

# Stock Market Database

## FINAL REPORT

---

### **Team Name: Stocks**

Team number 59

### **Team Members:**

Jingyu Zhou - 002926983, Jagruti Agrawal - 002784847, Utkarsha Bhoir - 002727603

### **Project Description**

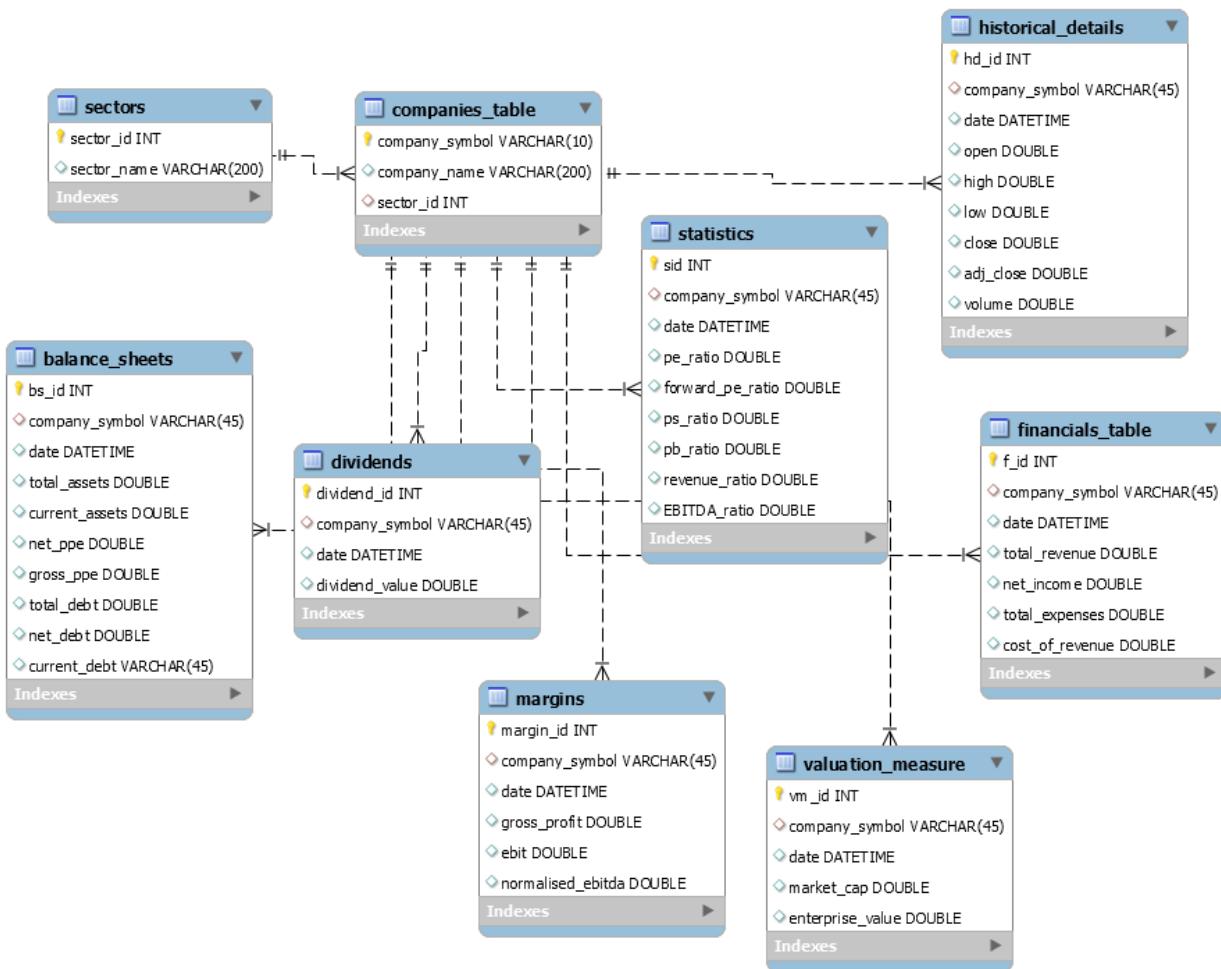
A database on the stock market for various sectors like healthcare, FMCG, and insurance. This database will aid in improving stock market comprehension for new investors. It will have the potential to help them decide which stocks to buy or sell by offering peer comparisons of stocks.

### **Project Features:**

- Our database predicts the stock trend of specific companies by providing the information about market capitalization, return over the years, market valuation, etc.
- This database will compare dividends, profit, sales, and return on investment of peer companies to help new investors buy potential stocks or sell underperforming stocks.

### **ER Diagram**

---



## Google Colab Link:

This colab link contains data cleaning scripts which handles null values and changes data types to it's desired one and also include data visualisation charts.

<https://colab.research.google.com/drive/1IMwMFGj8H9amhbHxhWPdnBiD1rOSU1eC#scrollTo=gn3osI0T26Ts>

## Files:

The below drive link includes all the csv files, google colab file, and sql queries file.

[https://drive.google.com/drive/folders/1tmwjHt1gcfzKCrfLRwpvRvjci2cUSL3R?usp=share\\_link](https://drive.google.com/drive/folders/1tmwjHt1gcfzKCrfLRwpvRvjci2cUSL3R?usp=share_link)

# Database Queries:

## Create Sector Table and Insert values

```
CREATE TABLE `sectors` (  
    `sector_id` int NOT NULL AUTO_INCREMENT,  
    `sector_name` varchar(200) DEFAULT NULL,  
    PRIMARY KEY (`sector_id`)  
)
```

insert into sectors(sector\_name) select distinct sector from companies

select \* from sectors

The screenshot shows the MySQL Workbench interface. In the top tab bar, 'SQL Queries' is selected. The left sidebar shows the 'SCHEMAS' tree, which includes 'stocks' and its sub-tables: 'balance\_sheet', 'balance\_sheets', 'companies', 'companies\_table', 'dividend', 'dividends', 'financials', 'financials\_table', 'historical\_data', 'historical\_details', 'historical\_prices', 'historical\_returns', 'industries', 'sectors', 'statistics', 'valuation\_measure', and 'valuation\_measures'. The main SQL editor window contains the following code:

```
-- Create Sectors Table and Insert values  
CREATE TABLE `sectors` (  
    `sector_id` int NOT NULL AUTO_INCREMENT,  
    `sector_name` varchar(200) DEFAULT NULL,  
    PRIMARY KEY (`sector_id`)  
);  
-- insert into sectors(sector_name) select distinct sector from companies  
SELECT * FROM sectors;  
-- Create Companies Table and Insert values  
CREATE TABLE `companies_table` (
```

Below the code, the 'Result Grid' shows the data inserted into the 'sectors' table:

sector_id	sector_name
1	Industrials
2	Health Care
3	Information Technology
4	Consumer Discretionary
5	Utilities
6	Financials
7	Materials
8	Real Estate
9	Energy
10	Consumer Staples

The bottom status bar indicates the operation completed successfully: '1 row(s) returned'.

## Create Companies Table and Insert values

```
CREATE TABLE `companies_table` (  
    `company_symbol` varchar(10) NOT NULL,
```

```

`company_name` varchar(200) DEFAULT NULL,
`sector_id` int DEFAULT NULL,
PRIMARY KEY (`company_symbol`),
KEY `comp_sec_idx` (`sector_id`),

CONSTRAINT `comp_sec` FOREIGN KEY (`sector_id`) REFERENCES `sectors` (`sector_id`) ON
DELETE CASCADE ON UPDATE CASCADE
)

```

insert into companies\_table(company\_symbol,company\_name,sector\_id) select symbol,name,`sector id` from companies;

select \* from companies\_table

The screenshot shows the MySQL Workbench interface with two queries being run:

```

SQL File 3* | SQL Queries* | SQL File 4*
19 -- KEY `comp_sec_idx` (`sector_id`),
20 -- CONSTRAINT `comp_sec` FOREIGN KEY (`sector_id`) REFERENCES `sectors` (`sector_id`) ON
21 -- DELETE CASCADE ON UPDATE CASCADE
22 --
23 -- insert into companies_table(company_symbol,company_name,sector_id) select symbol,name,`sector id` from companies;
24
25 -- Create Historical Details table and Insert values
26 -- CREATE TABLE `historical_details` (

```

The Result Grid shows the data inserted into the companies\_table:

	company_symbol	company_name	sector_id
1	A	Agilent Technologies Inc	2
2	AAL	American Airlines Group	1
3	AAP	Advance Auto Parts	4
4	AAPL	Apple Inc.	3
5	ABC	AmerisourceBergen Corp	2
6	ABT	Abbott Laboratories	2
7	ADI	Analog Devices, Inc.	3
8	ADM	Archer Daniels-Midland Co	10
9	ADP	Automatic Data Processing	3
10	AEE	Ameren Corp	5
11	AEP	American Electric Power	5
12	AES	AES Corp	5

The Action Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	16:24:14	select * from sectors - Create Companies Table and Insert values - CREATE TABLE `companies_table` ...	10 row(s) returned	0.073 sec / 0.000 sec
2	16:24:40	select * from companies_table - Create Historical Details table and Insert values - CREATE TABLE `historical_details` ...	53 row(s) returned	0.000 sec / 0.000 sec

## Create Dividends Table and Insert values

```

CREATE TABLE `dividends` (
`dividend_id` int NOT NULL AUTO_INCREMENT,

```

```

`company_symbol` varchar(45) DEFAULT NULL,  

`date` datetime DEFAULT NULL,  

`dividend_value` double DEFAULT NULL,  

PRIMARY KEY (`dividend_id`),  

KEY `comp_div_idx` (`company_symbol`),  

CONSTRAINT `comp_div` FOREIGN KEY (`company_symbol`) REFERENCES  

`companies_table` (`company_symbol`) ON DELETE CASCADE ON UPDATE CASCADE  

)

```

insert into dividends(company\_symbol,date,dividend\_value) select symbol,date,dividends  
from dividend;

select \* from dividends

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database schema with tables like aed\_project, ass\_b1, dbmsl, sdl\_tut, stock\_market, and stock.
- SQL Editor:** Contains the SQL code for creating the dividends table and inserting data. The code includes a primary key, a foreign key constraint, and a select statement to insert data from the dividend table into the dividends table.
- Result Grid:** Displays the results of the insert operation, showing 12 rows of data with columns: dividend\_id, company\_symbol, date, and dividend\_value.
- Action Output:** Shows the log of actions taken, including the creation of companies\_table, historical\_details, and valuation\_measure, and the execution of the select and insert statements.

dividend_id	company_symbol	date	dividend_value
1	A	2012-12-27 00:00:00	0.071531
2	A	2013-03-28 00:00:00	0.085837
3	A	2013-06-28 00:00:00	0.085837
4	A	2013-09-27 00:00:00	0.094421
5	A	2013-12-27 00:00:00	0.094421
6	A	2014-03-04 00:00:00	0.094421
7	A	2014-06-27 00:00:00	0.094421
8	A	2014-09-26 00:00:00	0.094421
9	A	2015-02-01 00:00:00	0.1
10	A	2015-03-27 00:00:00	0.1
11	A	2015-06-26 00:00:00	0.1
12	A	2015-09-25 00:00:00	0.1

## Create Valuation Measures Table and Insert values

CREATE TABLE `valuation\_measure` (

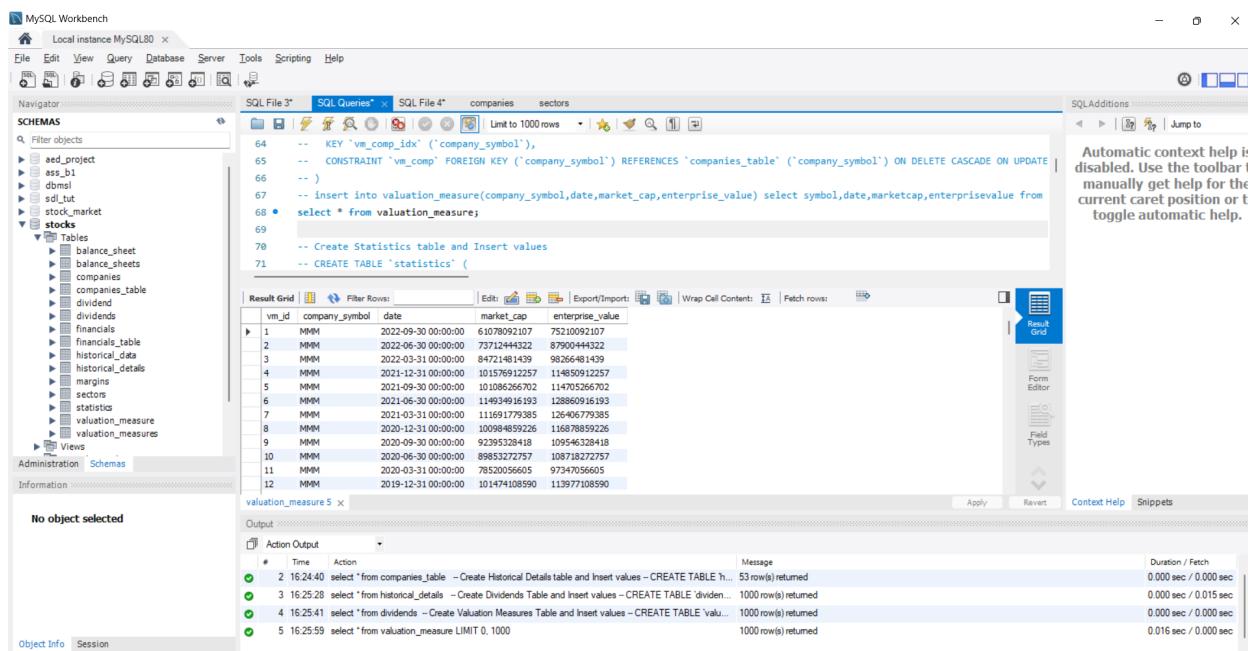
```

`vm_id` int NOT NULL AUTO_INCREMENT,
`company_symbol` varchar(45) DEFAULT NULL,
`date` datetime DEFAULT NULL,
`market_cap` double DEFAULT NULL,
`enterprise_value` double DEFAULT NULL,
PRIMARY KEY (`vm_id`),
KEY `vm_comp_idx` (`company_symbol`),
CONSTRAINT `vm_comp` FOREIGN KEY (`company_symbol`) REFERENCES
`companies_table`(`company_symbol`) ON DELETE CASCADE ON UPDATE CASCADE
)

```

insert into valuation\_measure(company\_symbol,date,market\_cap,enterprise\_value) select symbol,date,marketcap,enterprisevalue from valuation\_measures;

select \* from valuation\_measure;



The screenshot shows the MySQL Workbench interface with the following details:

- SQL Editor:** Contains the SQL code for creating the valuation\_measure table and inserting data from valuation\_measures.
- Result Grid:** Displays the 12 rows of data inserted into the valuation\_measure table.
- Output:** Shows the log of actions taken, including the creation of historical\_details, dividends, and valuation\_measures tables, and the insertion of data into valuation\_measure.

```

64 -- KEY `vm_comp_idx` (`company_symbol`),
65 -- CONSTRAINT `vm_comp` FOREIGN KEY (`company_symbol`) REFERENCES `companies_table`(`company_symbol`) ON DELETE CASCADE ON UPDATE CASCADE
66 )
67 -- insert into valuation_measure(company_symbol,date,market_cap,enterprise_value) select symbol,date,marketcap,enterprisevalue from valuation_measures;
68 • select * from valuation_measure;
69
70 -- Create Statistics table and Insert values
71 -- CREATE TABLE `statistics` (

```

vm_id	company_symbol	date	market_cap	enterprise_value
1	MMM	2022-09-30 00:00:00	61078092107	75210092107
2	MMM	2022-06-30 00:00:00	73712444322	87900444322
3	MMM	2022-03-31 00:00:00	84721481439	98266481439
4	MMM	2021-12-31 00:00:00	10157691257	11485091257
5	MMM	2021-09-30 00:00:00	101086265702	114705265702
6	MMM	2021-06-30 00:00:00	1149349516193	128860916193
7	MMM	2021-03-31 00:00:00	111691779385	126405779385
8	MMM	2020-12-31 00:00:00	100984859226	116878859226
9	MMM	2020-09-30 00:00:00	92395328418	109545328418
10	MMM	2020-06-30 00:00:00	89853272757	108718272757
11	MMM	2020-03-31 00:00:00	7852056605	97347056605
12	MMM	2019-12-31 00:00:00	101474108590	113977108590

```

# Time Action Message Duration / Fetch
2 16:24:40 select * from companies_table -- Create Historical Details table and Insert values --CREATE TABLE `historical_d... 53 row(s) returned 0.000 sec / 0.000 sec
3 16:25:28 select * from historical_details -- Create Dividends Table and Insert values --CREATE TABLE `dividen... 1000 row(s) returned 0.000 sec / 0.015 sec
4 16:25:41 select * from dividends -- Create Valuation Measures Table and Insert values --CREATE TABLE `valu... 1000 row(s) returned 0.000 sec / 0.000 sec
5 16:25:59 select * from valuation_measure LIMIT 0, 1000 1000 row(s) returned 0.016 sec / 0.000 sec

```

---

## Create Statistics table and Insert values

```
CREATE TABLE `statistics` (
    `sid` int NOT NULL AUTO_INCREMENT,
    `company_symbol` varchar(45) DEFAULT NULL,
    `date` datetime DEFAULT NULL,
    `pe_ratio` double DEFAULT NULL,
    `forward_pe_ratio` double DEFAULT NULL,
    `ps_ratio` double DEFAULT NULL,
    `pb_ratio` double DEFAULT NULL,
    `revenue_ratio` double DEFAULT NULL,
    `EBITDA_ratio` double DEFAULT NULL,
    PRIMARY KEY (`sid`),
    KEY `s_comp_idx` (`company_symbol`),
    CONSTRAINT `s_comp` FOREIGN KEY (`company_symbol`) REFERENCES `companies_table`(`company_symbol`) ON DELETE CASCADE ON UPDATE CASCADE
)
insert into
statistics(company_symbol,date,pe_ratio,Forward_Pe_Ratio,Ps_Ratio,pb_ratio,revenue_ratio,
EBITDA_ratio) select
symbol,date,PeRatio,ForwardPeRatio,PsRatio,PbRatio,EnterprisesValueRevenueRatio,Enter
prisesValueEBITDARatio from valuation_measures;
select * from statistics
```

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'Local instance MySQL80' is selected. The main area has three tabs: 'SQL File 3\*', 'SQL Queries\*', and 'SQL File 4\*'. The 'SQL Queries\*' tab contains the following SQL code:

```

82 -- KEY `s_comp_id`(`company_symbol`),
83 -- CONSTRAINT `s_comp` FOREIGN KEY (`company_symbol`) REFERENCES `companies_table`(`company_symbol`) ON DELETE CASCADE ON UPDATE
84 --
85 -- insert into statistics(company_symbol,date,pe_ratio,Forward_Pe_Ratio,ps_ratio,pb_ratio,revenue_ratio,EBITDA_ratio) select symbol,
86 select * from statistics
87
88 -- Create Financials table and and Insert values
89 CREATE TABLE `financials_table` (

```

The 'Result Grid' pane below displays the results of the 'select \* from statistics' query. The columns are: sid, company\_symbol, date, pe\_ratio, forward\_pe\_ratio, ps\_ratio, pb\_ratio, revenue\_ratio, EBITDA\_ratio. The data is as follows:

sid	company_symbol	date	pe_ratio	forward_pe_ratio	ps_ratio	pb_ratio	revenue_ratio	EBITDA_ratio
1	MMM	2022-09-30 00:00:00	15.433	10.142	1.822	4.452	8.726	16.043
2	MMM	2022-06-30 00:00:00	13.466	12.019	2.133	4.937	10.101	135.231
3	MMM	2022-03-31 00:00:00	14.711	14.306	2.465	5.631	11.13	45.18
4	MMM	2021-12-31 00:00:00	17.57	17.153	2.948	6.991	13.336	52.467
5	MMM	2021-09-30 00:00:00	17.351	16.287	2.956	6.964	12.828	48.811
6	MMM	2021-06-30 00:00:00	20.331	20.492	3.516	8.353	14.398	50.913
7	MMM	2021-03-31 00:00:00	20.83	19.96	3.486	8.6	14.282	49.825
8	MMM	2020-12-31 00:00:00	20.491	18.587	3.207	8.456	13.617	49.757
9	MMM	2020-09-30 00:00:00	18.182	17.036	2.974	8.51	13.119	45.38
10	MMM	2020-06-30 00:00:00	18.309	19.268	2.813	8.801	15.15	48.148
11	MMM	2020-03-31 00:00:00	17.479	14.599	2.485	7.803	12.055	45.703
12	MMM	2019-12-31 00:00:00	20.928	18.282	3.243	9.482	14.052	63.356

The 'Output' pane at the bottom shows the execution log with the following entries:

- 3 16:25:28 select \* from historical\_details --Create Dividends Table and Insert values --CREATE TABLE `dividen... 1000 row(s) returned 0.000 sec / 0.015 sec
- 4 16:25:41 select \*from dividends --Create Valuation Measures Table and Insert values --CREATE TABLE `valu... 1000 row(s) returned 0.000 sec / 0.000 sec
- 5 16:25:59 select \*from valuation\_measure LIMIT 0, 1000 1000 row(s) returned 0.016 sec / 0.000 sec
- 6 16:26:13 select \*from statistics -- Create Financials table and and Insert values --CREATE TABLE `financials\_ta... 1000 row(s) returned 0.000 sec / 0.000 sec

## Create Financials table and and Insert values

```

CREATE TABLE `financials_table` (
    `f_id` int NOT NULL AUTO_INCREMENT,
    `company_symbol` varchar(45) DEFAULT NULL,
    `date` datetime DEFAULT NULL,
    `total_revenue` double DEFAULT NULL,
    `net_income` double DEFAULT NULL,
    `total_expenses` double DEFAULT NULL,
    `cost_of_revenue` double DEFAULT NULL,
    PRIMARY KEY (`f_id`),
    KEY `financials_comp_idx`(`company_symbol`),
    CONSTRAINT `financials_comp` FOREIGN KEY (`company_symbol`) REFERENCES `companies_table`(`company_symbol`) ON DELETE CASCADE ON UPDATE CASCADE
)

```

)

insert into

```
financials_table(company_symbol,date,total_revenue,net_income,total_expenses,cost_of_revenue) select symbol,date,totalrevenue,netincome,totalexpenses,costofrevenue from financials;
```

select \* from financials\_table

The screenshot shows the MySQL Workbench interface with a script editor containing a SQL insert statement and its results.

**Script Editor:**

```
-- PRIMARY KEY ('f_id'),  
-- KEY `financials_comp_idx` ('company_symbol'),  
-- CONSTRAINT `financials_comp` FOREIGN KEY (`company_symbol`) REFERENCES `companies_table` (`company_symbol`) ON DELETE CASCADE  
-- )  
-- insert into financials_table(company_symbol,date,total_revenue,net_income,total_expenses,cost_of_revenue) select symbol,date,tota  
102 • select * from financials_table  
103  
104 -- Create margins table and Insert values
```

**Result Grid:**

f_id	company_symbol	date	total_revenue	net_income	total_expenses	cost_of_revenue
1	AAP	2022-09-30 00:00:00	2641341000	110982000	2464143000	1461490000
2	AAP	2022-06-30 00:00:00	2665426000	144402000	2463744000	1479707000
3	AAP	2022-03-31 00:00:00	3374210000	139791000	3170940000	1867690000
4	AAP	2021-12-31 00:00:00	2396975000	81669000	2284513000	1324858000
5	AAP	2021-09-30 00:00:00	2621229000	169813000	2392031000	1438775000
6	AAP	2021-06-30 00:00:00	2649415000	178696000	2404487000	1460164000
7	AAP	2021-03-31 00:00:00	3330370000	185930000	3078241000	1845444000
8	AAP	2020-12-31 00:00:00	2365131000	111997000	2213305000	1281435000
9	AAP	2020-09-30 00:00:00	2541928000	147476000	2285117000	1413457000
10	AAP	2020-06-30 00:00:00	2501380000	189960000	2238535000	1404666000
11	AAP	2020-03-31 00:00:00	2697882000	43588000	2619457000	1525149000
12	AAP	2019-12-31 00:00:00	2112614000	95907000	1986475000	1183845000

**Action Output:**

#	Time	Action	Message	Duration / Fetch
5	16:25:59	select * from valuation_measure LIMIT 0, 1000	1000 row(s) returned	0.016 sec / 0.000 sec
6	16:26:13	select * from statistics - Create Financials table and Insert values - CREATE TABLE financials_table (f_id int NOT NULL AUTO_INCREMENT, company_symbol varchar(45) DEFAULT NULL, date datetime DEFAULT NULL, open double DEFAULT NULL, high double DEFAULT NULL, low double DEFAULT NULL, close double DEFAULT NULL, volume int DEFAULT NULL, total_revenue double DEFAULT NULL, net_income double DEFAULT NULL, total_expenses double DEFAULT NULL, cost_of_revenue double DEFAULT NULL, margin double DEFAULT NULL, margin_percent double DEFAULT NULL, margin_change double DEFAULT NULL, margin_change_percent double DEFAULT NULL, margin_low double DEFAULT NULL, margin_high double DEFAULT NULL, margin_percent_low double DEFAULT NULL, margin_percent_high double DEFAULT NULL, margin_change_low double DEFAULT NULL, margin_change_high double DEFAULT NULL, margin_change_percent_low double DEFAULT NULL, margin_change_percent_high double DEFAULT NULL, created_at timestamp DEFAULT CURRENT_TIMESTAMP, updated_at timestamp DEFAULT CURRENT_TIMESTAMP);	1000 row(s) returned	0.000 sec / 0.000 sec
7	16:26:38	select * from financials_table - Create margins table and Insert values - CREATE TABLE margins (f_id int NOT NULL AUTO_INCREMENT, company_symbol varchar(45) DEFAULT NULL, date datetime DEFAULT NULL, open double DEFAULT NULL, high double DEFAULT NULL, low double DEFAULT NULL, close double DEFAULT NULL, volume int DEFAULT NULL, total_revenue double DEFAULT NULL, net_income double DEFAULT NULL, total_expenses double DEFAULT NULL, cost_of_revenue double DEFAULT NULL, margin double DEFAULT NULL, margin_percent double DEFAULT NULL, margin_change double DEFAULT NULL, margin_change_percent double DEFAULT NULL, margin_low double DEFAULT NULL, margin_high double DEFAULT NULL, margin_percent_low double DEFAULT NULL, margin_percent_high double DEFAULT NULL, margin_change_low double DEFAULT NULL, margin_change_high double DEFAULT NULL, margin_change_percent_low double DEFAULT NULL, margin_change_percent_high double DEFAULT NULL, created_at timestamp DEFAULT CURRENT_TIMESTAMP, updated_at timestamp DEFAULT CURRENT_TIMESTAMP);	1000 row(s) returned	0.016 sec / 0.000 sec
8	16:26:51	select * from financials_table - Create margins table and Insert values - CREATE TABLE margins (f_id int NOT NULL AUTO_INCREMENT, company_symbol varchar(45) DEFAULT NULL, date datetime DEFAULT NULL, open double DEFAULT NULL, high double DEFAULT NULL, low double DEFAULT NULL, close double DEFAULT NULL, volume int DEFAULT NULL, total_revenue double DEFAULT NULL, net_income double DEFAULT NULL, total_expenses double DEFAULT NULL, cost_of_revenue double DEFAULT NULL, margin double DEFAULT NULL, margin_percent double DEFAULT NULL, margin_change double DEFAULT NULL, margin_change_percent double DEFAULT NULL, margin_low double DEFAULT NULL, margin_high double DEFAULT NULL, margin_percent_low double DEFAULT NULL, margin_percent_high double DEFAULT NULL, margin_change_low double DEFAULT NULL, margin_change_high double DEFAULT NULL, margin_change_percent_low double DEFAULT NULL, margin_change_percent_high double DEFAULT NULL, created_at timestamp DEFAULT CURRENT_TIMESTAMP, updated_at timestamp DEFAULT CURRENT_TIMESTAMP);	1000 row(s) returned	0.000 sec / 0.000 sec

## Create Historical Details table and Insert values

```
CREATE TABLE `historical_details` (
```

```
    `hd_id` int NOT NULL AUTO_INCREMENT,  
  
    `company_symbol` varchar(45) DEFAULT NULL,  
  
    `date` datetime DEFAULT NULL,  
  
    `open` double DEFAULT NULL,  
  
    `high` double DEFAULT NULL,
```

```

`low` double DEFAULT NULL,
`close` double DEFAULT NULL,
`adj_close` double DEFAULT NULL,
`volume` double DEFAULT NULL,
PRIMARY KEY (`hd_id`),
KEY `comp_hd_idx` (`company_symbol`),
CONSTRAINT `comp_hd` FOREIGN KEY (`company_symbol`) REFERENCES `companies_table`(`company_symbol`)
ON DELETE CASCADE ON UPDATE CASCADE
)

```

```

insert into historical_details(company_symbol,date,open,high,low,close,adj_close,volume)
select symbol,Date,open,high,low,close,'Adj Close',volume from historical_data;

```

```

select * from historical_details

```

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The current schema is 'stocks'.
- SQL Editor:** Contains the following SQL code:

```

37 -- KEY `comp_hd_idx` ('company_symbol'),
38 -- CONSTRAINT `comp_hd` FOREIGN KEY (`company_symbol`) REFERENCES `companies_table`(`company_symbol`)
39 -- ON DELETE CASCADE ON UPDATE CASCADE
40 -- insert into historical_details(company_symbol,date,open,high,low,close,adj_close,volume) select symbol,Date,open,high,low,close,
41 -- select * from historical_details
42
43 -- Create Dividends Table and Insert values
44 -- CREATE TABLE `dividends` (

```
- Result Grid:** Displays 12 rows of historical data for symbol 'MMM' from January 2012 to November 2013. The columns are: hd\_id, company\_symbol, date, open, high, low, close, adj\_close, volume.

hd_id	company_symbol	date	open	high	low	close	adj_close	volume
1	MMM	2012-01-12 00:00:00	91.40004	94.139999	89.599998	92.849998	69.440949	54850200
2	MMM	2013-01-10 00:00:00	94.19002	101.949997	93.959999	100.550003	75.199509	62614800
3	MMM	2013-01-12 00:00:00	101.44002	104.559998	100.580002	104	77.779724	4790500
4	MMM	2013-01-13 00:00:00	103.389992	106.879997	102.589996	106.309998	79.998344	50128700
5	MMM	2013-01-14 00:00:00	106	108.720001	102.889999	104.709999	78.794319	67230500
6	MMM	2013-01-15 00:00:00	104.779999	112.379997	104.419998	110.269997	82.978232	62323200
7	MMM	2013-01-16 00:00:00	110.699997	113.25	107.150002	109.349999	82.755089	53296100
8	MMM	2013-01-17 00:00:00	108.370003	118.599998	108.209999	117.43	88.869934	52519700
9	MMM	2013-01-18 00:00:00	118.379997	119.260002	112.360001	115.580002	85.955030	44047400
10	MMM	2013-01-19 00:00:00	119.57	122.699997	112.949998	119.410004	90.868294	53217500
11	MMM	2013-01-10 00:00:00	119.690002	126.379997	116.650002	125.849998	95.769028	54356200
12	MMM	2013-01-11 00:00:00	126.760002	134.160004	124.860001	133.509995	101.598091	45827900
- Output:** Shows the results of three previous queries:

#	Time	Action	Message	Duration / Fetch
1	16:24:14	select * from sectors	-- Create Companies Table and Insert values - CREATE TABLE `companies_table` ... 10 row(s) returned	0.078 sec / 0.000 sec
2	16:24:40	select * from companies_table	-- Create Historical Details table and Insert values - CREATE TABLE `historical_details` ... 53 row(s) returned	0.000 sec / 0.000 sec
3	16:25:28	select * from historical_details	-- Create Dividends Table and Insert values - CREATE TABLE `dividends` ... 1000 row(s) returned	0.000 sec / 0.015 sec

## Create margins table and Insert values

---

```
CREATE TABLE `margins` (
    `margin_id` int NOT NULL AUTO_INCREMENT,
    `company_symbol` varchar(45) DEFAULT NULL,
    `date` datetime DEFAULT NULL,
    `gross_profit` double DEFAULT NULL,
    `ebit` double DEFAULT NULL,
    `normalised_ebitda` double DEFAULT NULL,
    PRIMARY KEY (`margin_id`),
    KEY `margin_comp_idx` (`company_symbol`),
    CONSTRAINT `margin_comp` FOREIGN KEY (`company_symbol`) REFERENCES
    `companies_table` (`company_symbol`) ON DELETE CASCADE ON UPDATE CASCADE
)
insert into margins(company_symbol,date,gross_profit,ebit,normalised_ebitda) select
symbol,date,GrossProfit,ebit,NormalizedEBITDA from financials;
select * from margins
```

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'Local instance MySQL80' is selected. The main area has four tabs: 'SQL File 3\*', 'SQL Queries\*', 'SQL File 4\*', and 'companies sectors'. The 'SQL Queries\*' tab contains the following SQL code:

```

115    -- )
116    -- insert into margins(company_symbol,date,gross_profit,ebit,normalised_ebitda) select symbol,date,GrossProfit,ebit,NormalizedEBITDA
117    select * from margins
118    |
119    -- Create balance sheets table and Insert values
120    -- CREATE TABLE `balance_sheets` (
121    -- `bs_id` int NOT NULL AUTO_INCREMENT,
122    -- `company_symbol` varchar(45) DEFAULT NULL,

```

The 'Result Grid' pane below displays the results of the last query, showing 12 rows of data:

margin_id	company_symbol	date	gross_profit	ebit	normalised_ebitda
1	AAP	2022-09-30 00:00:00	1179851000	159457000	225990000
2	AAP	2022-06-30 00:00:00	1185719000	200971000	264081000
3	AAP	2022-03-31 00:00:00	1506520000	195980000	288987000
4	AAP	2021-12-31 00:00:00	1072117000	109671000	174867000
5	AAP	2021-09-30 00:00:00	1182454000	231080000	290434000
6	AAP	2021-06-30 00:00:00	1189251000	246071000	304129000
7	AAP	2021-03-31 00:00:00	1484926000	256965000	334218000
8	AAP	2020-12-31 00:00:00	1083696000	150040000	207210000
9	AAP	2020-09-30 00:00:00	1128471000	209463000	315175000
10	AAP	2020-06-30 00:00:00	1096714000	265962000	322604000
11	AAP	2020-03-31 00:00:00	1172733000	72436000	151015000
12	AAP	2019-12-31 00:00:00	928769000	127875000	186681000

The 'Output' pane at the bottom shows the execution log:

#	Time	Action	Message	Duration / Fetch
6	16:26:13	select * from statistics	- Create Financials table and Insert values - CREATE TABLE `Financials`_ta... 1000 row(s) returned	0.000 sec / 0.000 sec
7	16:26:38	select * from financials_table	- Create margins table and Insert values - CREATE TABLE `margins` ( ... 1000 row(s) returned	0.016 sec / 0.000 sec
8	16:26:51	select * from financials_table	- Create margins table and Insert values - CREATE TABLE `margins` ( ... 1000 row(s) returned	0.000 sec / 0.000 sec
9	16:27:03	select * from margins	- Create balance sheets table and Insert values - CREATE TABLE `balance_sh... 1000 row(s) returned	0.000 sec / 0.000 sec

## Create balance sheets table and Insert values

```

CREATE TABLE `balance_sheets`(
`bs_id` int NOT NULL AUTO_INCREMENT,
`company_symbol` varchar(45) DEFAULT NULL,
`date` datetime DEFAULT NULL,
`total_assets` double DEFAULT NULL,
`current_assets` double DEFAULT NULL,
`net_ppe` double DEFAULT NULL,
`gross_ppe` double DEFAULT NULL,
`total_debt` double DEFAULT NULL,
`net_debt` double DEFAULT NULL,
`current_debt` varchar(45) DEFAULT NULL,

```

PRIMARY KEY (`bs\_id`),

KEY `bs\_comp\_idx` (`company\_symbol`),

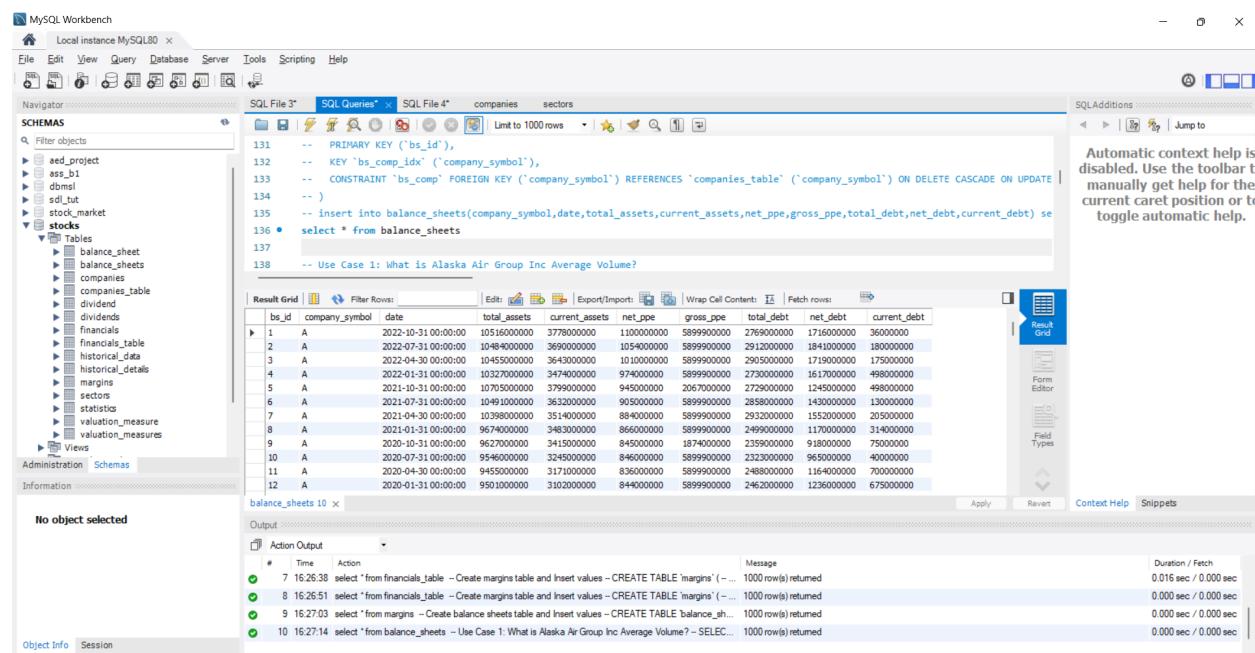
CONSTRAINT `bs\_comp` FOREIGN KEY (`company\_symbol`) REFERENCES `companies\_table`(`company\_symbol`) ON DELETE CASCADE ON UPDATE CASCADE

)

insert into

balance\_sheets(company\_symbol,date,total\_assets,current\_assets,net\_ppe,gross\_ppe,total\_debt,net\_debt,current\_debt) select  
symbol,quarter\_end\_date,totalassets,currentassets,netppe,grossppe,totaldebt,netdebt,curr  
entdebt from balance\_sheet;

select \* from balance\_sheets



The screenshot shows the MySQL Workbench interface. The SQL Editor tab contains the following code:

```
131 -- PRIMARY KEY ('bs_id'),
132 -- KEY `bs_comp_idx` (`company_symbol`),
133 -- CONSTRAINT `bs_comp` FOREIGN KEY (`company_symbol`) REFERENCES `companies_table`(`company_symbol`) ON DELETE CASCADE ON UPDATE CASCADE
134 --
135 -- insert into balance_sheets(company_symbol,date,total_assets,current_assets,net_ppe,gross_ppe,total_debt,net_debt,current_debt) se
136 • select * from balance_sheets
137
138 -- Use Case 1: What is Alaska Air Group Inc Average Volume?
```

The Results Grid displays the following data:

bs_id	company_symbol	date	total_assets	current_assets	net_ppe	gross_ppe	total_debt	net_debt	current_debt
1	A	2022-10-31 00:00:00	1051600000	377800000	110000000	589990000	279000000	171600000	3600000
2	A	2022-07-31 00:00:00	1049400000	369000000	105400000	589990000	291200000	181000000	18000000
3	A	2022-04-30 00:00:00	1045500000	364300000	101000000	589990000	290500000	171900000	17500000
4	A	2022-01-31 00:00:00	1032700000	347400000	97400000	589990000	273000000	161000000	49800000
5	A	2021-08-31 00:00:00	1070500000	379900000	94500000	589990000	205700000	272900000	124000000
6	A	2021-05-31 00:00:00	1050400000	365000000	96300000	589990000	226000000	143000000	19000000
7	A	2021-02-28 00:00:00	1039800000	351400000	88400000	589990000	291200000	155200000	20500000
8	A	2021-09-30 00:00:00	967490000	348200000	86500000	589990000	249900000	117000000	31400000
9	A	2020-12-31 00:00:00	962700000	341500000	84500000	589990000	187400000	239900000	91800000
10	A	2020-07-31 00:00:00	954690000	324600000	84600000	589990000	232300000	96500000	40000000
11	A	2020-04-30 00:00:00	945500000	317100000	83600000	589990000	248800000	116400000	70000000
12	A	2020-01-31 00:00:00	950100000	310200000	84400000	589990000	246200000	123600000	67500000

The Action Output pane shows the following log entries:

#	Time	Action	Message	Duration / Fetch
7	16:26:38	select * from financials_table -- Create margins table and Insert values - CREATE TABLE 'margins' ( ... ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4; -- 1000 row(s) returned		0.016 sec / 0.000 sec
8	16:26:51	select * from financials_table -- Create margins table and Insert values - CREATE TABLE 'margins' ( ... ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4; -- 1000 row(s) returned		0.000 sec / 0.000 sec
9	16:27:03	select * from margins -- Create balance sheets table and Insert values - CREATE TABLE 'balance_sh... -- 1000 row(s) returned		0.000 sec / 0.000 sec
10	16:27:14	select * from balance_sheets -- Use Case 1: What is Alaska Air Group Inc Average Volume? -- SELECT ... -- 1000 row(s) returned		0.000 sec / 0.000 sec

## Views:

## Use Case: What is Alaska Air Group Inc Average Volume?

Create view alaska\_average\_volume as

```
SELECT b.company_name, avg(a.Volume) AS Avg_Volume
```

FROM historical\_details a

INNER JOIN companies\_table b

ON a.company\_symbol=b.company\_symbol AND b.company\_name = 'Alaska Air Group Inc'

```
GROUP BY b.company_symbol;
```

```
Select * from alaska_average_volume
```

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Navigator:** Shows the database schema with the 'stocks' schema expanded, containing tables like 'balance\_sheet', 'balance\_sheets', 'companies', etc.
- SQL Editor:** Contains a query to create a view 'alaska\_average\_volume' and select from it.
- Result Grid:** Displays the output of the query, showing a single row for 'Alaska Air Group Inc' with an average volume of 35520132.5.
- Output Tab:** Shows the execution log with two entries: creating the view and selecting from it.

**Use Case: What is the average PeRatio where sector id is 2?**

Create view avg\_PeRatio as

```
SELECT b.sector_id, avg(a.pe_ratio) AS Average_PeRatio FROM statistics a
```

```
INNER JOIN companies_table b ON a.company_symbol=b.company_symbol AND b.sector_id = '2' GROUP BY b.sector_id;
```

Select \* from avg\_PeRatio

```
-- Create view avg_PeRatio as
SELECT b.sector_id, avg(a.pe_ratio) AS Average_PeRatio
FROM statistics a
INNER JOIN companies_table b
ON a.company_symbol=b.company_symbol
AND b.company_name = 'Alaska Air Group Inc'
GROUP BY b.sector_id;
Select * from avg_PeRatio
```

### Use Case: What is the maximum ebit of Carnival Corp. ?

Create view Max\_ebit as

```
SELECT b.company_name, max(a.ebit) FROM margins a
INNER JOIN companies_table b
ON a.company_symbol=b.company_symbol
AND b.company_name = 'Carnival Corp.'
group by b.company_name;
```

Select \* from Max\_ebit

The screenshot shows the MySQL Workbench interface with several tabs open:

- SQL File 3\***: Contains SQL code for calculating average volume.
- SQL Queries\***: Contains SQL code for calculating average PeRatio and maximum EBIT.
- SQL File 4\***: Contains SQL code for creating a view of maximum EBIT.

The results pane shows the output of the last query:

```
Max_ebit 2 x
Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid
+-----+
| company_name | max(a.ebit) |
+-----+
| Carnival Corp. | 1879000000 |
+-----+
```

The log pane shows the execution history:

#	Time	Action	Message	Duration / Fetch
1	21:51:12	Create view avg_PeRatio as SELECT b.sector_id, avg(a.pe_ratio) AS Average_PeRatio FROM statistics a INNER JOIN companies_table b ON a.company_symbol=b.company_symbol	0 row(s) affected	0.062 sec
2	21:51:12	Select * from avg_PeRatio -- Use Case 3 : What is the maximum ebt of Carnival Corp. ? - Create view ... 1 row(s) returned		0.031 sec / 0.000 sec
3	21:51:40	Create view Max_ebt as SELECT b.company_name, max(a.ebit) FROM margins a INNER JOIN companies_table b ON a.company_symbol=b.company_symbol	0 row(s) affected	0.016 sec
4	21:51:40	Select * from Max_ebt -- Use Case 4: What was the TotalAssets of American Airline Group on 9/30/2022? 1 row(s) returned		0.000 sec / 0.000 sec

## Use Case: What was the TotalAssets of American Airline Group on 9/30/2022?

Create view AA\_Total\_Assets as

```
SELECT b.company_name, a.total_assets FROM balance_sheets a INNER JOIN
companies_table b ON a.company_symbol=b.company_symbol
```

```
AND b.company_name = 'American Airlines Group' AND a.date = '2022-09-30';
```

Select \* from AA\_Total\_Assets

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'Local instance MySQL80' is selected. The main area contains four tabs: 'SQL File 3\*', 'SQL Queries\*', 'SQL File 4\*', and 'companies'. The 'SQL Queries\*' tab is active, displaying the following SQL code:

```

485 -- SELECT b.sector_id, avg(a.pe_ratio) AS Average_PeRatio FROM statistics a
486 -- INNER JOIN companies_table b ON a.company_symbol=b.company_symbol AND b.sector_id = '2' GROUP BY b.sector_id;
487 -- Select * from avg_PeRatio
488
489 -- Use Case 3 : What is the maximum ebit of Carnival Corp. ?
490 -- Create view Max_ebit as
491 -- SELECT b.company_name, max(a.ebit) FROM margins a INNER JOIN companies_table b ON a.company_symbol=b.company_symbol AND b.company_
492 -- Select * from Max_ebit
493
494 -- Use Case 4: What was the TotalAssets of American Airline Group on 9/30/2022?
495 • Create view AA_Total_Assets as
496 SELECT b.company_name, a.total_assets FROM balance_sheets a INNER JOIN companies_table b ON a.company_symbol=b.company_symbol
497 AND b.company_name = 'American Airlines Group' AND a.date = '2022-09-30';
498 • Select * from AA_Total_Assets
499

```

The results grid shows one row for 'American Airlines Group' with a total assets value of 66652000000.

On the right side of the interface, there is a message box stating: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

## Use case: Which companies have PsRatio between 2-3?

Create view ps\_ratio as

SELECT DISTINCT b.company\_name

FROM statistics a

INNER JOIN companies\_table b

ON a.company\_symbol=b.company\_symbol

WHERE a.ps\_ratio BETWEEN 2 AND 3;

Select \* from ps\_ratio;

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'Local instance MySQL80' is selected. The main area contains two tabs: 'SQL File 3\*' and 'doctor'. The 'doctor' tab is active, displaying the following SQL code:

```

507    -- WHERE
508    --     sectors.Sector_name = 'Energy';
509    -- Select * from pb_ratio
510
511
512    -- Use case 13: Which companies have PsRatio between 2-3?
513 • Create view ps_ratio as
514     SELECT DISTINCT b.company_name
515     FROM statistics a
516     INNER JOIN companies_table b
517         ON a.company_symbol=b.company_symbol
518         WHERE a.ps_ratio BETWEEN 2 AND 3;
519 • Select * from ps_ratio;
520

```

Below the code, the 'Result Grid' shows the output of the query:

company_name
3M Company
A.O. Smith Corp
Abbott Laboratories
Affiliated Managers Group Inc
Agilent Technologies Inc

At the bottom, the 'Output' section displays the execution log:

#	Time	Action	Message	Duration / Fetch
5	17:50:58	Create view pb_ratio as Select companies_table company_Symbol, sectors.Sector_name, statistics...	0 row(s) affected	0.015 sec
6	17:50:58	Select * from pb_ratio - Use case 13: Which companies have PsRatio between 2-3? - SELECT DIS...	15 row(s) returned	0.000 sec / 0.000 sec
7	17:51:58	Create view ps_ratio as SELECT DISTINCT b.company_name FROM statistics a INNER JOIN compan...	0 row(s) affected	0.031 sec
8	17:51:58	Select * from ps_ratio LIMIT 0, 1000	30 row(s) returned	0.015 sec / 0.000 sec

## Use case: What is the list of companies under Industrials?

Create view list\_companies as

```
select a.company_name, b.sector_name
```

```
from companies_table a
```

```
join sectors b
```

```
on a.sector_id = b.sector_id
```

```
where b.sector_name = "Industrials";
```

```
select * from list_companies;
```

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'Local instance MySQL80' is selected. The main area contains two tabs: 'SQL File 3\*' and 'doctor'. The 'doctor' tab displays the following SQL code:

```

516 -- INNER JOIN companies_table b
517 --   ON a.company_symbol=b.company_symbol
518 -- WHERE a.ps_ratio BETWEEN 2 AND 3;
519 -- Select * from ps_ratio;
520
521 -- Use case 14: What is the list of companies under Industrials?
522 • Create view list_companies as
523 select a.company_name, b.sector_name
524 from companies_table a
525 join sectors b
526 on a.sector_id = b.sector_id
527 where b.sector_name = "Industrials";
528 • select * from list_companies;
529

```

The 'Result Grid' shows the output of the 'list\_companies' view:

company_name	sector_name
American Airlines Group	Industrials
Alaska Air Group Inc	Industrials
AMETEK Inc	Industrials
A.O. Smith Corp	Industrials
Acuity Brands Inc	Industrials
Boeing Company	Industrials

The 'Output' section shows the execution log:

#	Time	Action	Message	Duration / Fetch
7	17:51:58	Create view ps_ratio as SELECT DISTINCT b.company_name FROM statistics a INNER JOIN companies_table b ON a.company_symbol = b.company_symbol WHERE a.ps_ratio >= 2 AND a.ps_ratio <= 3	0 row(s) affected	0.031 sec
8	17:51:58	Select * from ps_ratio LIMIT 0, 1000	30 row(s) returned	0.015 sec / 0.000 sec
9	17:53:09	Create view list_companies as select a.company_name, b.sector_name from companies_table a join sectors b on a.sector_id = b.sector_id where b.sector_name = 'Industrials'	0 row(s) affected	0.031 sec
10	17:53:09	select * from list_companies LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec

## Use case: What is the total assets of Health Care sector in the quarter ending '07/31/2021'

Create view total\_assets as

SELECT b.Sector\_name,

sum(a.total\_assets) as total\_assets

FROM balance\_sheets a

JOIN companies\_table c

join sectors b

ON a.company\_symbol = c.company\_symbol

and c.sector\_id = b.sector\_id

AND b.sector\_name = 'Health Care'

WHERE a.date = '2021-07-31'

GROUP BY b.sector\_name;

Select \* from total\_assets

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'Local instance MySQL80' is selected. The main area displays a SQL editor with the following code:

```
-- Use case 15: what is the total assets of Health Care sector in the quarter ending '07/31/2021'
Create view total_assets as
SELECT b.Sector_name,
       sum(a.total_assets) as total_assets
  FROM balance_sheets a
  JOIN companies_table c
  JOIN sectors b
     ON a.company_symbol = c.company_symbol
    and c.sector_id = b.sector_id
    AND b.sector_name = 'Health Care'
 WHERE a.date = '2021-07-31'
 GROUP BY b.sector_name;
Select * from total_assets;
```

The results grid shows one row:

Sector_name	total_assets
Health Care	1049100000

In the bottom pane, the 'Action Output' section shows the following log entries:

#	Action	Time	Message	Duration / Fetch
12	Select * from total LIMIT 0, 1000	17:54:03	1 row(s) returned	0.000 sec / 0.000 sec
13	Create view total as SELECT b.Sector_name, sum(a.total_assets) as total_assets FROM balance_sh...	17:54:47	Error Code: 1050. Table 'total' already exists	0.016 sec
14	Create view total_assets as SELECT b.Sector_name, sum(a.total_assets) as total_assets FROM bal...	17:55:06	0 row(s) affected	0.032 sec
15	Select * from total_assets LIMIT 0, 1000	17:55:06	1 row(s) returned	0.000 sec / 0.000 sec

## Use case: What was the close price of 3M Company on 1/9/2017?

Create view 3M\_Company as

```
SELECT b.company_name, a.close AS Close_Price FROM historical_details a INNER JOIN
companies_table b
```

```
ON a.company_symbol=b.company_symbol AND b.company_name = '3M Company' AND
a.Date = '2017-01-09';
```

Select \* from 3M\_Company

```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Navigator Schemas SQL File 3* SQL Queries* SQL File 4* companies sectors
SCHEMAS Filter objects
aed_project
asx_b1
dbmsl
sdltut
stocks_market
Tables
balance_sheet
balance_sheets
companies
companies_table
dividend
dividends
financials
financials_table
historical_data
historical_details
margins
sp500
statistics
valuation_measure
valuation_measures
Views
Administration Schemas Information
No object selected
3M_Company 4*
Output
Action Output
# Time Action Message Duration / Fetch
5 21:52:16 Create view AA_Total_Assets as SELECT b.company_name, a.total_assets FROM balance_sheets a INNER JOIN companies_table b ON a.company_symbol=b.company_symbol AND b.company_name = 'American Airline Group' AND a.date = '2022-09-30'; 0 row(s) affected 0.015 sec
6 21:52:16 Select * from AA_Total_Assets - Use case 4: What was the TotalAssets of American Airline Group on 9/30/2022? 1 row(s) returned 0.000 sec / 0.000 sec
7 21:52:31 Create view 3M_Company as SELECT b.company_name, a.close AS Close_Price FROM historical_details a INNER JOIN companies_table b ON a.company_symbol=b.company_symbol AND b.company_name = '3M Company' AND a.Date = '2017-01-09'; 0 row(s) affected 0.016 sec
8 21:52:31 Select * from 3M_Company - Use case 5: What was the close price of 3M Company on 1/9/2017? 1 row(s) returned 0.000 sec / 0.000 sec

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid

company_name	Close_Price
3M Company	209.89994

## Major Use Cases:

### Use case: Five companies with highest MarketCap

SELECT t1.company\_name,

t1.avg\_MarketCap

FROM (SELECT b.company\_name,

avg(a.market\_cap) AS AVG\_MarketCap

FROM valuation\_measure a

INNER JOIN companies\_table b

ON a.company\_symbol=b.company\_symbol

GROUP BY b.company\_name) t1

WHERE 5 > (SELECT count(\*)

```

FROM (SELECT b.company_name,
           avg(a.market_cap) AS AVG_MarketCap
      FROM valuation_measure a
     INNER JOIN companies_table b
           ON a.company_symbol=b.company_symbol
      GROUP BY b.company_name) t2

WHERE t1.avg_MarketCap < t2.avg_MarketCap);

```

The screenshot shows the MySQL Workbench interface with a query editor titled "doctor". The code in the editor is a complex SQL query involving multiple joins and subqueries. The results pane below shows a table named "historical\_data" with columns "company\_name" and "avg\_MarketCap", listing several companies and their average market caps. The status bar at the bottom indicates the query was executed in 0.046 sec / 0.000 sec.

```

-- Use case 6
SELECT t1.company_name,
       t1.avg_MarketCap
  FROM (SELECT b.company_name,
               avg(a.market_cap) AS AVG_MarketCap
          FROM valuation_measure a
         INNER JOIN companies_table b
               ON a.company_symbol=b.company_symbol
      GROUP BY b.company_name) t1
 WHERE 5 > (SELECT count(*)
              FROM (SELECT b.company_name,
                           avg(a.market_cap) AS AVG_MarketCap
                      FROM valuation_measure a
                     INNER JOIN companies_table b
                           ON a.company_symbol=b.company_symbol
                  GROUP BY b.company_name) t2
              WHERE t1.avg_MarketCap < t2.avg_MarketCap);

```

company_name	avg_MarketCap
Apple Inc.	1097536544420.675
Abbott Laboratories	115224050886.02563
Augen Inc	116276217129.3077
Boeing Company	111979471537.5
Bristol-Myers Squibb	101347933725.72728

## Use Case: How's healthcare stocks EBIT going in past 3 years?

```
SELECT companies_table.company_symbol,margins.Date,margins.ebit
```

```
FROM companies_table
```

```
INNER JOIN margins
```

```
join sectors
```

ON companies\_table.company\_symbol = margins.company\_symbol

WHERE sectors.Sector\_name = 'Health Care'

AND margins.Date BETWEEN '2019-01-01' AND '2022-12-01'

The screenshot shows the MySQL Workbench interface. The SQL editor tab contains the following query:

```
-- ON companies_table.company_symbol = financials_table.company_symbol
-- WHERE financials_table.Total_Expenses > '3,000,000,000'
-- AND financials_table.Date BETWEEN '2019-01-01' AND '2022-12-01'

-- 24. How's healthcare stocks' EBIT going in past 3 years?
SELECT companies_table.company_symbol,margins.Date,margins.ebit
FROM companies_table
INNER JOIN margins
join sectors
ON companies_table.company_symbol = margins.company_symbol
WHERE sectors.Sector_name = 'Health Care'
AND margins.Date BETWEEN '2019-01-01' AND '2022-12-01'
```

The Result Grid shows the following data:

company_symbol	Date	ebit
AAL	2022-09-30 00:00:00	1157000000
AAL	2022-06-30 00:00:00	1071000000
AAL	2022-03-31 00:00:00	-1623000000
AAL	2021-12-31 00:00:00	-723000000
AAL	2021-09-30 00:00:00	682000000

The Action Output pane shows the following log entries:

#	Time	Action	Message	Duration / Fetch
428	21:22:05	SELECT companies_table.company_symbol.financials_table.Date.financials_table.Total_Expenses FROM ...	789 row(s) returned	0.015 sec / 0.000 sec
429	21:26:26	SELECT companies_table.company_symbol.financials_table.Date.financials_table.ebit FROM companies_table	Error Code: 1054. Unknown column 'financials_table.ebit' in field list	0.078 sec
430	21:27:14	SELECT companies_table.company_symbol.margins.Date.margins.ebit FROM companies_table INNER JOIN margins	0 row(s) returned	0.078 sec / 0.000 sec
431	21:27:28	SELECT companies_table.company_symbol.margins.Date.margins.ebit FROM companies_table INNER JOIN margins	790 row(s) returned	0.000 sec / 0.000 sec

### Use case : Ten companies with highest TotalAssets

SELECT t1.company\_name,

t1.AVG\_TotalAssets

FROM (SELECT b.company\_name,

avg(a.total\_assets) AS AVG\_TotalAssets

FROM balance\_sheets a

INNER JOIN companies\_table b

ON a.company\_symbol=b.company\_symbol

GROUP BY b.company\_name) t1

WHERE 11 > (SELECT count(\*)

```
FROM (SELECT b.company_name,
           avg(a.total_assets) AS AVG_TotalAssets
      FROM balance_sheets a
     INNER JOIN companies_table b
            ON a.company_symbol=b.company_symbol
       GROUP BY b.company_name) t2
 WHERE t1.AVG_TotalAssets < t2.AVG_TotalAssets);
```

The screenshot shows the MySQL Workbench interface. The SQL Editor tab contains the complex query shown above. The Results Grid tab displays the output of the query, which includes 13 rows of company names and their average total assets. The bottom pane shows the execution history with four log entries.

company_name	AVG_TotalAssets
American Airlines Group	50840116279.06977
Apple Inc.	298436186046.51166
Abbott Laboratories	59779621139.53488
Archer-Daniels-Midland Co	4424266666.66664
American Electric Power	67570211627.06975

Action	Time	Action	Message	Duration / Fetch
464	22:12:24	Select companies_table.company_Symbol, sectors.Sector_name, statistics.date, statistics.Pb_Ratio ...	15 row(s) returned	0.000 sec / 0.000 sec
465	22:14:33	SELECT t1.company_name, t1.AVG_TotalAssets FROM (SELECT b.company_name, avg(a.total_assets) AS AVG_TotalAssets	Error Code: 1054. Unknown column 't1' in 'where clause'	0.079 sec
466	22:14:56	SELECT t1.company_name, t1.AVG_TotalAssets FROM (SELECT b.company_name, avg(a.total_assets) AS AVG_TotalAssets	Error Code: 1054. Unknown column 'date' in 'where clause'	0.000 sec
467	22:17:00	SELECT t1.company_name, t1.AVG_TotalAssets FROM (SELECT b.company_name, avg(a.total_assets) AS AVG_TotalAssets	avg(a.total_assets) AS AVG_TotalAssets	0.093 sec / 0.000 sec

## Use case: What is the max Dividend of Companies under utilities?

```
SELECT c.sector_name, b.company_name, max(a.dividend_value) AS max_divident
      FROM dividends a
     INNER JOIN companies_table b
```

join sectors c

ON a.company\_symbol=b.company\_symbol

and c.sector\_id = b.sector\_id

AND c.sector\_name = 'Utilities'

GROUP BY b.Sector\_id, b.company\_name;

The screenshot shows the MySQL Workbench interface with a SQL editor window containing the following query:

```
-- Use case 10
SELECT c.sector_name, b.company_name, max(a.dividend_value) AS max_dividend
FROM dividends a
INNER JOIN companies_table b
JOIN sectors c
    ON a.company_symbol=b.company_symbol
    and c.sector_id = b.sector_id
    AND c.sector_name = 'Utilities'
GROUP BY b.Sector_id, b.company_name;
```

The Result Grid shows the following data:

sector_name	company_name	max_dividend
Utilities	Ameren Corp	0.59
Utilities	American Electric Power	0.83
Utilities	AES Corp	0.158
Utilities	American Water Works Company Inc	0.655

The Action Output pane shows the following log entries:

#	Time	Action	Message	Duration / Fetch
352	19:25:33	SELECT t1.company_name, t1.avg_CostofRevenue FROM (SELECT b.company_name, avg(...)	51 row(s) returned	0.016 sec / 0.000 sec
353	19:27:31	select b.company_name,min(a.cost_of_revenue) as Min_Cost_of_Revenue from financials_table a inner join co...	53 row(s) returned	0.000 sec / 0.000 sec
354	19:28:10	select b.company_name,a.cost_of_revenue as Min_Cost_of_Revenue from financials_table a inner join co...	1 row(s) returned	0.016 sec / 0.000 sec
355	19:33:37	SELECT c.sector_name, b.company_name, max(a.Divident) AS max_dividend FROM dividends a INNER J...	Error Code: 1054. Unknown column 'a.Divident' in field list	0.016 sec
356	19:33:55	SELECT c.sector_name, b.company_name, max(a.dividend_value) AS max_dividend FROM dividends a IN...	4 row(s) returned	0.031 sec / 0.000 sec

**Use case: What is the total assets of Health Care sector in the quarter ending**

**'07/31/2021'**

SELECT b.Sector\_name,c.company\_name,

sum(a.total\_assets) as total\_assets

FROM balance\_sheets a

JOIN companies\_table c

join sectors b

ON a.company\_symbol = c.company\_symbol

and c.sector\_id = b.sector\_id

AND b.sector\_name = 'Health Care'

WHERE a.date = '2021-07-31'

GROUP BY b.sector\_name;

The screenshot shows the MySQL Workbench interface. In the top-left, the Navigator pane displays the database schema with tables like 'companies', 'balance\_sheets', 'sectors', and 'historical\_data'. The central area contains a SQL editor window titled 'doctor' with the following query:

```
-- Use case 15
SELECT b.Sector_name,c.company_name,
       sum(a.total_assets) as total_assets
  FROM balance_sheets a
  JOIN companies_table c
  JOIN sectors b
 WHERE a.company_symbol = c.company_symbol
   AND c.sector_id = b.sector_id
   AND b.sector_name = 'Health Care'
  WHERE a.date = '2021-07-31'
 GROUP BY b.sector_name;

```

Below the editor is a 'Result Grid' showing one row of data:

Sector_name	company_name	total_assets
Health Care	Agilent Technologies Inc	1049100000

At the bottom, the 'Result 111' tab shows the execution history:

#	Time	Action	Message	Duration / Fetch
358	19:39:53	SELECT DISTINCT b.company_name FROM valuation_measure a INNER JOIN companies_table b ON a...	Error Code: 1054. Unknown column 'ps_ratio' in 'where clause'	0.016 sec
359	19:40:12	SELECT DISTINCT b.company_name FROM statistics a INNER JOIN companies_table b ON a.company_...	30 row(s) returned	0.032 sec / 0.000 sec
360	19:43:38	select a.company_name,b.sector_name from companies_table a join sectors b on a.sector_id = b.sector_id	10 row(s) returned	0.016 sec / 0.000 sec
361	19:46:39	SELECT b.Sector_name,c.company_name, sum(a.total_assets) as total_assets FROM balance_sheets a ...	0 row(s) returned	0.031 sec / 0.000 sec
362	19:46:47	SELECT b.Sector_name,c.company_name, sum(a.total_assets) as total_assets FROM balance_sheets a ...	1 row(s) returned	0.016 sec / 0.000 sec

## Use Case: Which health care stock has the highest market capitalization in past ten years?

SELECT

companies\_table.company\_symbol,companies\_table.sector\_id,sectors.sector\_name,max(valuation\_measure.market\_cap)

FROM companies\_table

INNER JOIN valuation\_measure

join sectors

ON companies\_table.company\_symbol = valuation\_measure.company\_symbol

and sectors.sector\_id = companies\_table.sector\_id

AND sectors.sector\_name = 'HEALTH CARE'

GROUP BY companies\_table.company\_symbol

The screenshot shows the MySQL Workbench interface. In the top-left, the Navigator pane displays the schema structure for the 'doctor' database, including tables like 'companies', 'companies\_table', 'valuation\_measure', and 'sectors'. The main area contains a SQL editor window with the following query:

```
-- 1. Which health care stock has the highest market capitalization in past ten years?
SELECT companies_table.company_symbol, companies_table.sector_id, sectors.sector_name, max(valuation_measure.market_cap)
FROM companies_table
INNER JOIN valuation_measure
    ON companies_table.company_symbol = valuation_measure.company_symbol
        AND sectors.sector_id = companies_table.sector_id
        AND sectors.sector_name = 'HEALTH CARE'
GROUP BY companies_table.company_symbol
```

Below the SQL editor is a Result Grid showing the output of the query:

company_symbol	sector_id	sector_name	max(valuation_measure.market_cap)
A	2	Health Care	47675792983
ABC	2	Health Care	32596597243
ABT	2	Health Care	246276927842
AMGN	2	Health Care	148302360000
BAX	2	Health Care	44437702928

At the bottom, the Output pane shows the history of actions taken:

#	Time	Action	Message	Duration / Fetch
361	19:46:39	SELECT b.Sector_name,c.company_name, sum(a.total_assets) as total_assets FROM balance_sheets a ...	0 row(s) returned	0.031 sec / 0.000 sec
362	19:46:47	SELECT b.Sector_name,c.company_name, sum(a.total_assets) as total_assets FROM balance_sheets a ...	1 row(s) returned	0.016 sec / 0.000 sec
363	19:55:32	SELECT companies_table.company_symbol,companies_table.sector_id,sectors.sector_name,max(valuation... Error Code: 1054: Unknown column 'companies_table.symbol' in 'on clause'	Error Code: 1054: Unknown column 'companies_table.symbol' in 'on clause'	0.109 sec
364	19:55:43	SELECT companies_table.company_symbol,companies_table.sector_id,sectors.sector_name,max(valuation... Error Code: 1054: Unknown column 'companies_table.symbol' in 'group statement'	Error Code: 1054: Unknown column 'companies_table.symbol' in 'group statement'	0.000 sec
365	19:55:49	SELECT companies_table.company_symbol,companies_table.sector_id,sectors.sector_name,max(valuation... 8 row(s) returned	8 row(s) returned	0.063 sec / 0.000 sec

## Use Case: How does company BMY's stock price change from 2012 to 2022?

Select company\_symbol,date,open,high,low,close From historical\_details GROUP BY company\_symbol HAVING company\_symbol = 'BMY'

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'Local instance MySQL800' and 'MySQL Model' are selected. The main area contains a SQL editor titled 'doctor' with the following code:

```

181 -- AND sectors.sector_name = 'HEALTH CARE'
182 -- GROUP BY companies_table.company_symbol
183
184 -- 2. What is the total revenue and cost of revenue of company AAP in 2022?
185 -- SELECT company_symbol, date, total_revenue, cost_of_revenue
186 -- FROM financials_table
187 -- WHERE year(date) = '2022'
188 -- Group by Company_Symbol, date, Total_Revenue, Cost_Of_Revenue
189 -- Having Company_Symbol = 'AAP'
190
191 -- 3. How does company BMY's stock price change from 2012 to 2022?
192 • Select company_symbol, date, open, high, low, close From historical_details GROUP BY company_symbol HAVING company_symbol = 'BMY'

```

The 'Result Grid' shows the following data for BMY:

company_symbol	date	open	high	low	close
BMY	2012-01-12 00:00:00	32.709999	33.549999	31.85	32.59

The bottom section shows the 'historical\_details114' output log:

Action	Time	Action	Message	Duration / Fetch
365	19:55:49	SELECT companies_table.company_symbol, companies_table.sector_id, sectors.sector_name, max(valuation)	8 row(s) returned	0.063 sec / 0.000 sec
366	19:58:51	SELECT CompanySymbol, Date, TotalRevenue, CostOfRevenue FROM StockFinancials WHERE right(Date, ...)	Error Code: 1049. Unknown database 'stock'	0.078 sec
367	20:01:11	SELECT company_symbol, date, total_revenue, cost_of_revenue FROM financials_table WHERE year(date) ...	3 row(s) returned	0.015 sec / 0.000 sec
368	20:07:52	Select company_symbol,date,open,high,low,close From historical_details GROUP BY symbol HAVING symb...	Error Code: 1054. Unknown column 'symbol' in 'group statement'	0.015 sec
369	20:08:03	Select company_symbol,date,open,high,low,close From historical_details GROUP BY company_symbol HA...	1 row(s) returned	0.063 sec / 0.000 sec

## Use Case: Which company has the highest market capitalization in the materials sector?

```

SELECT
    companies_table.company_symbol, sectors.sector_name, valuation_measure.Market_Cap
FROM companies_table
INNER JOIN valuation_measure
    ON companies_table.company_symbol = valuation_measure.company_symbol
    AND companies_table.sector_id = sectors.sector_id
    AND sectors.sector_name = 'Materials'
GROUP BY companies_table.company_symbol
ORDER BY Market_Cap desc

```

The screenshot shows the MySQL Workbench interface. In the top navigation bar, 'Local instance MySQL800' and 'MySQL Model' are selected. The main area contains a SQL editor titled 'doctor' with the following code:

```

228 -- ORDER BY MAX(balance_sheets.Total_Assets) desc
229
230 -- 11. Which company has the highest market capitalization in the materials sector?
231 • SELECT companies_table.company_symbol,sectors.sector_name,valuation_measure.Market_Cap
232 FROM companies_table
233 INNER JOIN valuation_measure
234 JOIN sectors
235 ON companies_table.company_symbol = valuation_measure.company_symbol
236 and companies_table.sector_id = sectors.sector_id
237 AND sectors.sector_name = 'Materials'
238 GROUP BY companies_table.company_symbol
239 ORDER BY Market_Cap desc
240
241

```

Below the code is a 'Result Grid' showing the results of the query:

company_symbol	sector_name	Market_Cap
APD	Materials	51619277080
ALB	Materials	24476155560
CF	Materials	19032991958
AVY	Materials	13220417418

On the right side of the interface, there is a 'SQLAdditions' panel with a message: 'Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.'

## Use Case: Which healthcare companies have had positive net income over the years?

SELECT

companies\_table.company\_symbol,sectors.Sector\_name,financials\_table.Date,financials\_table.Net\_Income

FROM companies\_table

INNER JOIN financials\_table

join sectors

ON companies\_table.company\_symbol = financials\_table.company\_symbol

WHERE sectors.Sector\_name = 'Health Care'

```

MySQL Workbench - Local instance MySQL800 - MySQL Model* - File Edit View Query Database Server Tools Scripting Help
SQL File 3* doctor
282 -- AND statistics.date = '2020-06-30'
283
284 -- 20. Which healthcare companies have had positive net income over the years?
285 • SELECT companies_table.company_symbol,sectors.Sector_name,financials_table.Date,financials_table.Net_Income
286 FROM companies_table
287 INNER JOIN financials_table
288 join sectors
289 ON companies_table.company_symbol = financials_table.company_symbol
290 WHERE sectors.Sector_name = 'Health Care'
291
292
293
294
295
Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows: Result Grid
company_symbol Sector_name Date Net_Income
AAL Health Care 2022-09-30 00:00:00 483000000
AAL Health Care 2022-06-30 00:00:00 476000000
AAL Health Care 2022-03-31 00:00:00 -163500000
AAL Health Care 2021-12-31 00:00:00 -932000000
AAL Health Care 2021-09-30 00:00:00 169000000

```

Table: historical\_data

Columns:

- Symbol
- Open
- High
- Low
- Close
- Volume
- date
- date\_time
- gross\_profit
- gross\_margin
- high\_low
- low\_high
- open\_close
- volume\_gross
- volume\_low
- volume\_high

Object Info Session

## Use Case: What are the gross profit of materials stocks in the past three years?

SELECT

companies\_table.company\_symbol,sectors.Sector\_name,margins.Date,margins.Gross\_Profit  
FROM companies\_table

INNER JOIN margins

join sectors

ON companies\_table.company\_symbol = margins.company\_symbol

WHERE sectors.Sector\_name = 'Materials'

AND margins.Date

BETWEEN '2019-01-01' AND '2022-12-01'

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree, which includes 'dividend', 'dividends', 'company\_symbol', 'date', 'dividend\_value', 'financials\_table', 'f\_id', 'gross\_profit', 'net\_income', 'revenue', 'total\_revenue', and 'total\_expenses'. The main SQL editor window contains the following query:

```

293 -- SELECT company_symbol, Date, dividend_value FROM dividends WHERE dividend_value BETWEEN 2 AND 4
294
295 -- 22. What are the gross profit of materials stocks in the past three years?
296 • SELECT companies_table.company_symbol, sectors.Sector_name, margins.Date, margins.Gross_Profit
297 FROM companies_table
298 INNER JOIN margins
299 JOIN sectors
300 ON companies_table.company_symbol = margins.company_symbol
301 WHERE sectors.Sector_name = 'Materials'
302 AND margins.Date
303 BETWEEN '2019-01-01' AND '2022-12-01'
304
305
306

```

The results grid shows data for company\_symbol AAL across different dates and sectors. The output section shows the following log entries:

#	Time	Action	Message	Duration / Fetch
423	21:19:03	SELECT companies_table.company_symbol, sectors.Sector_name, margins.Date, margins.Gross_Profit, FRO...	0 row(s) returned	0.000 sec / 0.000 sec
424	21:19:14	SELECT companies_table.company_symbol, sectors.Sector_name, margins.Date, margins.Gross_Profit, FRO...	0 row(s) returned	0.000 sec / 0.000 sec
425	21:19:47	SELECT companies_table.company_symbol, sectors.Sector_name, margins.Date, margins.Gross_Profit, FRO...	790 row(s) returned	0.016 sec / 0.000 sec
426	21:19:54	SELECT companies_table.company_symbol, sectors.Sector_name, margins.Date, margins.Gross_Profit, FRO...	790 row(s) returned	0.000 sec / 0.000 sec

## Use Case: What enterprise value and market cap of company symbol AAPL in past ten years?

SELECT

companies\_table.company\_symbol, sectors.sector\_name, valuation\_measure.Date, valuation\_measure.enterprise\_value, valuation\_measure.market\_cap

FROM companies\_table

INNER JOIN valuation\_measure

join sectors

ON companies\_table.company\_symbol = valuation\_measure.company\_symbol

and sectors.sector\_id = companies\_table.sector\_id

WHERE companies\_table.company\_symbol = 'AAPL'

```

326 -- join sectors
327 -- ON companies_table.company_symbol = dividends.company_symbol
328 -- WHERE sectors.Sector_name IN ('Health Care','Industrials','Materials')
329
330 -- 26 . What enterprise value and market cap of company symbol AAPL in past ten years?
331 • SELECT companies_table.company_symbol,sectors.Sector_name,valuation_measure.Date,valuation_measure.enterprise_value,valuation_measure.market_cap
332 FROM companies_table
333 INNER JOIN valuation_measure
334 join sectors
335 ON companies_table.company_symbol = valuation_measure.company_symbol
336 and sectors.Sector_id = companies_table.Sector_id
337 WHERE companies_table.company_symbol = 'AAPL'
338
339

```

**Result Grid**

company_symbol	sector_name	Date	enterprise_value	market_cap
AAPL	Information Technology	2022-09-30 00:00:00	2274841335000	2203381335000
AAPL	Information Technology	2022-06-30 00:00:00	2269030080160	220560080160
AAPL	Information Technology	2022-03-31 00:00:00	2888888448480	283000348480
AAPL	Information Technology	2021-12-31 00:00:00	2963724912070	2901644912070
AAPL	Information Technology	2021-09-30 00:00:00	2384485219000	2324390219000

**Action Output**

#	Time	Action	Message	Duration / Fetch
438	21:43:20	SELECT companies_table.company_symbol,sectors.Sector_name,valuation_measure.Date,valuation_measure.enterprise_value,valuation_measure.market_cap	40 row(s) returned	0.016 sec / 0.000 sec
439	21:47:05	SELECT companies_table.company_symbol,sectors.Sector_name,valuation_measure.Date,valuation_measure.enterprise_value,valuation_measure.market_cap	0 row(s) returned	0.000 sec / 0.000 sec
440	21:47:12	SELECT companies_table.company_symbol,sectors.Sector_name,valuation_measure.Date,valuation_measure.enterprise_value,valuation_measure.market_cap	40 row(s) returned	0.000 sec / 0.000 sec
441	21:47:30	SELECT companies_table.company_symbol,sectors.Sector_name,valuation_measure.Date,valuation_measure.enterprise_value,valuation_measure.market_cap	40 row(s) returned	0.000 sec / 0.000 sec

## Steps followed:

1. The initial phase involved obtaining the dataset from Yahoo Finance, which consisted of 53 companies overall and 10 different sectors.
2. The second step involved web scraping from Twitter to hunt for news relevant to the stock market.
3. The third step involved carrying out data cleansing procedures and displaying data visualization graphs. The processes for cleaning the data included managing null values and duplicate values.
4. The fourth stage was to examine the table for normalizations to see if they were in the first, second, or third normal form.