## Table design

## video_data

| COLUMN_NAME | DATA_TYPE | PK | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|---|---|---|---|---|---|---|
| DATETIME | VARCHAR2(3 0 BYTE) | N | Yes | null | 1 | Data from raw file |
| VIDEOTITLE | VARCHAR2(2 00 BYTE) | N | Yes | null | 2 | Data from raw file |
| EVENTS | VARCHAR2(1 50 BYTE) | N | Yes | null | 3 | Data from raw file |

## staging

| COLUMN_NAME | DATA_TYPE | PK | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|---|---|---|---|---|---|---|
| DATETIME | DATETIME | N | NO | null | 1 | Data reformatted from video_data. DATETIME |
| PLATFORM | VARCHAR2(20 BYTE) | N | NO | null | 2 | Data derived from video_data. VIDEOTITLE |
| SITE | VARCHAR2(10 BYTE) | N | NO | null | 3 | Data derived from video_data. VIDEOTITLE |
| TITLE | VARCHAR2(2 00 BYTE) | N | NO | null | 4 | Data derived from video_data. VIDEOTITLE |

## FACTVIDEOSTART

| COLUMN_NAME | DATA_TYPE | PK | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|---|---|---|---|---|---|---|
| DATETIME_SKEY | NUMBER( 3 8,0) | N | NO | null | 1 | Data derived from DIMDATE. DATETIME_SKEY |
| PLATFORM_SKEY | NUMBER(38 ,0) | N | NO | null | 2 | Data derived from DIMPLATFORM. PLATFORM_SKEY |
| SITE_SKEY | NUMBER(38 ,0) | N | NO | null | 3 | Data derived from DIMSITE. SITE_SKEY |
| TITLE_SKEY | NUMBER(38 ,0) | N | NO | null | 4 | Data derived from DIMTITLE. TITLE_SKEY |
| DB_INSERT_TIMESTAMP | TIMESTAMP (6) | N | NO | null | 5 | TIMESTAMP when inserting the data |

## dim_time

| COLUMN_NAME | DATA_TYPE | PK | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|---|---|---|---|---|---|---|
| DATETIME | DATETIME | N | NO | null | 1 | Data reformatted from fact_dlt. DataTime |

## dim_platform

| COLUMN_NAME | DATA_TYPE | PK | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|---|---|---|---|---|---|---|
| PLATFORM | VARCHAR2(20 BYTE) | N | NO | null | 1 | Data derived from fact_dlt. platform |

## dim_site

| COLUMN_NAME | DATA_TYPE | PK | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|---|---|---|---|---|---|---|
| SITE | VARCHAR2(10BYTE) | N | NO | null | 1 | Data derived from fact_dlt. site |

## dim_title

| COLUMN_NAME | DATA_TYPE | PK | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|---|---|---|---|---|---|---|
| TITLE | VARCHAR2(200BYTE) | N | NO | null | 1 | Data derived from fact_dlt. tilte |

## DIMDATE

| COLUMN_NAME | DATA_TYPE | PK | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|---|---|---|---|---|---|---|
| DATETIME_SKEY | NUMBER(38,0) | Y | NO | | 1 | Data derived from dim_time. DataTime |
| DATETIME | DATETIME | | | | | |
| YEAR | smallint | N | NO | null | 2 | Data derived from dim_time. DataTime |
| MONTH | smallint | N | NO | null | 3 | Data derived from dim_time. DataTime |
| DAY | smallint | N | NO | null | 4 | Data derived from dim_time. DataTime |
| HOUR | smallint | N | NO | null | 5 | Data derived from dim_time. DataTime |
| MINU | smallint | N | NO | null | 6 | Data derived from dim_time. DataTime |

## DIMPLATFORM

| COLUMN_NAME | DATA_TYPE | PK | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|---|---|---|---|---|---|---|
| PLATFORM_SKEY | NUMBER(38,0) | Y | NO | | 1 | |
| PLATFORM | VARCHAR2(20 BYTE) | N | NO | null | 2 | Data derived from staging. PLATFORM |

## DIMSITE

| COLUMN_NAME | DATA_TYPE | PK | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|---|---|---|---|---|---|---|
| SITE_SKEY | NUMBER(38,0) | Y | NO | | 1 | |
| SITE | VARCHAR2(10 BYTE) | N | NO | null | 2 | Data derived from staging. SITE |

## DIMTITLE

| COLUMN_NAME | DATA_TYPE | PK | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|---|---|---|---|---|---|---|
| TITLE_SKEY | NUMBER(38,0) | Y | NO | | 1 | |
| TITLE | VARCHAR2(200 BYTE) | N | NO | null | 2 | Data derived from dim_title. TITLE |

## Process design

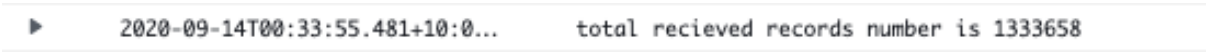
once upload raw data into S3, it will invoke lambda(python script) to extract data and transform data, generate dimension delta and fact delta csv file, store back to S3 and then invokes lambda(python script) to load processed data into OLAP database redshift.

**Extraction and transformation(detailed code please find relative code)**

1. Change encoding into UTF-8, otherwise the special characters will show.
2. Upload raw data video_data.csv into S3
3. Lambda will be invoked (Python script ) to do data auditing

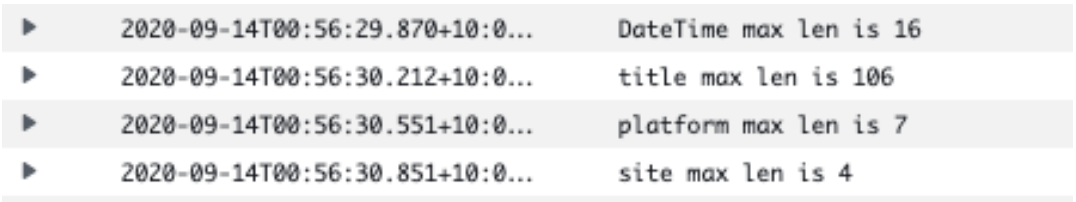
```
#data auditing: count total #records
print(f"total recieved records number is {len(raw)}")
```

CloudWatch Log Events:




```
#data auditing: find max length of all field in fact
for col in fact.columns:
  print(f"{col}  max len is {max([len(i) for i in fact[col] if i])}")
```

CloudWatch Log Events:




Use the result to adjust the length of columns in table in Redshift.
Then wash raw data and load into fact delta and dimension delta tables.

4. The lambda will populate all dimension delta tables and fact delta table as following separately: dim_time.csv, dim_platform.csv, dim_site.csv, dim_title.csv, fact_dlt.csv.
5. Redshift SQL query to COPY data in processed folder in S3 into delta tables and staging table.
6. Insert and generate SKEY for DIM tables DIMDATE, DIMPLATFORM, DIMSITE and DIMTITLE
7. Use staging table, DIMDATE, DIMPLATFORM, DIMSITE and DIMTITLE to generate output data and append the data into fact table FACTVIDEOSTART.

## Load data from S3 to Redshift(detailed code please find relative code)

1. SQL create dimension tables in Redshift if those dimension tables not exist.
2. SQL create dimension delta tables and staging table if not exist in Redshift
3. Load dimension delta table data to dimension delta table, fact table data to staging table:

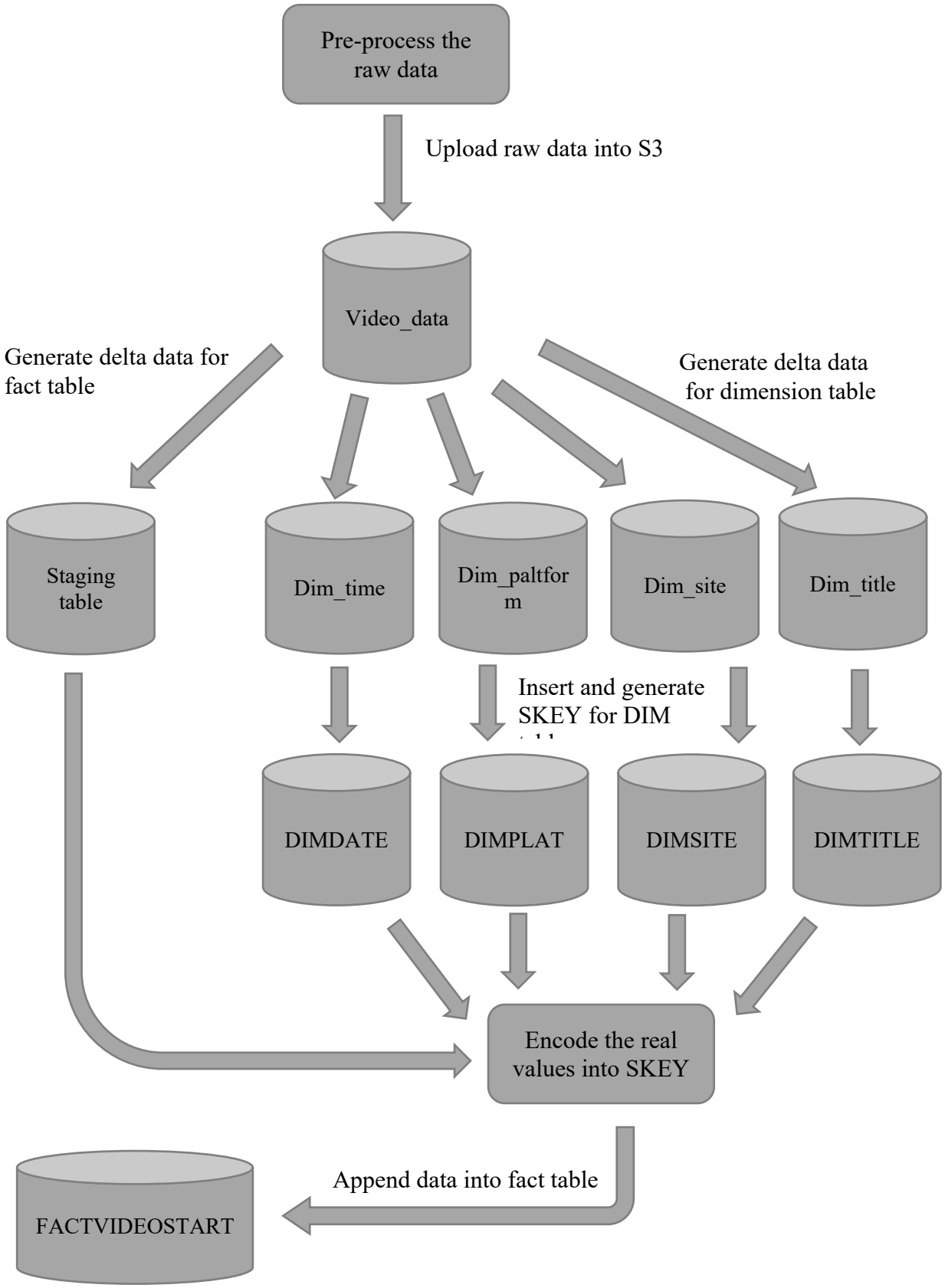
   COPY time_dlt ("timeid", "DateTime", "year", "month", "day", "hour", "minute")
   FROM 's3://project4de/processed/dim_time.csv'
   credentials 'aws_iam_role=my_arn '
   CSV
   EXPLICIT_IDS
   IGNOREHEADER 1;
4. Insert and generate SKEY for all dimension tables:

   insert DLT data for DIMDATE right join timeDLT whenDIMDATE is null(Type 1)
5. Load data from staging table to fact table:

```
insert into FACTVIDEOSTART (DATETIME_SKEY, TITLE_SKEY, SITE_SKEY, PLATFORM_SKEY)\
select a.DATETIME_SKEY, b.TITLE_SKEY, c.SITE_SKEY, d.PLATFORM_SKEY\
from staging e\
left join DIMDATE a \
on e.DATETIME = a.DATETIME\
left join DIMTITLE b\
on e.TITLE = b.TITLE\
left join DIMSITE c\
on e.SITE = e.SITE\
left join DIMPLATFORM d\
n e.PLATFORM = d.PLATFORM;
```

6. Truncate staging, time_dlt, site_dlt, platform_dlt, title_dlt.

## On-going process



## NOTE:

1. **SKEY stands for surrogate key.**
2. **The current design is Dimension Type One.**
3. **If the source dimension data contains other attributes other than just PK, and we want to track changes of those attributes, eg. sometimes we want to track the product's price changing, then we should use Dimension Type Two.**
   One sample of Dimension Type Two
   Delta Data from 06/04/2017:

| Product_ID | Product_Name | Price | Location |
|---|---|---|---|
| P001 | Iphone6 | 750 | Townhall Shop |
| P003 | Iphone7 | 1000 | Townhall Shop |

Data in dimension table:

| Product_SKEY | Product_ID | Product_Name | Price | Location | Current_Flag | Start_Date | End_Date |
|---|---|---|---|---|---|---|---|
| 111 | P001 | Iphone6 | 800 | Townhall Shop | Y | 31/12/2016 | 31/12/9999 |
| 112 | P002 | Iphone6Plus | 900 | Townhall Shop | Y | 20/01/2017 | 31/12/9999 |

Add new product (P003) and update product (P001) in dimension table:

| Product_SKEY | Product_ID | Product_Name | Price | Location | Current_Flag | Start_Date | End_Date |
|---|---|---|---|---|---|---|---|
| 111 | P001 | Iphone6 | 800 | Townhall Shop | N | 31/12/2016 | 05/04/2017 |
| 112 | P002 | Iphone6Plus | 900 | Townhall Shop | Y | 20/01/2017 | 31/12/9999 |
| 113 | P003 | Iphone7 | 1000 | Townhall Shop | Y | 06/04/2017 | 31/12/9999 |
| 114 | P001 | Iphone6 | 800 | Townhall Shop | Y | 06/04/2017 | 31/12/9999 |

Yellow part is updated, red part is insertion.
When there is a new record coming, we generate new record with new SKEY, Current_Flag='Y',Start_Date = Current_Date, End_Date=31/12/9999
When there is a updated record coming, we generate new record with new SKEY, Current_Flag='Y',Start_Date = Current_Date, End_Date=31/12/9999, and at the same time, we update the old record with Current_Flag='N', End_Date= Current_Date-1.

Therefore, when we populate new records into fact table, we need a filter eg. Current_Flag='Y' in order to get the correct SKEY;
If we want to track the history data for certain days, we use a time filter between Start_Date and End_Date.

For example, if in fact table we see a transaction like customer purchased product(P001) on 01/04/2017, by looking at product dimension table, we could find the price that customer paid at that moment was 800 not 750, although 750 is the current price of P001