Projet Data Warehouse Avancé

Credit Card Approval Prediction

Lien GitHub :

https://github.com/chokrihamza/credit-card-data-warehouse.git

Réalisés par :

Hamza Chokri (mailto:chokrihamza.ia@gmail.com).

Malek Ghanem (mailto:malek.ghanem1920@gmail.com).

1. Introduction :

Le credit scoring est un outil largement utilisé par les prestataires de crédit pour évaluer la solvabilité et la fiabilité des emprunteurs potentiels. Il est basé sur des données financières et de crédit collectées auprès de différentes sources, comme les agences de crédit et les banques. Ces données sont utilisées pour calculer un score de crédit, qui est ensuite utilisé par les prestataires de crédit pour déterminer la probabilité qu'un emprunteur rembourse ses dettes de manière fiable.

Le crédit scoring est souvent utilisé par les banques et les organismes de crédit pour évaluer la solvabilité des personnes qui demandent un prêt, mais il est également utilisé dans d'autres contextes, tels que l'évaluation de la fiabilité des locataires potentiels par les propriétaires d'immeubles ou l'évaluation de la solvabilité des entreprises par les investisseurs potentiels.

Le crédit scoring peut être utilisé de manière positive, en aidant les prestataires de crédit à identifier les emprunteurs les plus fiables et en leur permettant de leur offrir des taux d'intérêt plus avantageux. Cependant, il peut également avoir des effets négatifs, en limitant l'accès au crédit pour certaines personnes ou entreprises qui ont un score de crédit faible mais qui pourraient être en mesure de rembourser leurs dettes de manière fiable.

2. Les facteurs pris en compte dans le calcul d'un crédit score aux États-Unis :

Plusieurs facteurs qui peuvent être pris en compte dans le calcul d'un crédit score aux États-Unis.

Voici quelques informations supplémentaires sur chacun de ces facteurs :

- *L'utilisation de la réserve de crédit attachée à une carte de crédit* est généralement exprimée en pourcentage de la limite de crédit accordée. Si vous utilisez une grande partie de votre limite de crédit, cela peut être interprété comme un signe de surendettement et votre crédit score peut en souffrir. Cela peut être particulièrement vrai si vous utilisez une grande partie de la limite de crédit de plusieurs cartes de crédit. Il est recommandé de ne pas utiliser plus de 30 % de la limite de crédit accordée par chaque émetteur de carte de crédit pour minimiser l'impact sur votre credit score.

- *L'historique de crédit* est un facteur important dans le calcul de votre crédit score. Il comprend les prêts et les cartes de crédit que vous avez utilisés dans le passé, ainsi que la façon dont vous avez remboursé ces dettes. Si vous avez des antécédents de paiements réguliers et de remboursement complet de vos dettes, cela peut améliorer votre crédit score. En revanche, si vous avez eu des difficultés à rembourser vos dettes ou si vous avez eu des retards de paiement, cela peut nuire à votre crédit score.
- *La régularité des paiements effectués* est également prise en compte dans le calcul de votre crédit score. Si vous payez vos factures et vos loyers à temps, cela peut améliorer votre crédit score. En revanche, si vous avez des retards de paiement réguliers, cela peut nuire à votre crédit score.
- *Le type de crédit demandé* peut également avoir un impact sur votre crédit score. Si vous demandez un prêt à haut risque, comme un prêt à taux variable ou un prêt à court terme, cela peut nuire à votre crédit score. En revanche, si vous demandez un prêt à faible risque, comme un prêt à taux fixe ou un prêt à long terme, cela peut améliorer votre crédit score.
- *Le nombre de recherches effectuées pour obtenir un prêt*, également appelé *"credit inquiry"*, peut également avoir un impact sur votre crédit score. Si vous effectuez de nombreuses demandes de crédit en peu de temps, cela peut être interprété comme un signe de besoin de crédit urgent.

3. Description des données utilisées pour l'analyse :

Dans notre étude on va utiliser :

3.1. Les données des emprunteurs :

Feature name	Explanation	Remarks
ID	Client number	
CODE_GENDER	Gender	
FLAG_OWN_CAR	Is there a car	
FLAG_OWN_REALTY	Is there a property	
CNT_CHILDREN	Number of children	
AMT_INCOME_TOTAL	Annual income	
NAME_INCOME_TYPE	Income category	
NAME_EDUCATION_TYPE	Education level	
NAME_FAMILY_STATUS	Marital status	
NAME_HOUSING_TYPE	Way of living	
DAYS_BIRTH	Birthday	Count backwards from current day (0), -1 means yesterday
DAYS_EMPLOYED	Start date of employment	Count backwards from current day (0). If positive, it means the person currently unemployed.
FLAG_MOBIL	Is there a mobile phone	
FLAG_WORK_PHONE	Is there a work phone	
FLAG_PHONE	Is there a phone	
FLAG_EMAIL	Is there an email	

OCCUPATION_TYPE	Occupation	
CNT_FAM_MEMBERS	Family size	

3.2. Credit record:

Feature name	Explanation	Remarks
ID	Client number	
MONTHS_BALANCE	Record month	The month of the extracted data is the starting point, backwards, 0 is the current month, -1 is the previous month, and so on
STATUS	Status	0: 1-29 days past due 1: 30-59 days past due 2: 60-89 days overdue 3: 90-119 days overdue 4: 120-149 days overdue 5: Overdue or bad debts, write-offs for more than 150 days C: paid off that month X: No loan for the month

Lien des données utilisées : https://www.kaggle.com/datasets/rikdifos/credit-card-approval-prediction

4. La modélisation des données:

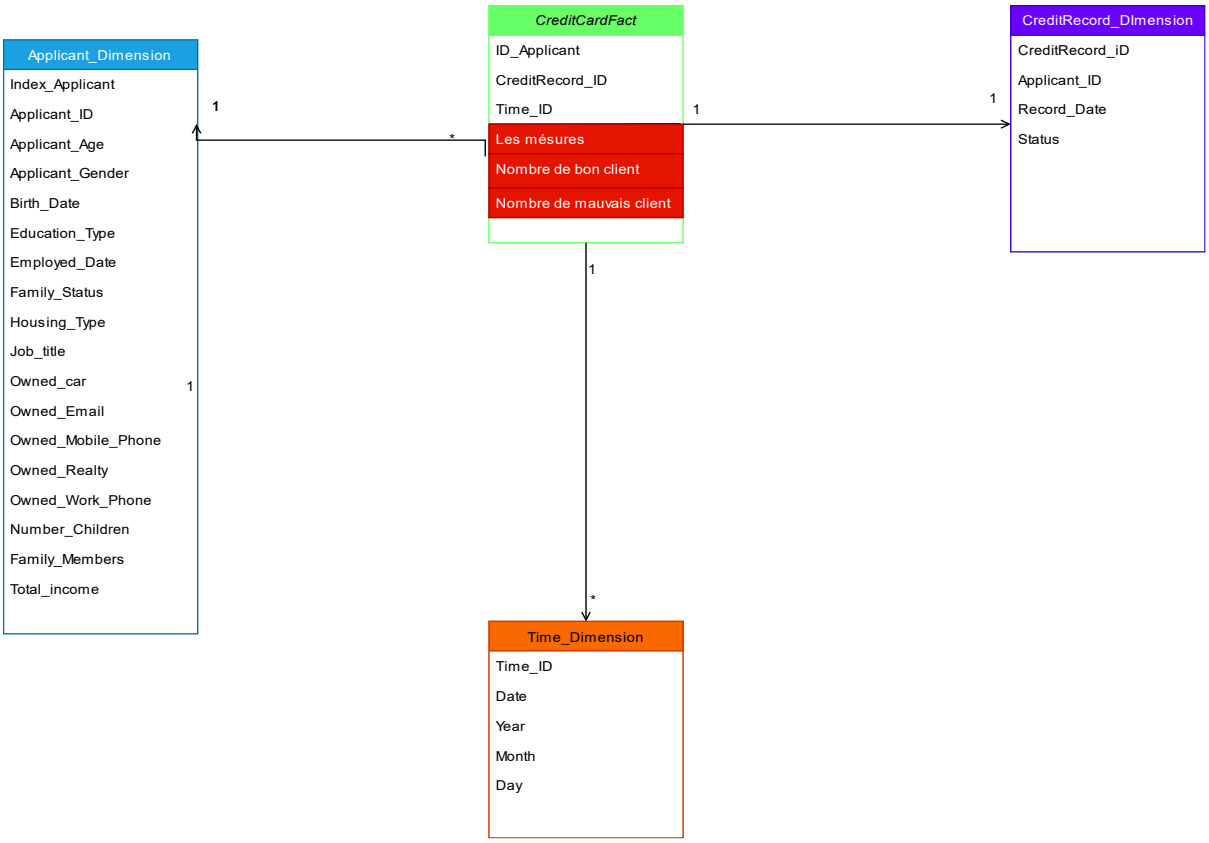


Fig. 1 : Schéma en étoile

Les mesures, qu'on va utiliser pour notre analyse sont :

- Le nombre des bons et mouvais clients.

5. L'alimentation des données :

Prétraitement des données (Dimension emprunteur) :

Les étapes qu'on a suivi pour alimenter notre data Warehouse sont comme suit :

- Extraire la base de données des emprunteurs de Kaggle.
- Trier les données par ID.
- Filtrer les ID dupliqués.
- Remplacer les valeurs.
- Ajouter la date actuelle.
- Calculer la date.

- Filtrer les enregistrements nuls.
- Ajouter un index pour l'emprunteur.
- Sélectionner des valeurs.
- Charger les données dans la dimension de l'emprunteurs (Applicant Dimension)

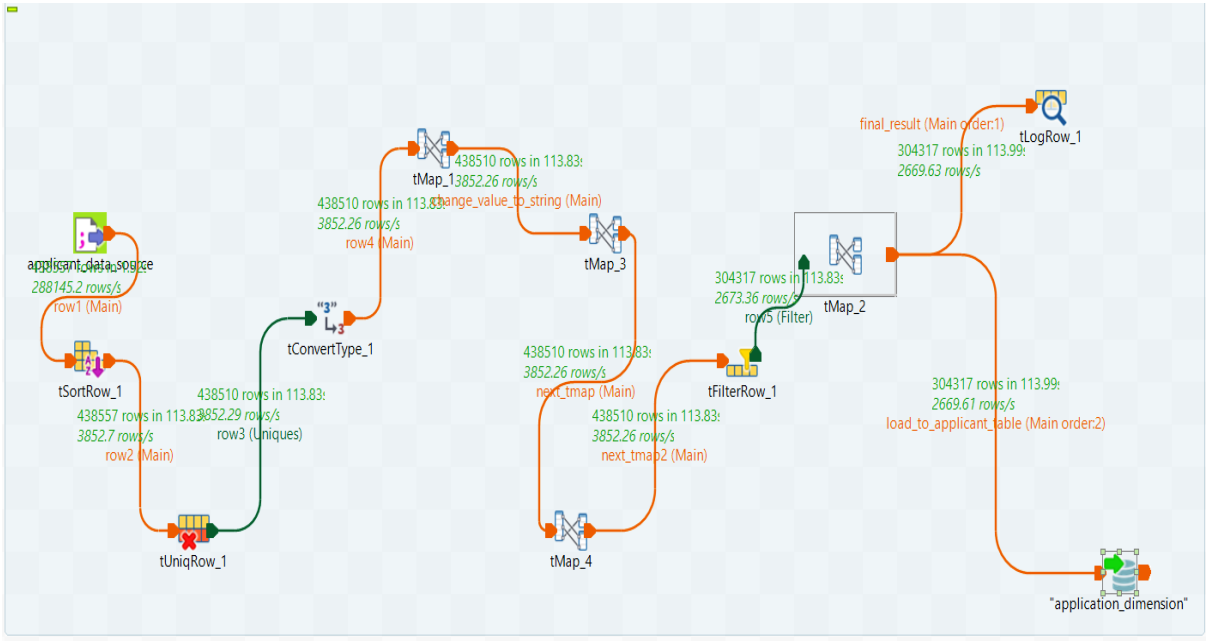
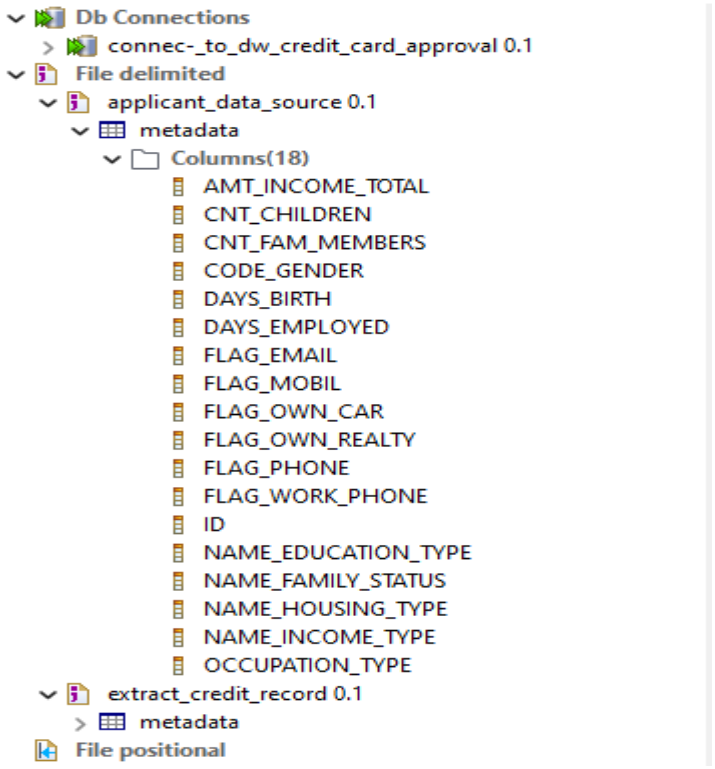
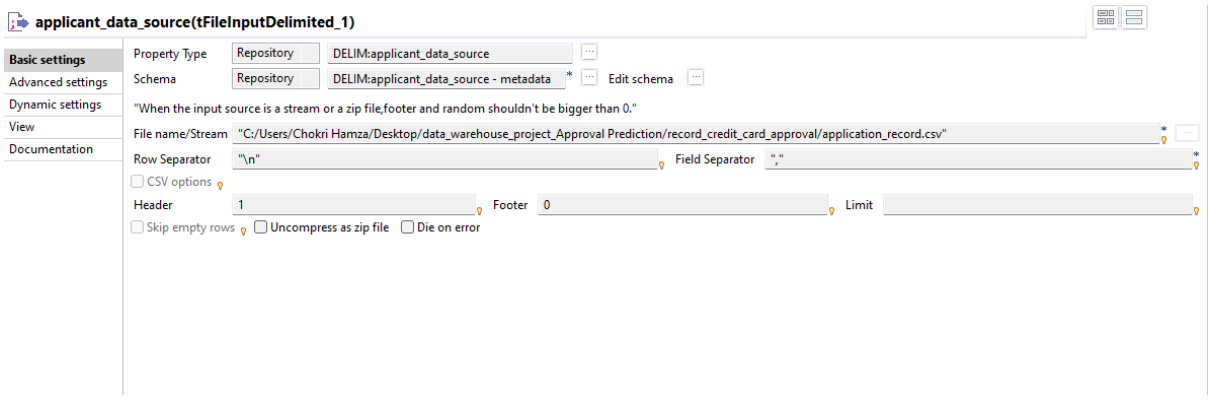
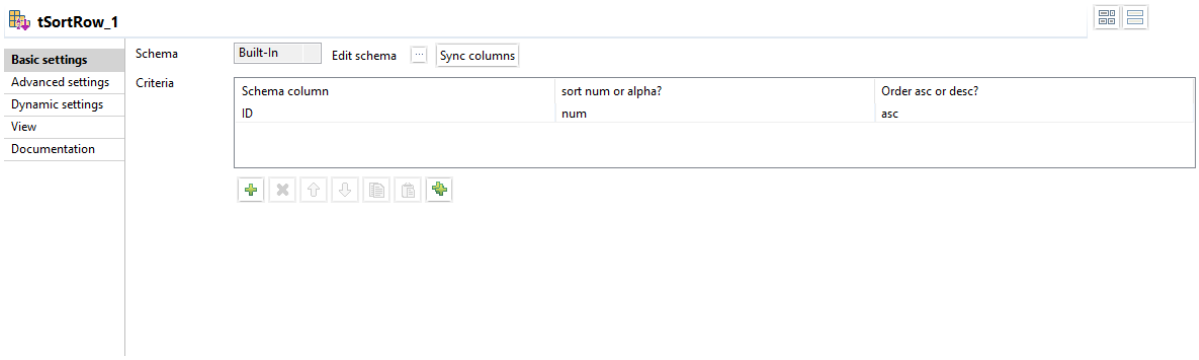


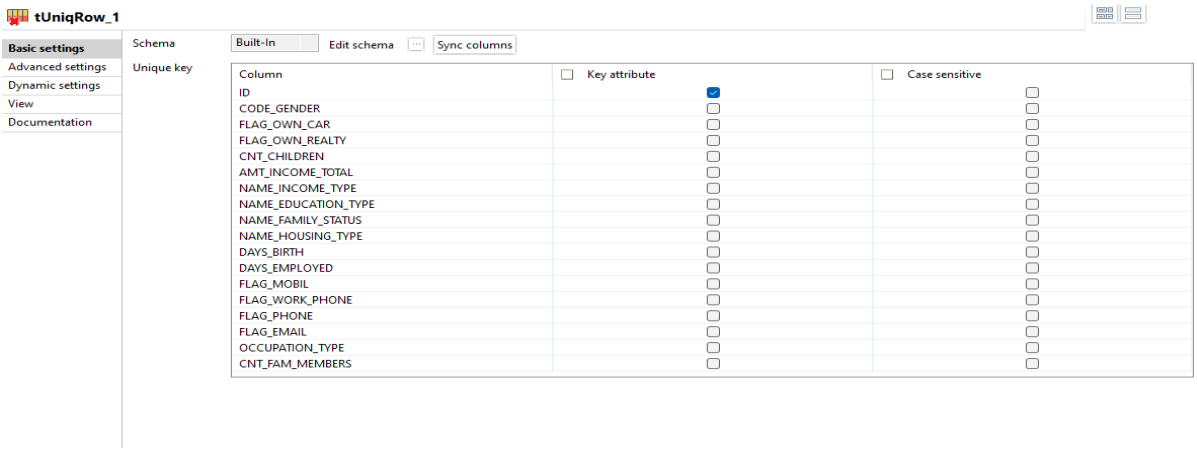
Fig.2 : ETL Applicant dimension

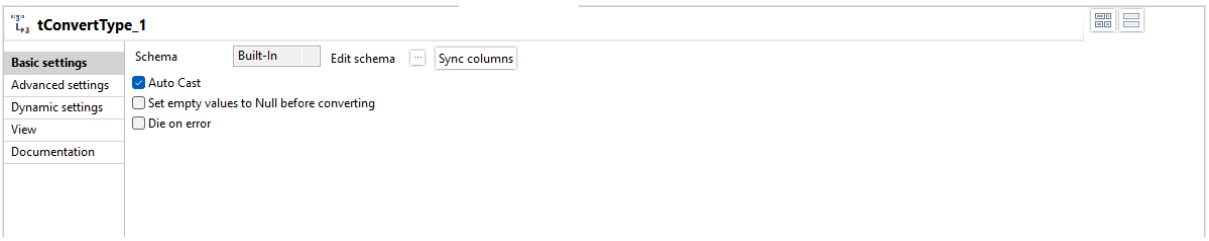
6. Le détail des étapes de l'ETL :

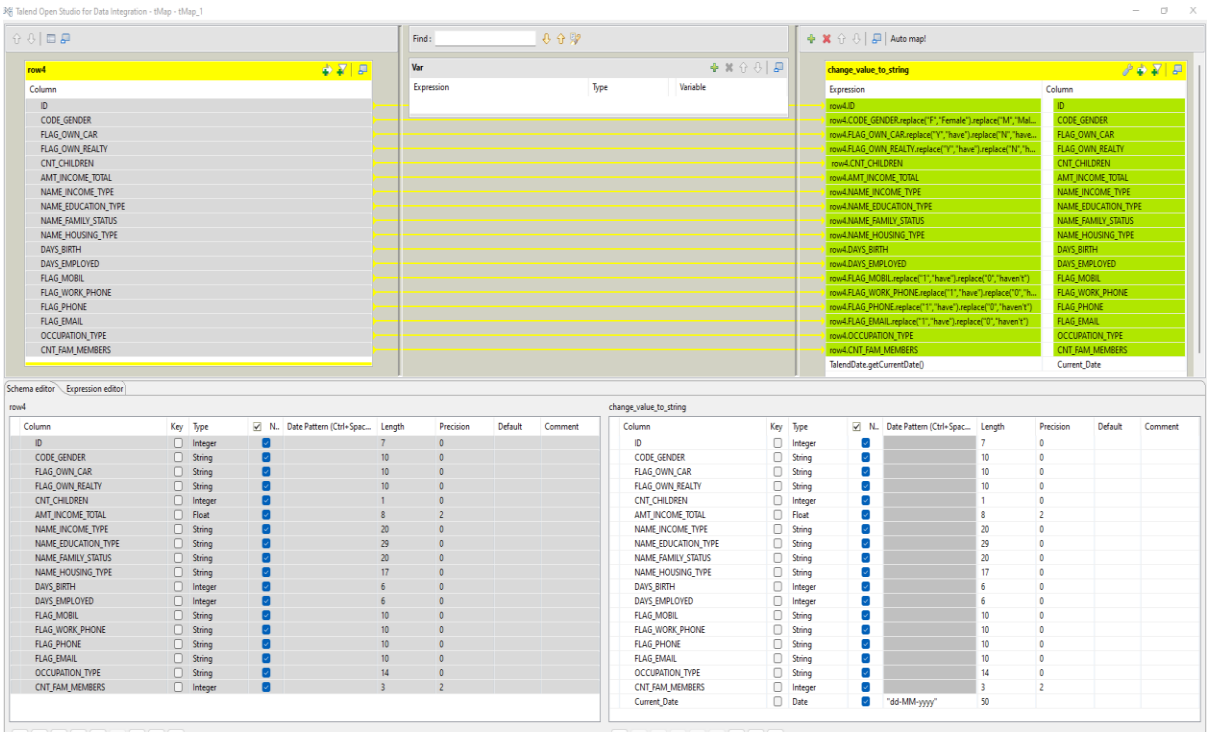

Db Connections
connec-_to_dw_credit_card_approval 0.1
File delimited
applicant_data_source 0.1
metadata
Columns(18)
AMT_INCOME_TOTAL
CNT_CHILDREN
CNT_FAM_MEMBERS
CODE_GENDER
DAYS_BIRTH
DAYS_EMPLOYED
FLAG_EMAIL
FLAG_MOBIL
FLAG_OWN_CAR
FLAG_OWN_REALTY
FLAG_PHONE
FLAG_WORK_PHONE
ID
NAME_EDUCATION_TYPE
NAME_FAMILY_STATUS
NAME_HOUSING_TYPE
NAME_INCOME_TYPE
OCCUPATION_TYPE
extract_credit_record 0.1
metadata
File positional

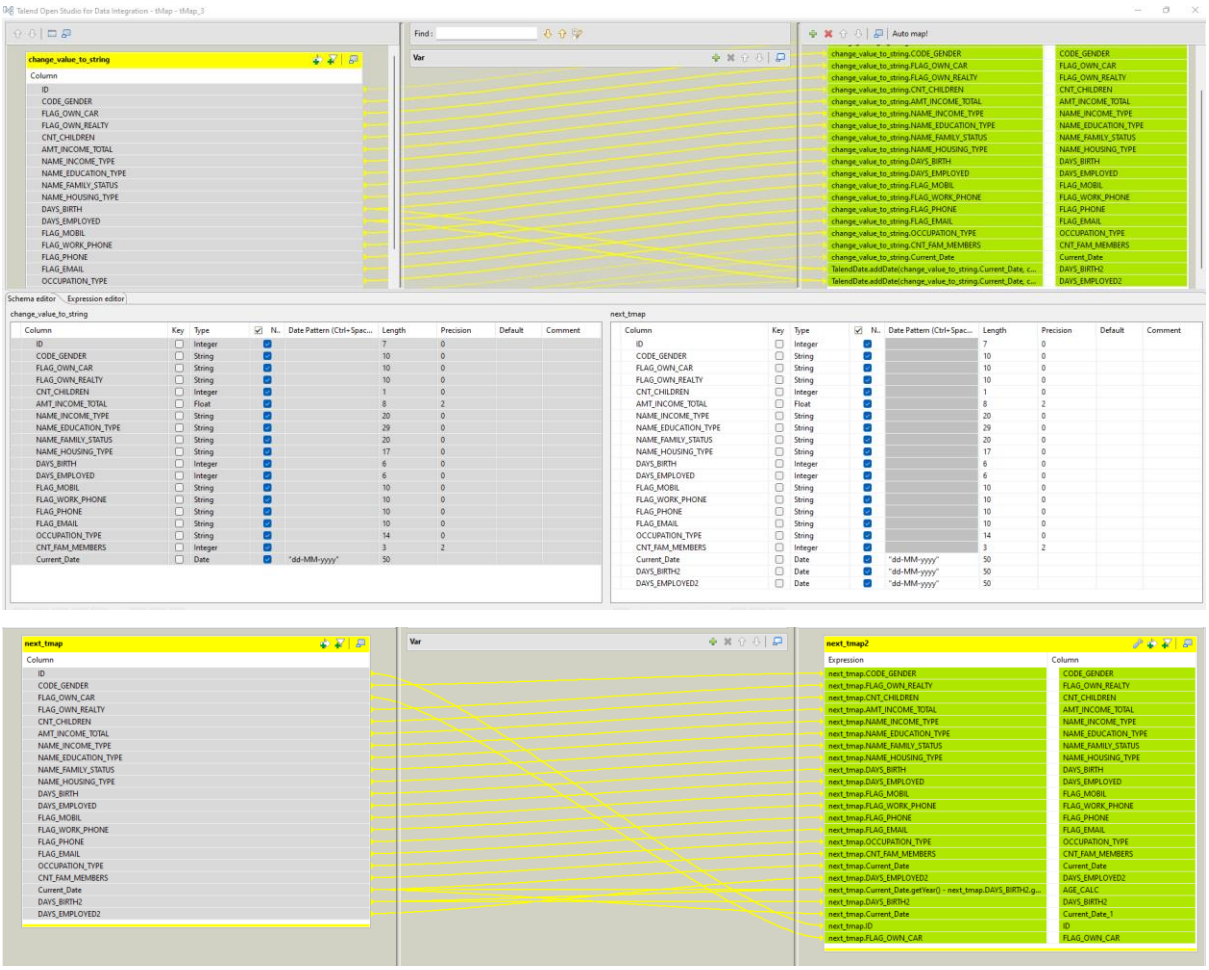

applicant_data_source(tFileInputDelimited_1)
Basic settings
Advanced settings
Dynamic settings
View
Documentation
Property Type
Repository
DELIM:applicant_data_source
Schema
Repository
DELIM:applicant_data_source - metadata
Edit schema
"When the input source is a stream or a zip file,footer and random shouldn't be bigger than 0."
File name/Stream
"C:/Users/Chokri Hamza/Desktop/data_warehouse_project_Approval Prediction/record_credit_card_approval/application_record.csv"
Row Separator
"\n"
Field Separator
","
CSV options
Header
1
Footer
0
Limit
Skip empty rows
Uncompress as zip file
Die on error

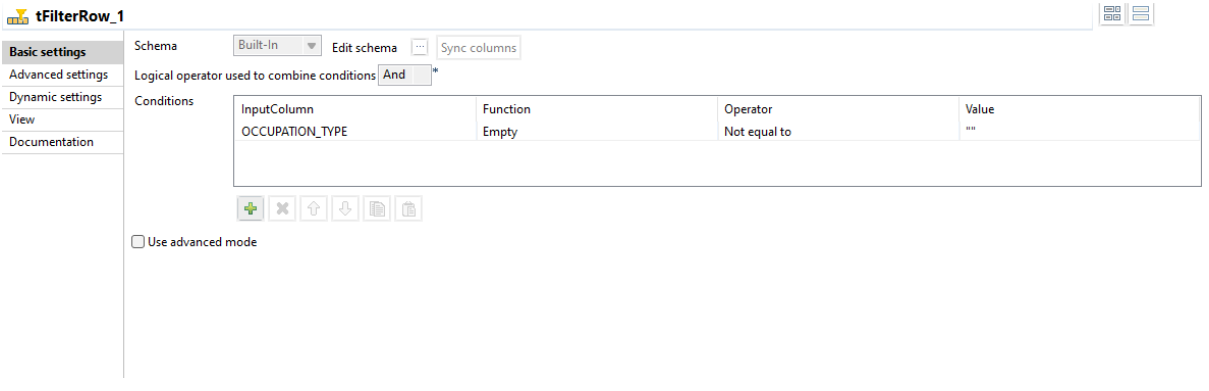

tSortRow_1
Basic settings
Advanced settings
Dynamic settings
View
Documentation
Schema
Built-In
Edit schema
Sync columns
Criteria
Schema column
sort num or alpha?
Order asc or desc?
ID
num
asc


tUniqRow_1
Basic settings
Advanced settings
Dynamic settings
View
Documentation
Schema
Built-In
Edit schema
Sync columns
Unique key
Column
Key attribute
Case sensitive
ID
CODE_GENDER
FLAG_OWN_CAR
FLAG_OWN_REALTY
CNT_CHILDREN
AMT_INCOME_TOTAL
NAME_INCOME_TYPE
NAME_EDUCATION_TYPE
NAME_FAMILY_STATUS
NAME_HOUSING_TYPE
DAYS_BIRTH
DAYS_EMPLOYED
FLAG_MOBIL
FLAG_WORK_PHONE
FLAG_PHONE
FLAG_EMAIL
OCCUPATION_TYPE
CNT_FAM_MEMBERS


tConvertType_1
Basic settings
Advanced settings
Dynamic settings
View
Documentation
Schema
Built-In
Edit schema
Sync columns
Auto Cast
Set empty values to Null before converting
Die on error


Talend Open Studio for Data Integration - tMap - tMap_1
row4
Column
ID
CODE_GENDER
FLAG_OWN_CAR
FLAG_OWN_REALTY
CNT_CHILDREN
AMT_INCOME_TOTAL
NAME_INCOME_TYPE
NAME_EDUCATION_TYPE
NAME_FAMILY_STATUS
NAME_HOUSING_TYPE
DAYS_BIRTH
DAYS_EMPLOYED
FLAG_MOBIL
FLAG_WORK_PHONE
FLAG_PHONE
FLAG_EMAIL
OCCUPATION_TYPE
CNT_FAM_MEMBERS
Find:
Var
Expression
Type
Variable
Auto map!
change_value_to_string
Expression
Column
row4.ID
row4.CODE_GENDER.replace("F","Female").replace("M","Mal...
row4.FLAG_OWN_CAR.replace("Y","have").replace("N","have...
row4.FLAG_OWN_REALTY.replace("Y","have").replace("N","h...
row4.CNT_CHILDREN
row4.AMT_INCOME_TOTAL
row4.NAME_INCOME_TYPE
row4.NAME_EDUCATION_TYPE
row4.NAME_FAMILY_STATUS
row4.NAME_HOUSING_TYPE
row4.DAYS_BIRTH
row4.DAYS_EMPLOYED
row4.FLAG_MOBIL.replace("1","have").replace("0","haven't")
row4.FLAG_WORK_PHONE.replace("1","have").replace("0","h...
row4.FLAG_PHONE.replace("1","have").replace("0","haven't")
row4.FLAG_EMAIL.replace("1","have").replace("0","haven't")
row4.OCCUPATION_TYPE
row4.CNT_FAM_MEMBERS
TalendDate.getCurrentDate()
Current_Date
Schema editor
Expression editor
row4
Column
Key
Type
N..
Date Pattern (Ctrl+Spac...
Length
Precision
Default
Comment
ID Integer 7 0
CODE_GENDER String 10 0
FLAG_OWN_CAR String 10 0
FLAG_OWN_REALTY String 10 0
CNT_CHILDREN Integer 1 0
AMT_INCOME_TOTAL Float 8 2
NAME_INCOME_TYPE String 20 0
NAME_EDUCATION_TYPE String 29 0
NAME_FAMILY_STATUS String 20 0
NAME_HOUSING_TYPE String 17 0
DAYS_BIRTH Integer 6 0
DAYS_EMPLOYED Integer 6 0
FLAG_MOBIL String 10 0
FLAG_WORK_PHONE String 10 0
FLAG_PHONE String 10 0
FLAG_EMAIL String 10 0
OCCUPATION_TYPE String 14 0
CNT_FAM_MEMBERS Integer 3 2
change_value_to_string
Current_Date Date "dd-MM-yyyy" 50





tFilterRow_1

- Basic settings
- Advanced settings
- Dynamic settings
- View
- Documentation

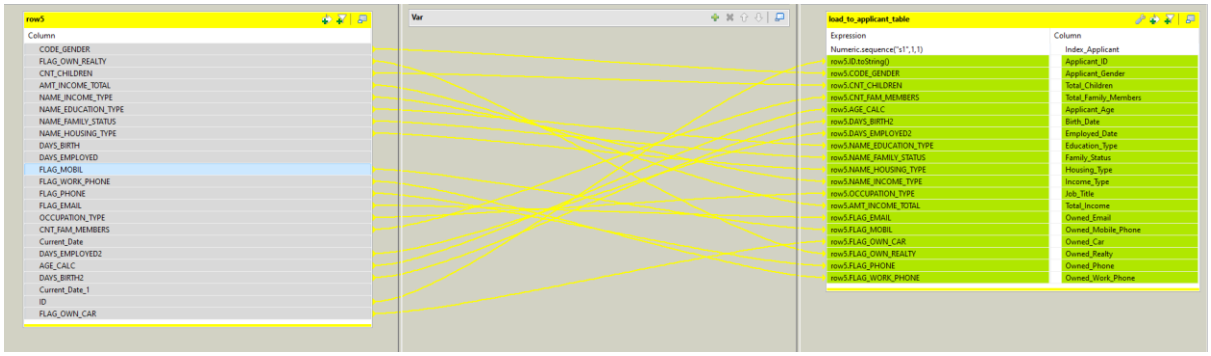
Schema: Built-In | Edit schema | Sync columns

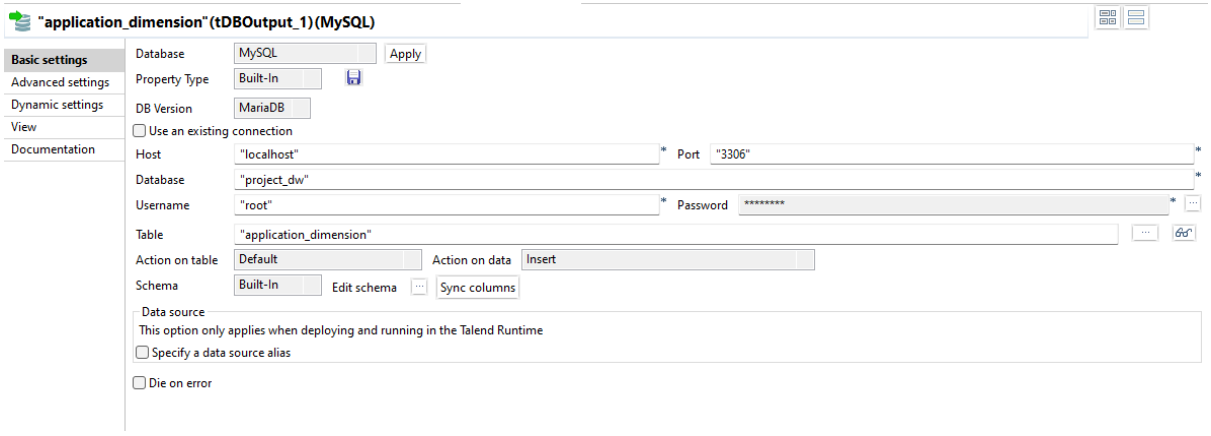
Logical operator used to combine conditions: And *

Conditions

InputColumn	Function	Operator	Value
OCCUPATION_TYPE	Empty	Not equal to	""

Use advanced mode





7. La creation de Data warehouse:

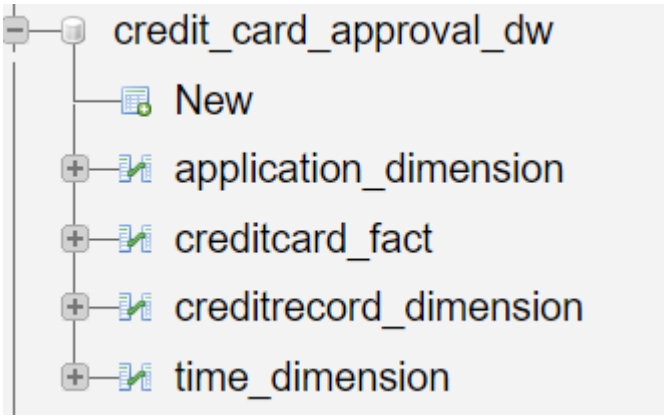
7.1. Le code du chaque table :

```
CREATE TABLE `Time_Dimension`
(
`Time_ID` BIGINT
, `Day` INT
, `Month` INT
, `Year` INT
, `Date` DATETIME
)ENGINE=MyISAM DEFAULT CHARSET=latin1;
-- SQLINES DEMO *** tion_Dimension
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE `Application_Dimension`
(
Index_Applicant BIGINT
, `Applicant_ID` VARCHAR(10)
, `Applicant_Gender`VARCHAR(7)
, `Total_Children` INT
```

```
, `Total_Family_Members` INT
, `Applicant_Age` INT
, `Birth_Date` DATETIME
, `Employed_Date` DATETIME
, `Education_Type` VARCHAR(50)
, `Family_Status` VARCHAR(50)
, `Housing_Type` VARCHAR(50)
, `Income_Type` VARCHAR(50)
, `Job_Title` VARCHAR(50)
, `Total_Income` BIGINT
, `Owned_Email` VARCHAR(10)
, `Owned_Mobile_Phone` VARCHAR(10)
, `Owned_Car` VARCHAR(10)
, `Owned_Realty` VARCHAR(10)
, `Owned_Phone` VARCHAR(10)
, `Owned_Work_Phone` VARCHAR(10)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

CREATE TABLE `CreditRecord_Dimension`
(
CreditRecord_ID BIGINT
, `Record_Date` DATETIME
, `Applicant_ID` VARCHAR(10)
, `Status` VARCHAR(5)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

-- SQLINES DEMO *** ard_Fact
-- SQLINES LICENSE FOR EVALUATION USE ONLY
CREATE TABLE `CreditCard_Fact`
(
  CreditRecord_ID VARCHAR(15)
, `Applicant_ID` VARCHAR(10)
, `Time_ID` BIGINT
, `Good_Debt` INT
, `Bad_Debt` INT
, `Last_ETL` DATETIME
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

COMMIT;

7.2. Le résultat:

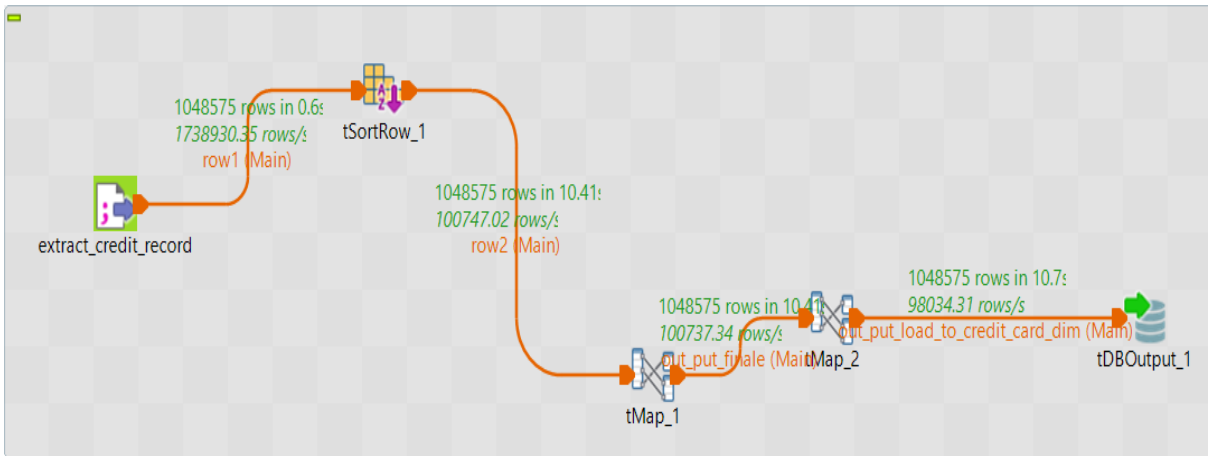


8. Structure de notre application_dimension :

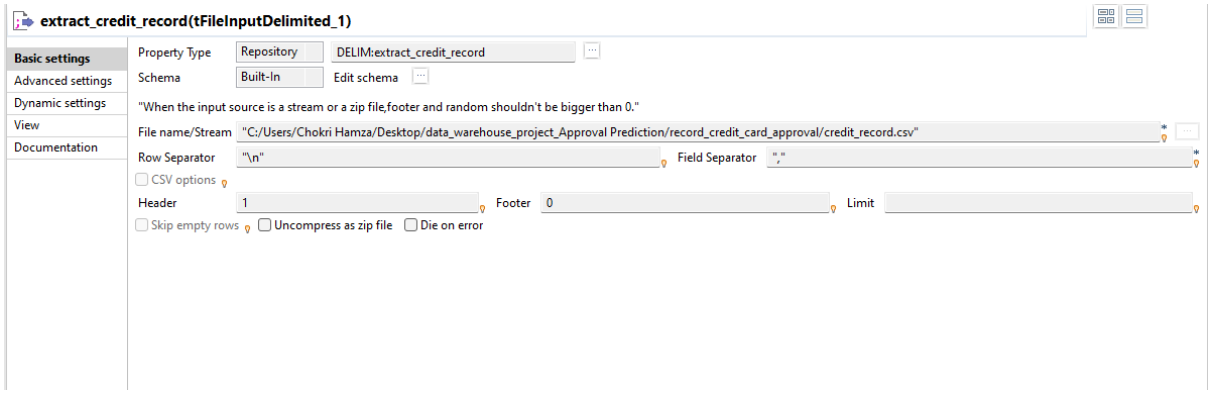
Index_Applicant	Applicant_ID	Applicant_Gender	Total_Children	Total_Family_Members	Applicant_Age	Birth_Date	Employed_Date	Education_Type	Family_Status
1	5008806	Male	0	2	58	1964-02-09 23:05:19	2019-10-18 23:05:19	Secondary / secondary special	Married
2	5008808	Female	0	1	52	1970-07-31 23:05:19	2014-07-19 23:05:19	Secondary / secondary special	Single / not married
3	5008809	Female	0	1	52	1970-07-31 23:05:19	2014-07-19 23:05:19	Secondary / secondary special	Single / not married
4	5008810	Female	0	1	52	1970-07-31 23:05:19	2014-07-19 23:05:19	Secondary / secondary special	Single / not married
5	5008811	Female	0	1	52	1970-07-31 23:05:19	2014-07-19 23:05:19	Secondary / secondary special	Single / not married
6	5008815	Male	0	2	46	1976-09-15 23:05:19	2020-10-17 23:05:19	Higher education	Married
7	5008819	Male	0	2	48	1974-03-24 23:05:19	2019-08-19 23:05:19	Secondary / secondary special	Married
8	5008820	Male	0	2	48	1974-03-24 23:05:19	2019-08-19 23:05:19	Secondary / secondary special	Married
9	5008821	Male	0	2	48	1974-03-24 23:05:19	2019-08-19 23:05:19	Secondary / secondary special	Married
10	5008822	Male	0	2	48	1974-03-24 23:05:19	2019-08-19 23:05:19	Secondary / secondary special	Married
11	5008823	Male	0	2	48	1974-03-24 23:05:19	2019-08-19 23:05:19	Secondary / secondary special	Married
12	5008824	Male	0	2	48	1974-03-24 23:05:19	2019-08-19 23:05:19	Secondary / secondary special	Married
13	5008825	Female	0	2	29	1993-09-09 23:05:19	2019-11-18 23:05:19	Incomplete higher	Married
14	5008826	Female	0	2	29	1993-09-09 23:05:19	2019-11-18 23:05:19	Incomplete higher	Married
15	5008830	Female	0	2	27	1995-06-09 23:05:19	2018-11-17 23:05:19	Secondary / secondary special	Married

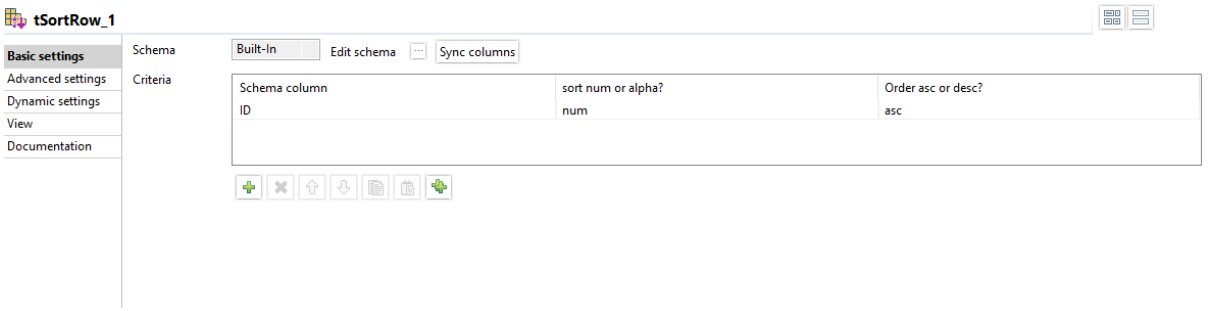
Housing_Type	Income_Type	Job_Title	Total_Income	Owned_Email	Owned_Mobile_Phone	Owned_Car	Owned_Realty	Owned_Phone	Owned_Work_Phone
House / apartment	Working	Security staff	112500	haven't	have	have	have	haven't	haven't
House / apartment	Commercial associate	Sales staff	270000	have	have	haven't	have	have	haven't
House / apartment	Commercial associate	Sales staff	270000	have	have	haven't	have	have	haven't
House / apartment	Commercial associate	Sales staff	270000	have	have	naven't	have	have	haven't
House / apartment	Commercial associate	Sales staff	270000	have	have	haven't	have	have	haven't
House / apartment	Working	Accountants	270000	have	have	have	have	have	have
House / apartment	Commercial associate	Laborers	135000	haven't	have	have	have	haven't	haven't
House / apartment	Commercial associate	Laborers	135000	haven't	have	have	have	haven't	haven't
House / apartment	Commercial associate	Laborers	135000	haven't	have	have	have	haven't	haven't
House / apartment	Commercial associate	Laborers	135000	haven't	have	have	have	haven't	haven't
House / apartment	Commercial associate	Laborers	135000	haven't	have	have	have	haven't	haven't
House / apartment	Commercial associate	Laborers	135000	haven't	have	have	have	haven't	haven't
House / apartment	Working	Accountants	130500	haven't	have	have	haven't	haven't	haven't
House / apartment	Working	Accountants	130500	haven't	have	have	haven't	haven't	haven't
House / apartment	Working	Laborers	157500	haven't	have	haven't	have	have	haven't

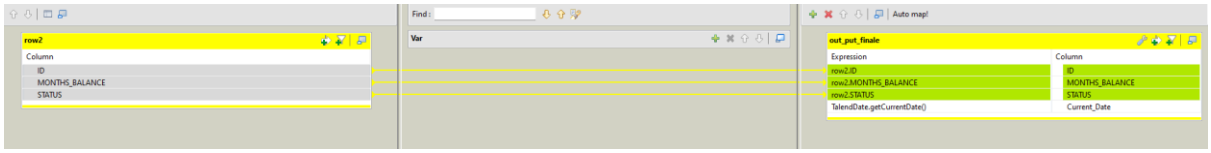
9. ETL du record dimension :

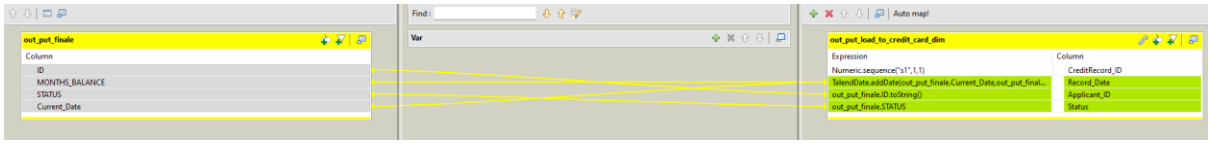


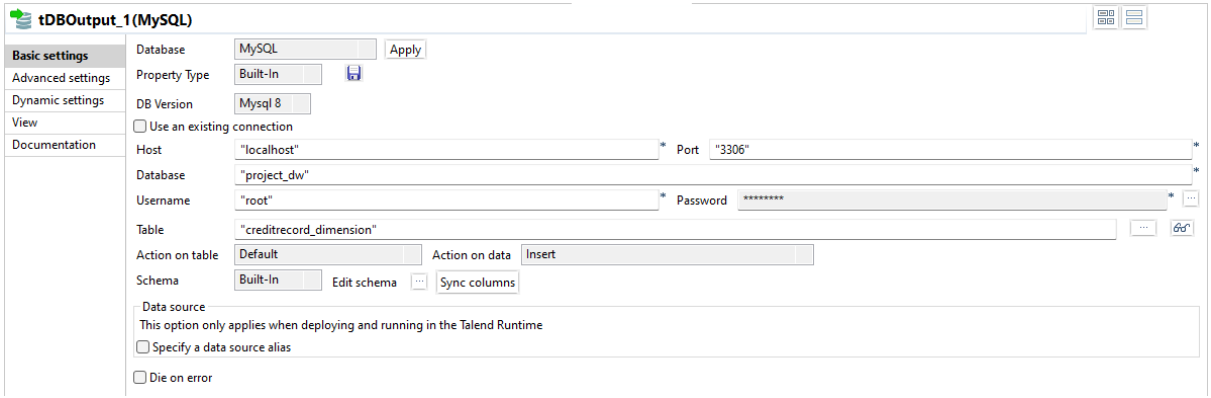
10. Détail de l'ETL pour l'alimentation de record dimension :







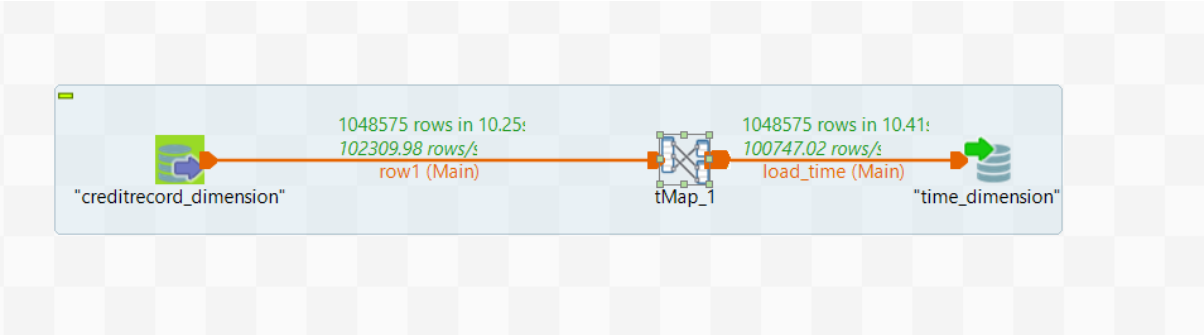




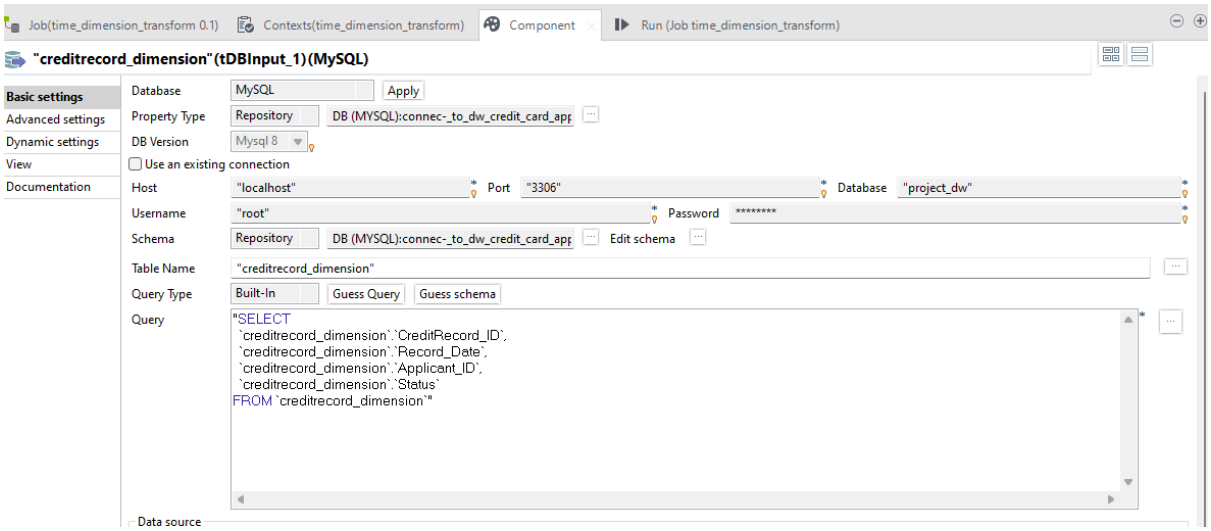
11. Structure de record_dimension :

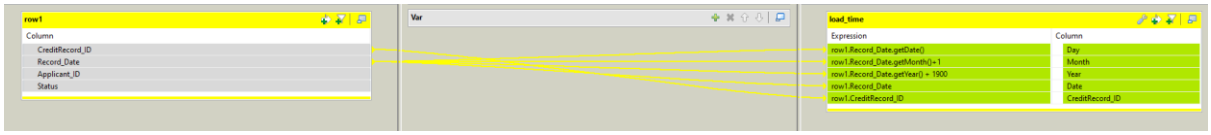
CreditRecord_ID	Record_Date	Applicant_ID	Status
1	2022-11-25	5001711	X
2	2022-10-25	5001711	0
3	2022-09-25	5001711	0
4	2022-08-25	5001711	0
5	2022-11-25	5001712	C
6	2022-10-25	5001712	C
7	2022-09-25	5001712	C
8	2022-08-25	5001712	C
9	2022-07-25	5001712	C
10	2022-06-25	5001712	C
11	2022-05-25	5001712	C
12	2022-04-25	5001712	C
13	2022-03-25	5001712	C
14	2022-02-25	5001712	0
15	2022-01-25	5001712	0
16	2021-12-25	5001712	0
17	2021-11-25	5001712	0
18	2021-10-25	5001712	0
19	2021-09-25	5001712	0
20	2021-08-25	5001712	0
21	2021-07-25	5001712	0
22	2021-06-25	5001712	0
23	2021-05-25	5001712	0
24	2022-11-25	5001713	X
25	2022-10-25	5001713	X

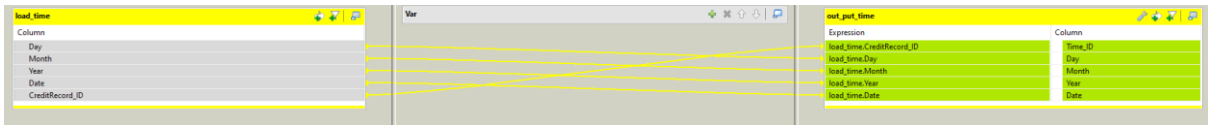
12. ETL pour le Time Dimension :

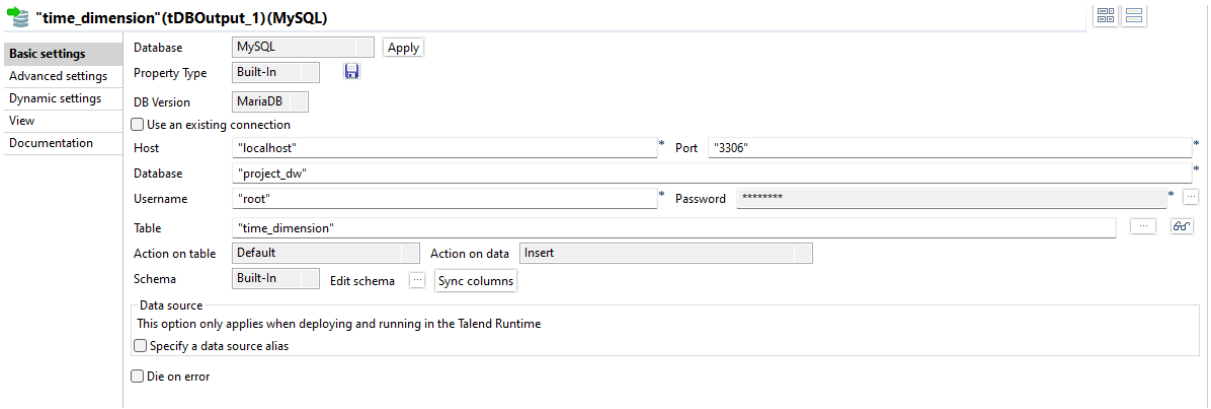


13. Détails ETL pour l'alimentation du Time dimension :





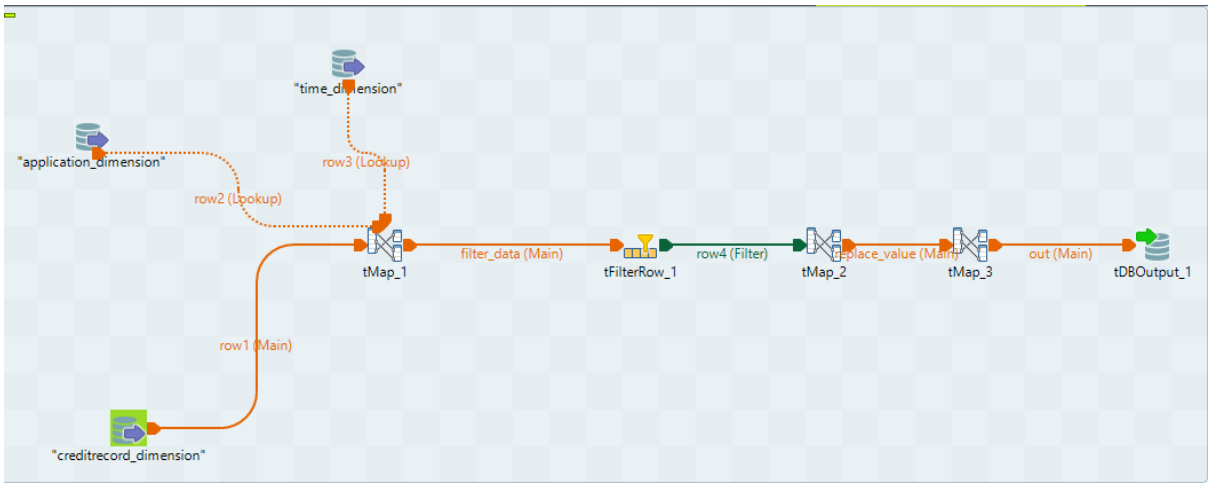




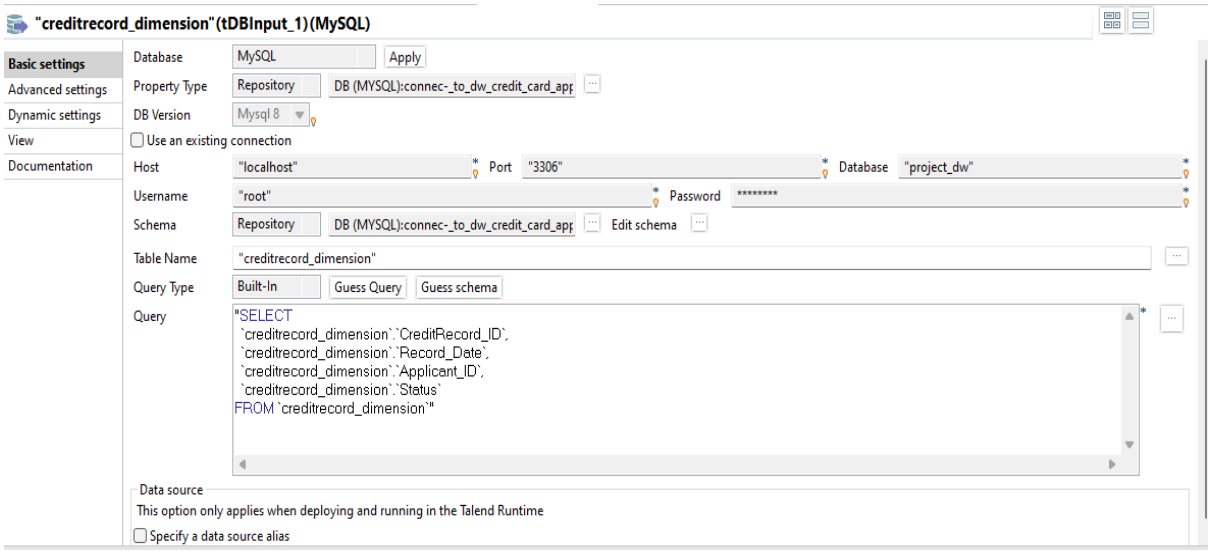
14. Structure de notre Time_dimension :

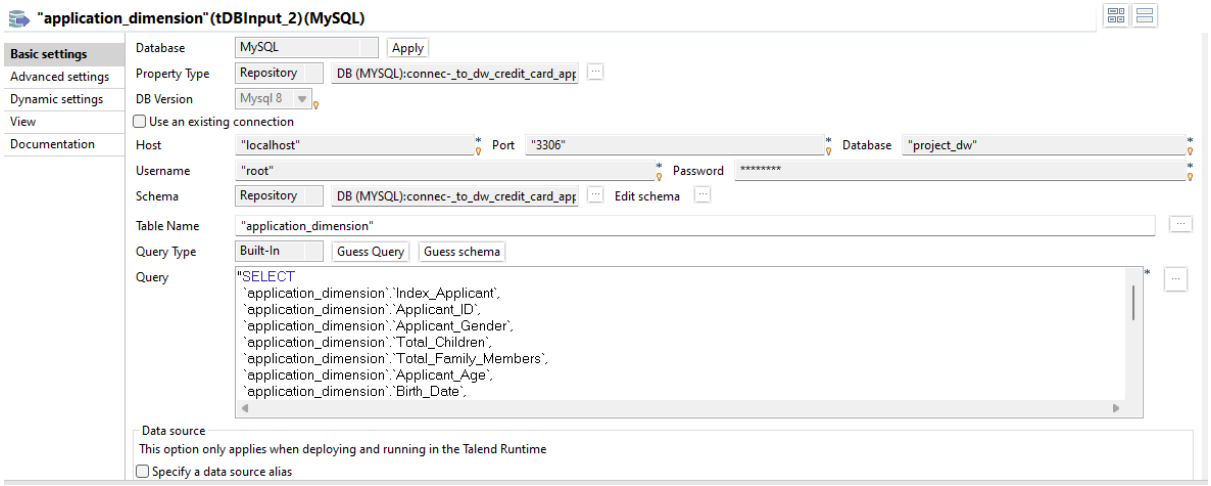
Time_ID	Day	Month	Year	Date
1	25	11	2022	2022-11-25
2	25	10	2022	2022-10-25
3	25	9	2022	2022-09-25
4	25	8	2022	2022-08-25
5	25	11	2022	2022-11-25
6	25	10	2022	2022-10-25
7	25	9	2022	2022-09-25
8	25	8	2022	2022-08-25
9	25	7	2022	2022-07-25
10	25	6	2022	2022-06-25
11	25	5	2022	2022-05-25
12	25	4	2022	2022-04-25
13	25	3	2022	2022-03-25
14	25	2	2022	2022-02-25
15	25	1	2022	2022-01-25
16	25	12	2021	2021-12-25
17	25	11	2021	2021-11-25
18	25	10	2021	2021-10-25
19	25	9	2021	2021-09-25
20	25	8	2021	2021-08-25
21	25	7	2021	2021-07-25
22	25	6	2021	2021-06-25
23	25	5	2021	2021-05-25
24	25	11	2022	2022-11-25
25	25	10	2022	2022-10-25

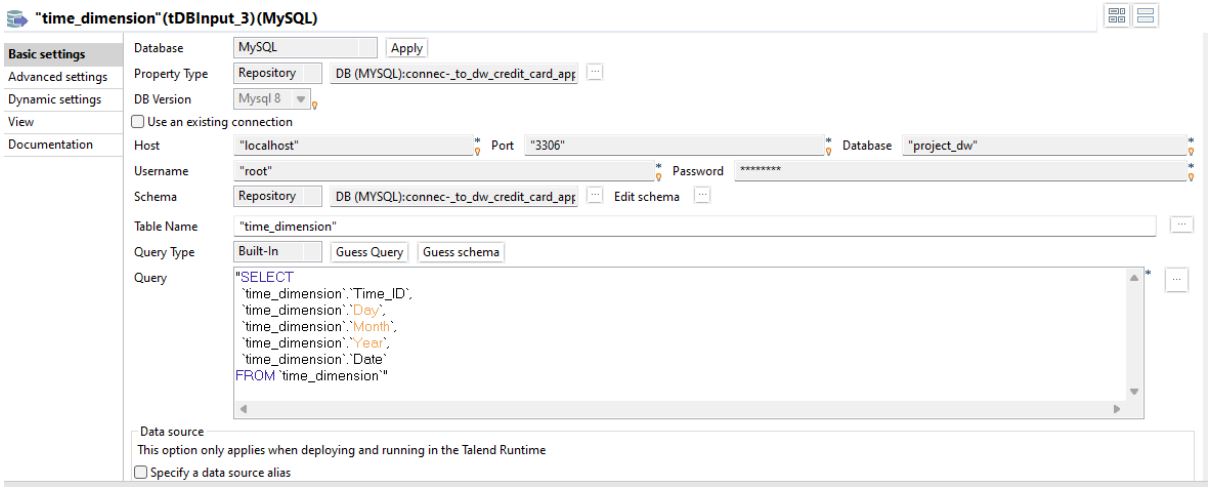
15. ETL pour le credit card fact dimension :

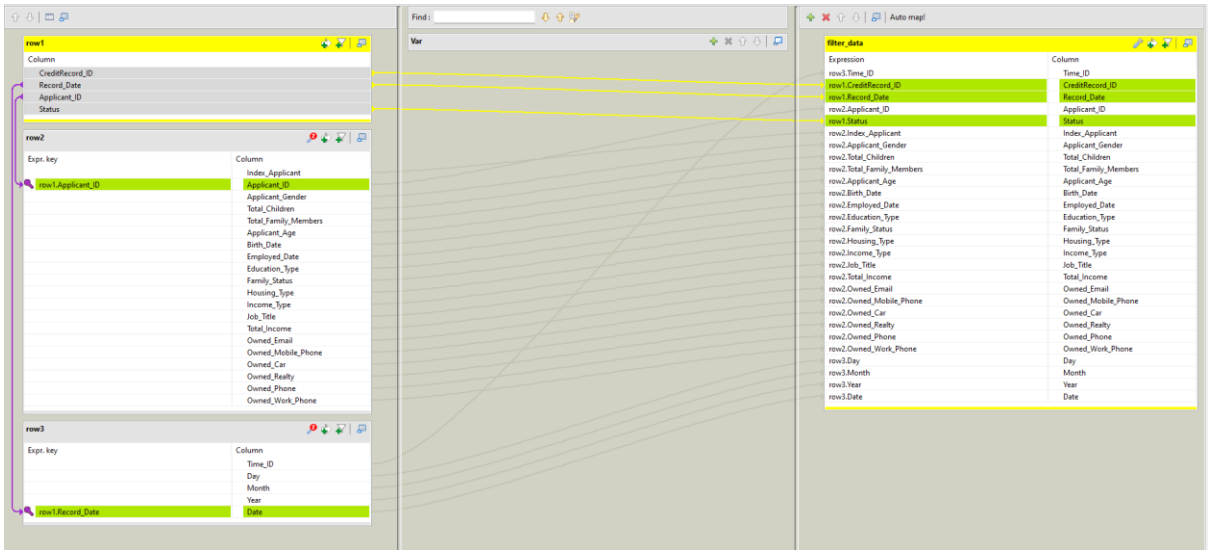


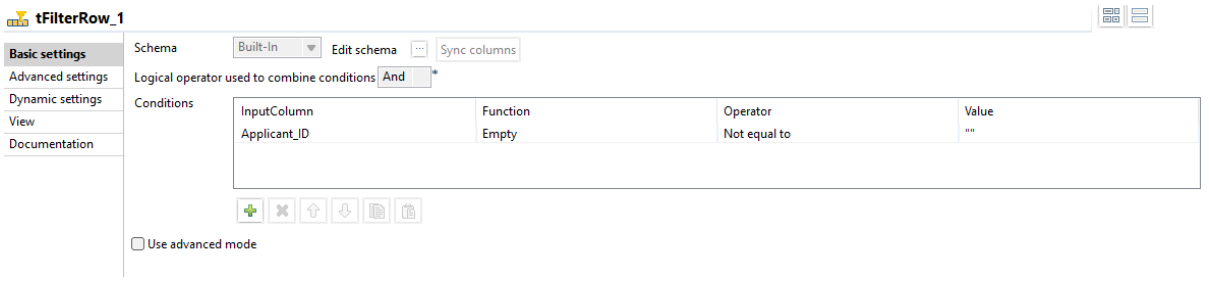
16. Détail de l’ETL pour alimentation du credit card fact dimension :

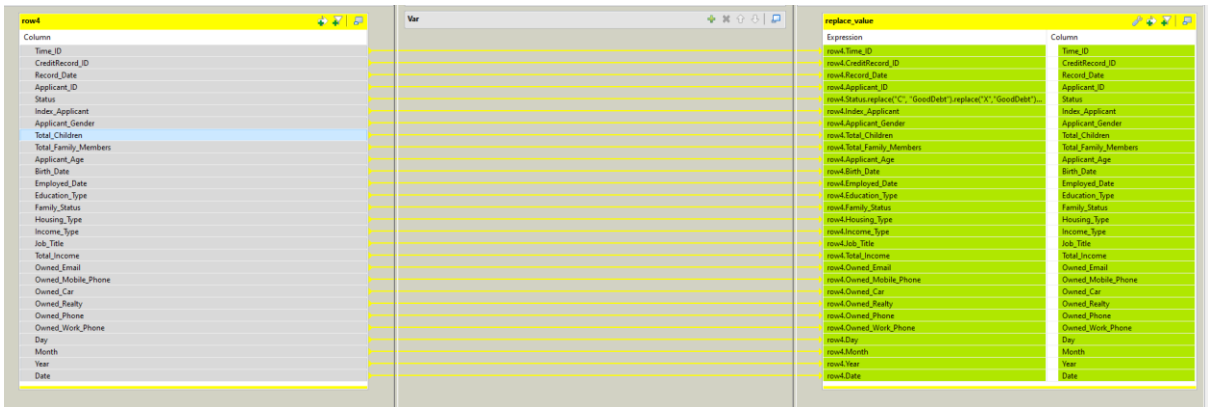


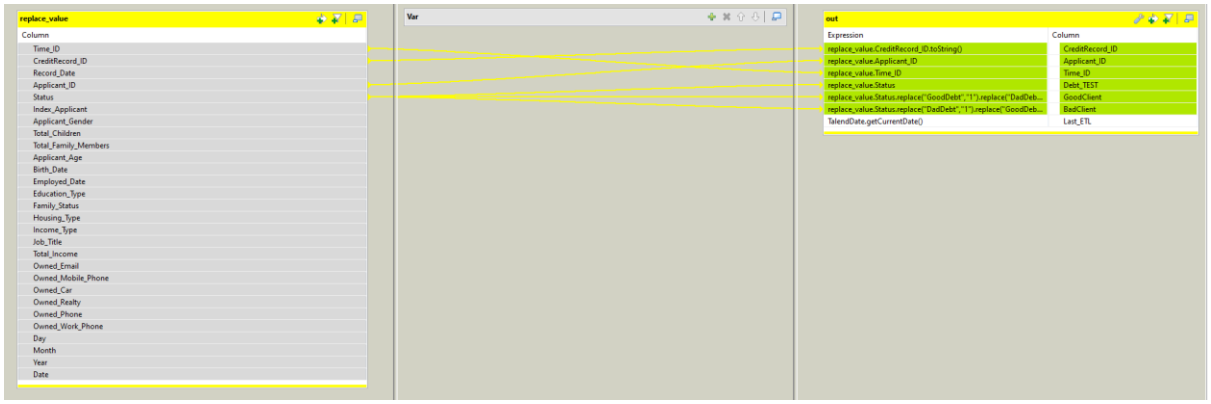


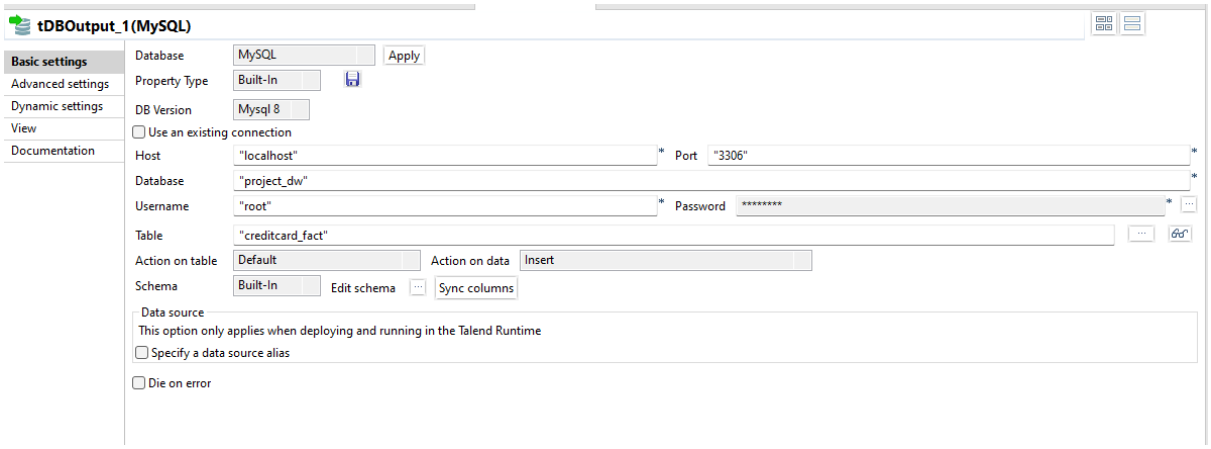



row1
Column
CreditRecord_ID
Record_Date
Applicant_ID
Status
row2
Expr. key
Column
row1.Applicant_ID
Index_Applicant
Applicant_ID
Applicant_Gender
Total_Children
Total_Family_Members
Applicant_Age
Birth_Date
Employed_Date
Education_Type
Family_Status
Housing_Type
Income_Type
Job_Title
Total_Income
Owned_Email
Owned_Mobile_Phone
Owned_Car
Owned_Realty
Owned_Phone
Owned_Work_Phone
row3
Expr. key
Column
row1.Record_Date
Time_ID
Day
Month
Year
Date
Find:
Var
Auto map!
filter_data
Expression
Column
row3.Time_ID
row1.CreditRecord_ID
row1.Record_Date
row2.Applicant_ID
row1.Status
row2.Index_Applicant
row2.Applicant_Gender
row2.Total_Children
row2.Total_Family_Members
row2.Applicant_Age
row2.Birth_Date
row2.Employed_Date
row2.Education_Type
row2.Family_Status
row2.Housing_Type
row2.Income_Type
row2.Job_Title
row2.Total_Income
row2.Owned_Email
row2.Owned_Mobile_Phone
row2.Owned_Car
row2.Owned_Realty
row2.Owned_Phone
row2.Owned_Work_Phone
row3.Day
row3.Month
row3.Year
row3.Date


tFilterRow_1
Basic settings
Advanced settings
Dynamic settings
View
Documentation
Schema
Built-In
Edit schema
Sync columns
Logical operator used to combine conditions
And
Conditions
InputColumn
Function
Operator
Value
Applicant_ID
Empty
Not equal to
""
Use advanced mode


row4
Column
Time_ID
CreditRecord_ID
Record_Date
Applicant_ID
Status
Index_Applicant
Applicant_Gender
Total_Children
Total_Family_Members
Applicant_Age
Birth_Date
Employed_Date
Education_Type
Family_Status
Housing_Type
Income_Type
Job_Title
Total_Income
Owned_Email
Owned_Mobile_Phone
Owned_Car
Owned_Realty
Owned_Phone
Owned_Work_Phone
Day
Month
Year
Date
Var
replace_value
Expression
Column
row4.Time_ID
row4.CreditRecord_ID
row4.Record_Date
row4.Applicant_ID
row4.Status.replace("C", "GoodDebt").replace("X","GoodDebt")...
row4.Index_Applicant
row4.Applicant_Gender
row4.Total_Children
row4.Total_Family_Members
row4.Applicant_Age
row4.Birth_Date
row4.Employed_Date
row4.Education_Type
row4.Family_Status
row4.Housing_Type
row4.Income_Type
row4.Job_Title
row4.Total_Income
row4.Owned_Email
row4.Owned_Mobile_Phone
row4.Owned_Car
row4.Owned_Realty
row4.Owned_Phone
row4.Owned_Work_Phone
row4.Day
row4.Month
row4.Year
row4.Date





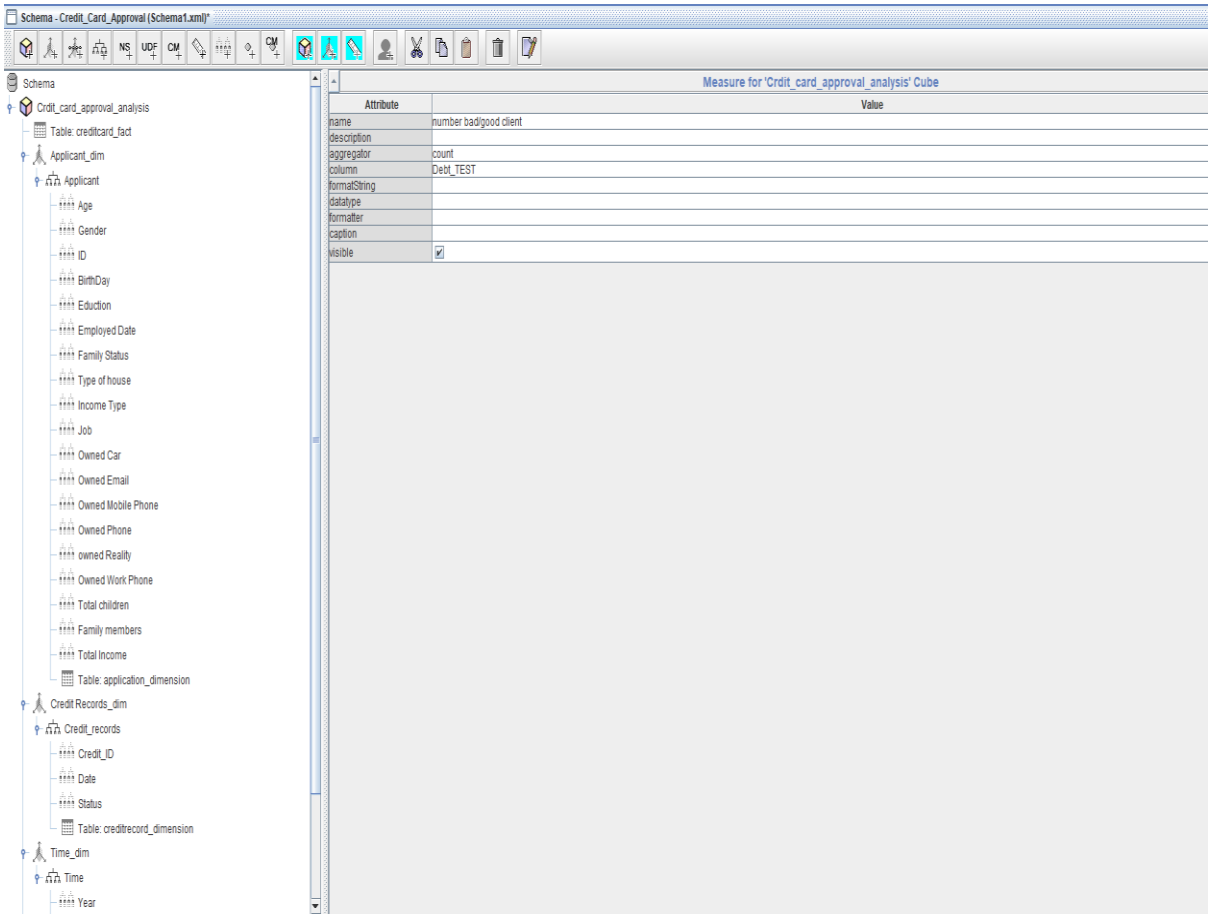
17. Structure de notre credit card fact dimension :

+ Options

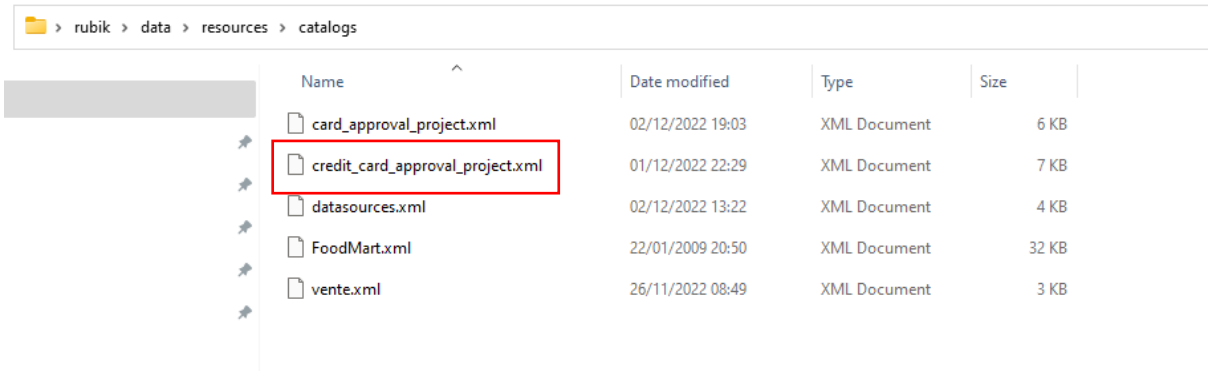
CreditRecord_ID	Applicant_ID	Time_ID	Debt_TEST	GoodClient	BadClient	Last_ETL
32	5008806	189	DadDebt	0	1	2022-12-19 16:13:16
33	5008806	190	DadDebt	0	1	2022-12-19 16:13:16
34	5008806	191	DadDebt	0	1	2022-12-19 16:13:16
35	5008806	192	DadDebt	0	1	2022-12-19 16:13:16
36	5008806	193	DadDebt	0	1	2022-12-19 16:13:16
37	5008806	194	DadDebt	0	1	2022-12-19 16:13:16
38	5008806	177	DadDebt	0	1	2022-12-19 16:13:16
39	5008806	178	GoodDebt	1	0	2022-12-19 16:13:16
40	5008806	179	GoodDebt	1	0	2022-12-19 16:13:16
41	5008806	180	GoodDebt	1	0	2022-12-19 16:13:16
42	5008806	181	GoodDebt	1	0	2022-12-19 16:13:16
43	5008806	182	GoodDebt	1	0	2022-12-19 16:13:16
44	5008806	183	GoodDebt	1	0	2022-12-19 16:13:16
45	5008806	184	GoodDebt	1	0	2022-12-19 16:13:16
46	5008806	185	GoodDebt	1	0	2022-12-19 16:13:16
47	5008806	186	GoodDebt	1	0	2022-12-19 16:13:16
48	5008806	187	GoodDebt	1	0	2022-12-19 16:13:16
49	5008806	188	GoodDebt	1	0	2022-12-19 16:13:16
50	5008806	152	GoodDebt	1	0	2022-12-19 16:13:16
51	5008806	153	GoodDebt	1	0	2022-12-19 16:13:16
52	5008806	154	GoodDebt	1	0	2022-12-19 16:13:16
53	5008806	120	GoodDebt	1	0	2022-12-19 16:13:16
54	5008806	121	GoodDebt	1	0	2022-12-19 16:13:16
55	5008806	122	GoodDebt	1	0	2022-12-19 16:13:16
56	5008806	123	GoodDebt	1	0	2022-12-19 16:13:16

18. La construction des cubes OLAP :

On a utilisé l’outil ” Schema workbench “ pour faire la création de notre cube OLAP.



19. Exporter le schéma sous forme d’un fichier .XML :



20. Le code source:

```
<Schema name="Credit_Card_Approval">
  <Cube name="Crdit_card_approval_analysis" visible="true" cache="true" enabled="true">
    <Table name="creditcard_fact">
    </Table>
    <Dimension type="StandardDimension" visible="true" foreignKey="Applicant_ID"
highCardinality="false" name="Applicant_dim">
      <Hierarchy name="Applicant" visible="true" hasAll="true" >
        <Table name="application_dimension">
        </Table>

<Level name="Gender" visible="true" table="application_dimension" column="Applicant_Gender"
nameColumn="Applicant_Gender" type="String" uniqueMembers="false" levelType="Regular"
hideMemberIf="Never">
        </Level>
<Level name="Eduction" visible="true" table="application_dimension" column="Education_Type"
nameColumn="Education_Type" type="String" uniqueMembers="false" levelType="Regular"
hideMemberIf="Never">
        </Level>
        <Level name="Age" visible="true" table="application_dimension" column="Applicant_Age"
nameColumn="Applicant_Age" type="String" uniqueMembers="false" levelType="Regular"
hideMemberIf="Never">
        </Level>


        <Level name="BirthDay" visible="true" table="application_dimension" column="Birth_Date"
nameColumn="Birth_Date" type="String" uniqueMembers="false" levelType="Regular"
hideMemberIf="Never">
        </Level>

        <Level name="Employed Date" visible="true" table="application_dimension"
column="Employed_Date" nameColumn="Employed_Date" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never">
        </Level>
        <Level name="Family Status" visible="true" table="application_dimension"
column="Family_Status" nameColumn="Family_Status" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never">
```

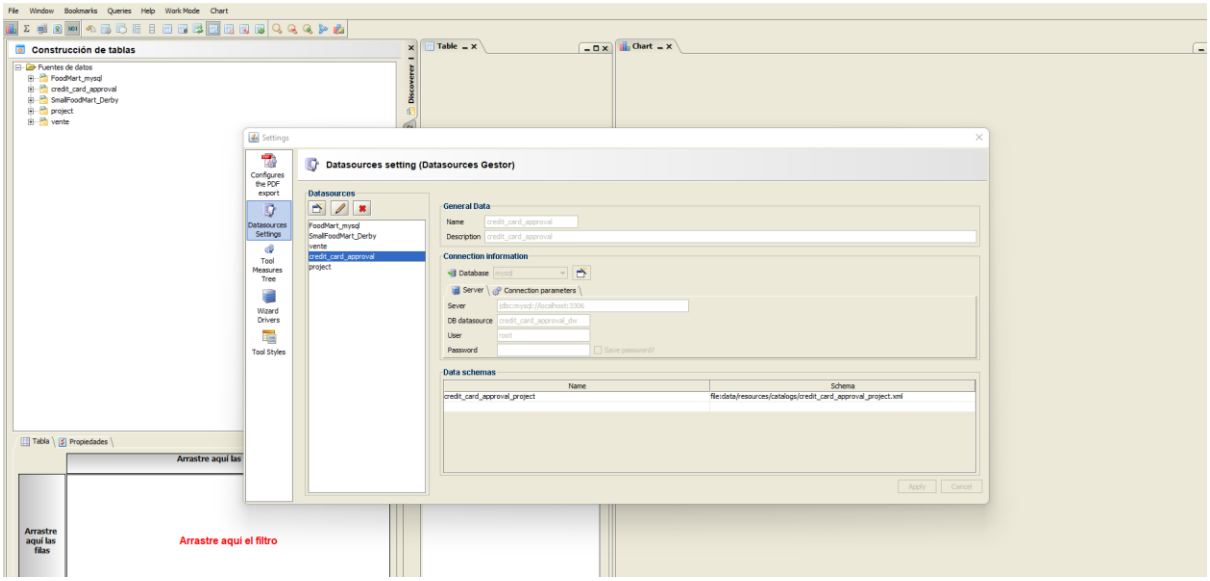
```
</Level>
<Level name="Type of house" visible="true" table="application_dimension"
column="Housing_Type" nameColumn="Housing_Type" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never">
</Level>
<Level name="Income Type" visible="true" table="application_dimension"
column="Income_Type" nameColumn="Income_Type" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never">
</Level>
<Level name="Job" visible="true" table="application_dimension" column="Job_Title"
nameColumn="Job_Title" type="String" uniqueMembers="false" levelType="Regular"
hideMemberIf="Never">
</Level>
 <Level name="Total Income" visible="true" table="application_dimension"
column="Total_Income" nameColumn="Total_Income" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never">
</Level>
<Level name="Owned Car" visible="true" table="application_dimension"
column="Owned_Car" nameColumn="Owned_Car" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never">
</Level>


<Level name="owned Reality" visible="true" table="application_dimension"
column="Owned_Realty" nameColumn="Owned_Realty" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never">
</Level>

<Level name="Total children" visible="true" table="application_dimension"
column="Total_Children" nameColumn="Total_Children" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never">
</Level>
<Level name="Family members" visible="true" table="application_dimension"
column="Total_Family_Members" nameColumn="Total_Family_Members" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
</Level>
```

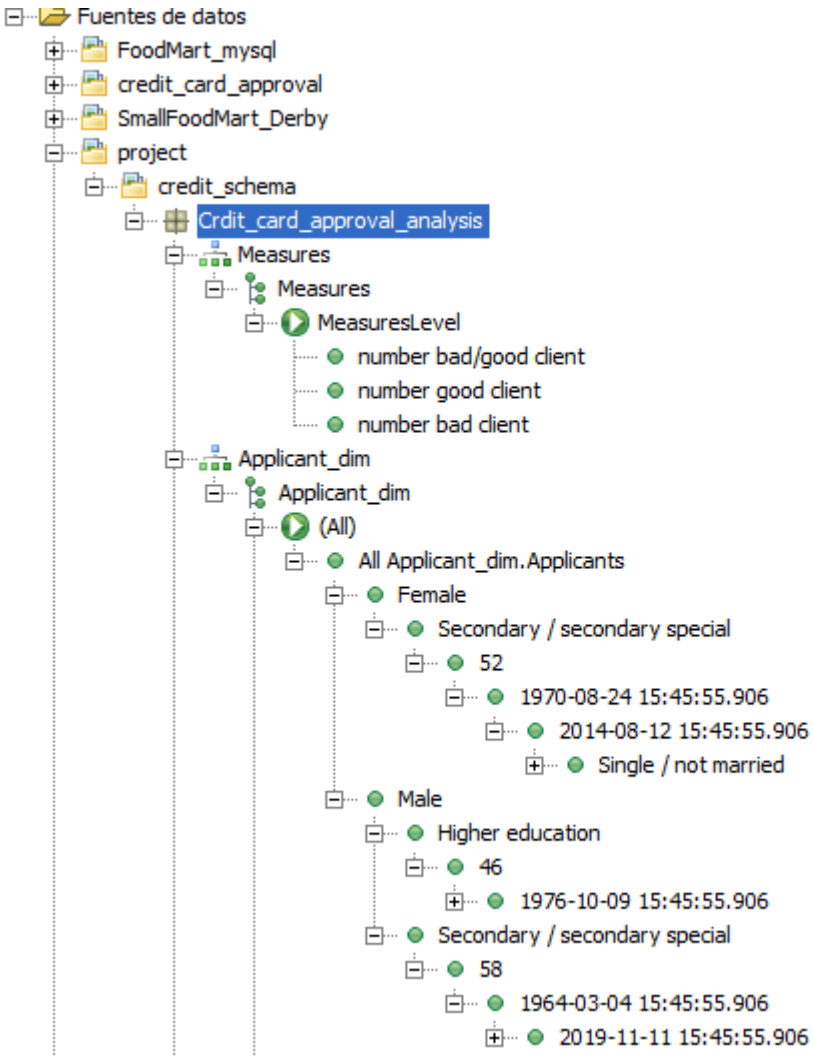
```
    </Hierarchy>
  </Dimension>
  <Dimension type="StandardDimension" visible="true" foreignKey="CreditRecord_ID"
highCardinality="false" name="Credit Records_dim">
    <Hierarchy name="Credit_records" visible="true" hasAll="true"
primaryKey="CreditRecord_ID">
 <Table name="creditrecord_dimension">
      </Table>
       <Level name="Status" visible="true" table="creditrecord_dimension" column="Status"
nameColumn="Status" type="String" uniqueMembers="false" levelType="Regular"
hideMemberIf="Never">
      </Level>
      <Level name="Credit_ID" visible="true" table="creditrecord_dimension"
column="CreditRecord_ID" nameColumn="CreditRecord_ID" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
      </Level>


    </Hierarchy>
  </Dimension>
  <Dimension type="StandardDimension" visible="true" foreignKey="Time_ID"
highCardinality="false" name="Time_dim">
    <Hierarchy name="Time" visible="true" hasAll="true" primaryKey="Time_ID">
      <Table name="time_dimension">
      </Table>
      <Level name="Year" visible="true" table="time_dimension" column="Year"
nameColumn="Year" type="String" uniqueMembers="false" levelType="Regular"
hideMemberIf="Never">
      </Level>
      <Level name="Month" visible="true" table="time_dimension" column="Month"
nameColumn="Month" type="String" uniqueMembers="false" levelType="Regular"
hideMemberIf="Never">
      </Level>
      <Level name="Day" visible="true" table="time_dimension" column="Day"
nameColumn="Day" type="String" uniqueMembers="false" levelType="Regular"
hideMemberIf="Never">
      </Level>
```

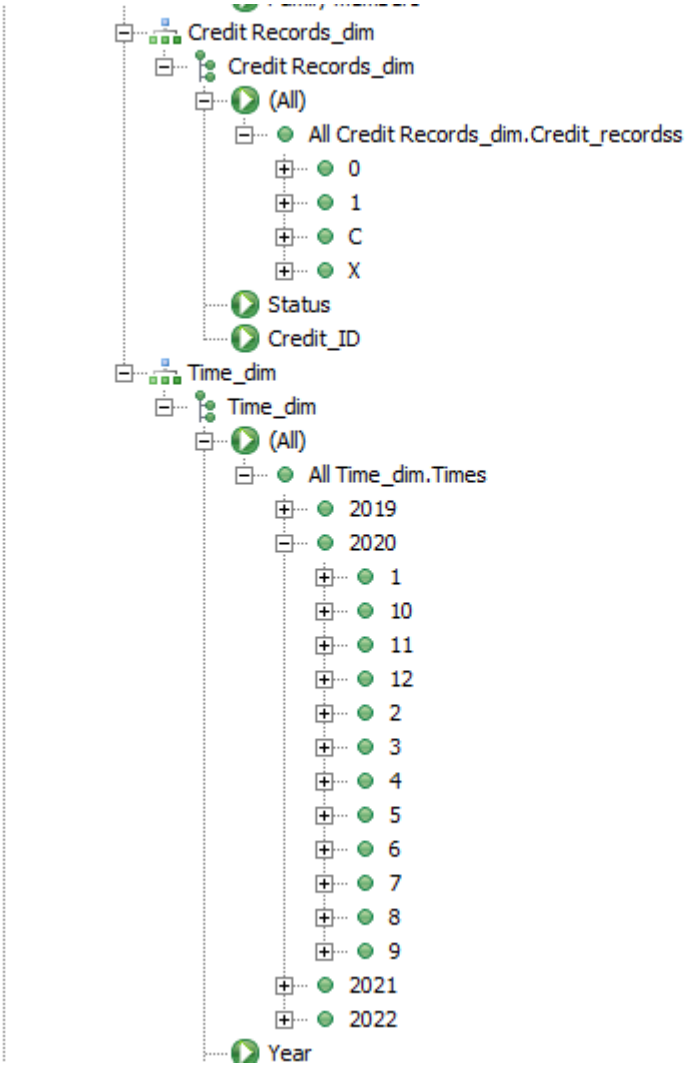
```
        <Level name="Date" visible="true" table="time_dimension" column="Date"
nameColumn="Date" type="Timestamp" uniqueMembers="false" levelType="Regular"
hideMemberIf="Never">
        </Level>
      </Hierarchy>
    </Dimension>

    <Measure name="number good client" column="GoodClient" aggregator="sum" visible="true">
    </Measure>
    <Measure name="number bad client" column="BadClient" aggregator="sum" visible="true">
    </Measure>
  </Cube>
</Schema>
```

21. L'utilisation de l'outil « JRubik » pour faire les différentes analyses et visualisation :

21.1. Faire la connexion avec notre data Warehouse :




Fuentes de datos
FoodMart_mysql
credit_card_approval
SmallFoodMart_Derby
project
credit_schema
Crdit_card_approval_analysis
Measures
Measures
MeasuresLevel
number bad/good client
number good client
number bad client
Applicant_dim
Applicant_dim
(All)
All Applicant_dim.Applicants
Female
Secondary / secondary special
52
1970-08-24 15:45:55.906
2014-08-12 15:45:55.906
Single / not married
Male
Higher education
46
1976-10-09 15:45:55.906
Secondary / secondary special
58
1964-03-04 15:45:55.906
2019-11-11 15:45:55.906



22. Les différentes analyses et visualisation :

22.1. Les requêtes MDX appliquées :

22.1.1. Requête pour visualiser le nombre du « Bad Client » et « Good Client »

```
select {[Measures]. [number good client], [Measures]. [number bad client]}
ON COLUMNS,
  Hierarchize({[Applicant_dim.Applicant].[All Applicant_dim.Applicants]})
ON ROWS
from [Crdit_card_approval_analysis]
```

22.1.2. Résultat :

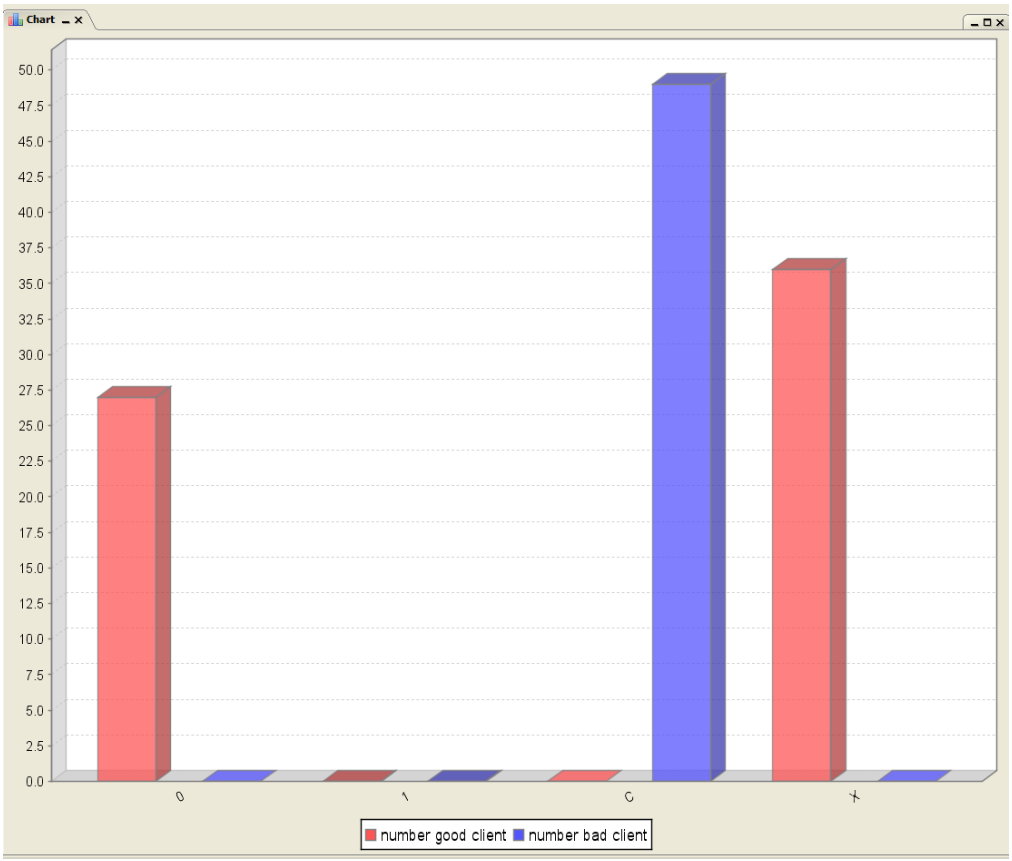
Applicant_dim	number good client	number bad client
+All Applicant_dim.Applicants	63	49

22.1.3. Requête pour visualiser le nombre de « Good Client » « Bad Client » par Statut :

```
select {[Measures]. [number good client], [Measures]. [number bad client]}}
ON COLUMNS,
  Hierarchize ([Credit Records_dim.Credit_records].[All Credit
Records_dim.Credit_recordss].Children)
ON ROWS
from [Crdit_card_approval_analysis]
```

22.1.4. Résultat:

Credit Records_dim	number good client	number bad client
+0	27	0
+1		
+C	0	49
+X	36	0

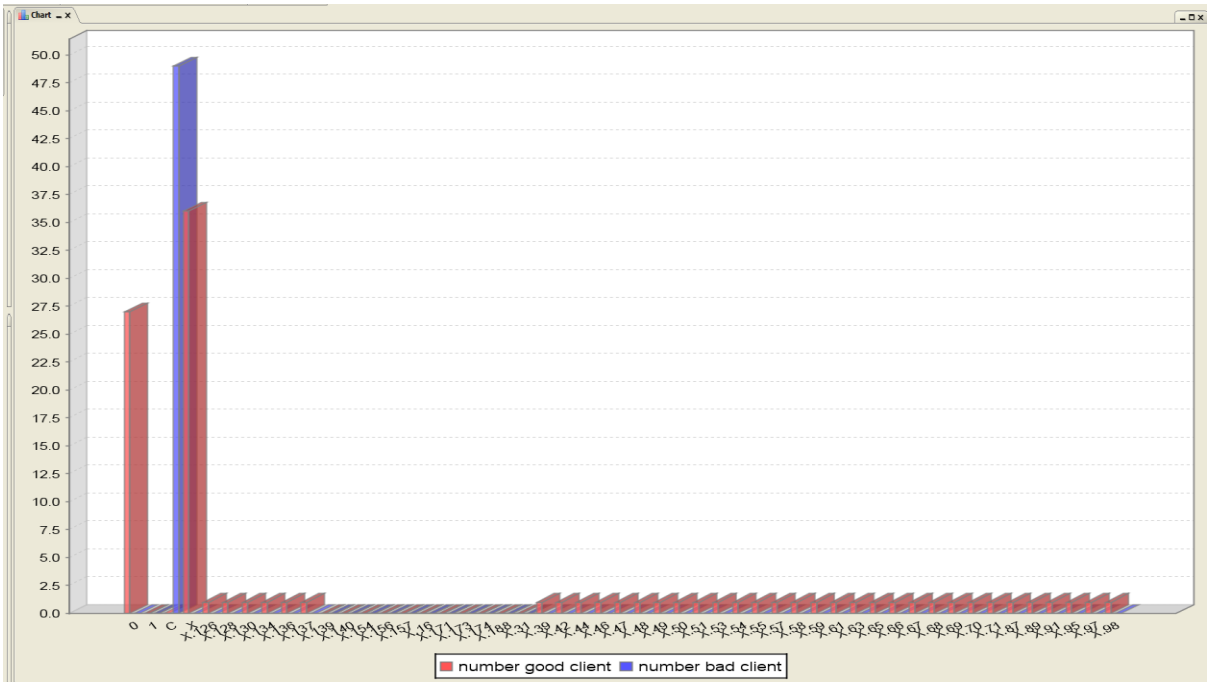


22.1.5. Requête pour visualiser le nombre du « Good Client » « Bad Client » par Client pour le statut X :

```
select {[Measures]. [number good client],
 [Measures]. [number bad client]}
ON COLUMNS,
  Hierarchize(Union({[Credit Records_dim.Credit_records].[All Credit
Records_dim.Credit_recordss].[0],
```

```
 [Credit Records_dim.Credit_records].[All Credit Records_dim.Credit_recordss].[1],
 [Credit Records_dim.Credit_records].[All Credit Records_dim.Credit_recordss].[C],
 [Credit Records_dim.Credit_records].[All Credit Records_dim.Credit_recordss].[X]},
 [Credit Records_dim.Credit_records].[All Credit Records_dim.Credit_recordss].[X].Children))
ON ROWS
from [Crdit_card_approval_analysis]
```

22.1.6. Résultat :



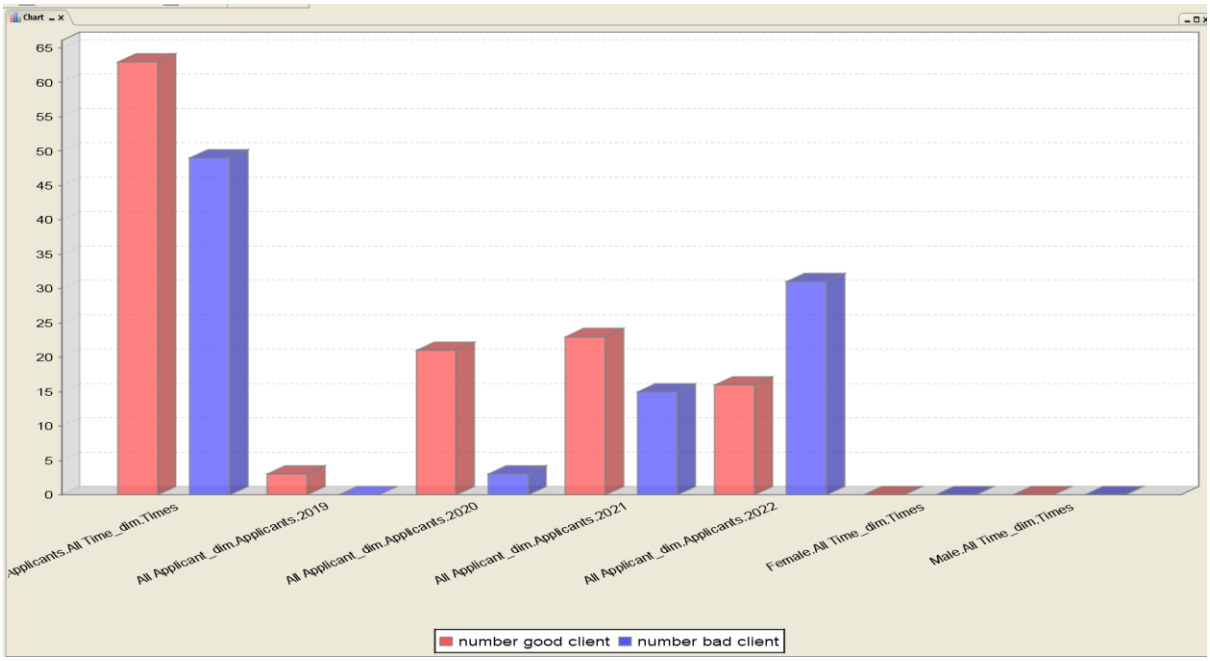
22.1.7. Requête pour visualiser le nombre du « Good Client » « Bad Client » par Client pour chaque année :

```
select {[Measures].[number good client],
 [Measures].[number bad client]}
ON COLUMNS,
  Hierarchize(Union(Union(Crossjoin({[Applicant_dim.Applicant].[All Applicant_dim.Applicants]},
 {[Time_dim.Time].[All Time_dim.Times]}),
 Crossjoin({[Applicant_dim.Applicant].[All Applicant_dim.Applicants]},
 [Time_dim.Time].[All Time_dim.Times].Children)),
 Crossjoin([Applicant_dim.Applicant].[All Applicant_dim.Applicants].Children, {[Time_dim.Time].[All
Time_dim.Times]})))
ON ROWS
```

```
from [Crdit_card_approval_analysis]
```

22.1.8. Résultat:

Applicant_dim	Time_dim	number good client	number bad client
−All Applicant_dim.Applicants	−All Time_dim.Times	63	49
	+2019	3	0
	+2020	21	3
	+2021	23	15
	+2022	16	31
+Female	+All Time_dim.Times		
+Male	+All Time_dim.Times		

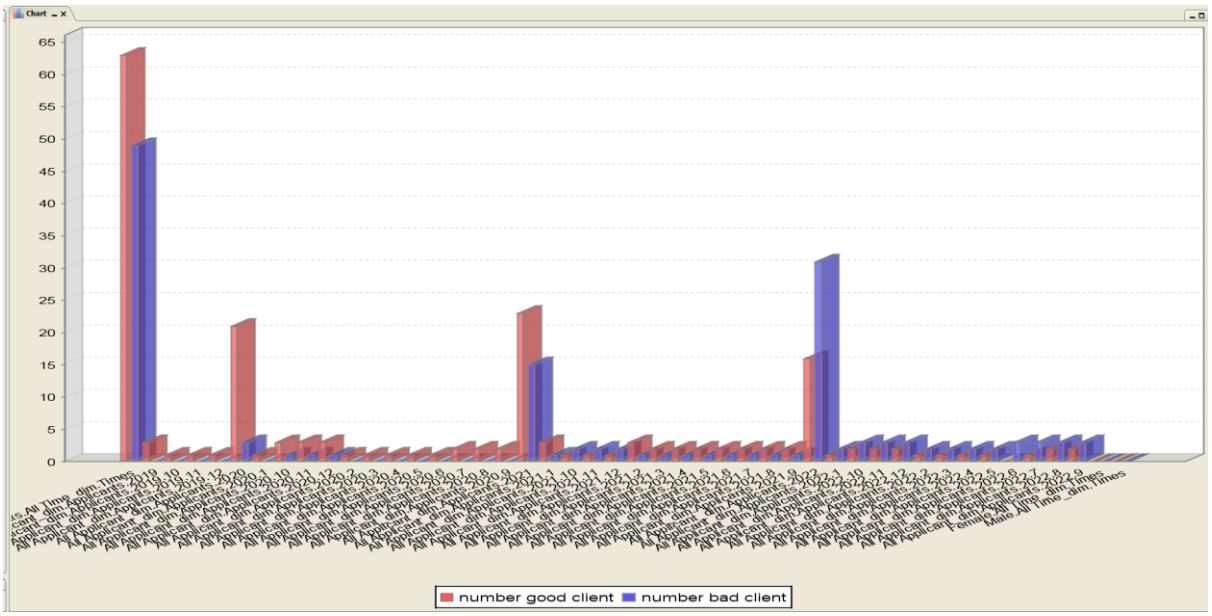


22.1.9. Requête pour visualiser le nombre du « Good Client » « Bad Client » par Client et par mois :

```
select {[Measures]. [number good client],
 [Measures]. [number bad client]}
ON COLUMNS,
  Hierarchize (Union (Union (Union (Union (Union (Crossjoin({[Applicant_dim.Applicant].[All
Applicant_dim.Applicants]},
 {[Time_dim.Time].[All Time_dim.Times],
 [Time_dim.Time].[All Time_dim.Times].[2019],
```

```
 [Time_dim.Time].[All Time_dim.Times].[2020],
 [Time_dim.Time].[All Time_dim.Times].[2021],
 [Time_dim.Time].[All Time_dim.Times].[2022]}),
 Crossjoin({[Applicant_dim.Applicant].[All Applicant_dim.Applicants]},
 [Time_dim.Time].[All Time_dim.Times].[2019].Children)),
 Crossjoin({[Applicant_dim.Applicant].[All Applicant_dim.Applicants]},
 [Time_dim.Time].[All Time_dim.Times].[2020].Children)),
 Crossjoin({[Applicant_dim.Applicant].[All Applicant_dim.Applicants]},
 [Time_dim.Time].[All Time_dim.Times].[2021].Children)),
 Crossjoin({[Applicant_dim.Applicant].[All Applicant_dim.Applicants]},
 [Time_dim.Time].[All Time_dim.Times].[2022].Children)),
 {([Applicant_dim.Applicant].[All Applicant_dim.Applicants].[Female],
 [Time_dim.Time].[All Time_dim.Times]),
 ([Applicant_dim.Applicant].[All Applicant_dim.Applicants].[Male],
 [Time_dim.Time].[All Time_dim.Times])}))
ON ROWS
from [Crdit_card_approval_analysis]
```

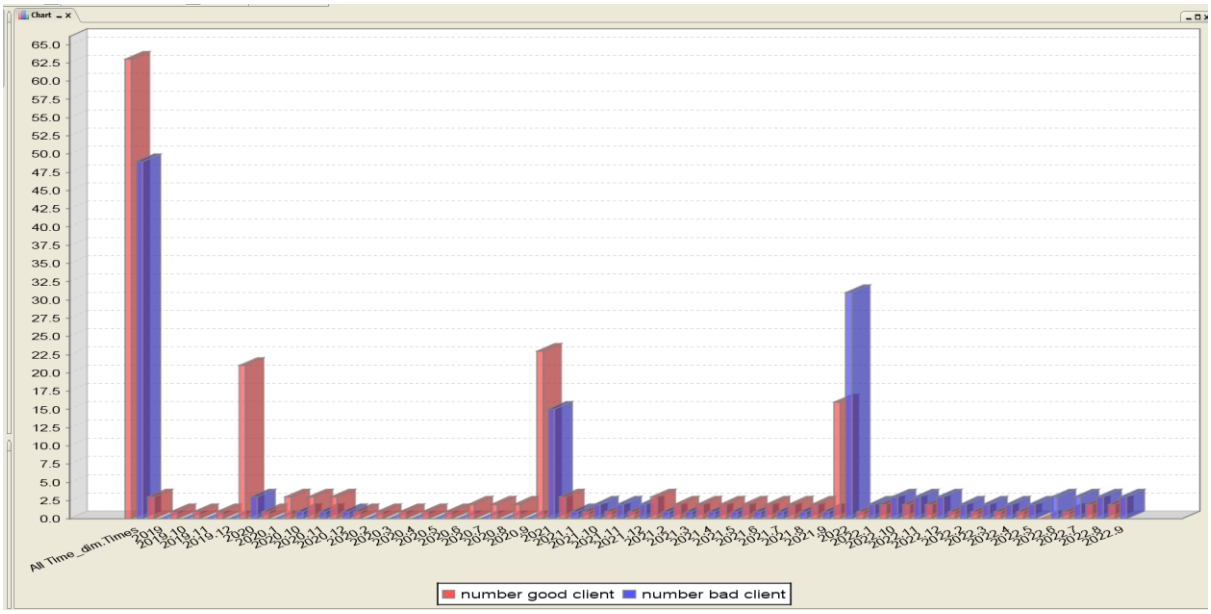
Applicant_dim	Time_dim	number good client	number bad client
All Applicant_dim.Applicants	All Time_dim.Times	63	49
	2019	3	0
	10	1	0
	11	1	0
	12	1	0
	2020	21	3
	1	1	0
	10	3	1
	11	3	1
	12	3	1
	2	1	0
	3	1	0
	4	1	0
	5	1	0
	6	1	0
	7	2	0
	8	2	0
	9	2	0
	2021	23	15
	1	3	1
	10	1	2
	11	1	2
	12	1	2
	2	3	1
	3	2	1
	4	2	1
	5	2	1
	6	2	1
	7	2	1
	8	2	1
	9	2	1
	2022	16	31
	1	1	2
	10	2	3
	11	2	3
	12	2	3
	2	1	2
	3	1	2
	4	1	2
	5	1	2
	6	0	3
	7	1	3
	8	2	3
	9	2	3



22.1.10. Requête pour visualiser le nombre de « Good Client » « Bad Client » par mois :

```
select {[Measures]. [number good client],
 [Measures]. [number bad client]}
ON COLUMNS,
  Hierarchize (Union (Union (Union (Union (Union({[Time_dim.Time].[All Time_dim.Times]},
 [Time_dim.Time].[All Time_dim.Times].Children),
 [Time_dim.Time].[All Time_dim.Times].[2020].Children),
 [Time_dim.Time].[All Time_dim.Times].[2019].Children),
 [Time_dim.Time].[All Time_dim.Times].[2021].Children),
 [Time_dim.Time].[All Time_dim.Times].[2022].Children))
ON ROWS
from [Crdit_card_approval_analysis]
```

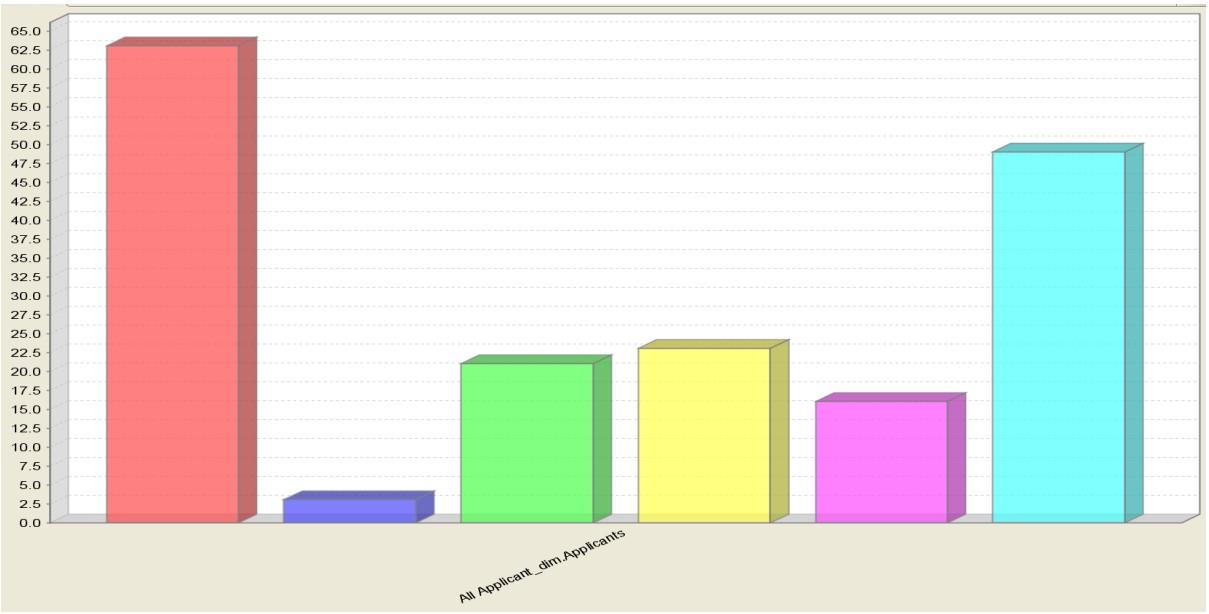
Time_dim	number good client	number bad client
−All Time_dim.Times	63	49
−2019	3	0
+10	1	0
+11	1	0
+12	1	0
−2020	21	3
+1	1	0
+10	3	1
+11	3	1
+12	3	1
+2	1	0
+3	1	0
+4	1	0
+5	1	0
+6	1	0
+7	2	0
+8	2	0
+9	2	0
−2021	23	15
+1	3	1
+10	1	2
+11	1	2
+12	1	2
+2	3	1
+3	2	1
+4	2	1
+5	2	1
+6	2	1
+7	2	1
+8	2	1
+9	2	1
−2022	16	31
+1	1	2



22.1.11. Requête pour visualiser le nombre de « Good Client » « Bad Client » pour chaque année :

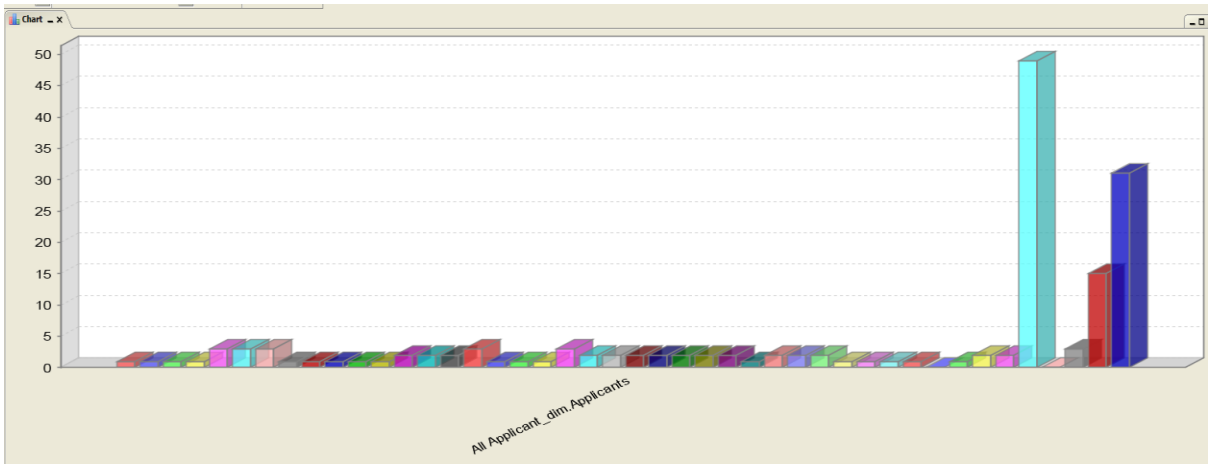
```
select Union (Crossjoin({[Measures]. [number good client]},
 {[Time_dim.Time].[All Time_dim.Times],
 [Time_dim.Time].[All Time_dim.Times].[2019],
 [Time_dim.Time].[All Time_dim.Times].[2020],
 [Time_dim.Time].[All Time_dim.Times].[2021],
 [Time_dim.Time].[All Time_dim.Times].[2022]}),
 {([Measures].[number bad client],
 [Time_dim.Time].[All Time_dim.Times])})
ON COLUMNS,
  Hierarchize({[Applicant_dim.Applicant].[All Applicant_dim.Applicants]})
ON ROWS
from [Crdit_card_approval_analysis]
```

	number good client				
	Time_dim				
Applicant_dim	All Time_dim.Times	2019	2020	2021	2022
All Applicant_dim.Applicants	63	3	21	23	16



22.1.12. Requête pour visualiser le nombre du « Bad Client » pour chaque mois :

```
select Hierarchize (Union (Union (Union (Union (Union (Crossjoin({[Measures]. [number bad
client]},
 {[Time_dim.Time].[All Time_dim.Times],
 [Time_dim.Time].[All Time_dim.Times].[2019],
 [Time_dim.Time].[All Time_dim.Times].[2020],
 [Time_dim.Time].[All Time_dim.Times].[2021],
 [Time_dim.Time].[All Time_dim.Times].[2022]}),
 Crossjoin({[Measures].[number good client]},
 [Time_dim.Time].[All Time_dim.Times].[2019].Children)),
 Crossjoin({[Measures].[number good client]},
 [Time_dim.Time].[All Time_dim.Times].[2020].Children)),
 Crossjoin({[Measures].[number good client]},
 [Time_dim.Time].[All Time_dim.Times].[2021].Children)),
 Crossjoin({[Measures].[number good client]},
 [Time_dim.Time].[All Time_dim.Times].[2022].Children)),
 {([Measures].[number bad client],
 [Time_dim.Time].[All Time_dim.Times])}))
ON COLUMNS,
  Hierarchize({[Applicant_dim.Applicant].[All Applicant_dim.Applicants]})
ON ROWS
from [Crdit_card_approval_analysis]
```



22.1.13.Requête pour visualiser le nombre du « Good Client » « Bad Client » par statut pour chaque année :

```
select {[Measures]. [number bad client],
 [Measures]. [number good client]}
ON COLUMNS,
  Hierarchize (Union (Union (Union (Union (Union (Union (Crossjoin({[Credit
Records_dim.Credit_records].[All Credit Records_dim.Credit_recordss]},
 {[Time_dim.Time].[All Time_dim.Times]}),
 Crossjoin({[Credit Records_dim.Credit_records].[All Credit Records_dim.Credit_recordss]},
 [Time_dim.Time].[All Time_dim.Times].Children)),
 Crossjoin([Credit Records_dim.Credit_records].[All Credit Records_dim.Credit_recordss].Children,
{[Time_dim.Time].[All Time_dim.Times]})),
 Crossjoin({[Credit Records_dim.Credit_records].[All Credit Records_dim.Credit_recordss].[0]},
 [Time_dim.Time].[All Time_dim.Times].Children)),
 Crossjoin({[Credit Records_dim.Credit_records].[All Credit Records_dim.Credit_recordss].[1]},
 [Time_dim.Time].[All Time_dim.Times].Children)),
 Crossjoin({[Credit Records_dim.Credit_records].[All Credit Records_dim.Credit_recordss].[C]},
 [Time_dim.Time].[All Time_dim.Times].Children)),
 Crossjoin({[Credit Records_dim.Credit_records].[All Credit Records_dim.Credit_recordss].[X]},
 [Time_dim.Time].[All Time_dim.Times].Children)))
ON ROWS
from [Crdit_card_approval_analysis]
```

Credit Records_dim	Time_dim	number bad client	number good client
−All Credit Records_dim.Credit_recordss	−All Time_dim.Times	49	63
	+2019	0	3
	+2020	3	21
	+2021	15	23
	+2022	31	16
+0	−All Time_dim.Times	0	27
	+2019	0	1
	+2020	0	8
	+2021	0	7
	+2022	0	11
+1	−All Time_dim.Times		
	+2019		
	+2020		
	+2021		
	+2022		
+C	−All Time_dim.Times	49	0
	+2019		
	+2020	3	0
	+2021	15	0
	+2022	31	0
+X	−All Time_dim.Times	0	36
	+2019	0	2
	+2020	0	13
	+2021	0	16
	+2022	0	5