Screening Test for Data Engineer

Part 1. Knowledge of SQL

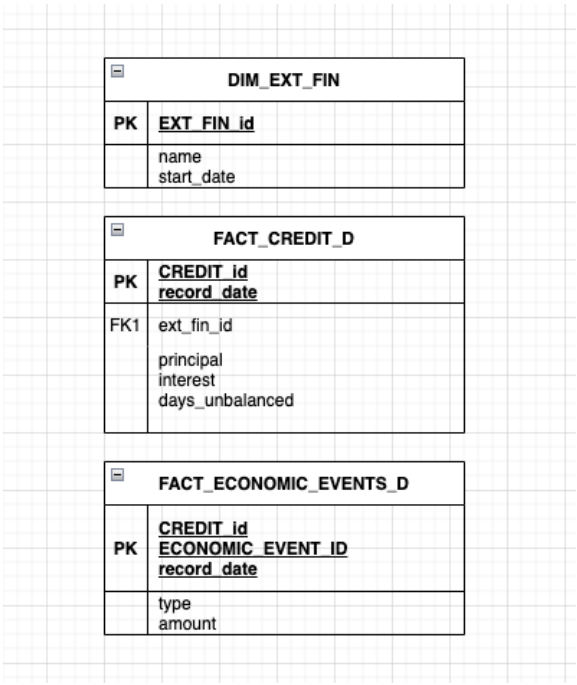
Reason your answers, assumptions, …

Question1

Two fact tables are provided: FACT_CREDIT_D, which contains **daily snapshots** of the state of all the granted loans since the start of the company (every day all the credits info is loaded, approx size 900M records) and the economic events related to each of the loans, FACT_ECONOMIC_EVENTS_D.

A dimension is also provided, DIM_EXT_FIN, that contains the information of the external financiers that buy and sell loans.

You are asked to extract all the events of type “Chargeback” that the external Financier “Fund2” has produced during 2022



Data Sample

DIM_EXT_FIN
Ext_fin_id;name;start_date
1;”Inv1”;2020-04-18

2;”Fund2”;2021-10-11

FACT_CREDIT_D

Credit_id;record_date;ext_fin_id;principal;interest;days_unbalanced
15;2022-01-05;1;100;3;0
15;2022-01-06;1;100;3;0
16;2022-01-06;1;80;2;1
15;2022-01-07;1;100;3;0
16;2022-01-07;1;80;2;1
17;2022-01-07;2;300;3
18;2022-01-07;2;50;3;0
15;2022-01-08;1;100;3;0
16;2022-01-08;1;80;2;1
17;2022-01-08;2;300;3
18;2022-01-08;2;50;3;0
19;2022-01-08;2;65;3;0

FACT_ECONOMIC_EVENTS_D

Credit_id;economic_event_id;record_date;type;amount
15;1;2022-01-05;”installment”;5
15;2;2022-01-05;”chargeback”;10
17;3;2022-01-07;”chargeback”;10
18;4;2022-01-08;”payment”;50
18;5;2022-01-10;”chargeback”;50
19;6;2022-01-10;”payment”;30

Question 2

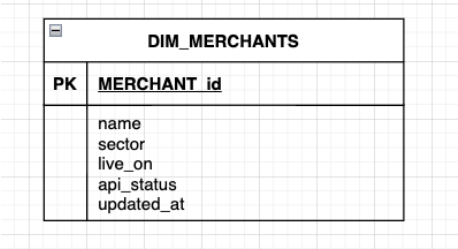
Merchants Growth

Estimate the growth of the merchant contracts signed each year using the live_on date and the api status field.

The rate of growth is calculated by taking ((number of merchants with api_status allowed and live_on in the current year - number of merchants with api_status allowed and live_on in the previous year) / number of merchants with api_status allowed and live_on in the previous year) * 100.

Output the year, number of merchants in the current year, number of merchants in the previous year, and the rate of growth. Round the rate of growth to the nearest percent and order the result in the ascending order based on the year.

Assume that the dataset consists only of unique merchants



Data Sample

DIM_MERCHANTS
Merchant_id;Name;Sector;Live_on;Api_status;Updated_at
1;’trajesamedida’;’retail’;2020-07-25;’allowed’;2020-07-25
25;’decoracionMark’;’home’;2018-06-08;’allowed’;2021-02-17
4;’superdiario’;’food’;2018-11-05;’allowed’;2021-02-17
10;’ropainterior’;’retail’;;’allowed’;2021-02-17
11;’proteinasaltorendimiento’;’health’;2018-03-15;’denied’;2021-02-17
12;’gimnasioMoreno’;’health’;2019-08-17;’allowed’;2021-02-17
6;’cuerpoyvida’;’health’;2021-01-14;’supervised’;2021-02-20
2;’dietasana’;’health’;2021-01-15;’allowed’;2021-02-20
3;’ropadesalon’;’retail’;2019-01-02;’supervised’;2021-02-20
13;’menajecocina’;’home’;;’allowed’;2021-02-06

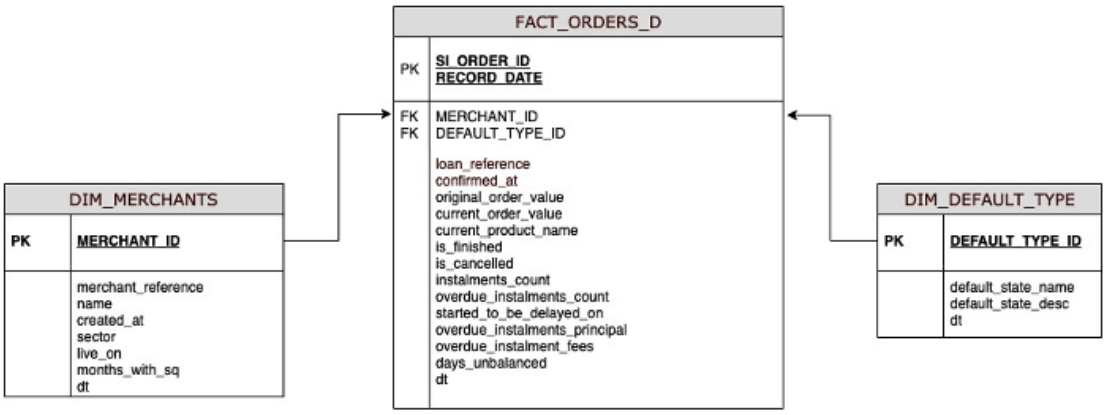
Part 2. Knowledge of Data Modeling

With the objective of controlling the optimum Risk levels of the company, a dashboard is requested, in our case Tableau, to be able to analyze the state of each defaulted loan by cohorts. This means visualizing how loans evolve by sector, merchant, default type and current_product_name. (Example: Matrix of month to be analyzed x loan confirmation month)

Given the following table images and tips on the size of the fact table, how would you model and optimize the table/s needed in the DWH to be used by Tableau?

Tips:

- FACT table ‘**FACT_ORDERS_D**’ with daily photos of the state of each credit from 2017, it contains more than 1,4B of rows
- ‘**DIM_Merchants'**
- **‘DIM_Type of Default’** with the classification of the arrears credits



Part 3. Pyspark

Given a batch process that runs daily, the following logic needs to be implemented in Pyspark code:

We have an aggregate table in the DWH, where monthly credit status snapshots and yesterday's snapshot are stored. Every day the current snapshot should be inserted and the previous snapshot deleted except for when the previous photo was on the last day of the month.

Example:
AGG_table

T0:
Id; record_date; **
Xxxxxx; 2021-03-31; **
Yyyyyy; 2021-04-30; **
T1:
Id; record_date; **
Xxxxxx; 2021-03-31; **
Yyyyyy; 2021-04-30; **
Zzzzzz; 2021-05-01; **
T2:
Id; record_date; **
Xxxxxx; 2021-03-31; **
Yyyyyy; 2021-04-30; **

Zzzzzz; 2021-05-02; **

…

T31:

Id; record_date; **
Xxxxxx; 2021-03-31; **
Yyyyyy; 2021-04-30; **
Zzzzzz; 2021-05-31; **

T32:

Id; record_date; **
Xxxxxx; 2021-03-31; **
Yyyyyy; 2021-04-30; **
Zzzzzz; 2021-05-31; **
Aaaaaa; 2021-06-01; **

You are asked to write the extraction and loading sentences in pyspark so that the data is stored as previously explained.
Please elaborate on reasons for your chosen partitioning, file format, etc.