*** KGiSL INSTITUTE OF TECHNOLOGY***

***INSTITUTION CODE : 7117***

***NAAN MUDHALVAN***

***PROJECT TITLE:***

*Data Warehousing with IBM Cloud Db2 Warehouse.*

***MENTOR:***

*Mrs. R. Indu Poornima*

***TEAM MEMBERS:***

1. *Keerthana. J*
2. *Saruthi. T*
3. *Kishore. K*
4. *Jeevanandham. S*

***IMPLEMENTING ADVANCED ANALYTICS AND MACHINE LEARNING MODELS FOR PREDICTIVE ANALYSIS IN THE DATA WAREHOUSE***

***PROBLEM STATEMENT***

*The current state of the data warehouse lacks the capability to provide predictive insights from historical data. As a result, we are unable to anticipate trends, make proactive decisions, or optimize our operations effectively. This presents a significant challenge in an increasingly competitive business landscape.* *The main objective of the project focused on data warehousing using IBM Cloud Db2 Warehouse is to enable efficient data management and analysis for decision-making processes within an organization.*

***STEP 1:*** *Define the Schema and Structure of Data Warehouse Tables*

*Here's a basic schema for the data warehouse tables that is used in project:*

***1.Customer Table:***

*customer\_id (Primary Key)*

*name*

*email*

*address*

*phone*

***2.Product Table:***

*product\_id (Primary Key)*

*product\_name*

*category*

*price*

*description*

***3.Orders Table:***

*order\_id (Primary Key)*

*customer\_id (Foreign Key referencing Customer Table)*

*order\_date*

*total\_amount*

*status*

***4.Order\_Items Table:***

*order\_item\_id (Primary Key)*

*order\_id (Foreign Key referencing Orders Table)*

*product\_id (Foreign Key referencing Product Table)*

*quantity*

*unit\_price*

***STEP 2:*** *Identify Data Sources and Design Integration Strategy*

***1.CSV Files:***

*Customer data (customer.csv)*

*Product data (product.csv)*

*Order data (order.csv)*

*Order items data (order\_items.csv)*

***2.Databases:***

*Existing transactional databases (e.g., MySQL, PostgreSQL) containing customer, product, order, and order items data.*

***STEP 3:*** *Integration Strategy*

***1****.Using Db2 Warehouse Data Import Functionality: Utilize the Db2 Warehouse data import functionality to load CSV files directly into the tables created in the schema.*

***2****.Data Transformation and Loading (ETL): Use an ETL tool like IBM DataStage to extract data from existing transactional databases, transform it according to the data warehouse schema, and load it into Db2 Warehouse.*

***3****.Scheduled Data Syncing: Implement scheduled data syncing processes to ensure that the data warehouse remains updated with the latest data from the transactional databases.*

***4****.Data Quality and Consistency Checks: Implement data quality checks to ensure that the data being integrated into the data warehouse is accurate and consistent.*

***CODING:***

*The following libraries must be installed to perform the given operations,*

***1.ibm\_db***

***2.pandas***

pip install ibm\_db

pip install pandas

*The following code in python is used to create tables, import data from CSV files, and load data from an external database into IBM Cloud Db2 Warehouse.*

***import ibm\_db***

***import pandas as pd***

***# Connect to IBM Cloud Db2 Warehouse***

***dsn\_driver = "{IBM DB2 ODBC DRIVER}"***

***dsn\_database = "YOUR\_DATABASE\_NAME"***

***dsn\_hostname = "YOUR\_HOST\_NAME"***

***dsn\_port = "YOUR\_PORT"***

***dsn\_protocol = "TCPIP"***

***dsn\_uid = "YOUR\_USER\_ID"***

***dsn\_pwd = "YOUR\_PASSWORD"***

***dsn = (***

***"DRIVER={0};"***

***"DATABASE={1};"***

***"HOSTNAME={2};"***

***"PORT={3};"***

***"PROTOCOL={4};"***

***"UID={5};"***

***"PWD={6};").format(dsn\_driver, dsn\_database, dsn\_hostname, dsn\_port, dsn\_protocol, dsn\_uid, dsn\_pwd)***

***conn = ibm\_db.connect(dsn, "", "")***

***# Define the schema and structure of the data warehouse tables***

***create\_customer\_table\_query = '''***

***CREATE TABLE CUSTOMER (***

***customer\_id INT PRIMARY KEY,***

***name VARCHAR(255),***

***email VARCHAR(255),***

***address VARCHAR(255),***

***phone VARCHAR(20)***

***);***

***'''***

***create\_product\_table\_query = '''***

***CREATE TABLE PRODUCT (***

***product\_id INT PRIMARY KEY,***

***product\_name VARCHAR(255),***

***category VARCHAR(255),***

***price DECIMAL(10, 2),***

***description TEXT***

***);***

***'''***

***create\_orders\_table\_query = '''***

***CREATE TABLE ORDERS (***

***order\_id INT PRIMARY KEY,***

***customer\_id INT,***

***order\_date DATE,***

***total\_amount DECIMAL(10, 2),***

***status VARCHAR(50),***

***FOREIGN KEY (customer\_id) REFERENCES CUSTOMER(customer\_id)***

***);***

***'''***

***create\_order\_items\_table\_query = '''***

***CREATE TABLE ORDER\_ITEMS (***

***order\_item\_id INT PRIMARY KEY,***

***order\_id INT,***

***product\_id INT,***

***quantity INT,***

***unit\_price DECIMAL(10, 2),***

***FOREIGN KEY (order\_id) REFERENCES ORDERS(order\_id),***

***FOREIGN KEY (product\_id) REFERENCES PRODUCT(product\_id)***

***);***

***'''***

***# Execute the create table queries***

***stmt = ibm\_db.exec\_immediate(conn, create\_customer\_table\_query)***

***stmt = ibm\_db.exec\_immediate(conn, create\_product\_table\_query)***

***stmt = ibm\_db.exec\_immediate(conn, create\_orders\_table\_query)***

***stmt = ibm\_db.exec\_immediate(conn, create\_order\_items\_table\_query)***

***# Load data from CSV files into the tables***

***customer\_data = pd.read\_csv('customer.csv')***

***product\_data = pd.read\_csv('product.csv')***

***orders\_data = pd.read\_csv('order.csv')***

***order\_items\_data = pd.read\_csv('order\_items.csv')***

***customer\_data.to\_sql('CUSTOMER', conn, if\_exists='append', index=False)***

***product\_data.to\_sql('PRODUCT', conn, if\_exists='append', index=False)***

***orders\_data.to\_sql('ORDERS', conn, if\_exists='append', index=False)***

***order\_items\_data.to\_sql('ORDER\_ITEMS', conn, if\_exists='append', index=False)***

***# Close the connection***

***ibm\_db.close(conn)***

*This code establishes a connection to IBM Cloud Db2 Warehouse, creates tables as defined in the schema, and loads data from CSV files into the tables.*