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.