Vehicle Insurance Data Warehouse

TANMAY ATKEKAR

Company Description

- The insurance company XYZ provides various insurances to different Automobiles across USA.

- The company has different covers for each vehicle depending on Driver's driving history, points on license, vehicle details and type etc.

Data Warehouse mission statement

The objective behind building a data warehouse is connecting data from source systems, applying transformations and storing into one repository for better data flow across the organization.

This finally helps in better analytics considering vast data coming from different sources, which results in data-based decision making across the organization.

The data warehouse repository will also be used across different departments to serve their data and reporting needs

Business case

- The need of central data warehouse repository is imminent because data is spread across various sources, causing inconsistent data flow throughout the organization.

- This results in poor analytical operations being performed across organization, ultimately resulting in slow growth of organization.

Project Scope

The scope consists of four major activities:

1. Creating data warehouse
2. Breaking the DW down into data marts
3. Creating BI plan
4. Maintenance of DW

Project Stakeholders

- Project Sponsor: Director of XYZ company who is funding the project

- Project Team: Team responsible for implementing the data warehouse

- End Users: Employees using the DW for data and analytical purposes

- Review board: The team responsible to check whether the DW meets the requirements

Interview Questions

- What are the data sources and type of data?

- What regulations do we need to adhere to?

- What kind of reporting and analytical services it should provide?

- Do we need Data marts or just one consolidated DW?

- What are the existing data security measures?

- Do you need the team to support the data warehouse or it can be done in house?

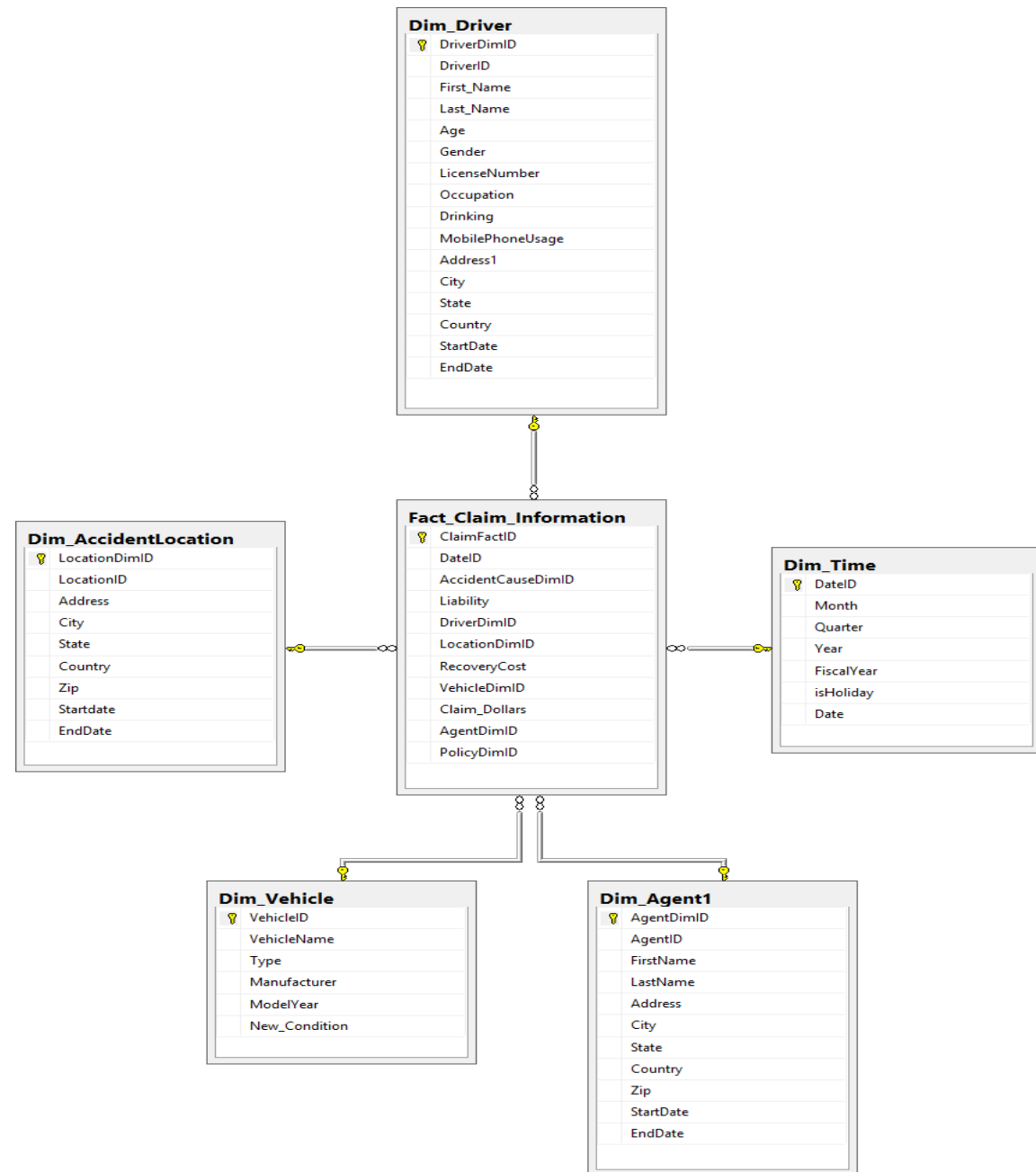- What are current problems while extracting data from different sources for analytical purposes?
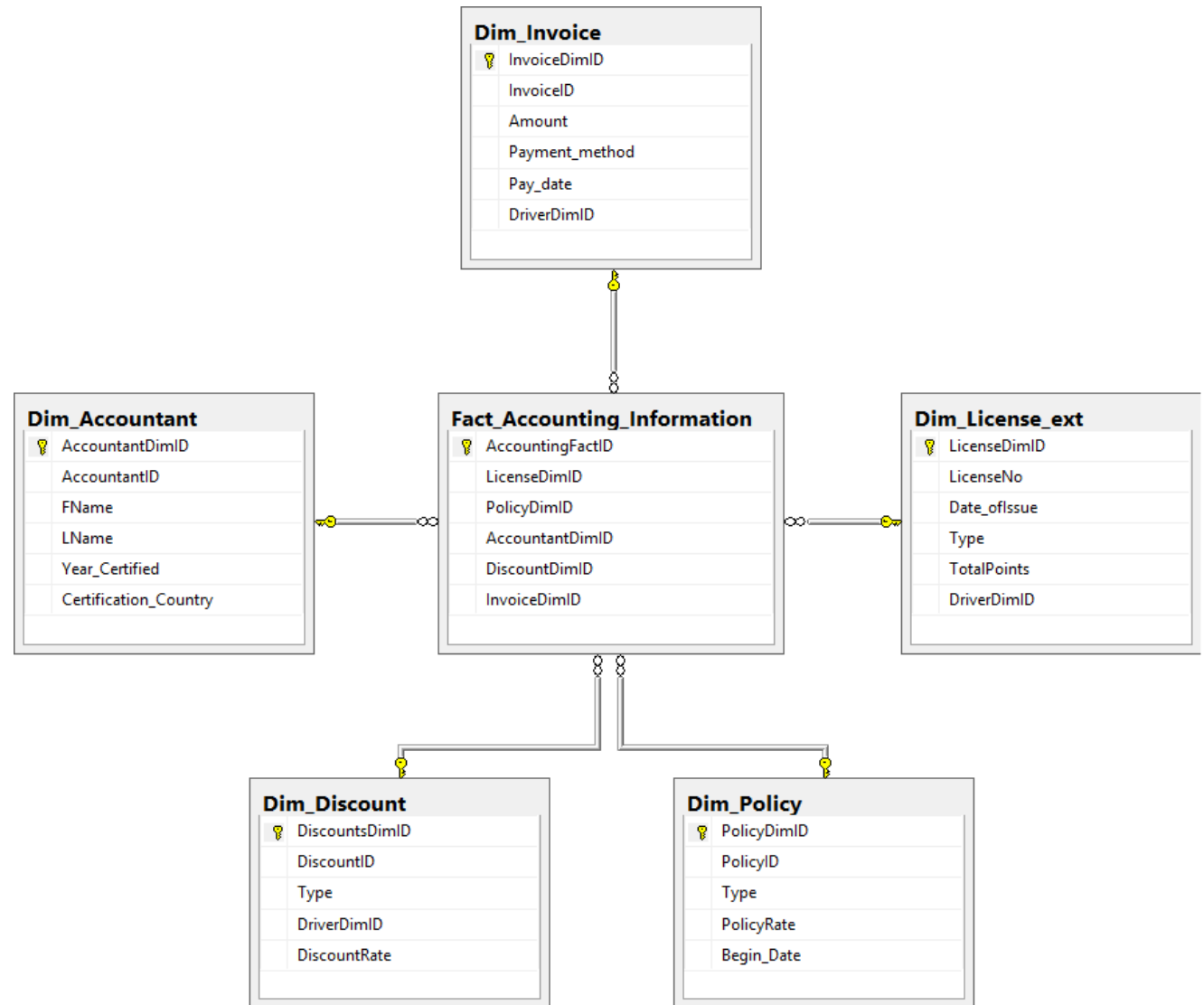
Data Warehouse Schema Design
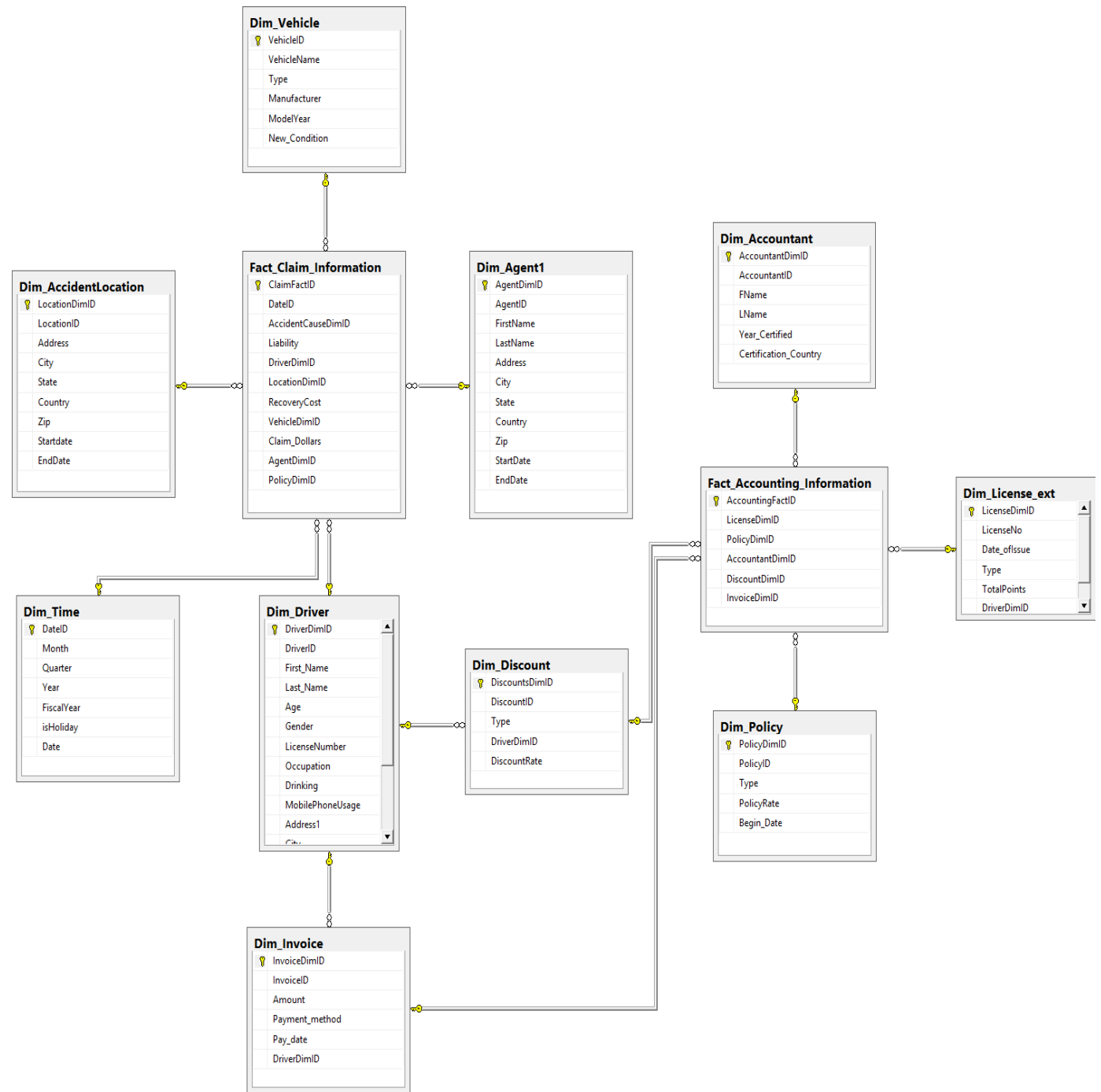
Data Marts
Schema Design

2. Accounting DM

SQL-Build (create)Dim_Tables

```sql
/****** Object:  Table [dbo].[Dim_Driver]    Script Date: 11/26/2018 11:35:38 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Dim_Driver](
    [DriverDimID] [int] NOT NULL,
    [DriverID] [nvarchar](50) NULL,
    [First_Name] [nvarchar](50) NULL,
    [Last_Name] [nvarchar](50) NULL,
    [Age] [int] NULL,
    [Gender] [nvarchar](50) NULL,
    [LicenseNumber] [nvarchar](50) NULL,
    [Occupation] [nvarchar](50) NULL,
    [Drinking] [bit] NULL,
    [MobilePhoneUsage] [text] NULL,
    [Address1] [nvarchar](50) NULL,
    [City] [nvarchar](50) NULL,
    [State] [nvarchar](50) NULL,
    [Country] [nvarchar](50) NULL,
    [StartDate] [datetime] NULL,
    [EndDate] [datetime] NULL,
 CONSTRAINT [PK_Dim_Driver] PRIMARY KEY CLUSTERED
(
    [DriverDimID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
```

```sql
/****** Object:  Table [dbo].[Dim_Accountant]    Script Date: 11/26/2018 11:35:14 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Dim_Accountant](
    [AccountantDimID] [int] NOT NULL,
    [AccountantID] [nvarchar](50) NULL,
    [FName] [nvarchar](50) NULL,
    [LName] [nvarchar](50) NULL,
    [Year_Certified] [datetime] NULL,
    [Certification_Country] [nvarchar](50) NULL,
 CONSTRAINT [PK_Dim_Accountant] PRIMARY KEY CLUSTERED
(
    [AccountantDimID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
```

SQL-Build (create) Fact_Accounting

```sql
/****** Object:  Table [dbo].[Fact_Accounting_Information]    Script Date: 11/26/2018 11:36:13 PM ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Fact_Accounting_Information](
    [AccountingFactID] [int] NOT NULL,
    [LicenseDimID] [int] NOT NULL,
    [PolicyDimID] [int] NOT NULL,
    [AccountantDimID] [int] NOT NULL,
    [DiscountDimID] [int] NOT NULL,
    [InvoiceDimID] [int] NOT NULL,
 CONSTRAINT [PK_Fact_Accounting_Information] PRIMARY KEY CLUSTERED
(
    [AccountingFactID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Fact_Accounting_Information]  WITH CHECK ADD  CONSTRAINT [FK_Fact_Accounting_Information_Dim_Accountant] FOREIGN KEY([Account
REFERENCES [dbo].[Dim_Accountant] ([AccountantDimID])
GO

ALTER TABLE [dbo].[Fact_Accounting_Information] CHECK CONSTRAINT [FK_Fact_Accounting_Information_Dim_Accountant]
GO

ALTER TABLE [dbo].[Fact_Accounting_Information]  WITH CHECK ADD  CONSTRAINT [FK_Fact_Accounting_Information_Dim_Discount] FOREIGN KEY([DiscountD
REFERENCES [dbo].[Dim_Discount] ([DiscountsDimID])
GO
```

```sql
ALTER TABLE [dbo].[Fact_Accounting_Information] CHECK CONSTRAINT [FK_Fact_Accounting_Information_Dim_Discount]
GO

ALTER TABLE [dbo].[Fact_Accounting_Information]  WITH CHECK ADD  CONSTRAINT [FK_Fact_Accounting_Information_Dim_Invoice] FOREIGN KEY([InvoiceDim
REFERENCES [dbo].[Dim_Invoice] ([InvoiceDimID])
GO

ALTER TABLE [dbo].[Fact_Accounting_Information] CHECK CONSTRAINT [FK_Fact_Accounting_Information_Dim_Invoice]
GO

ALTER TABLE [dbo].[Fact_Accounting_Information]  WITH CHECK ADD  CONSTRAINT [FK_Fact_Accounting_Information_Dim_License_ext] FOREIGN KEY([Licens
REFERENCES [dbo].[Dim_License_ext] ([LicenseDimID])
GO

ALTER TABLE [dbo].[Fact_Accounting_Information] CHECK CONSTRAINT [FK_Fact_Accounting_Information_Dim_License_ext]
GO

ALTER TABLE [dbo].[Fact_Accounting_Information]  WITH CHECK ADD  CONSTRAINT [FK_Fact_Accounting_Information_Dim_Policy] FOREIGN KEY([PolicyDimID
REFERENCES [dbo].[Dim_Policy] ([PolicyDimID])
GO

ALTER TABLE [dbo].[Fact_Accounting_Information] CHECK CONSTRAINT [FK_Fact_Accounting_Information_Dim_Policy]
GO
```

SQL-Build (insert)

ETL Script using Python

```python
from auxiliary import *

directory = r"C:\Users\re70296\Downloads\IDR Project\sample\\"
filename = 'CF1 Q1 2018'
file_location = directory + filename + ".xlsx"

CF_Name = filename.split(' ',1)[0]
Quarter_Name = filename.split(' ',1)[1]

# Open the workbook and define the worksheet
book = xlrd.open_workbook(file_location)

SHEETS_index = {'Accountant': 'Dim_Accountant','Agent': 'Dim_Agent',
'Discount': 'Dim_Discount','Driver': 'Dim_Driver',
'Invoice': 'Dim_Invoice','License':'Dim_Licence_Ext',
'Policy': 'Dim_Policy','Time': 'Dim_Time',
'Vehicle': 'Dim_Vehicle'}

database, cursor = create_connection()

if(result==0):
    ###Update all tables
    for sheetname in SHEETS_index:
        insert_values(book = book, sheetname = sheetname, SHEETS_index = SHEETS_index,
            database = database, cursor = cursor, file_location = file_location)

close_database(database, cursor)
input('Press ENTER to exit')
```

```python
import xlrd
from openpyxl import load_workbook
import pymysql as MySQLdb

import numpy as np
import pandas as pd

def create_connection():
    # Establish a MySQL connection
    database = MySQLdb.connect(host="localhost", user = "root", passwd = "Tanmay9830!", db = "idr_db")

    # Get the cursor, which is used to traverse the database, line by line
    cursor = database.cursor()
    cursor.execute("USE idr_db") # select the database
    cursor.execute("SET SESSION sql_mode = ''")
    database.commit()
    print('Database Connection Opened')
    return database, cursor

def return_pk_index(tablename, database, cursor, pkid='PKID'):
    '''
    To get the maximum primary key for a specific table
    '''
    query_pk = """SELECT MAX(PKID) FROM """ + tablename
    cursor.execute(query_pk)
    query_pk_result = cursor.fetchone()[0]

    if(query_pk_result):
        return query_pk_result
    else:
        return 0

def insert_value_in_cell(cell, query, cursor, sheet):
    '''
    Inserting values in cells in Excel
    '''
    cursor.execute(query)
    val = cursor.fetchone()[0]
    sheet[cell] = val
```

Lookup Transformation in SSIS

ETL Implementation SCD-2

- Before change in Policy Rate

PolicyDimID	PolicyID	Type	PolicyRate	Begin_Date	End_Date
1	123	Lifelong	20	2019-01-01 00:00:00.000	NULL
2	234	1 year	25	2018-09-18 00:00:00.000	NULL
3	345	2 years	10	2014-10-03 00:00:00.000	NULL
4	567	6 months	5	2017-09-09 00:00:00.000	NULL
5	489	Lifelong	18	2016-01-13 00:00:00.000	NULL
NULL	NULL	NULL	NULL	NULL	NULL

- After change in Policy Rate

	PolicyDimID	PolicyID	Type	PolicyRate	Begin_Date	End_Date
1	1	123	Lifelong	25	2019-01-01 00:00:00.000	2018-11-27 20:02:26.000
2	2	234	1 year	35	2018-09-18 00:00:00.000	2018-11-27 20:02:26.000
3	3	345	2 years	10	2014-10-03 00:00:00.000	NULL
4	4	567	6 months	5	2017-09-09 00:00:00.000	NULL
5	5	489	Lifelong	40	2016-01-13 00:00:00.000	2018-11-27 20:02:26.000

SCD-2 Dataflow in SSIS
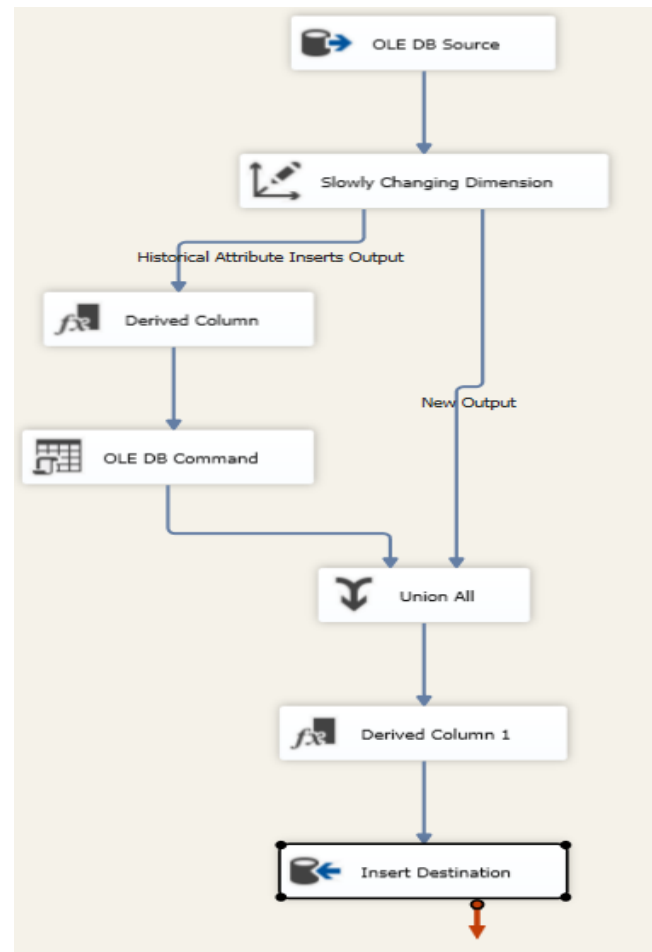
ETL Implementation – SCD 3

Before change in Address

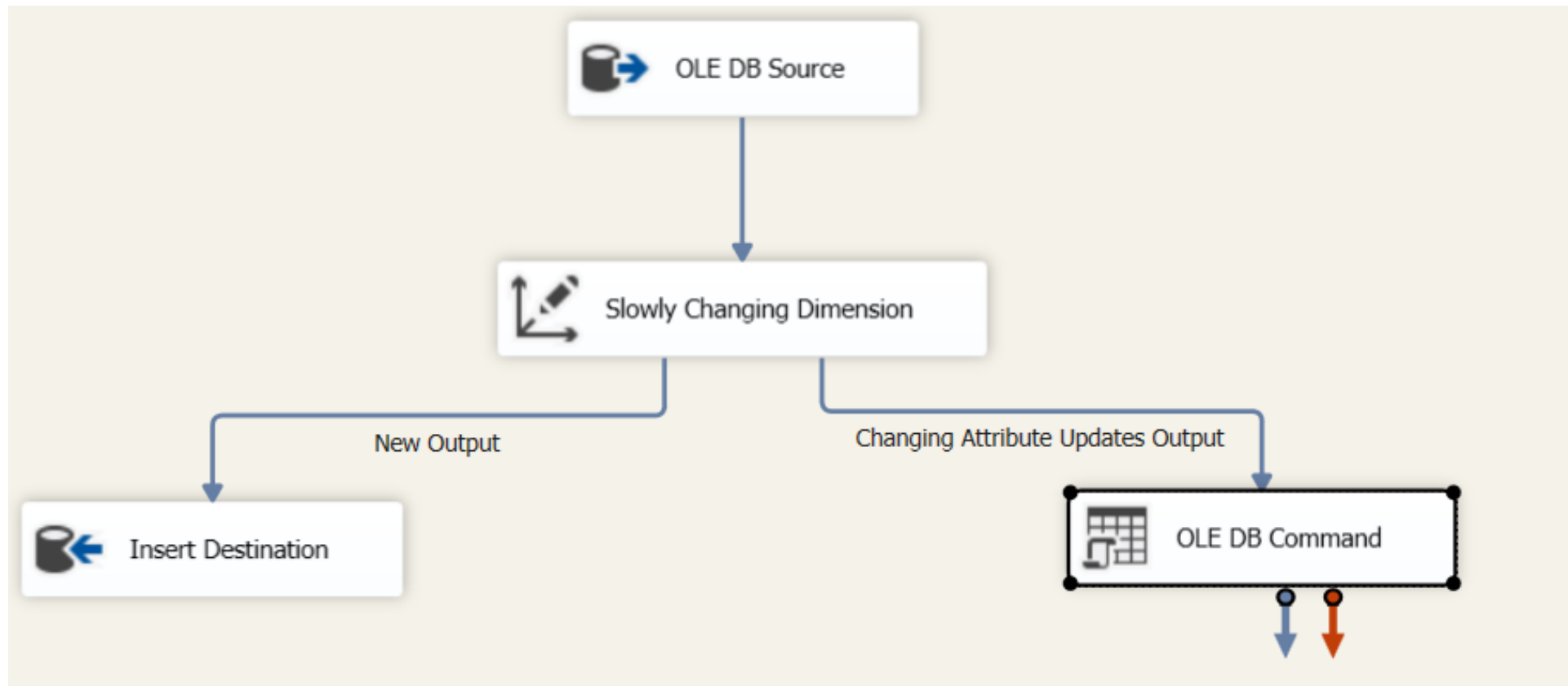| AgentID	FirstName	LastName	Address	City	State	Country	Zip	StartDate	EndDate
102	Adam	Paul	223 Oak Ave	Manhattan	New York	United States	10012	2018-09-11 00:0...	NULL
101	Eva	Dream	223 Oak Ave	Manhattan	New York	United States	10012	2018-08-13 00:0...	NULL
103	Jennifer	Dias	223 Oak Ave	Manhattan	New York	United States	10012	2018-09-13 00:0...	NULL
104	Noah	Williams	122 Adam St	Dallas	Texas	United States	10010	2018-09-13 00:0...	NULL
105	Rachel	Green	123 Brooklyn St	Brooklyn	New York	United States	10020	2018-09-13 00:0...	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

After change in Address

AgentDimID	AgentID	FirstName	LastName	Address	City	State	Country	Zip	StartDate
1	102	Adam	Paul	504 Greenwood...	Syracuse	NY	United States	10012	2018-09-11 00:
2	101	Eva	Dream	437 Columbus ...	Syracuse	NY	United States	10012	2018-08-13 00:
3	103	Jennifer	Dias	223 Oak Ave	Manhattan	New York	United States	10012	2018-09-13 00:
4	104	Noah	Williams	122 Adam St	Dallas	Texas	United States	10010	2018-09-13 00:
5	105	Rachel	Green	123 Brooklyn St	Brooklyn	New York	United States	10020	2018-09-13 00:
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

SCD 3 Data Flow

Maintenance Plan

Assign

Assign data loading times into the data warehouse

Purge

Purge data: get rid of unwanted data in the data warehouse as it is not an unlimited repository

Tune

Tune the system: Periodically review how the data warehouse is being used and fine tune the configuration to optimize its performance

Conclusion

- This newly formed central repository i.e. data warehouse is summarized data coming from various sources, without loss of any valuable information which is used for better decision making across the organization.

Thank You