cognizant

HOSPITAL MANAGEMENT SYSTEM

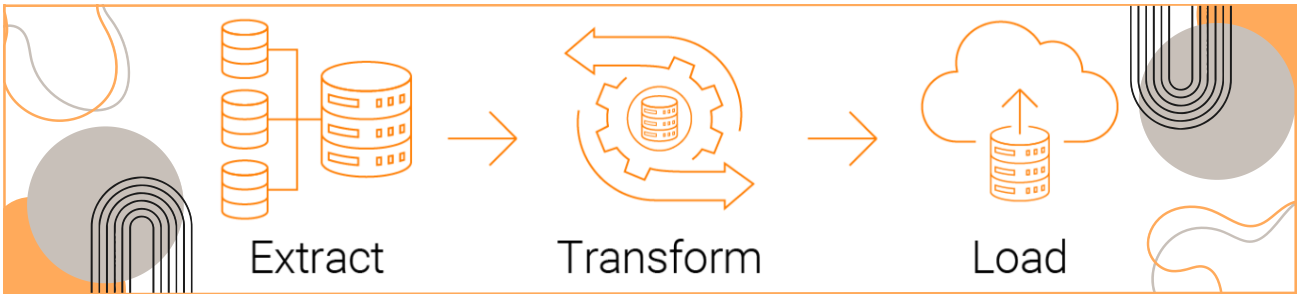
	Prepared By	Reviewed By	Approved By
Name	Swapnendu Chakraborty		
Role	Intern: INTAIA23TA001		
Signature	Swapnendu Chakraborty		
Date	04.09.2023		

Contents

Introduction

This project is based on the Talend ETL tool to achieve the client's requirements. ETL (Extract, Transform, Load) is a crucial process in the world of data integration, enabling us to extract data from various sources, transform it into a consistent and meaningful format, and load it into a destination system for analysis and reporting. In this project, we aim to showcase the significance of Talend as a powerful ETL tool, facilitating seamless data movement, efficient data manipulation, and ultimately empowering data-driven decision-making.

Introduction



Objective

Extracting data from different sources (XML, Flat File, Delimited File, Database, etc.) and Integrating the Data in Data Warehouse.

Transforming the extracted data in the staging area to gain Uniform and Structured data. Transform includes:

- Data value unification
- Data type and Size unification
- De-duplication
- Vertical/Horizontal Slicing

Loading the data into a Datamart to extract the desired data easily. Loading processes are:

- Append
- In place update
- complete replacement
- rolling append

Project Requirement

ABC Hospital is a leading multispecialty hospital headquartered in Chennai. It offers comprehensive medical care in more than 40 specialties. It has vast pool of talented and experienced team of doctors who are further supported by team of highly qualified, experienced and dedicated support staff and cutting edge technology.

The following are the modules in this proposed system

a) Patient Module

b) Physician Module

c) Appointment Module

d) Insurance Module

e) Billing Module

The Enterprise Data Warehouse for Hospital Management is an independent software system developed to store the history on all activities happening in five departments of the hospital – Patient details, physician, Patient booking appointment, Billing and health insurance. The data stored could be subsequently used for reporting.

- The primary focus should be technical aspect of EIM, i.e. data warehousing concepts. Business functions and calculations are secondary.

- Use lexicons for table/column names for consistency.

- Common dimensions like customer should be same for all business areas. One of the teams should take care of developing/refreshing common dimensions.

- DW Data model should be a physical one, i.e. should identify table/column names along with size, primary keys (surrogate keys), business keys and foreign keys.

- Data model is not required for staging tables. However these tables don't need to have surrogate/foreign keys.

- Date dimension should be a common one across all business areas.

- All dimensions and facts should be prefixed with DIM and FCT respectively.

- Suffix surrogate keys, business keys and foreign keys with SK, BK and FK respectively.

- Insert and Update Timestamps apart from the start and end dates for dimensions

- Lexicon for ETL's

System Architecture

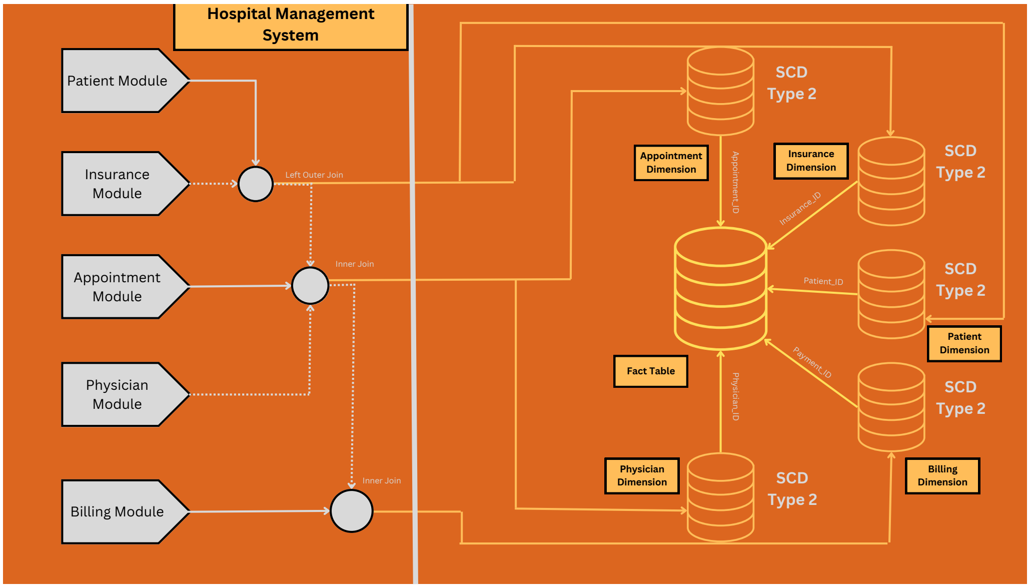
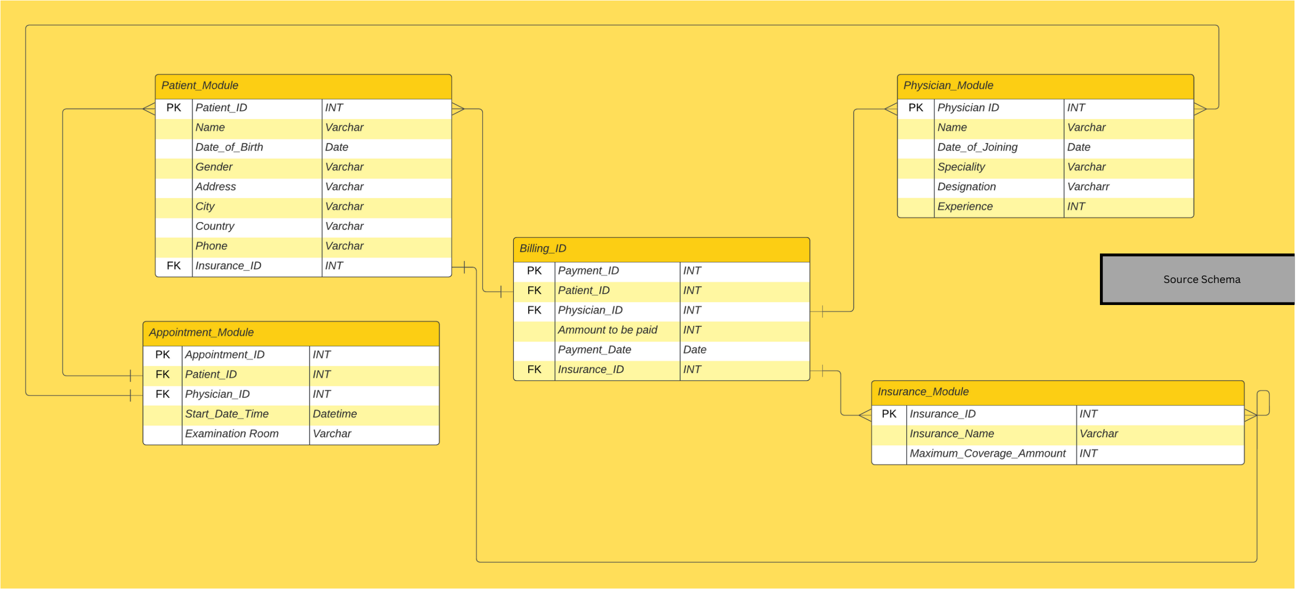


Table Definitions and Mapping

Table Name	Table Type	Table Lookup
Patient_Dim2282133	Dimension Type 2	Patient_Module
Insurance_Dim2282133	Dimension Type 2	Patient_Module + Insurance_Module
Physician_Dim2282133	Dimension Type 2	Physician_Module
Appointment_Dim2282133	Dimension Type 2	Appointment_Module + Patient_Module + Physician_Module
Billing_Dim2282133	Dimension Type 2	Billing_Module + Patient_Module + Physician_Module
Hospital_Management_Fact	Fact Table	Billing_Module + Appointment_Module + Patient_Module + Physican_Module + Insurance_Module

Source Files and Schema for Testing

Source Schema:



Patient_Module		
PK	Patient_ID	INT
	Name	Varchar
	Date_of_Birth	Date
	Gender	Varchar
	Address	Varchar
	City	Varchar
	Country	Varchar
	Phone	Varchar
FK	Insurance_ID	INT

Physician_Module		
PK	Physician ID	INT
	Name	Varchar
	Date_of_Joining	Date
	Speciality	Varchar
	Designation	Varcharr
	Experience	INT

Billing_ID		
PK	Payment_ID	INT
FK	Patient_ID	INT
FK	Physician_ID	INT
	Ammount to be paid	INT
	Payment_Date	Date
FK	Insurance_ID	INT

Appointment_Module		
PK	Appointment_ID	INT
FK	Patient_ID	INT
FK	Physician_ID	INT
	Start_Date_Time	Datetime
	Examination Room	Varchar

Insurance_Module		
PK	Insurance_ID	INT
	Insurance_Name	Varchar
	Maximum_Coverage_Ammount	INT

Testing files:

Patient Module - Patient_Module.csv

Physician Module - Physician_Module.csv

Insurance Module - Insurance_Module.csv

Appointment Module - Appointment_Module.csv

Billing Module - Billing_Module.csv

Functional Requirement

Patient Module

Produce report on patient details who are not from USA, not covered by any insurance

Produce report on patient details above age 50

Dimension Patient is used to get the Patient Details of the ABC Hospital including insurance details

Insurance Module

To store patient info who have insurance amount greater than 200000

To store and fetch count of patients covered by each insurance

Dimension Insurance is used to get the Patient Insurance details

Physician Module

Report of physician details who are surgeon who joined hospital between 2000 to 2010

To display physician details based on

i) Name

ii) Experience

To verify whether the physicians are head of the department

Dimension credit is used to get the Physician information

Appointment Module

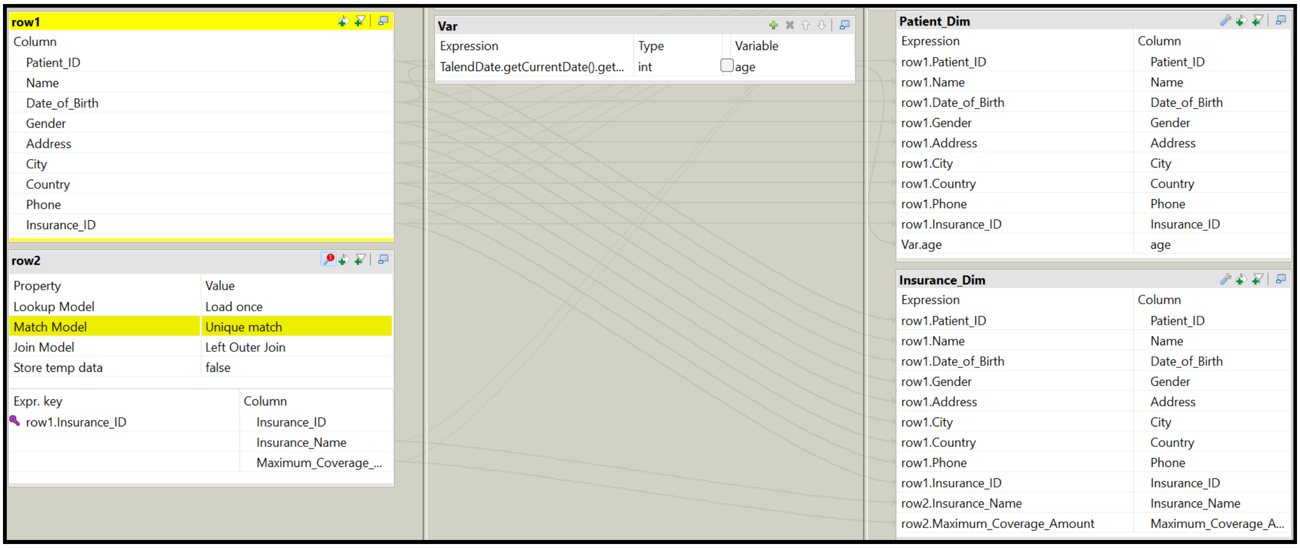
To store patient info who have appointment today
To fetch Patient details based on physician and examination room
To display patient who have appointment booked in future and covered by insurance
Dimension Appointment is used to get the Appointment details

Billing Module

To store the patient details who don’t have insurance coverage and need to pay the complete amount
To fetch the patient and the actual amount to be paid (Amount to be paid – Covered by insurance)
To fetch physician who had maximum amount to be paid
To fetch the list of patient who needs to pay this month
Dimension Billing is to report the Billing details of the patients

Table-wise Operations

Patient and Insurance Dimension Table Creation



row1

Column
Patient_ID
Name
Date_of_Birth
Gender
Address
City
Country
Phone
Insurance_ID

row2

Property	Value
Lookup Model	Load once
Match Model	Unique match
Join Model	Left Outer Join
Store temp data	false

Expr. key	Column
row1.Insurance_ID	Insurance_ID
	Insurance_Name
	Maximum_Coverage_...

Var

Expression	Type	Variable
TalendDate.getCurrentDate().get...	int	□age

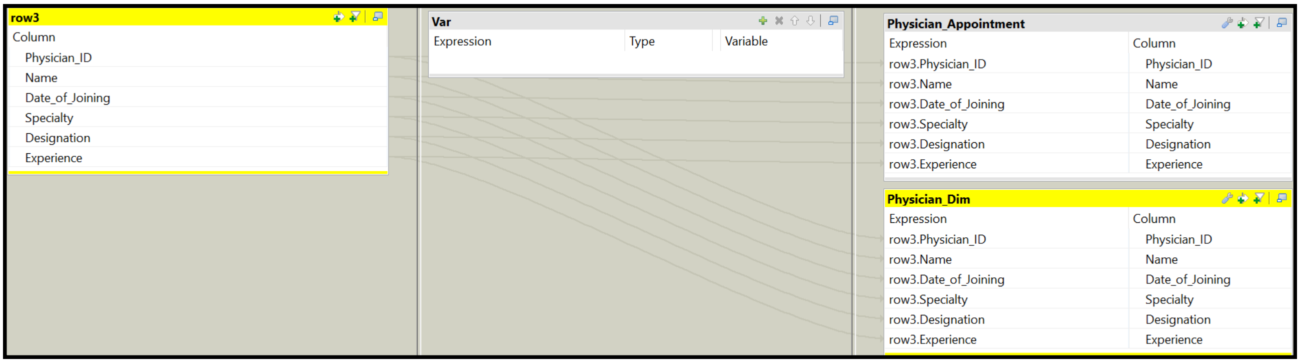
Patient_Dim

Expression	Column
row1.Patient_ID	Patient_ID
row1.Name	Name
row1.Date_of_Birth	Date_of_Birth
row1.Gender	Gender
row1.Address	Address
row1.City	City
row1.Country	Country
row1.Phone	Phone
row1.Insurance_ID	Insurance_ID
Var.age	age

Insurance_Dim

Expression	Column
row1.Patient_ID	Patient_ID
row1.Name	Name
row1.Date_of_Birth	Date_of_Birth
row1.Gender	Gender
row1.Address	Address
row1.City	City
row1.Country	Country
row1.Phone	Phone
row1.Insurance_ID	Insurance_ID
row2.Insurance_Name	Insurance_Name
row2.Maximum_Coverage_Amount	Maximum_Coverage_A...

Physician Dimension Table Creation



row3

Column
Physician_ID
Name
Date_of_Joining
Specialty
Designation
Experience

Var

Expression	Type	Variable

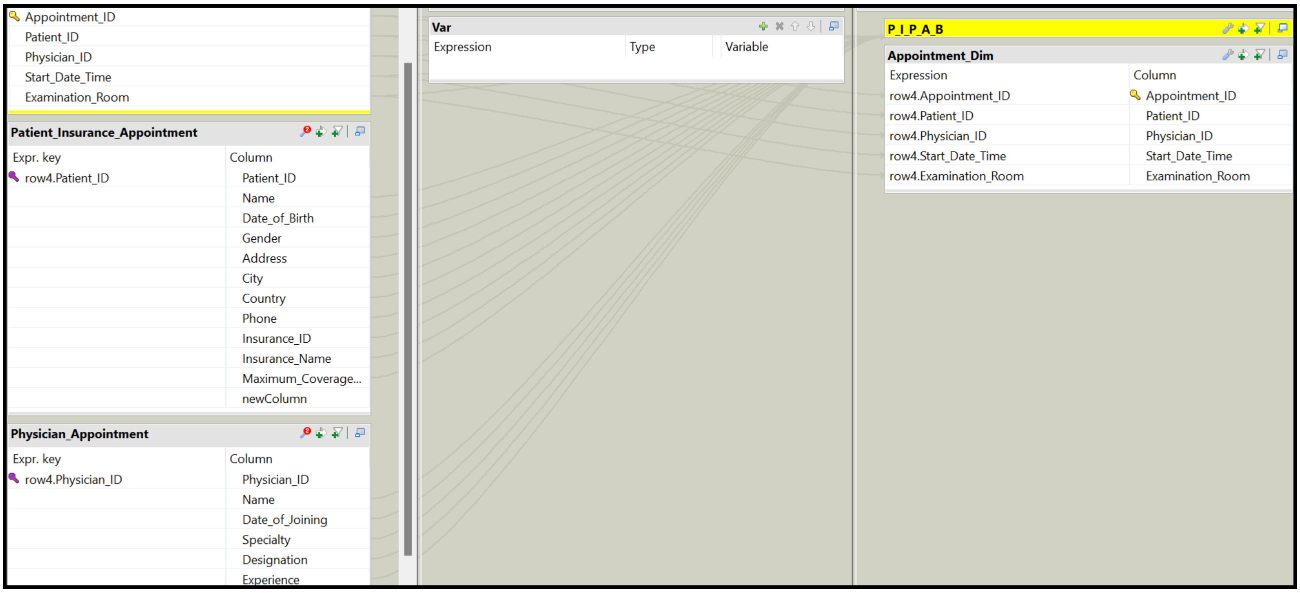
Physician_Appointment

Expression	Column
row3.Physician_ID	Physician_ID
row3.Name	Name
row3.Date_of_Joining	Date_of_Joining
row3.Specialty	Specialty
row3.Designation	Designation
row3.Experience	Experience

Physician_Dim

Expression	Column
row3.Physician_ID	Physician_ID
row3.Name	Name
row3.Date_of_Joining	Date_of_Joining
row3.Specialty	Specialty
row3.Designation	Designation
row3.Experience	Experience

Appointment Dimension Table Creation



Appointment_ID
Patient_ID
Physician_ID
Start_Date_Time
Examination_Room

Patient_Insurance_Appointment

Expr. key	Column
row4.Patient_ID	Patient_ID
	Name
	Date_of_Birth
	Gender
	Address
	City
	Country
	Phone
	Insurance_ID
	Insurance_Name
	Maximum_Coverage...
	newColumn

Physician_Appointment

Expr. key	Column
row4.Physician_ID	Physician_ID
	Name
	Date_of_Joining
	Specialty
	Designation
	Experience

Var

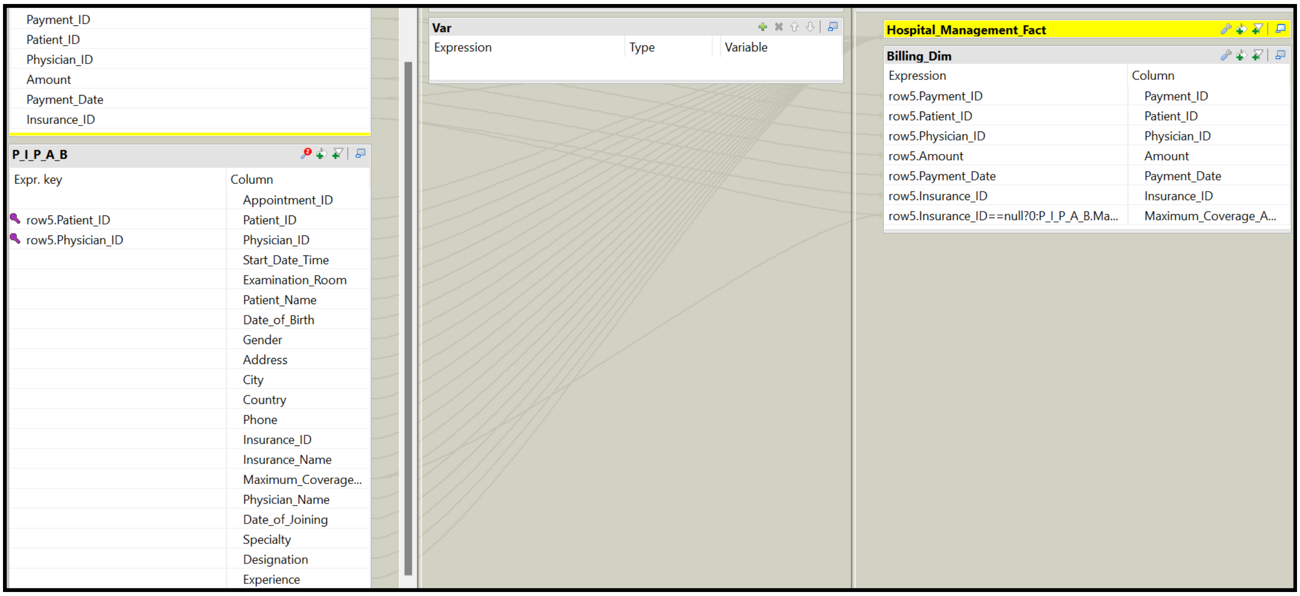
Expression	Type	Variable

P_I_P_A_B

Appointment_Dim

Expression	Column
row4.Appointment_ID	Appointment_ID
row4.Patient_ID	Patient_ID
row4.Physician_ID	Physician_ID
row4.Start_Date_Time	Start_Date_Time
row4.Examination_Room	Examination_Room

Billing Dimension Table Creation



Payment_ID
Patient_ID
Physician_ID
Amount
Payment_Date
Insurance_ID

P_I_P_A_B

Expr. key	Column
	Appointment_ID
row5.Patient_ID	Patient_ID
row5.Physician_ID	Physician_ID
	Start_Date_Time
	Examination_Room
	Patient_Name
	Date_of_Birth
	Gender
	Address
	City
	Country
	Phone
	Insurance_ID
	Insurance_Name
	Maximum_Coverage...
	Physician_Name
	Date_of_Joining
	Specialty
	Designation
	Experience

Var

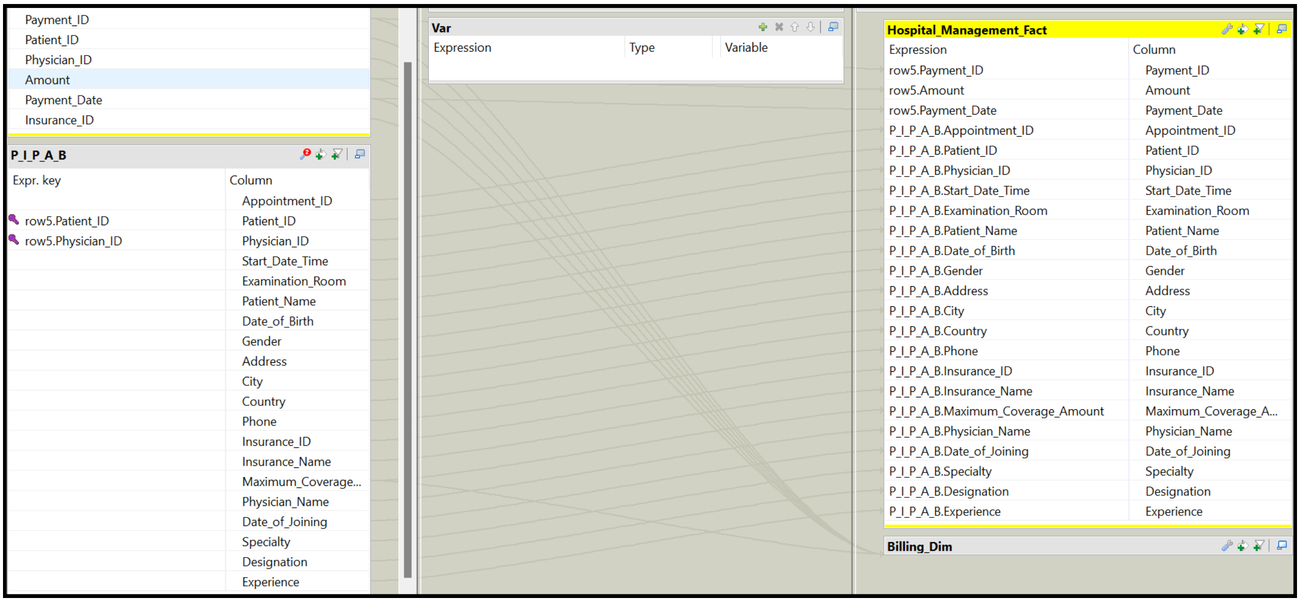
Expression	Type	Variable

Hospital_Management_Fact

Billing_Dim

Expression	Column
row5.Payment_ID	Payment_ID
row5.Patient_ID	Patient_ID
row5.Physician_ID	Physician_ID
row5.Amount	Amount
row5.Payment_Date	Payment_Date
row5.Insurance_ID	Insurance_ID
row5.Insurance_ID==null?0:P_I_P_A_B.Ma...	Maximum_Coverage_A...

Fact Table Creation



Payment_ID
Patient_ID
Physician_ID
Amount
Payment_Date
Insurance_ID

P_I_P_A_B

Expr. key	Column
	Appointment_ID
row5.Patient_ID	Patient_ID
row5.Physician_ID	Physician_ID
	Start_Date_Time
	Examination_Room
	Patient_Name
	Date_of_Birth
	Gender
	Address
	City
	Country
	Phone
	Insurance_ID
	Insurance_Name
	Maximum_Coverage...
	Physician_Name
	Date_of_Joining
	Specialty
	Designation
	Experience

Var

Expression	Type	Variable

Hospital_Management_Fact

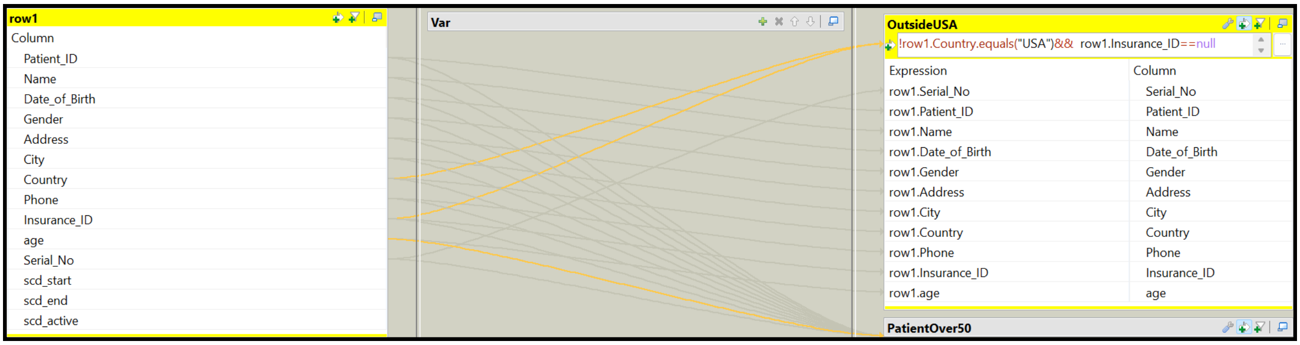
Expression	Column
row5.Payment_ID	Payment_ID
row5.Amount	Amount
row5.Payment_Date	Payment_Date
P_I_P_A_B.Appointment_ID	Appointment_ID
P_I_P_A_B.Patient_ID	Patient_ID
P_I_P_A_B.Physician_ID	Physician_ID
P_I_P_A_B.Start_Date_Time	Start_Date_Time
P_I_P_A_B.Examination_Room	Examination_Room
P_I_P_A_B.Patient_Name	Patient_Name
P_I_P_A_B.Date_of_Birth	Date_of_Birth
P_I_P_A_B.Gender	Gender
P_I_P_A_B.Address	Address
P_I_P_A_B.City	City
P_I_P_A_B.Country	Country
P_I_P_A_B.Phone	Phone
P_I_P_A_B.Insurance_ID	Insurance_ID
P_I_P_A_B.Insurance_Name	Insurance_Name
P_I_P_A_B.Maximum_Coverage_Amount	Maximum_Coverage_A...
P_I_P_A_B.Physician_Name	Physician_Name
P_I_P_A_B.Date_of_Joining	Date_of_Joining
P_I_P_A_B.Specialty	Specialty
P_I_P_A_B.Designation	Designation
P_I_P_A_B.Experience	Experience

Billing_Dim

Report Generation

Produce report on patient details who are not from USA, not covered by any insurance

Code	Purpose
!Patient.Country.equals(”USA”)	to check if patient is not from USA
Patient.Insurance_ID != null	to check if the insurance id is null
Patient.scd_active	to check the active datas
PatientReport.xlsx	



row1

Column
Patient_ID
Name
Date_of_Birth
Gender
Address
City
Country
Phone
Insurance_ID
age
Serial_No
scd_start
scd_end
scd_active

Var

OutsideUSA

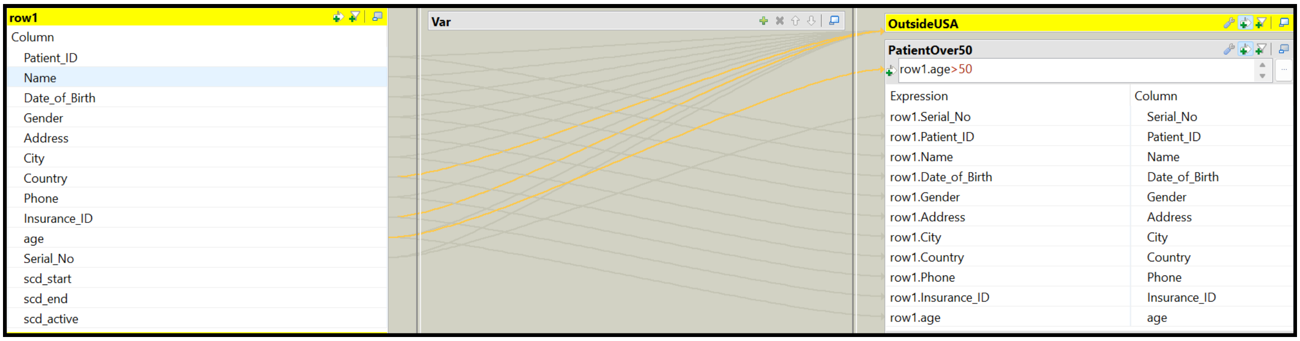
```
!row1.Country.equals("USA")&&  row1.Insurance_ID==null
```

Expression	Column
row1.Serial_No	Serial_No
row1.Patient_ID	Patient_ID
row1.Name	Name
row1.Date_of_Birth	Date_of_Birth
row1.Gender	Gender
row1.Address	Address
row1.City	City
row1.Country	Country
row1.Phone	Phone
row1.Insurance_ID	Insurance_ID
row1.age	age

PatientOver50

Produce report on patient details above age 50

Code	Purpose
Patient.age>50	to check if patient age is more than 50
Patient.scd_active	to check the active datas
PatientReport.xlsx	



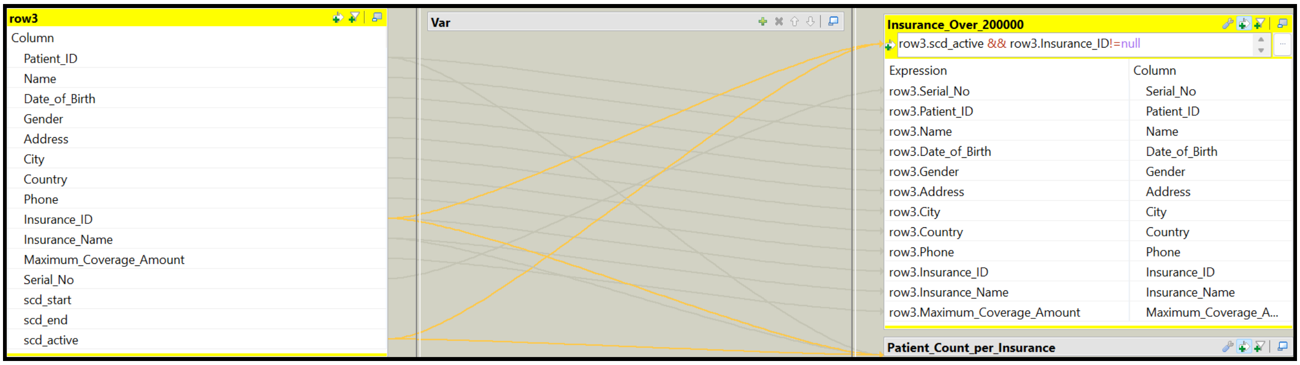
Column
Patient_ID
Name
Date_of_Birth
Gender
Address
City
Country
Phone
Insurance_ID
age
Serial_No
scd_start
scd_end
scd_active

row1.age>50

Expression	Column
row1.Serial_No	Serial_No
row1.Patient_ID	Patient_ID
row1.Name	Name
row1.Date_of_Birth	Date_of_Birth
row1.Gender	Gender
row1.Address	Address
row1.City	City
row1.Country	Country
row1.Phone	Phone
row1.Insurance_ID	Insurance_ID
row1.age	age

To store patient info who have insurance amount greater than 200000

Code	Component	Purpose
Insurance.Insurance_ID!=null	tMap	To check if the patient
Insurance.scd_active	tMap	to check the active datas
Insurance.Maximum_Coverage_Amount>200000	tFilterRow	To check if coverage amount is greater than 200000
InsuranceReport.xlsx		



row3

Column
Patient_ID
Name
Date_of_Birth
Gender
Address
City
Country
Phone
Insurance_ID
Insurance_Name
Maximum_Coverage_Amount
Serial_No
scd_start
scd_end
scd_active

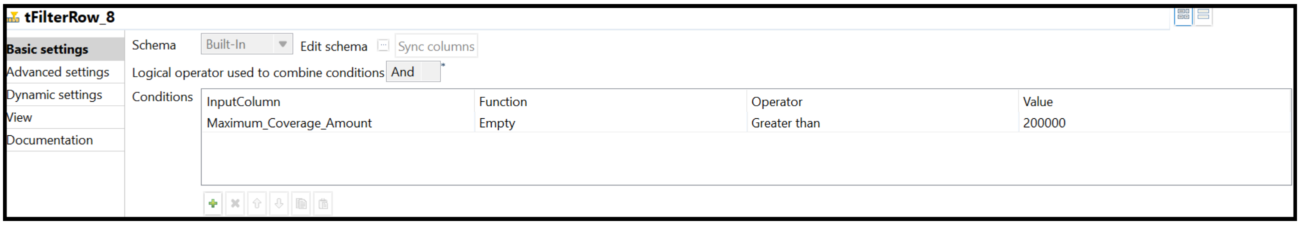
Var

Insurance_Over_200000

row3.scd_active && row3.Insurance_ID!=null

Expression	Column
row3.Serial_No	Serial_No
row3.Patient_ID	Patient_ID
row3.Name	Name
row3.Date_of_Birth	Date_of_Birth
row3.Gender	Gender
row3.Address	Address
row3.City	City
row3.Country	Country
row3.Phone	Phone
row3.Insurance_ID	Insurance_ID
row3.Insurance_Name	Insurance_Name
row3.Maximum_Coverage_Amount	Maximum_Coverage_A...

Patient_Count_per_Insurance



tFilterRow_8

Schema Built-In Edit schema □ Sync columns

Logical operator used to combine conditions And

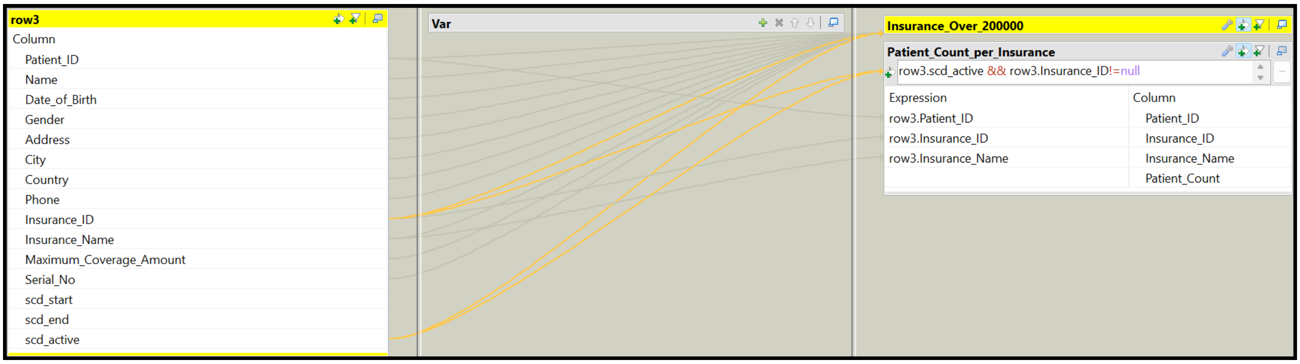
Conditions

InputColumn	Function	Operator	Value
Maximum_Coverage_Amount	Empty	Greater than	200000

To store and fetch count of patients covered by each insurance

Code	Component	Purpose
Insurance.Insurance_ID!=null	tMap	to get patient info who have insurance
Insurance.scd_active	tMap	to get active records of insurance dimension table
group by Insurance.Insurance_ID and Insurance.Insurance_Name	tAggregateRow	to group by Insurance_ID and Insurance_Name
Count patient_id, store in patient_count	tAggregateRow	to count number of patient per insurance

[InsuranceReport.xlsx]



row3

Column
Patient_ID
Name
Date_of_Birth
Gender
Address
City
Country
Phone
Insurance_ID
Insurance_Name
Maximum_Coverage_Amount
Serial_No
scd_start
scd_end
scd_active

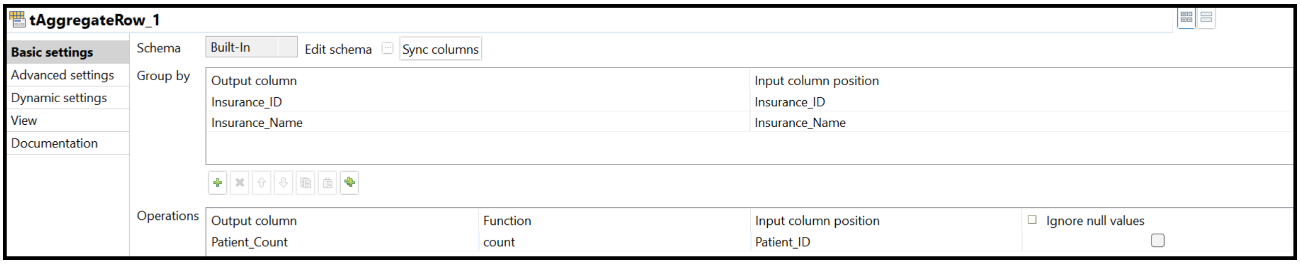
Var

Insurance_Over_200000

Patient_Count_per_Insurance

```
row3.scd_active && row3.Insurance_ID!=null
```

Expression	Column
row3.Patient_ID	Patient_ID
row3.Insurance_ID	Insurance_ID
row3.Insurance_Name	Insurance_Name
	Patient_Count



tAggregateRow_1

Schema: Built-In   Edit schema   Sync columns

Group by

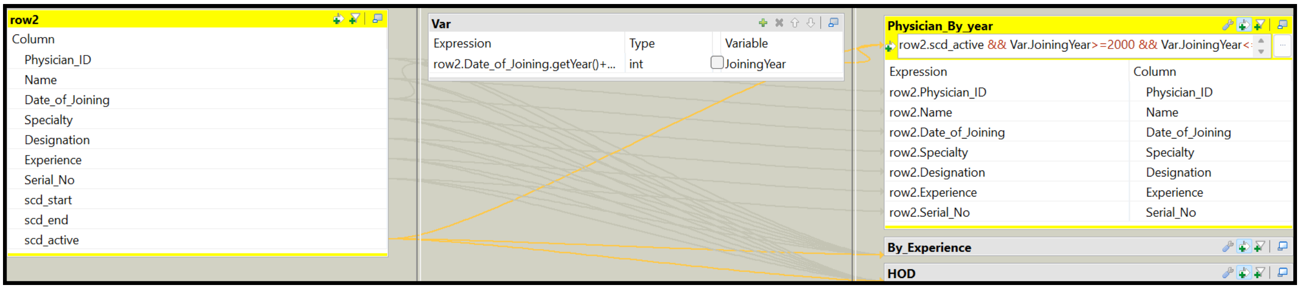
Output column	Input column position
Insurance_ID	Insurance_ID
Insurance_Name	Insurance_Name

Operations

Output column	Function	Input column position	□ Ignore null values
Patient_Count	count	Patient_ID	□

Report of physician details who are surgeon who joined hospital between 2000 to 2010

Code	Purpose
Physician.scd_active	to get the active records from Physician Dimension
Physician.Date_of_joining.getYear()>=2000 && Physician.Date_of_joining.getYear()<=2010	to get physician whose joining year is between 2000 and 2010
PhysicianReport.xlsx	



row2

Column
Physician_ID
Name
Date_of_Joining
Specialty
Designation
Experience
Serial_No
scd_start
scd_end
scd_active

Var

Expression	Type	Variable
row2.Date_of_Joining.getYear()+...	int	□ JoiningYear

Physician_By_year

```
row2.scd_active && Var.JoiningYear>=2000 && Var.JoiningYear<
```

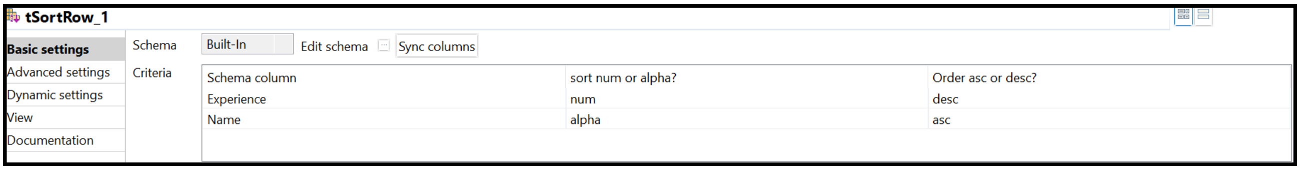
Expression	Column
row2.Physician_ID	Physician_ID
row2.Name	Name
row2.Date_of_Joining	Date_of_Joining
row2.Specialty	Specialty
row2.Designation	Designation
row2.Experience	Experience
row2.Serial_No	Serial_No

To display physician details based on

i) Experience

ii) Name

Code	Component	Purpose
Physician.scd_active	tMap	to get the active records from Physician Dimension
Sort by Experience – num – desc	tSortRow	to sort by Experience
Sort by Name – alpha – asc	tSortRow	to sort by name
PhysicianReport.xlsx		



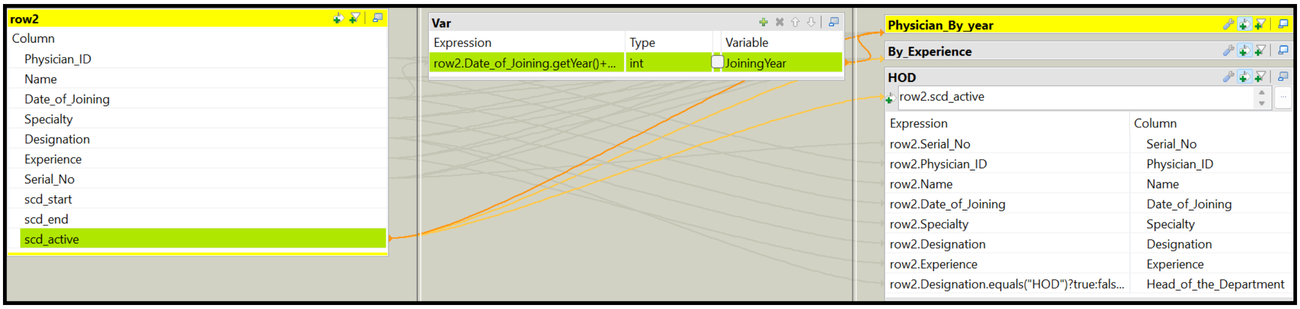
Schema: Built-In

Criteria

Schema column	sort num or alpha?	Order asc or desc?
Experience	num	desc
Name	alpha	asc

To verify whether the physicians are head of the department

Code	Purpose
Physician.scd_active	to get active records from Physician Dimension table
Physician.Designation.equals(”HOD”)	to check whether the designation is HOD or not
PhysicianReport.xlsx	



row2

Column
Physician_ID
Name
Date_of_Joining
Specialty
Designation
Experience
Serial_No
scd_start
scd_end
scd_active

Var

Expression	Type	Variable
row2.Date_of_Joining.getYear()+...	int	JoiningYear

Physician_By_year

By_Experience

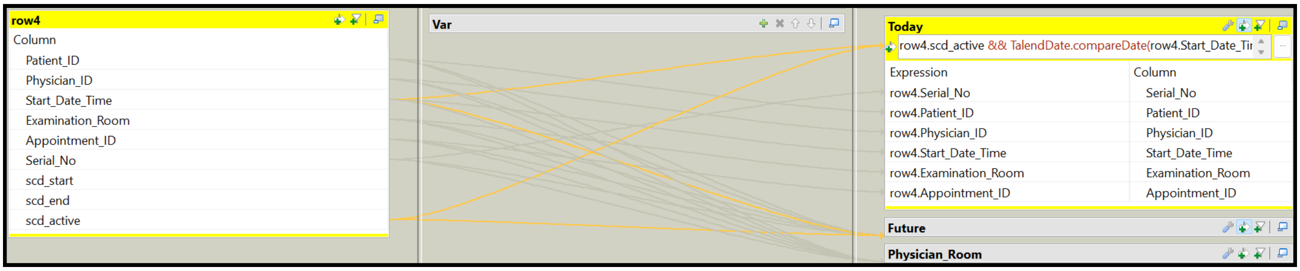
HOD

row2.scd_active

Expression	Column
row2.Serial_No	Serial_No
row2.Physician_ID	Physician_ID
row2.Name	Name
row2.Date_of_Joining	Date_of_Joining
row2.Specialty	Specialty
row2.Designation	Designation
row2.Experience	Experience
row2.Designation.equals("HOD")?true:fals...	Head_of_the_Department

To store patient info who have appointment today

Code	Purpose
Appointment.scd_active	to get the active records from Appointment dimension table
TalendDate.compareDate(Appointment.Start_Date_Time,Talend.getCurrentDate,”dd-MM-yyyy”)==0	to get patients who have appointment today
[AppointmentReport.xlsx]()	



row4

Column
Patient_ID
Physician_ID
Start_Date_Time
Examination_Room
Appointment_ID
Serial_No
scd_start
scd_end
scd_active

Var

Today

```
row4.scd_active && TalendDate.compareDate(row4.Start_Date_Tir
```

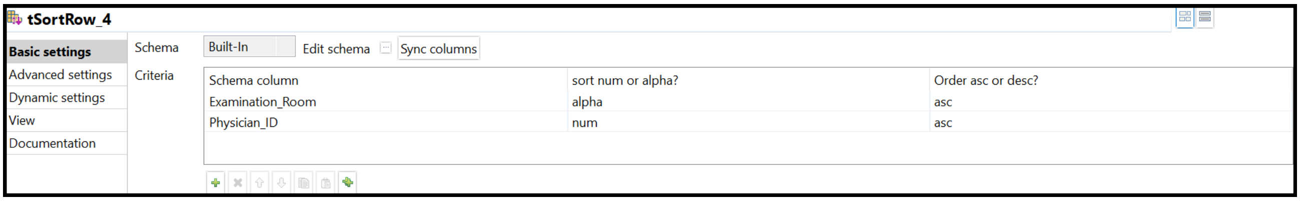
Expression	Column
row4.Serial_No	Serial_No
row4.Patient_ID	Patient_ID
row4.Physician_ID	Physician_ID
row4.Start_Date_Time	Start_Date_Time
row4.Examination_Room	Examination_Room
row4.Appointment_ID	Appointment_ID

Future

Physician_Room

To fetch Patient details based on physician and examination room

Code	Components	Purpose
Appointment.scd_active	tMap	to get active records from Appointment dimension
Sort by Examination_Room – alpha – asc	tSortRow	to sort records by Examination_Room
Sort by Physician_ID – num – asc	tSortRow	to sort records by Physician_ID
[AppointmentReport.xlsx]()		



tSortRow_4

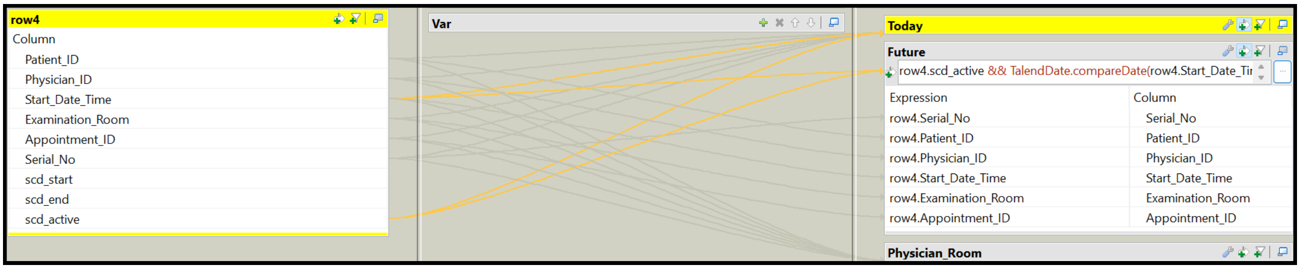
Schema: Built-In Edit schema □ Sync columns

Criteria

Schema column	sort num or alpha?	Order asc or desc?
Examination_Room	alpha	asc
Physician_ID	num	asc

To display patient who have appointment booked in future and covered by insurance

Code	Purpose
Appointment.scd_active	to get the active records from Appointment dimension table
TalendDate.compareDate(Appointment.Start_Date_Time,Talend.getCurrentDate,”dd-MM-yyyy”)==1	to get patients who have appointment in the future
AppointmentReport.xlsx	



row4

Column
Patient_ID
Physician_ID
Start_Date_Time
Examination_Room
Appointment_ID
Serial_No
scd_start
scd_end
scd_active

Var

Today

Future

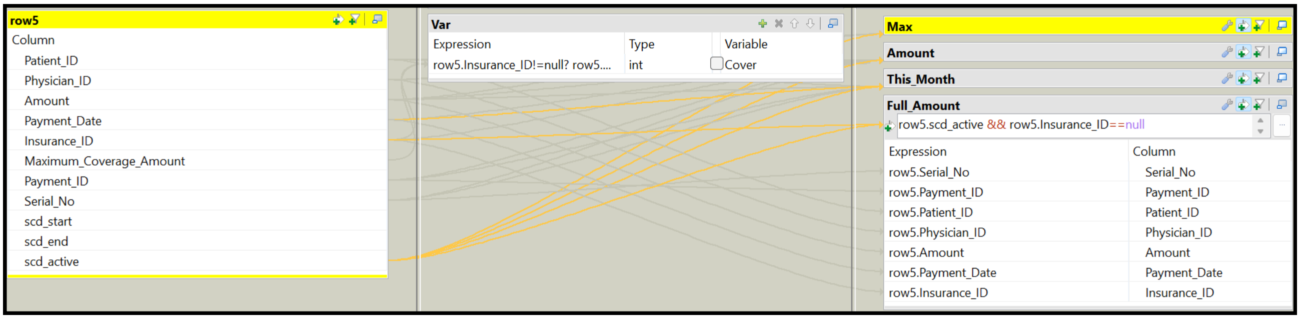
row4.scd_active && TalendDate.compareDate(row4.Start_Date_Tir

Expression	Column
row4.Serial_No	Serial_No
row4.Patient_ID	Patient_ID
row4.Physician_ID	Physician_ID
row4.Start_Date_Time	Start_Date_Time
row4.Examination_Room	Examination_Room
row4.Appointment_ID	Appointment_ID

Physician_Room

To store the patient details who don’t have insurance coverage and need to pay the complete amount

Code	Purpose
Billing.scd_active	to get the active records from Billing dimension table
Billing.Insurance_ID==null	to get records of patients who do not have insurance
BillingReport.xlsx	



row5

Column
Patient_ID
Physician_ID
Amount
Payment_Date
Insurance_ID
Maximum_Coverage_Amount
Payment_ID
Serial_No
scd_start
scd_end
scd_active

Var

Expression	Type	Variable
row5.Insurance_ID!=null? row5....	int	□ Cover

Max

Amount

This_Month

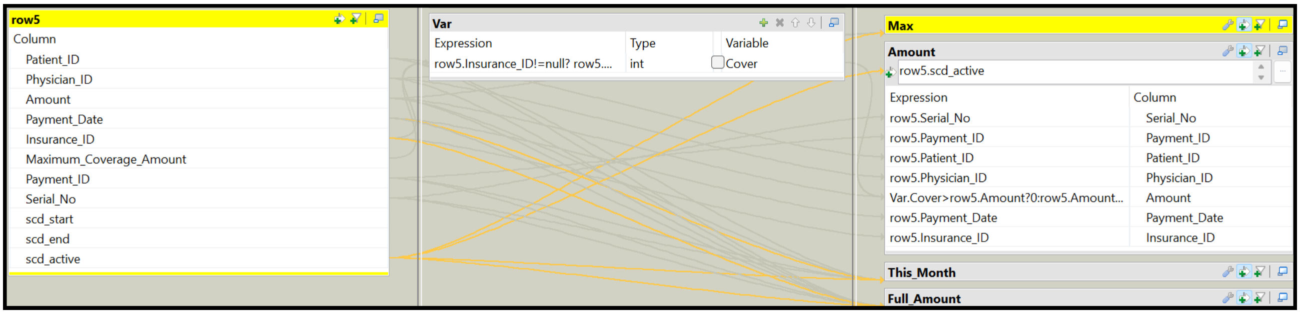
Full_Amount

row5.scd_active && row5.Insurance_ID==null

Expression	Column
row5.Serial_No	Serial_No
row5.Payment_ID	Payment_ID
row5.Patient_ID	Patient_ID
row5.Physician_ID	Physician_ID
row5.Amount	Amount
row5.Payment_Date	Payment_Date
row5.Insurance_ID	Insurance_ID

To fetch the patient and the actual amount to be paid (Amount to be paid – Covered by insurance)

Code	Purpose
`Billing.scd_active`	to get active records from Billing Dimension
`Billing.Insurance_ID==null?0:Billing.Maximum_Coverage_Amount`	if the Insurance is null then the Maximum coverage amount will be 0
`Billing.Maximum_Coverage_Amount>Billing.Amount?0:Billing.Amount-Billing.Maximum_Coverage_Amount`	if the coverage amount is more than amount than the bill amount, the actual payable amount will be 0 else it'll be Amount – Maximum_Coverage_Amount
BillingReport.xlsx	



row5

Column
Patient_ID
Physician_ID
Amount
Payment_Date
Insurance_ID
Maximum_Coverage_Amount
Payment_ID
Serial_No
scd_start
scd_end
scd_active

Var

Expression	Type	Variable
row5.Insurance_ID!=null? row5....	int	□ Cover

Max

Amount

row5.scd_active

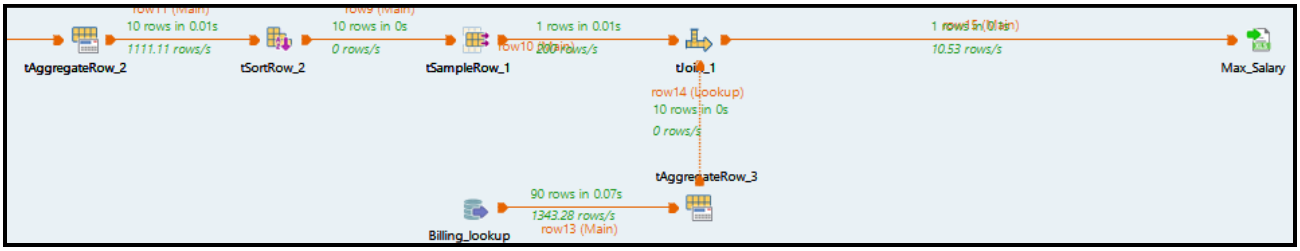
Expression	Column
row5.Serial_No	Serial_No
row5.Payment_ID	Payment_ID
row5.Patient_ID	Patient_ID
row5.Physician_ID	Physician_ID
Var.Cover>row5.Amount?0:row5.Amount...	Amount
row5.Payment_Date	Payment_Date
row5.Insurance_ID	Insurance_ID

This_Month

Full_Amount

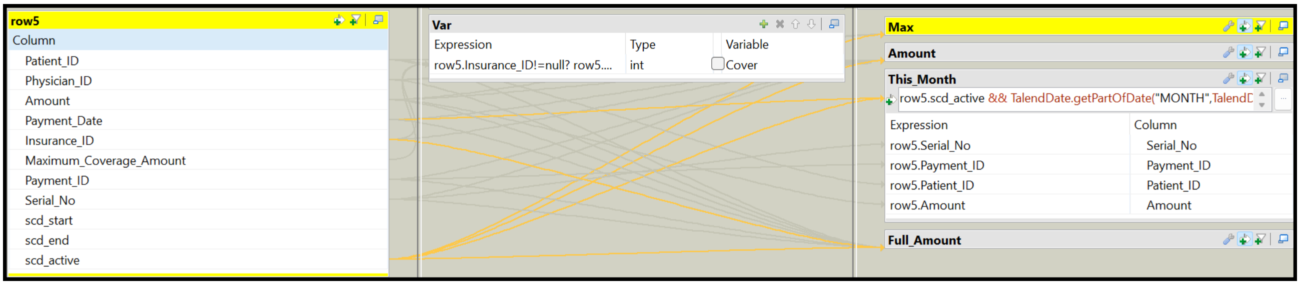
To fetch physician who had maximum amount to be paid

Code	Components	Purpose
group by Billing.Physician_id	tAggregateRow	to group by physician_id
sum of Billing.Amount	tAggregateRow	to get sum of amount for each physician
sort by Billing.Amount – num – desc	tSortRow	to arrange the sorted data based on amount in descending order
row number “1”	tSampleRow	to get the 1st record of the data
Inner Join over Billing.Amount	tJoin	to join the both tables over Aggregate amount
BillingReport.xlsx		



To fetch the list of patient who needs to pay this month

Code	Purpose
Billing.scd_active	to get the active records from billing dimension
TalendDate.getPartOfDate(”YEAR”, Billing.Payement_Date) == TalendDate.getPartOfDate(”YEAR”, Talend.getCurrentDate())	to get the records with payments of current year
TalendDate.getPartOfDate(”MONTH”, Billing.Payement_Date) == TalendDate.getPartOfDate(”MONTH”, Talend.getCurrentDate())	to get the records with payments of current month
BillingReport.xlsx	



row5

Column
Patient_ID
Physician_ID
Amount
Payment_Date
Insurance_ID
Maximum_Coverage_Amount
Payment_ID
Serial_No
scd_start
scd_end
scd_active

Var

Expression	Type	Variable
row5.Insurance_ID!=null? row5....	int	□Cover

Max

Amount

This_Month

row5.scd_active && TalendDate.getPartOfDate("MONTH",TalendD

Expression	Column
row5.Serial_No	Serial_No
row5.Payment_ID	Payment_ID
row5.Patient_ID	Patient_ID
row5.Amount	Amount

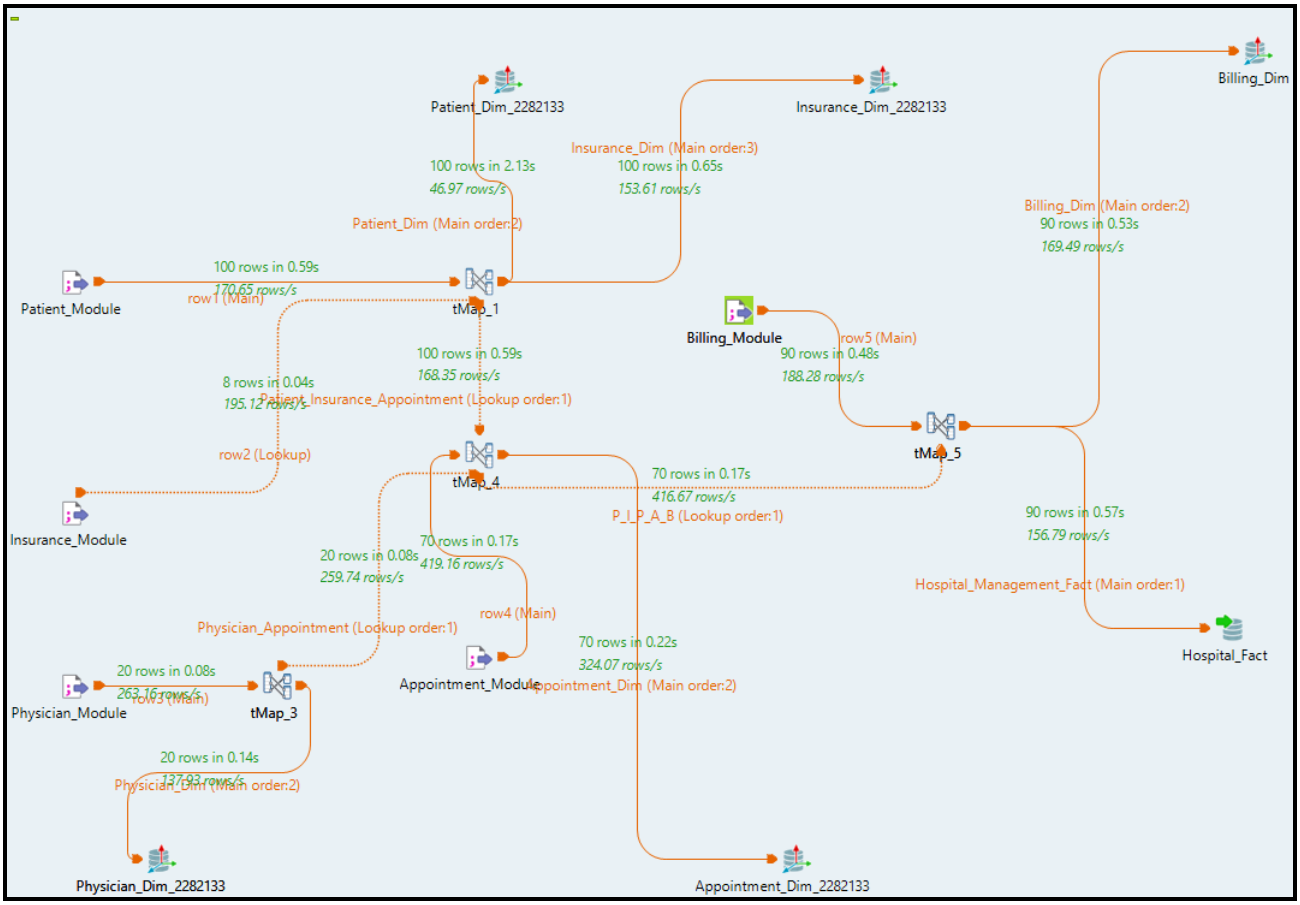
Full_Amount

Project Overview

Job 1

Purpose

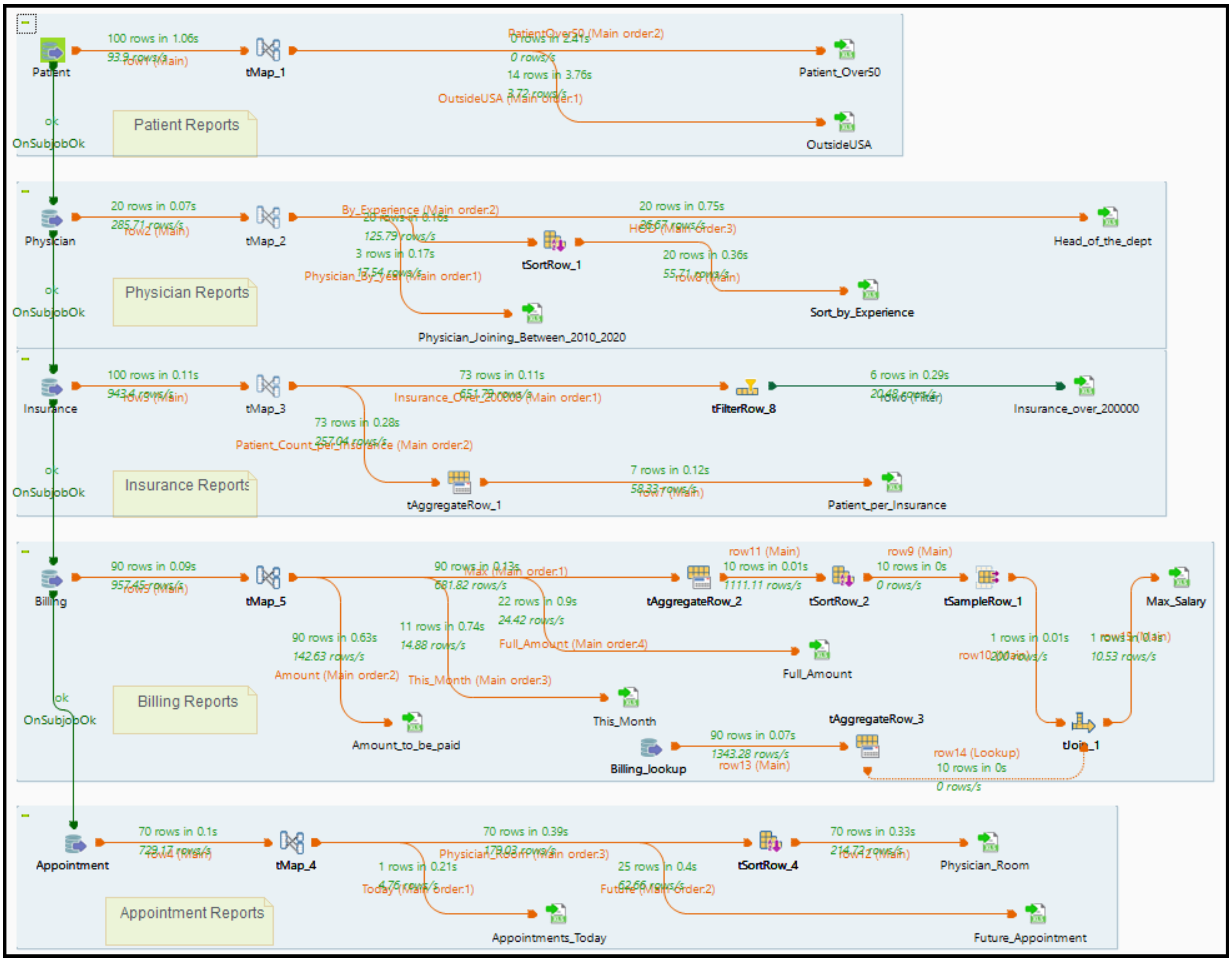
- Transform data according to standard
- Create Dimension Tables and Implement Type 2 Dimension
- Create Fact Table



Job 2

Purpose

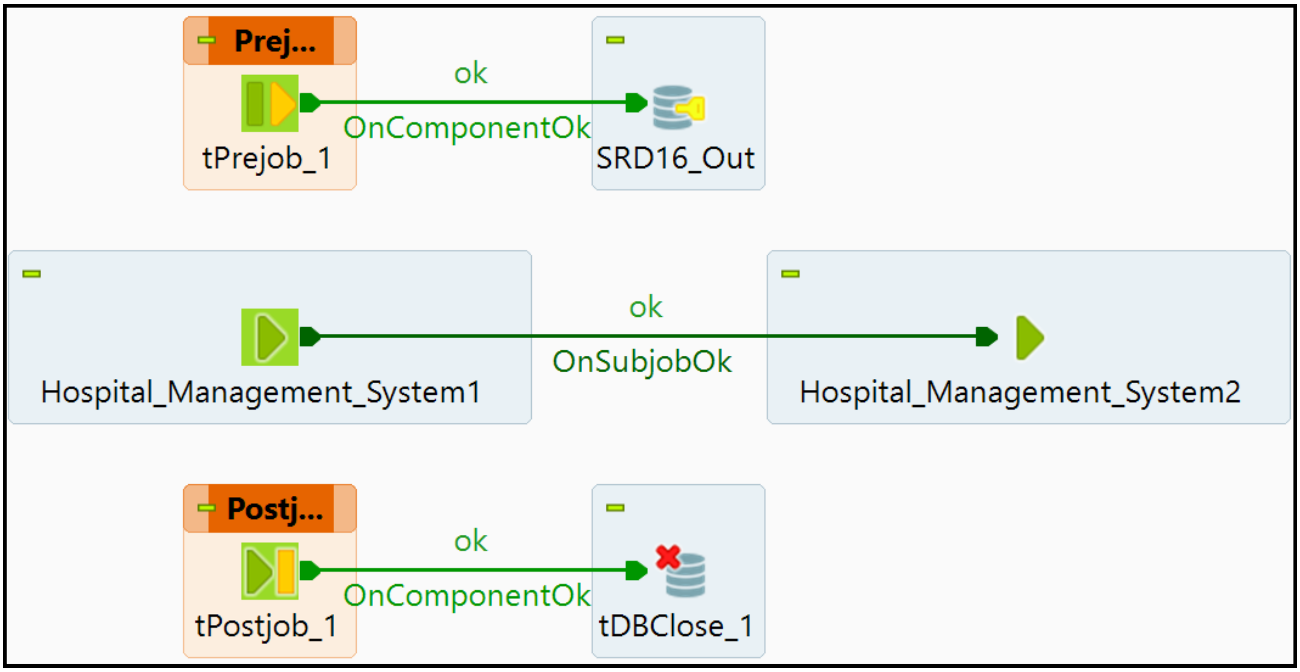
- Take Dimension Tables as Input
- Generate Report based on requirement
- Store the reports in form of CSV



Job 3

Purpose

- Establish the Connections with Database
- Run Job 1 to create Dimension and Fact Tables
- Run Job 2 to extract data from Database and Create report
- Close the connection



Annexure

Licensing Information:

The ETL project using Talend is subject to specific licensing terms and conditions. These terms govern the usage, distribution, and modification of the project deliverables. The licensing Information is outlined as follows:

1. The ETL project’s source code and documentation are the intellectual property of Cognizant Technology Solutions Ltd. and are protected under applicable copyright laws.
2. The project’s deliverables, including Talend job designs, ETL workflows, and documentation, are solely for internal use by Cognizant Technology Solutions Ltd. and may not be distributed or shared with external parties without prior written consent.
3. Any modifications or enhancements to the ETL project must comply with Cognizant Technology Solutions Ltd. ’s change management process and be approved by the designated project authorities.
4. Cognizant Technology Solutions Ltd. shall not hold Talend responsible for any issues or challenges arising from the usage of the Talend software, as per the terms and conditions set forth by Talend’s licensing agreement.

Data Privacy and Security Policies:

As part of the ETL project’s implementation, data privacy and security policies are paramount to safeguard sensitive information. The following policies are to be adhered to:

1. Data Encryption: All sensitive data transmitted between systems and during ETL processes must be encrypted to prevent unauthorized access.
2. Access Controls: Access to the Data Warehouse and related systems shall be granted based on the principle of least privilege, ensuring that only authorized personnel can access sensitive data.
3. Data Anonymization: Personally identifiable information (PII) and other sensitive data must be anonymized or pseudonymized when not required for specific reporting or analysis.
4. Data Retention: Data retention policies shall be defined to manage the storage and archival of data in compliance with legal and regulatory requirements.
5. Audit Logging: Comprehensive audit logs shall be maintained to track data access, changes, and user activities for accountability and troubleshooting purposes.
6. Disaster Recovery: Robust data backup and disaster recovery mechanisms shall be established to ensure data availability and continuity in case of unforeseen events.