
HQ_LOCATION : VARCHAR2(50)
NOTE : VARCHAR2(200)

Fact Tables

A fact table is a central table in a data warehouse that stores quantitative data, also known as facts, related to a business process. It is typically surrounded by dimension tables, which contain descriptive attributes that add context to the facts. Together, they form the basis for dimensional modeling, a popular approach to data warehousing and business intelligence (BI).

Fact tables in above ERD:

SP500_EOD_FACTS
SP500_EOD_STOCK_FACTS

Dimension Tables

Dimension tables are the companions to fact tables, storing descriptive attributes that add context and meaning to the quantitative data (facts) found in the fact table. Think of them as the labels on the data points in a chart or graph.

Dimension Tables in above ERD:

SP500_STOCKS
SP500_DATES
TRADER_INFO_DIM
JULIAN_DAYS
MARKET_SECTOR_DIM
STOCK_SPLIT_DIM
MERGER_ACQUSITION_DIM
SECTOR_HISTORY_DIM

Handling Stock Splits and Reverse Splits

Stock splits and reverse splits affect the stock price and volume, making historical comparisons challenging without adjustment. To handle this, we have created another dimension table.
STOCK_SPLIT_DIM table to record stock splits and reverse splits, including the date, split ratio, and type (split or reverse split). This table would have foreign keys linking back to the stock symbol.

Handling Mergers and Acquisitions

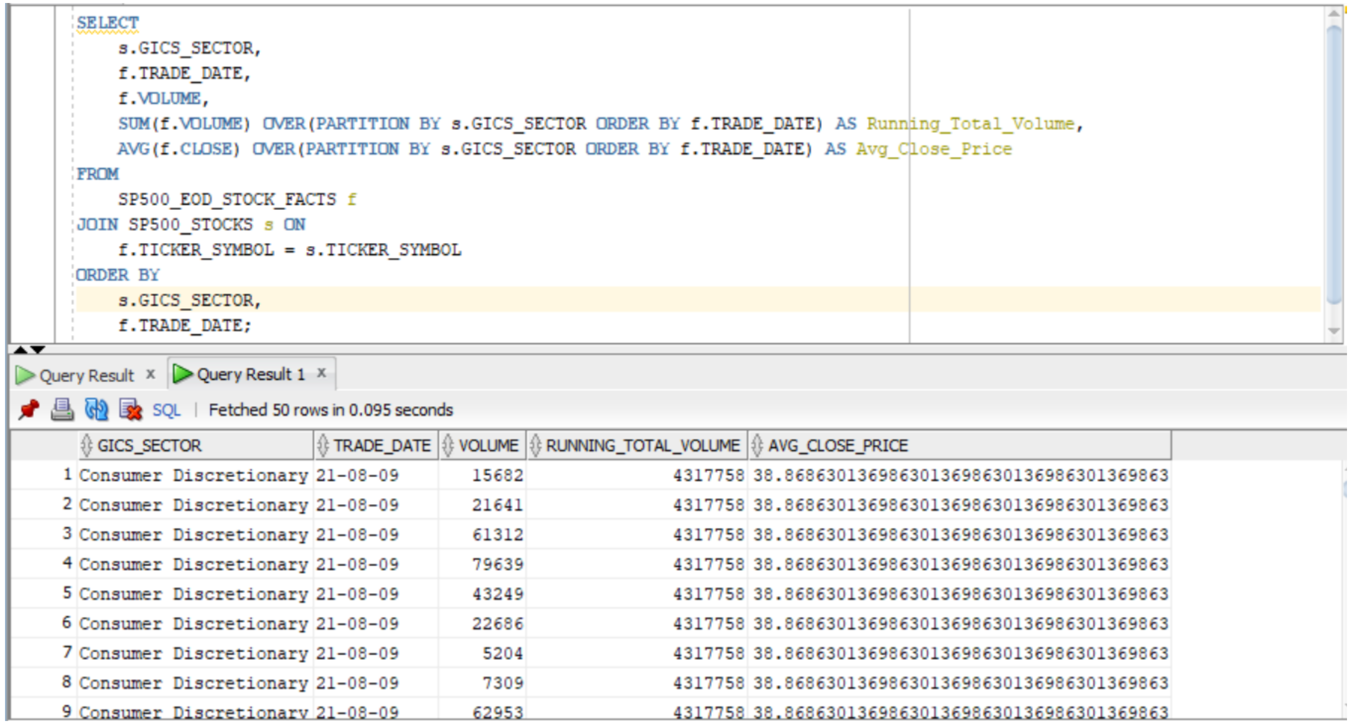
Mergers and acquisitions can result in the consolidation of companies, affecting historical data tracking.
For this we have created a MERGER_ACQUISITION_DIM table to log details about mergers and acquisitions, including the effective date, companies involved, and terms.

Handling Changes in Business Sectors

- Businesses evolve, and their sector classifications might change over time. To accommodate this:
- we have created another dimension history table,
- Sector History Table: Instead of updating the SP500_STOCKS table directly, maintain a SECTOR_HISTORY_DIM table that tracks the history of sector changes for each stock.

Business Queries

- Query using OVER for window function analysis:
- This query calculates the running total of trading volume and the average closing price over time within each GICS sector, using the SP500_EOD_STOCK_FACTS table, categorized by the GICS_SECTOR from SP500_STOCKS and ordered by trade date.

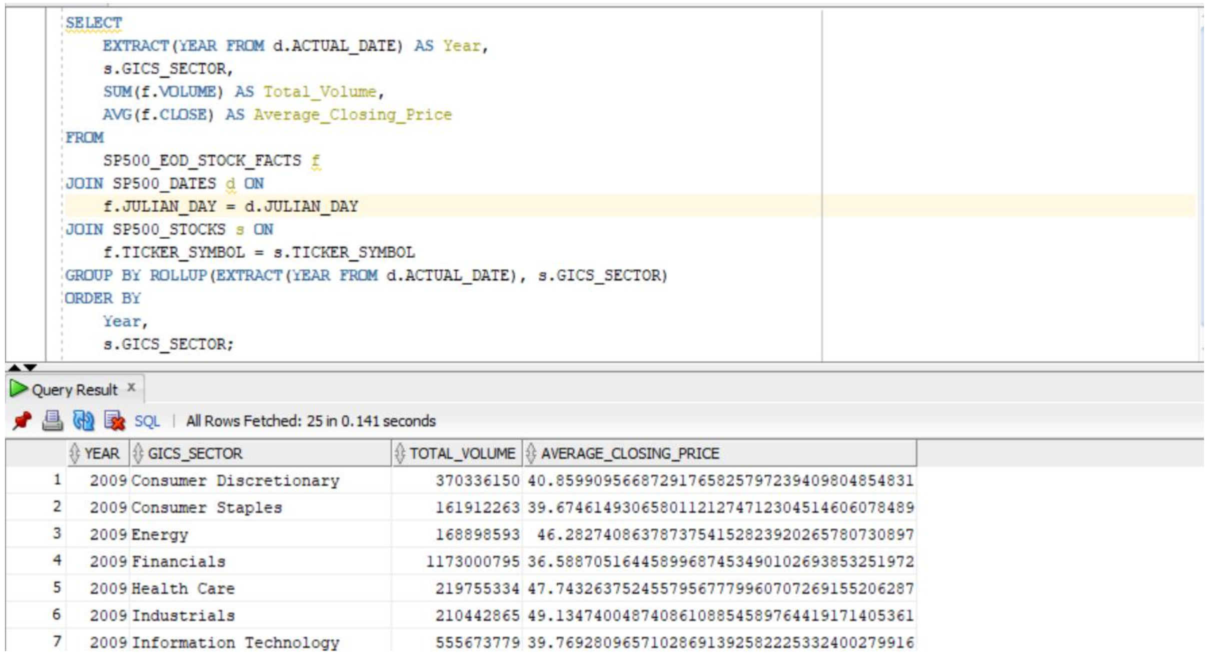


```
SELECT
    s.GICS_SECTOR,
    f.TRADE_DATE,
    f.VOLUME,
    SUM(f.VOLUME) OVER(PARTITION BY s.GICS_SECTOR ORDER BY f.TRADE_DATE) AS Running_Total_Volume,
    AVG(f.CLOSE) OVER(PARTITION BY s.GICS_SECTOR ORDER BY f.TRADE_DATE) AS Avg_Close_Price
FROM
    SP500_EOD_STOCK_FACTS f
JOIN SP500_STOCKS s ON
    f.TICKER_SYMBOL = s.TICKER_SYMBOL
ORDER BY
    s.GICS_SECTOR,
    f.TRADE_DATE;
```

Fetched 50 rows in 0.095 seconds

	GICS_SECTOR	TRADE_DATE	VOLUME	RUNNING_TOTAL_VOLUME	AVG_CLOSE_PRICE
1	Consumer Discretionary	21-08-09	15682	4317758	38.868630136986301369863013698630136986301369863
2	Consumer Discretionary	21-08-09	21641	4317758	38.868630136986301369863013698630136986301369863
3	Consumer Discretionary	21-08-09	61312	4317758	38.868630136986301369863013698630136986301369863
4	Consumer Discretionary	21-08-09	79639	4317758	38.868630136986301369863013698630136986301369863
5	Consumer Discretionary	21-08-09	43249	4317758	38.868630136986301369863013698630136986301369863
6	Consumer Discretionary	21-08-09	22686	4317758	38.868630136986301369863013698630136986301369863
7	Consumer Discretionary	21-08-09	5204	4317758	38.868630136986301369863013698630136986301369863
8	Consumer Discretionary	21-08-09	7309	4317758	38.868630136986301369863013698630136986301369863
9	Consumer Discretionary	21-08-09	62953	4317758	38.868630136986301369863013698630136986301369863

Query using ROLLUP for Sales Analysis by Date and Sector
This query aggregates total volumes and average closing prices for stocks, summarized by year and GICS sector, and then further aggregates these metrics at higher levels of the hierarchy (yearly, sector-wise, and overall).

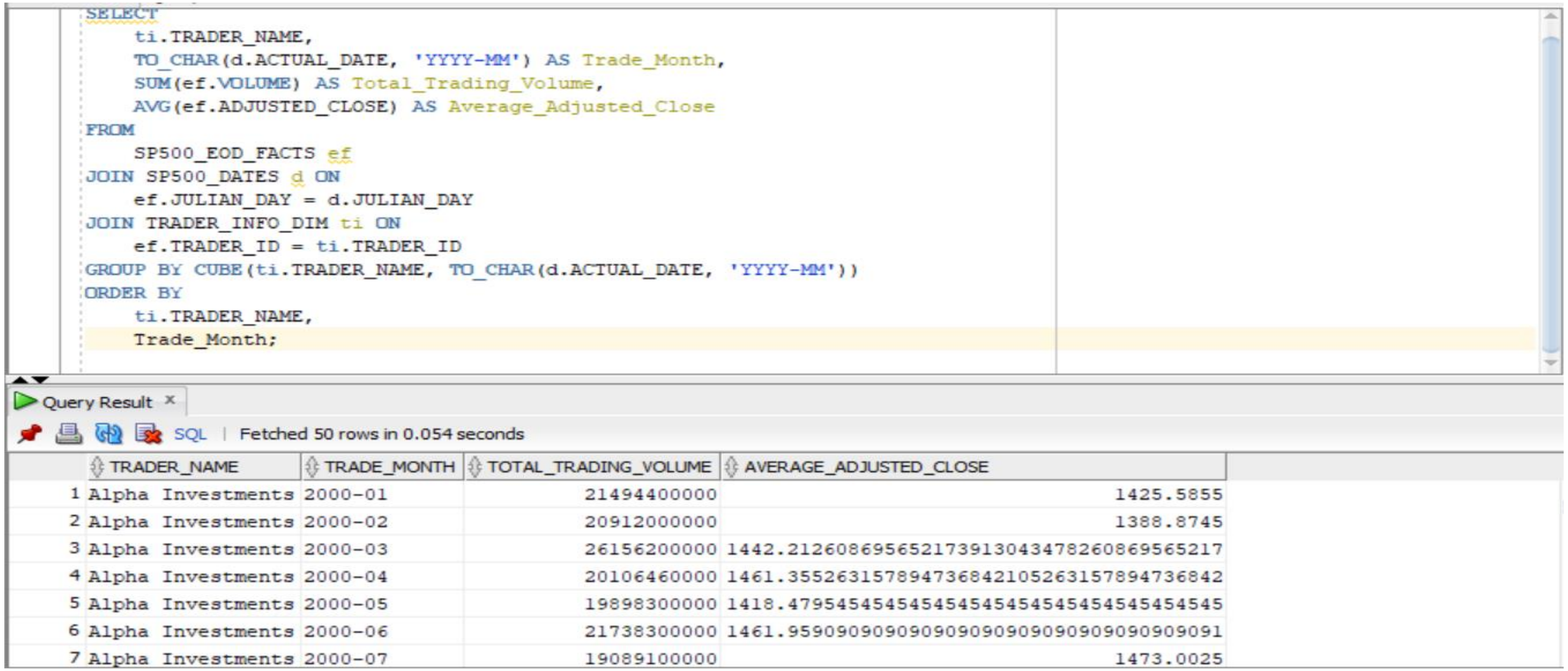


```
SELECT
    EXTRACT(YEAR FROM d.ACTUAL_DATE) AS Year,
    s.GICS_SECTOR,
    SUM(f.VOLUME) AS Total_Volume,
    AVG(f.CLOSE) AS Average_Closing_Price
FROM
    SP500_EOD_STOCK_FACTS f
JOIN SP500_DATES d ON
    f.JULIAN_DAY = d.JULIAN_DAY
JOIN SP500_STOCKS s ON
    f.TICKER_SYMBOL = s.TICKER_SYMBOL
GROUP BY ROLLUP(EXTRACT(YEAR FROM d.ACTUAL_DATE), s.GICS_SECTOR)
ORDER BY
    Year,
    s.GICS_SECTOR;
```

All Rows Fetched: 25 in 0.141 seconds

	YEAR	GICS_SECTOR	TOTAL_VOLUME	AVERAGE_CLOSING_PRICE
1	2009	Consumer Discretionary	370336150	40.85990956687291765825797239409804854831
2	2009	Consumer Staples	161912263	39.67461493065801121274712304514606078489
3	2009	Energy	168898593	46.2827408637873754152823920265780730897
4	2009	Financials	1173000795	36.58870516445899687453490102693853251972
5	2009	Health Care	219755334	47.74326375245579567779960707269155206287
6	2009	Industrials	210442865	49.13474004874086108854589764419171405361
7	2009	Information Technology	555673779	39.76928096571028691392582225332400279916

Query using CUBE for Aggregated Trading Activity by Trader and Month:
This query provides a multi-dimensional aggregation of trading activity from the SP500_EOD_FACTS table, analyzed by trader name (from TRADER_INFO_DIM) and month. It calculates total trading volumes and the average adjusted close price, allowing for analysis across different combinations of trader and time.



```
SELECT
    ti.TRADER_NAME,
    TO_CHAR(d.ACTUAL_DATE, 'YYYY-MM') AS Trade_Month,
    SUM(ef.VOLUME) AS Total_Trading_Volume,
    AVG(ef.ADJUSTED_CLOSE) AS Average_Adjusted_Close
FROM
    SP500_EOD_FACTS ef
JOIN SP500_DATES d ON
    ef.JULIAN_DAY = d.JULIAN_DAY
JOIN TRADER_INFO_DIM ti ON
    ef.TRADER_ID = ti.TRADER_ID
GROUP BY CUBE(ti.TRADER_NAME, TO_CHAR(d.ACTUAL_DATE, 'YYYY-MM'))
ORDER BY
    ti.TRADER_NAME,
    Trade_Month;
```

Fetched 50 rows in 0.054 seconds

	TRADER_NAME	TRADE_MONTH	TOTAL_TRADING_VOLUME	AVERAGE_ADJUSTED_CLOSE
1	Alpha Investments	2000-01	21494400000	1425.5855
2	Alpha Investments	2000-02	20912000000	1388.8745
3	Alpha Investments	2000-03	26156200000	1442.2126086956521739130434782608695652 17
4	Alpha Investments	2000-04	20106460000	1461.3552631578947368421052631578947368 42
5	Alpha Investments	2000-05	19898300000	1418.4795454545454545454545454545454545 45
6	Alpha Investments	2000-06	21738300000	1461.9590909090909090909090909090909090 91
7	Alpha Investments	2000-07	19089100000	1473.0025

PART B: Dimensional Data Model for Analyzing School Student Absences

Objectives

Understand the key factors contributing to student absenteeism (e.g., illness, academic struggles, family issues).

Identify students at risk of chronic absenteeism.

Develop targeted interventions to improve student attendance.

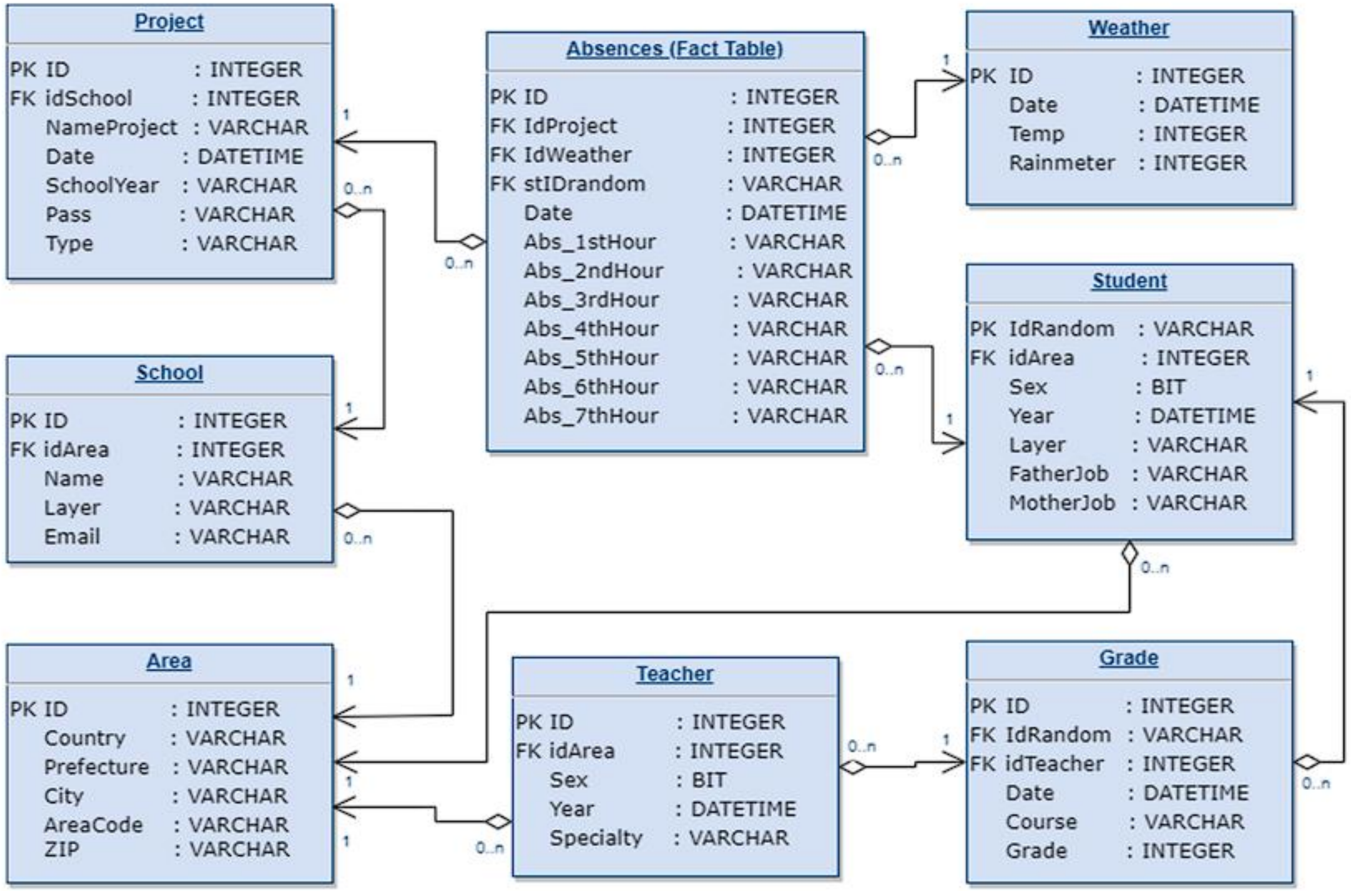
Conclusions

A dimensional database model can provide valuable insights into student absenteeism patterns and trends.

By analyzing data from multiple dimensions (e.g., time, student demographics, reason for absence), we can gain a deeper understanding of the root causes of absenteeism.

This information can be used to develop more effective strategies to improve student attendance and academic success.

Entity Relationship Diagram:



Project

PK	ID	: INTEGER
FK	idSchool	: INTEGER
	NameProject	: VARCHAR
	Date	: DATETIME
	SchoolYear	: VARCHAR
	Pass	: VARCHAR
	Type	: VARCHAR

Absences (Fact Table)

PK	ID	: INTEGER
FK	IdProject	: INTEGER
FK	IdWeather	: INTEGER
FK	stIDrandom	: VARCHAR
	Date	: DATETIME
	Abs_1stHour	: VARCHAR
	Abs_2ndHour	: VARCHAR
	Abs_3rdHour	: VARCHAR
	Abs_4thHour	: VARCHAR
	Abs_5thHour	: VARCHAR
	Abs_6thHour	: VARCHAR
	Abs_7thHour	: VARCHAR

Weather

PK	ID	: INTEGER
	Date	: DATETIME
	Temp	: INTEGER
	Rainmeter	: INTEGER

School

PK	ID	: INTEGER
FK	idArea	: INTEGER
	Name	: VARCHAR
	Layer	: VARCHAR
	Email	: VARCHAR

Student

PK	IdRandom	: VARCHAR
FK	idArea	: INTEGER
	Sex	: BIT
	Year	: DATETIME
	Layer	: VARCHAR
	FatherJob	: VARCHAR
	MotherJob	: VARCHAR

Area

PK	ID	: INTEGER
	Country	: VARCHAR
	Prefecture	: VARCHAR
	City	: VARCHAR
	AreaCode	: VARCHAR
	ZIP	: VARCHAR

Teacher

PK	ID	: INTEGER
FK	idArea	: INTEGER
	Sex	: BIT
	Year	: DATETIME
	Specialty	: VARCHAR

Grade

PK	ID	: INTEGER
FK	IdRandom	: VARCHAR
FK	idTeacher	: INTEGER
	Date	: DATETIME
	Course	: VARCHAR
	Grade	: INTEGER

THANK YOU