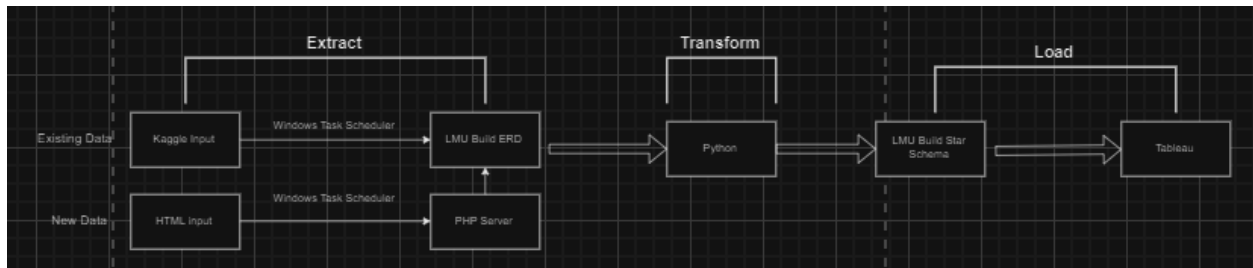


Automated ETL Pipeline:

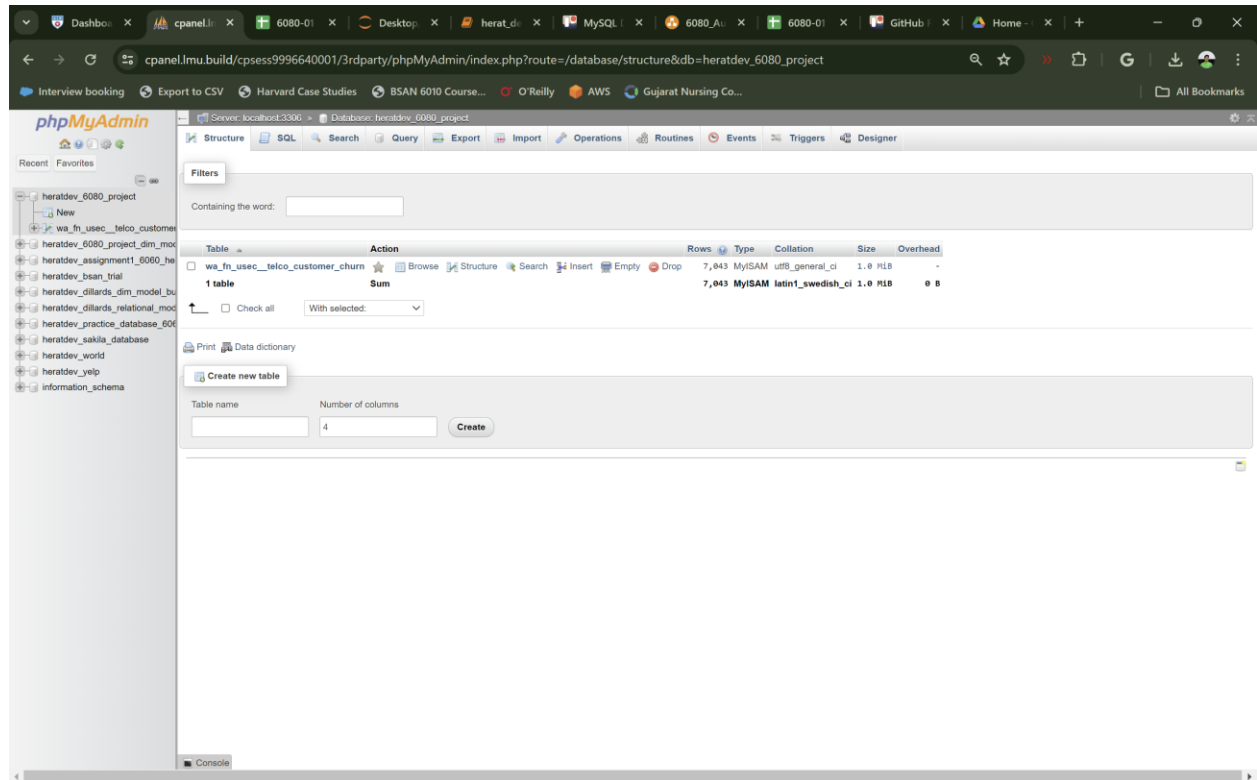
lontok	Running	At 16:02 every day	4/24/2024 16:02:54	11/30/1999 00:00:00	The task has not yet r
General	Triggers	Actions	Conditions	Settings	History
Number of events: 7					
Level	Date and Time	Event ID	Task Category	Operational Code	Correlation Id
Information	4/23/2024 16:04:19	102	Task completed	(2)	06308472-7ed8-4eb1-b94d-1d1c213d0d62
Information	4/23/2024 16:04:19	201	Action completed	(2)	06308472-7ed8-4eb1-b94d-1d1c213d0d62
Information	4/23/2024 16:04:19	110	Task triggered by user	Info	06308472-7ed8-4eb1-b94d-1d1c213d0d62
Information	4/23/2024 16:04:19	200	Action started	(1)	06308472-7ed8-4eb1-b94d-1d1c213d0d62
Information	4/23/2024 16:04:19	100	Task Started	(1)	06308472-7ed8-4eb1-b94d-1d1c213d0d62
Information	4/23/2024 16:04:19	129	Created Task Process	Info	
Information	4/23/2024 16:03:32	106	Task registered	Info	

The Entire ETL pipeline is also mentioned below:



1. Extract:

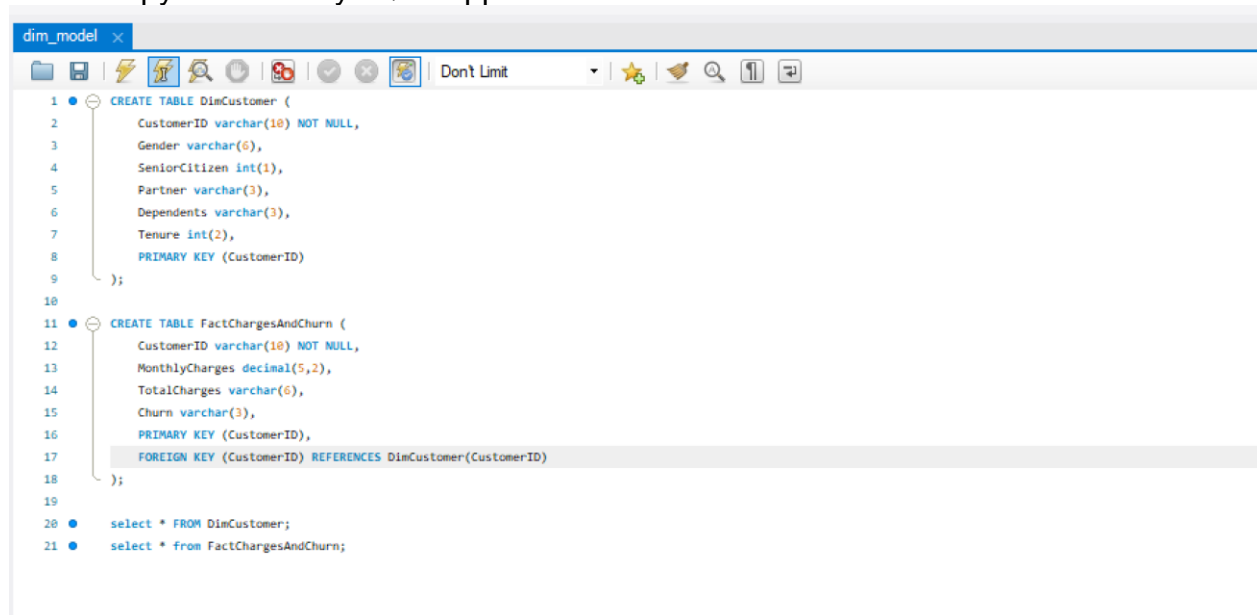
- There are 2 data sources, one is the existing data from Kaggle and the other is from a HTML format.
- The data from Kaggle goes directly into the LMU Build database and here are it's login credentials:
- Here are the connection details to our LMU Build for existing data:
- Hostname: heratdevisha.lmu.build
Username: heratdev_6080
Password: \$HEMAproject
Schema: heratdevisha_6080_project
Port: 3306



- We don't have an HTML form but if we are inputting new data then we can create the form and have the data host in an PHP server and then transfer it into the LMU build.

2. Transform:

- We made the connection into Python and then made our Star-Schema in it, here is a python and MySQL snippet.



```
In [13]: dim_customer_df = df_master_table[['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents', 'tenure']].copy()
```

```
In [14]: dim_customer_df
```

Out[14]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure
0	7590-VHVEG	Female	0	Yes	No	1
1	5575-GNVDE	Male	0	No	No	34
2	3668-QPYBK	Male	0	No	No	2
3	7795-CFOCW	Male	0	No	No	45
4	9237-HQITU	Female	0	No	No	2
...
7038	6840-RESVB	Male	0	Yes	Yes	24
7039	2234-XADUH	Female	0	Yes	Yes	72
7040	4801-JZAZL	Female	0	Yes	Yes	11
7041	8361-LTMKD	Male	1	Yes	No	4
7042	3186-AJIEK	Male	0	No	No	66

```
In [15]: fact_charges_and_churn_df = df_master_table[['customerID', 'MonthlyCharges', 'TotalCharges', 'Churn']].copy()
```

```
In [17]: fact_charges_and_churn_df
```

3. Load:

- Then we make the connection to our Dim_Model schema for loading the tables in our Star Schema database in LMU Build. Here are the login credentials:
- Hostname: heratdevisha.lmu.build
Username: heratdev_admin
Password: \$2608HBdevisha
Schema: heratdevisha_6080_project
Port: 3306
- Here is a python snippet for loading the tables:

```
In [18]: # Setting the Login credentials for dimension modelb
```

```
MYSQL_HOST1 = 'heratdevisha.lmu.build'  
MYSQL_USER1 = 'heratdev_admin'  
MYSQL_PASSWORD1 = '$2608HBdevisha'  
MYSQL_DB1 = 'heratdev_6080_project_dim_model'  
  
lmubuildengine1 = create_engine(f'mysql+mysqlconnector://{MYSQL_USER1}:{MYSQL_PASSWORD1}@{MYSQL_HOST1}/{MYSQL_DB1}')
```

```
In [19]: # Upload the DataFrame to the MySQL database
```

```
dim_customer_df.to_sql('DimCustomer', con=lmubuildengine1, if_exists='append', index=False)
```

Out[19]: 7043

```
In [20]: # Upload the DataFrame to the MySQL database
```

```
fact_charges_and_churn_df.to_sql('FactChargesAndChurn', con=lmubuildengine1, if_exists='append', index=False)
```

Out[20]: 7043

- Lastly we make our Tableau Dashboards by connecting to the Dim_Model database.