

## Tables Design:

VIDEOSTART\_RAW: raw data is directly loaded into the landing table from original file

COLUMN_NAME	DATA_TYPE	PK	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
DATETIME	VARCHAR(30)	N	Yes	null	1	Data from raw file
VIDEOTITLE	VARCHAR(200)	N	Yes	null	2	Data from raw file
EVENTS	VARCHAR(150)	N	Yes	null	3	Data from raw file

VIDEOSTART\_DLT: delta is a middle table after data transformation from raw table

COLUMN_NAME	DATA_TYPE	PK	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
DATETIME	TIMESTAMP	N	No	null	1	Data reformatted from VIDEOSTART_RAW. DATETIME
PLATFORM	VARCHAR(200)	N	No	null	2	Data derived from VIDEOSTART_RAW. VIDEOTITLE
SITE	VARCHAR(200)	N	No	null	3	Data derived from VIDEOSTART_RAW. VIDEOTITLE
VIDEO	VARCHAR(200)	N	No	null	4	Data derived from VIDEOSTART_RAW. VIDEOTITLE

DIMDATE\_DLT: delta table is used to record the date info to minute grain(yyyymmddhhii)

COLUMN_NAME	DATA_TYPE	PK	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
DATETIME	VARCHAR(12)	N	No	null	1	Data reformatted from VIDEOSTART_DLT. DATETIME

DIMPLATFORM\_DLT: delta table is used to record the platform info

COLUMN_NAME	DATA_TYPE	PK	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
PLATFORM	VARCHAR(200)	N	No	null	1	Data derived from VIDEOSTART_DLT. PLATFORM

DIMSITE\_DLT: delta table is used to record the site info

COLUMN_NAME	DATA_TYPE	PK	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
SITE	VARCHAR(200)	N	No	null	1	Data derived from VIDEOSTART_DLT. SITE

DIMVIDEO\_DLT: delta table is used to record the video info

COLUMN_NAME	DATA_TYPE	PK	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
VIDEO	VARCHAR(200)	N	No	null	1	Data derived from VIDEOSTART_DLT. VIDEO

MIDDATE: using date time as a surrogate key since they can be considered as unique

COLUMN_NAME	DATA_TYPE	PK	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
DATETIME_SKEY	VARCHAR(12)	N	No	null	1	Data reformatted from VIDEOSTART_DLT. DATETIME

MIDPLATFORM: platform dimension table with surrogate key

COLUMN_NAME	DATA_TYPE	PK	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
PLATFORM_SKEY	INT	Y	No	AUTO_INCREMENT	1	Surrogate key
PLATFORM	VARCHAR(200)	N	No	null	2	Data derived from DIMPLATFORM_DLT. PLATFORM

DIMSITE: site dimension table with unique surrogate key

COLUMN_NAME	DATA_TYPE	PK	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
SITE_SKEY	INT	Y	No	AUTO_INCREMENT	1	Surrogate key
SITE	VARCHAR(200)	N	No	null	2	Data derived from DIMSITE_DLT. SITE

DIMVIDEO: video dimension table with unique surrogate key.

COLUMN_NAME	DATA_TYPE	PK	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
VIDEO_SKEY	INT	Y	No	AUTO_INCREMENT	1	Surrogate key
VIDEO	VARCHAR(200)	N	No	null	2	Data derived from DIMVIDEO_DLT.VIDEO

FACTVIDEOSTART: using VIDEOSTART\_DLT table left join with each dimension table to get the surrogate key to generate this fact table.

COLUMN_NAME	DATA_TYPE	PK	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
DATETIME_SKEY	VARCHAR(12)	N	No	null	1	Data derived from DIMDATE. DATETIME_SKEY
PLATFORM_SKEY	INT	N	No	null	2	Data derived from DIMPLATFORM. PLATFORM_SKEY
SITE_SKEY	INT	N	No	null	3	Data derived from DIMSITE. SITE_SKEY
VIDEO_SKEY	INT	N	No	null	4	Data derived from DIMVIDEO. VIDEO_SKEY
SYSTIMESTAMP	TIMESTAMP(6)	N	No	null	5	TIMESTAMP when inserting the data

There are 6 stages designed in sequence to do the ETL process on mysql workbench with multiple tools:

## 1. Data Landing: load the raw videostarts file into videostart\_raw table

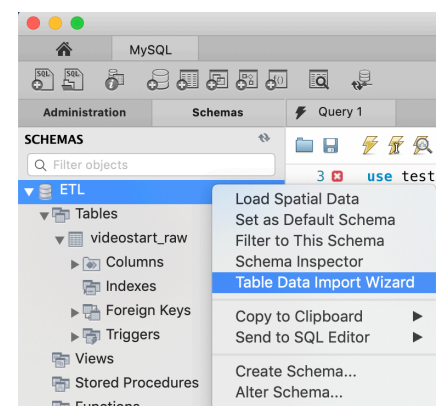
Load the video data into RStudio and have a quick review of the data in video\_data.csv file. There are 2686002 observations with 3 columns for each and the content of each observation is showing as below snapshot:

```
> dim(videodata)
[1] 2686002      3
> |
```

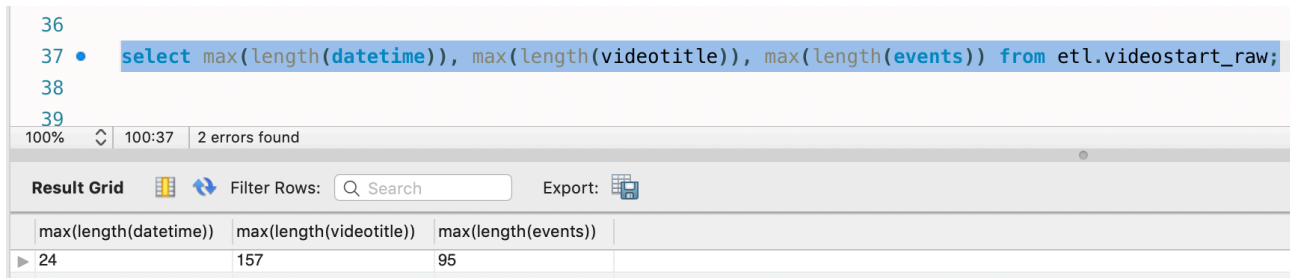
```
> head(videodata)
  DateTime                               VideoTitle                                events
1 2017-01-11T00:00:31.000Z App WebClipsIa-current-affair;2016IWilliam Tyrrell twist 157,120,160,104,162,161,163,164,165,166,171,229
2 2017-01-11T00:00:53.000Z newsI Shark attacks spearfisherman 127,157,120,160,104,162,161,171,206
3 2017-01-11T00:00:21.000Z newsI Shark attacks spearfisherman 127,157,120,160,104,162,161,163,164,165,166,171,229
4 2017-01-11T00:01:27.000Z newsI Chilean navy films UFO 157,120,160,104,162,161,170,171,237
5 2017-01-11T00:00:33.000Z newsI Video shows alleged axe attack at Sydney service station 157,120,160,104,162,161,163,164,165,166,171,229
6 2017-01-11T00:00:18.000Z App WebClipsItoday;2017IPint-sized Aussie surfer hits the global stage 127,157,120,160,104,162,161,163,164,165,166,171,229
```

The landing stage is only designed to swallow the raw data information as original and quick as possible. Therefore, a table videostart\_raw table is designed for data landing with 3 attributes : date\_time, video\_title and events.

Using MySQL Workbench tool to do table data import from video\_data.csv into new created table videostart\_raw:



Data auditing: The maximum lengths for date time, video title and events in VIDEOSTART\_RAW table are 24, 157 and 95 bytes, therefore, the corresponding field can be defined as 30, 200 and 150 bytes.



The screenshot shows a SQL IDE interface. At the top, a SQL query is entered in a text area: `select max(length(datetime)), max(length(video title)), max(length(events)) from etl.videostart_raw;`. Below the query, a status bar indicates '100%' zoom, '100:37' cursor position, and '2 errors found'. The main area displays the 'Result Grid' with a table of query results. The table has three columns: 'max(length(datetime))', 'max(length(video title))', and 'max(length(events))'. The first row shows the values 24, 157, and 95 respectively. The interface also includes a 'Filter Rows' search bar and an 'Export' button.

max(length(datetime))	max(length(video title))	max(length(events))
24	157	95

Run SQL script to create all designed tables.

**2. Using truncate function to clean all intermediate delta tables.**

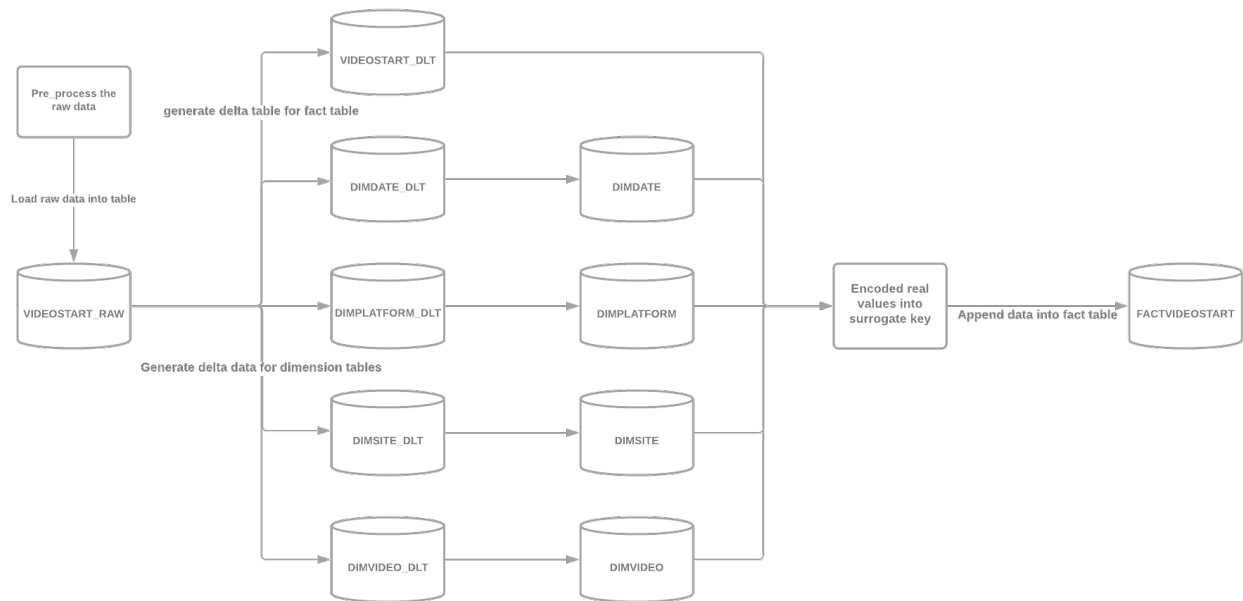
**3. Wash the date in VIDEOSTART\_RAW table and load into VIDEOSTART\_DLT table.**

**4. Populate dimension delta tables.**

**5. Insert data into dimension tables.**

**6. Append data into fact table.**

## On-going process workflow:



The following processes should be designed based on the above workflow:

1. Create raw data table on MySQLWorkbench.
2. Load the raw data file into raw data table.
3. Populate the dimension delta table first before dimension tables.
4. Populate dimension tables based on the delta tables.
5. Populate the fact table based on the dimension tables.

NB:

1. The current design is dimension type one: overwrite the changing content.
2. If the source dimension data contains not only the primary key but some other attributes, and the changes of the attributes need to be tracked, then dimension type two should be applied. Dimension type two is usually applied through adding a new row.

Here is one example for dimension two:

The product information at 2020-12-02:

Product_ID	Product_Name	Price	Location
P001	Huawei P4	550	EastVillage Shop
P003	Huawei P5	950	EastVillage Shop

Data in dimension table:

Product_SK EY	Product_ID	Product_Na me	Price	Location	Current_Flag	Start_Date	End_Date
11	P001	Huawei P4	750	EastVillage Shop	Y	2020-01-01	9999-12-31
12	P003	Huawei S5	1000	EastVillage Shop	Y	2020-05-01	9999-12-31

The dimension table after changes:

Product_SK EY	Product_ID	Product_Na me	Price	Location	Current_Flag	Start_Date	End_Date
11	P001	Huawei P4	750	EastVillage Shop	N	2020-01-01	2020-12-01
12	P003	Huawei S5	1000	EastVillage Shop	Y	2020-05-01	9999-12-31
13	P004	Huawei P5	950	EastVillage Shop	Y	2020-12-01	9999-12-31
14	P001	Huawei P4	550	EastVillage Shop	Y	2020-12-02	9999-12-31

The price for product Huawei P4 was reduced from \$750 to \$500 from 2020-12-02 due to the new product Huawei P5 was on the market. Therefore, the original product price was ended at 2020-12-01 and new price \$550 had been using from 2020-12-02.

In the above table, the highlighted yellow one is showing the price changing records and the red one is showing the new insertion record.

Current_Flag	Start_Date	End_Date
N	2020-01-01	2020-12-01

When the price got changed, the current\_flag and end\_date of the original record is changed, which means this record was in effective from 2020-01-02 to 2020-12-01 and currently is no long used.

Current_Flag	Start_Date	End_Date
Y	2020-12-02	9999-12-31

A new added record with same Product\_ID P001 but new Product\_SKEY 14 is in effective now. Therefore, when new records is populated into fact table, the filter Current\_Flag = 'Y' must be set to fetch the correct surrogate key.

If the historical data in dimension table is needed, the event date of the record to be retrieved should be between Start\_Date and End\_Date. For example, a customer purchased a Huawei P4 at 2020-05-01 and the purchase price should be \$750 rather than the currently effective price \$550.