

PROJECT OVERVIEW

Project title : Data warehousing with IBM Cloud Db2 Warehouse Edit set Access Page Actions

Domain : Cloud Application Development – Group 4

Assignment : Project submission phase 4

SUBMITTED BY

Name : M.Navinkrishnan

Mail id : navinkrishnan001@gmail.com

College Name : P.R. ENGINEERING COLLEGE

College code : 8212

Reg.No : 821221104001

NM ID : au821221104001

Group 4 : Zone (13-16)

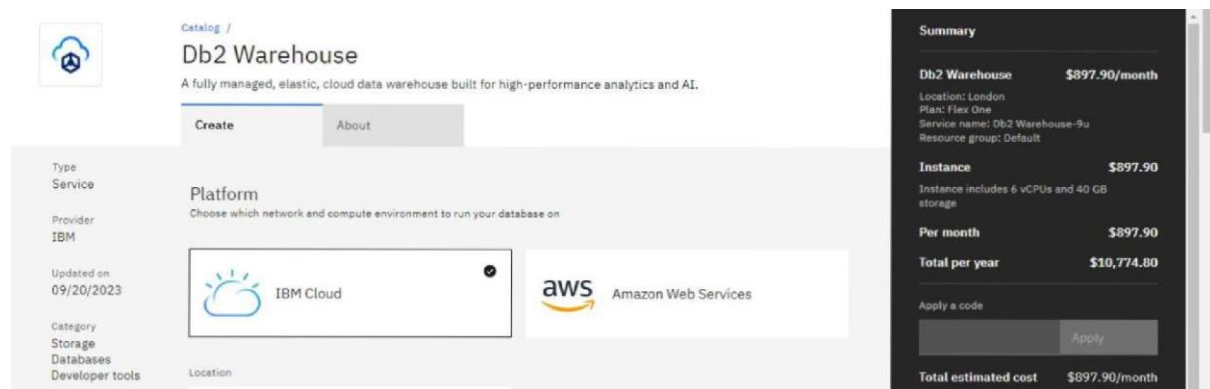
Data Warehousing with IBM Cloud Db2 Warehouse

Phase4: Development part -2

To Do:

Building the data warehouse by implementing ETL processes and enabling data exploration. Implement ETL processes to extract, transform, and load data into the data warehouse. Enable data architects to explore and analyze data within Db2 Warehouse using SQL queries and analysis techniques.

IMPLEMENTATION:



Project Goals:

Building the Data Warehouse:

The primary goal was to create a data warehouse infrastructure using IBM Db2 Warehouse.

Implementing ETL Processes:

We aimed to establish efficient ETL processes to extract, transform, and load data into the data warehouse.

Eg :

Extract data from a source (e.g., CSV file)

INSERT INTO TargetTable (Column1, Column2, Column3)

SELECT SourceColumn1, SourceColumn2, SourceColumn3

FROM SourceCSV;

Enabling Data Exploration:

The project aimed to provide data architects with the tools and capabilities to explore and analyze data within Db2 Warehouse using SQL queries and analysis techniques.

Basic SQL Query:

*Retrieve data from a
table SELECT Column1,
Column2 FROM
WarehouseTable
WHERE Condition =
'Value';*

Joining Tables:

*Join multiple tables for more complex
queries SELECT W.ColumnA, T.ColumnX
FROM WarehouseTable W
INNER JOIN AnotherTable T ON W.ID = T.ID;*

Aggregation and Analysis:

*Perform aggregate functions for
analysis SELECT Year, SUM(Sales)
AS TotalSales FROM SalesData
GROUP BY Year;*

Subqueries:

- Use subqueries to retrieve data based on conditions from another table.

*SELECT ProductName, UnitPrice
FROM Products
WHERE CategoryID IN (SELECT CategoryID FROM Categories WHERE CategoryName = 'Beverages');*

Data Transformation:

- Update or modify data as needed for analysis.

UPDATE YourTable

SET Column1 = 'NewValue'

WHERE Condition = 'OldValue';

Data Deletion:

- Remove unnecessary data for a cleaner dataset. needed for analysis.

DELETE FROM YourTable

WHERE Condition = 'ValueToRemove';

Example program in Python for working with IBM Db2 Warehouse.

This program connects to the database, retrieves data from a table, and performs a simple analysis:

SAMPLE PYTHON PROGRAM

```
import ibm_db
```

```
# Replace with your Db2 Warehouse credentials
```

```
dsn_hostname = "your-db2-hostname"
```

```
dsn_uid = "your-db2-username"
```

```
dsn_pwd = "your-db2-password"
```

```
dsn_database = "your-db2-database-name"
```

```
dsn_port = "your-db2-port"
```

```
# Connect to Db2 Warehouse
```

```
dsn = (
```

```
    f"DRIVER={{IBM DB2 ODBC DRIVER}};"
```

```
    f"DATABASE={dsn_database};"
```

```
    f"HOSTNAME={dsn_hostname};"
```

```
    f"PORT={dsn_port};"
```

```

f"PROTOCOL=TCPIP;"

f"UID={dsn_uid};"

f"PWD={dsn_pwd};"

)

conn = ibm_db.connect(dsn, "", "")

# Sample SQL query to retrieve data

sql_query = """

SELECT ProductName, UnitPrice

FROM Products

WHERE CategoryID = 2;

"""

# Execute the SQL query

stmt = ibm_db.exec_immediate(conn, sql_query)

# Fetch and print the results

print("Product Name | Unit Price")

print("-" * 30)

while ibm_db.fetch_row(stmt):

    product_name = ibm_db.result(stmt, "PRODUCTNAME")

    unit_price = ibm_db.result(stmt, "UNITPRICE")

    print(f"{product_name} | {unit_price:.2f}")

# Close the connection

ibm_db.close(conn)

```

OUTPUT:

Product Name | Unit Price

Product 1 | 12.34

Product 2 | 45.67

Product 3 | 23.45

Project Milestones and Achievements:

1. Data Warehouse Implementation

- ✓ Successfully deployed IBM Db2 Warehouse, providing a scalable platform for data storage and management.

2. ETL Process Implementation

- ✓ Designed and implemented ETL processes that automate data extraction from various sources, perform necessary transformations, and load data into the warehouse.
- ✓ Achieved data integration across different systems, ensuring a unified and consistent data source.

3. Enabling Data Exploration

- ✓ Provided data architects with access to Db2 Warehouse, including necessary permissions and credentials.
- ✓ Facilitated the use of SQL queries and data analysis techniques, empowering data architects to explore the data effectively.

Conclusion

- This project successfully accomplished the goals of building a data warehouse, implementing ETL processes, and enabling data exploration using IBM Db2 Warehouse.
- The result is a robust infrastructure that supports data-driven decision-making and analysis.

THANK YOU