DATA MINING

ASSIGNMENT: MILESTONE 2

NORAISHA YUSUF (WQD180008)

NORBAIZURA MOHAMAD (WQD180043)

MARINA SHAH MUHAMMAD ZABRI TAN (WQD180011)

AMIRA AN-NUR BINTI RUSLI (WQD180016)

AMINAH SOFIA BINTI MAHAYUDIN (WQD180032)

Step 1: Data Crawling

Current

1. **Stock data** - source: The Star

2. **News -** source: Investor.com

3. **Blogs -** source: Investor.com

4. **Financial Performance -** source: Investor.com

Future

1. Twitter posts – reaction to news events.

Step 2: Staging

- **Use MySQL**
- **All data are inserted automatically when crawled - mysql.connector package in python**
  - **All data inserted as string (temporarily) to avoid error**
- **4 sources of data transmitted into 3 tables**

Step 3: Data pre-processing

Data cleaning activities included:

- Correcting and standardize the data format e.g. date, numeric etc. across all tables

- Removing irrelevant special character, symbols e.g. –

- Remove trailing and leading spaces

- Splitting columns with more than one values e.g. price and stock volume column

Step 4: Design our data warehouse schema

- Using MySQL workbench
- Snowflake schema
- 3 fact tables & Inter-connected dimension/look-up tables
- Will be updated as an when data requirements are revised or changed

dim_month
- month_id INT(11)
- month_name VARCHAR(20)
- month_of_year_id INT(11)
- qtr_id INT(11)
- Indexes

dim_period
- date DATE
- day INT(11)
- qtr_id INT(11)
- Indexes

fact_ne...
- num INT(1...)
- date DATE
- time VARC...
- title TEXT
- author VA...
- category V...
- Indexes

fact_financial
- num INT(11)
- year INT(11)
- company_stockcode VARCHAR(10)
- company_sname VARCHAR(20)
- anniversary_date DATE
- financial_year DATE
- quarter_date DATE
- quarter_no INT(11)
- financial_revenue DECIMAL(20,5)
- financial_pbt DECIMAL(20,5)
- financial_np DECIMAL(20,5)
- financial_NptoSh DECIMAL(20,5)
- financial_dividend DECIMAL(20,5)
- financial_networth DECIMAL(20,5)
- ratio_divpayout DECIMAL(20,5)
- ratio_npmargin DECIMAL(20,5)
- ratio_roe DECIMAL(20,5)
- per_share_rps DECIMAL(20,5)
- per_share_eps DECIMAL(20,5)
- per_share_dps DECIMAL(20,5)
- per_share_naps DECIMAL(20,5)
- performance_QoY DECIMAL(20,5)
- performance_YoY DECIMAL(20,5)
- Indexes

dim_qtr
- qtr_id INT(11)
- qtr_name VARCHAR(20)
- qtr_of_year_id INT(11)
- year INT(11)
- Indexes

dim_year
- year INT(11)
- Indexes

dim_company
- company_stockcode VARCHAR(10)
- company_sname VARCHAR(20)
- company_name VARCHAR(200)
- board VARCHAR(255)
- Indexes

fact_stock
- num INT(11)
- date DATE
- time VARCHAR(1...)
- company_stockc...
- company_sname VA...
- open_price DECI...
- high_price DECIM...
- low_price DECIM...
- last_price DECIM...
- change_price_p ...
- volume DECIMAL...
- buy_price DECIM...
- buy_volume DEC...
- sell_price D...
- sell_volum...
- Indexes

Database ER Diagram

dim_month
- 🔑 month_id INT(11)
- ◇ month_name VARCHAR(20)
- ◇ month_of_year_id INT(11)
- ◇ qtr_id INT(11)
- Indexes

dim_period
- 🔑 date DATE
- ◇ day INT(11)
- ◇ month_id INT(11)
- ◇ qtr_id INT(11)
- ◇ year INT(11)
- Indexes

fact_news
- 🔑 num INT(11)
- 🔑 date DATE
- ◇ time VARCHAR(10)
- ◇ title TEXT
- ◇ author VARCHAR(20)
- ◇ category VARCHAR(20)
- Indexes

fact_financial
- 🔑 num INT(11)
- year INT(11)
- ◇ company_stockcode VARCHAR(10)
- company_sname VARCHAR(20)
- ◇ anniversary_date DATE
- ◇ financial_year DATE
- ◇ quarter_date DATE
- ◇ quarter_no INT(11)
- ◇ financial_revenue DECIMAL(20,5)
- ◇ financial_pbt DECIMAL(20,5)
- ◇ financial_np DECIMAL(20,5)
- ◇ financial_NptoSh DECIMAL(20,5)
- ◇ financial_dividend DECIMAL(20,5)
- ◇ financial_networth DECIMAL(20,5)
- ◇ ratio_divpayout DECIMAL(20,5)
- ◇ ratio_npmargin DECIMAL(20,5)
- ◇ ratio_roe DECIMAL(20,5)
- ◇ per_share_rps DECIMAL(20,5)
- ◇ per_share_eps DECIMAL(20,5)
- ◇ per_share_dps DECIMAL(20,5)
- ◇ per_share_naps DECIMAL(20,5)
- ◇ performance_QoY DECIMAL(20,5)
- ◇ performance_YoY DECIMAL(20,5)
- Indexes

dim_qtr
- 🔑 qtr_id INT(11)
- ◇ qtr_name VARCHAR(20)
- ◇ qtr_of_year_id INT(11)
- ◇ year INT(11)
- Indexes

dim_year
- 🔑 year INT(11)
- Indexes

dim_company
- 🔑 company_stockcode VARCHAR(10)
- 🔑 company_sname VARCHAR(20)
- ◇ company_name VARCHAR(200)
- ◇ board VARCHAR(255)
- Indexes

fact_stock
- 🔑 num INT(11)
- date DATE
- 🔑 time VARCHAR(10)
- ◇ company_stockcode VARCHAR(10)
- company_sname VARCHAR(20)
- ◇ open_price DECIMAL(20,5)
- ◇ high_price DECIMAL(20,5)
- ◇ low_price DECIMAL(20,5)
- ◇ last_price DECIMAL(20,5)
- ◇ change_price_p DECIMAL(20,5)
- ◇ volume DECIMAL(20,5)
- ◇ buy_price DECIMAL(20,5)
- ◇ buy_volume DECIMAL(20,5)
- ◇ sell_price DECIMAL(20,5)
- ◇ sell_volume DECIMAL(20,5)
- Indexes

Step 5: Install Hadoop etc.

On Windows

- **Install Hadoop (as per tutorial last week)**

- **Install Hive**

- **Install Derby – to ensure connection with MySQL**

- **Install Sqoop – to transfer data from MySQL to HDFS**

Step 6: Transfer data to Hadoop

- From MySQL to HDFS

- Run sqoop import commands

- check in Hive

- Future tasks: May explore Apache Zeppelin

Administrator: Command Prompt - hive

```
OK
city
fact_financial
fact_news
fact_stock
4 rows selected (0.348 seconds)
hive>
hive>
hive>
hive> select count(*) from fact_financial;
13:24:45.304 [6a063269-8344-4d55-b13f-eb2c73785034
uri with key [hadoop security key.provider.path] to
WARNING: Hive-on-MR is deprecated in Hive 2 and may
execution engine (i.e. spark, tez) or using Hive 1.
Query ID = win_20190318132444_11d8368c-11fd-4a86-9a
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1552896509462_0003, Tracking URL
Kill Command = C:\hadoop-2.8.0\bin\hadoop.cmd job
Hadoop job information for Stage-1: number of mapper
2019-03-18 13:25:56,737 Stage-1 map = 0%,  reduce =
2019-03-18 13:26:21,138 Stage-1 map = 100%,  reduce
2019-03-18 13:26:43,901 Stage-1 map = 100%,  reduce
MapReduce Total cumulative CPU time: 18 seconds 324
Ended Job = job_1552896509462_0003
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 1
Total MapReduce CPU Time Spent: 18 seconds 324 msec
OK
503
1 row selected (122.63 seconds)
hive>
```

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

stg_stockdetails   stg_newsblog   stg_financial   fact_stock   fact_news

Navigator

SCHEMAS

Filter objects

- sakila
- sys
- world
- wqd7005
  - Tables
    - dim_company
    - fact_financial
    - fact_news
    - fact_stock
    - stg_financial
    - stg_newsblog
    - stg_stockdetails
  - Views
  - Stored Procedures
  - Functions

Administration   Schemas

Information

Table:
dim_company

Columns:
num
company_stockcode

Limit to 1000 rows

```
1 •  select count(*) from wqd7005.fact_financial;
```

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

count(*)
503

THE END