IBM NAAN MUDHALVAN

CLOUD APPLICATION DEVELOPMENT

PROJECT

Data Warehousing with IBM Cloud Db2Warehouse

PHASE 5 : Documentation

Project Objective:

The objective of this data warehousing project using IBM Db2 is to create a robust and scalable data warehouse solution to enable efficient data storage, retrieval, and analysis for the organization. This project aims to:

1. Centralize Data: Consolidate data from various sources, providing a single source of truth for the organization.

2. Improve Reporting and Analytics: Enhance the reporting and analytical capabilities by providing a structured and easily accessible data repository.

3. Enable Data-Driven Decision-Making: Empower decision-makers with timely, accurate, and actionable insights.

4. Optimize Performance: Ensure that data retrieval and processing are efficient and support real-time or near-real-time analytics.

Design Thinking Process:

The design thinking process for this project involves the following stages:

1. **Empathize:**
   Understand the needs and pain points of stakeholders, including data analysts, business users, and IT teams. Conduct interviews and workshops to gather requirements.

2. **Define:**
   Define the scope, goals, and key performance indicators (KPIs) for the data warehouse. Create user personas and use cases to guide design.

3. **Ideate:**
   Brainstorm and generate creative ideas for the data warehouse architecture, data modeling, and integration strategies. Explore various possibilities for optimizing data storage and retrieval.

4. **Prototype:**
   Create a prototype or proof of concept to visualize the data warehouse's structure and user interface. Test and iterate on the design based on user feedback.

5. **Test:**
   Evaluate the prototype for usability, performance, and data quality. Identify and address any issues or bottlenecks.

6. **Implement:**
   Develop and deploy the data warehouse using IBM Db2, following the best practices and design principles established during the previous stages.

7. **Iterate**:
   Continuously improve the data warehouse based on user feedback and changing business needs. Implement enhancements and optimizations as required.

- Throughout the design thinking process, the project team should maintain close collaboration with stakeholders and ensure that the data warehouse solution aligns with the organization's objectives.

- This outline provides a framework for defining the project's objectives and the design thinking process that guides its development.

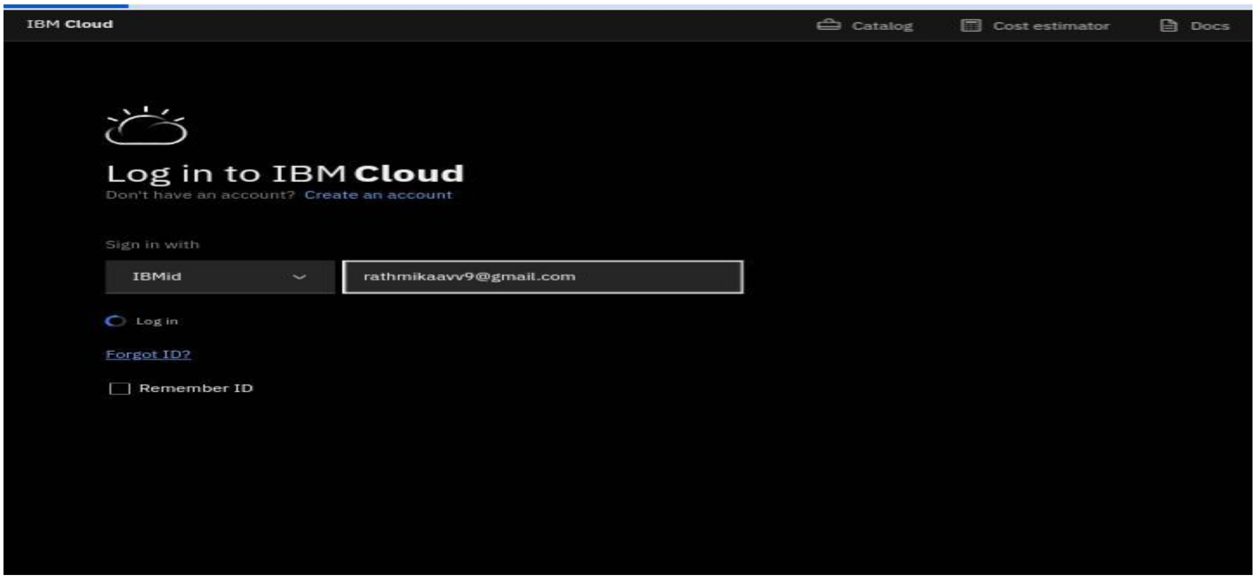
DEVELOPMENT STEPS:

In this part you will begin building your project.
Start building the data warehouse using IBM Cloud Db2 Warehouse.

Define the schema and structure of the data warehouse tables.Identify data sources (e.g., CSV files, databases) and design a strategy to integrate them into the data warehouse.

1. IBM Cloud Setup:

- Sign in to your IBM Cloud account or create one if you haven't already.



Log in to IBM Cloud

Don't have an account? Create an account

Sign in with

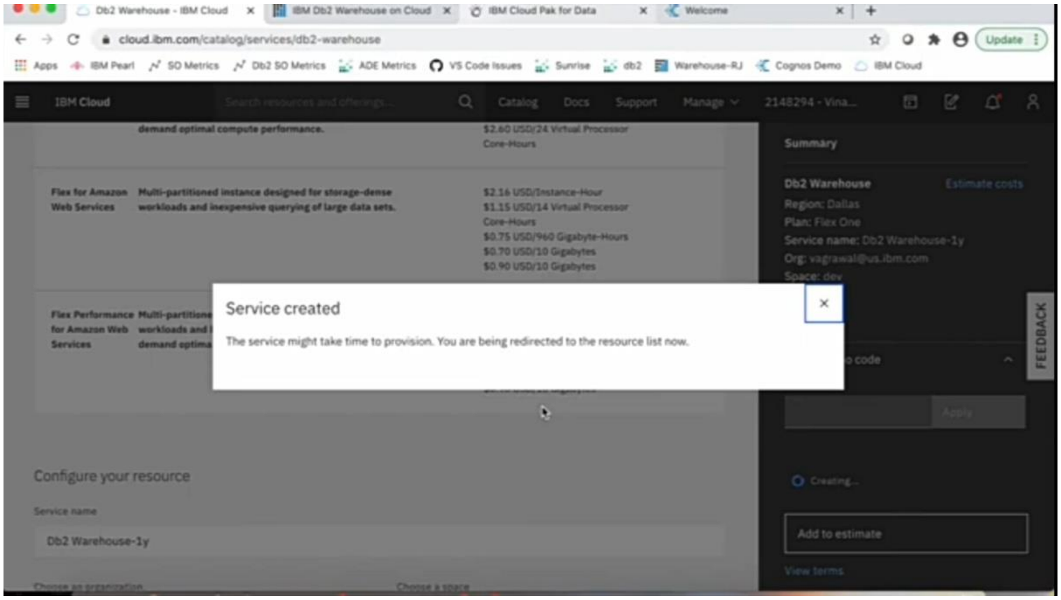
IBMid

rathmikaavv9@gmail.com

□ Remember ID

2. Provision Db2 Warehouse:

- In the IBM Cloud console, provision an instance of Db2 Warehouse.



cloud.ibm.com/catalog/services/db2-warehouse

	demand optimal compute performance.	$2.60 USD/24 Virtual Processor Core-Hours
Flex for Amazon Web Services	Multi-partitioned instance designed for storage-dense workloads and inexpensive querying of large data sets.	$2.16 USD/Instance-Hour $1.15 USD/14 Virtual Processor Core-Hours $0.75 USD/960 Gigabyte-Hours $0.70 USD/10 Gigabytes $0.90 USD/10 Gigabytes

Service created

The service might take time to provision. You are being redirected to the resource list now.

Summary

Db2 Warehouse

Region: Dallas

Plan: Flex One

Service name: Db2 Warehouse-1y

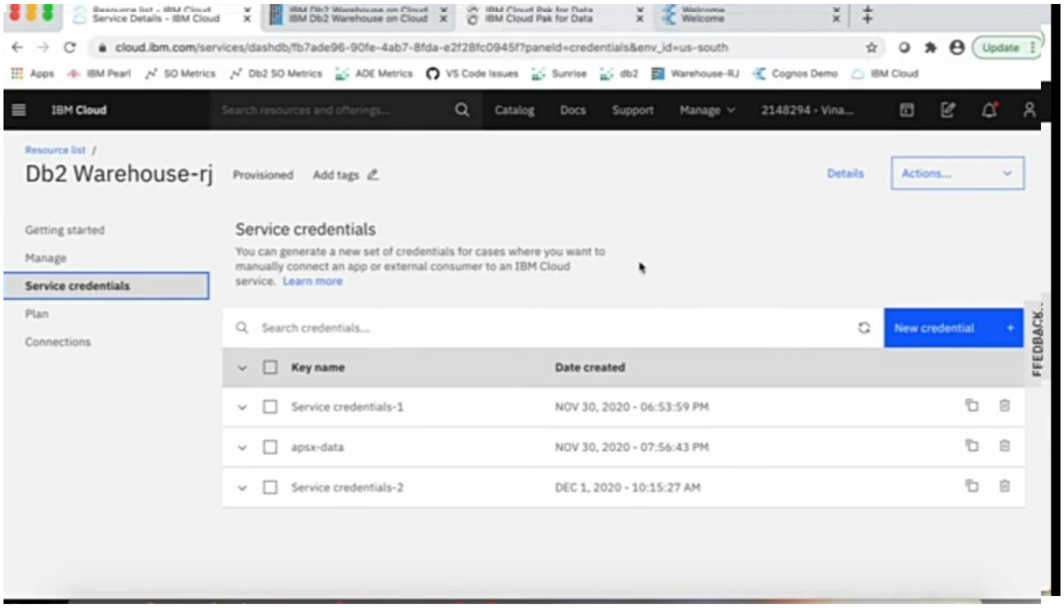
Org: vagrawal@us.ibm.com

Space: dev

Configure your resource

Service name: Db2 Warehouse-1y

-can be viewed in resource list



cloud.ibm.com/services/dashdb/fb7ade96-90fe-4ab7-8fda-e2f28fc0945f?paneld=credentials&env_id=us-south

Resource list /

Db2 Warehouse-rj Provisioned

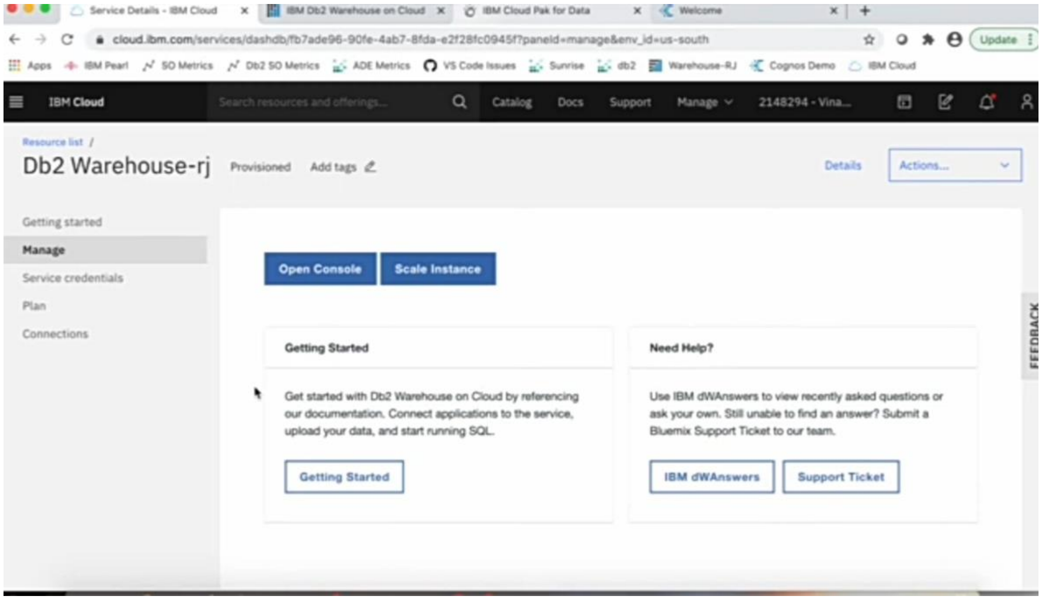
Service credentials

You can generate a new set of credentials for cases where you want to manually connect an app or external consumer to an IBM Cloud service. Learn more

Key name	Date created
Service credentials-1	NOV 30, 2020 - 06:53:59 PM
apsx-data	NOV 30, 2020 - 07:56:43 PM
Service credentials-2	DEC 1, 2020 - 10:15:27 AM

3. Connect to Db2 Warehouse:

- Use a SQL client like Db2 Data Studio or command-line tools to connect to your Db2 Warehouse instance.



cloud.ibm.com/services/dashdb/fb7ade96-90fe-4ab7-8fda-e2f28fc0945f?paneld=manage&env_id=us-south

Resource list /

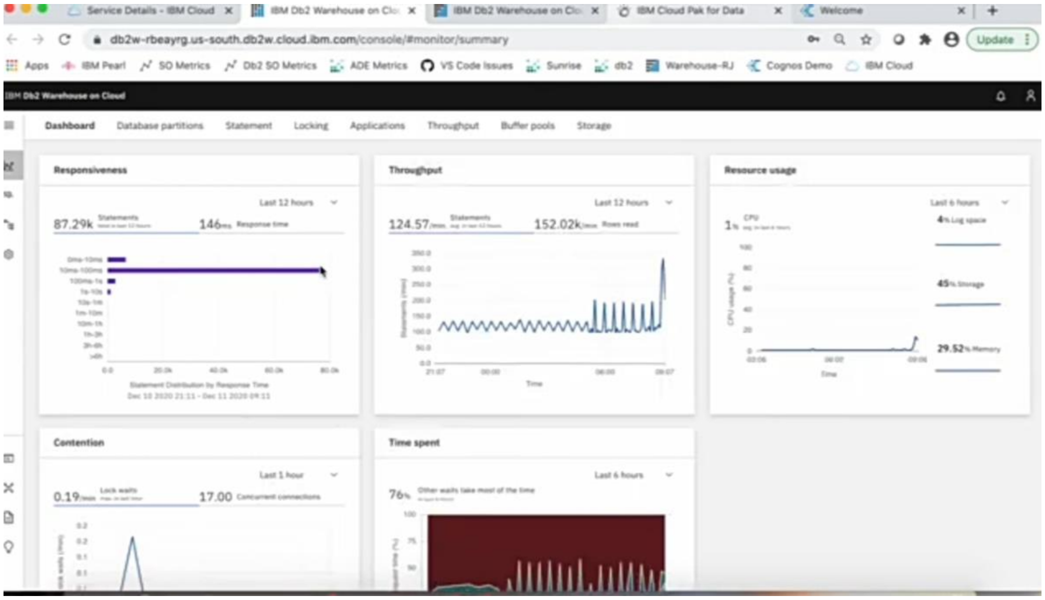
Db2 Warehouse-rj

Getting Started

Get started with Db2 Warehouse on Cloud by referencing our documentation. Connect applications to the service, upload your data, and start running SQL.

Need Help?

Use IBM dWAnswers to view recently asked questions or ask your own. Still unable to find an answer? Submit a Bluemix Support Ticket to our team.



db2w-rbeayrg.us-south.db2w.cloud.ibm.com/console/#monitor/summary

4. Schema Definition:

- Decide on the schema(s) for your data warehouse to logically organize data.

- Create a schema using SQL. For example:

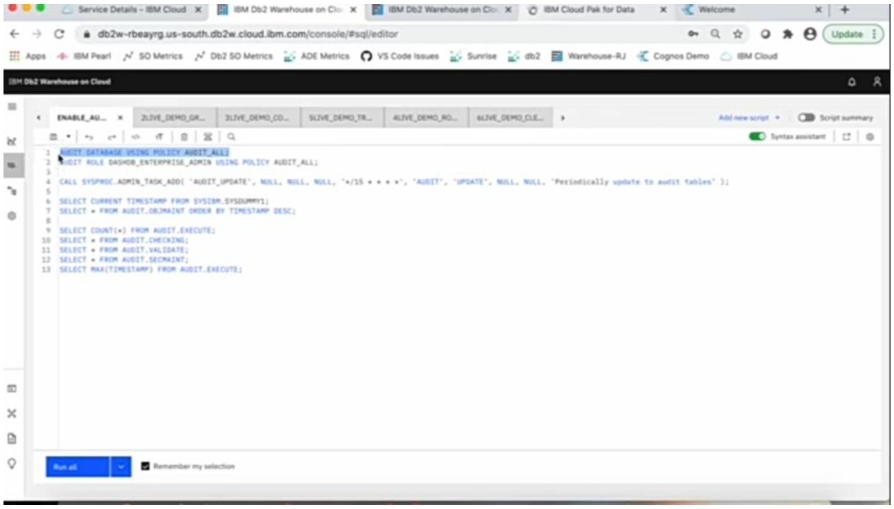
```sql
CREATE SCHEMA YOUR_SCHEMA;
```

5. Table Structure:

- Design the structure of your data warehouse tables. Determine which data attributes will be stored in each table.

- Create tables using SQL, specifying column names and data types. For example:

```sql
CREATE TABLE YOUR_SCHEMA.YOUR_TABLE (
  COLUMN1 DATATYPE,
  COLUMN2 DATATYPE,
  .
….
```

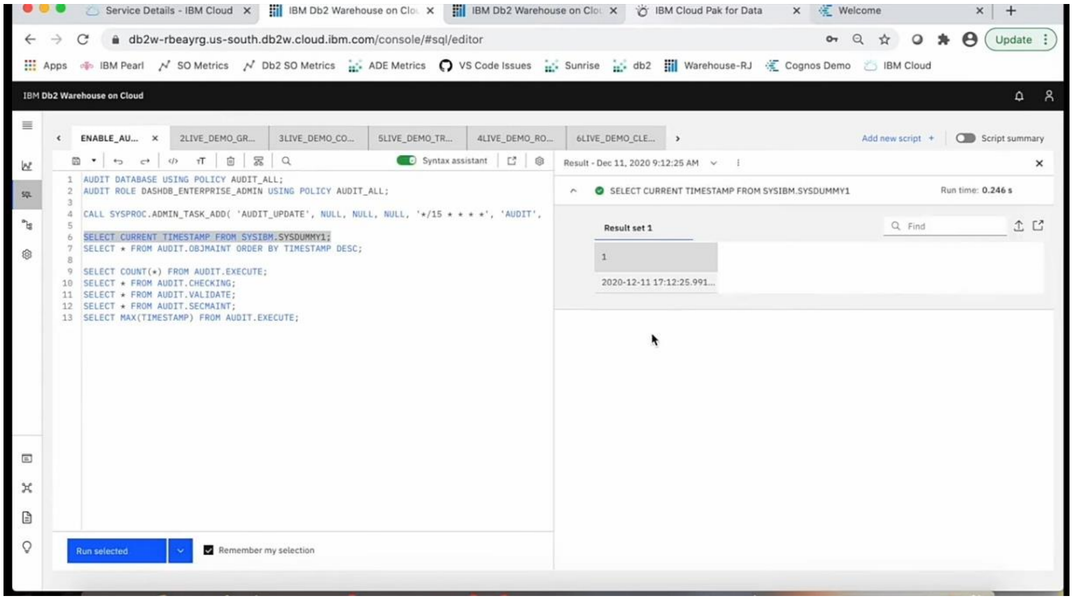


```
AUDIT DATABASE USING POLICY AUDIT_ALL;
AUDIT ROLE DASHDB_ENTERPRISE_ADMIN USING POLICY AUDIT_ALL;

CALL SYSPROC.ADMIN_TASK_ADD( 'AUDIT_UPDATE', NULL, NULL, NULL, '*/15 * * * *', 'AUDIT', 'UPDATE', NULL, NULL, 'Periodically update to audit tables' );

SELECT CURRENT TIMESTAMP FROM SYSIBM.SYSDUMMY1;
SELECT * FROM AUDIT.OBJMAINT ORDER BY TIMESTAMP DESC;

SELECT COUNT(*) FROM AUDIT.EXECUTE;
SELECT * FROM AUDIT.CHECKING;
SELECT * FROM AUDIT.VALIDATE;
SELECT * FROM AUDIT.SECMAINT;
SELECT MAX(TIMESTAMP) FROM AUDIT.EXECUTE;
```

.

....

);



```
AUDIT DATABASE USING POLICY AUDIT_ALL;
AUDIT ROLE DASHDB_ENTERPRISE_ADMIN USING POLICY AUDIT_ALL;

CALL SYSPROC.ADMIN_TASK_ADD( 'AUDIT_UPDATE', NULL, NULL, NULL, '*/15 * * * *', 'AUDIT',

SELECT CURRENT TIMESTAMP FROM SYSIBM.SYSDUMMY1;
SELECT * FROM AUDIT.OBJMAINT ORDER BY TIMESTAMP DESC;

SELECT COUNT(*) FROM AUDIT.EXECUTE;
SELECT * FROM AUDIT.CHECKING;
SELECT * FROM AUDIT.VALIDATE;
SELECT * FROM AUDIT.SECMAINT;
SELECT MAX(TIMESTAMP) FROM AUDIT.EXECUTE;
```

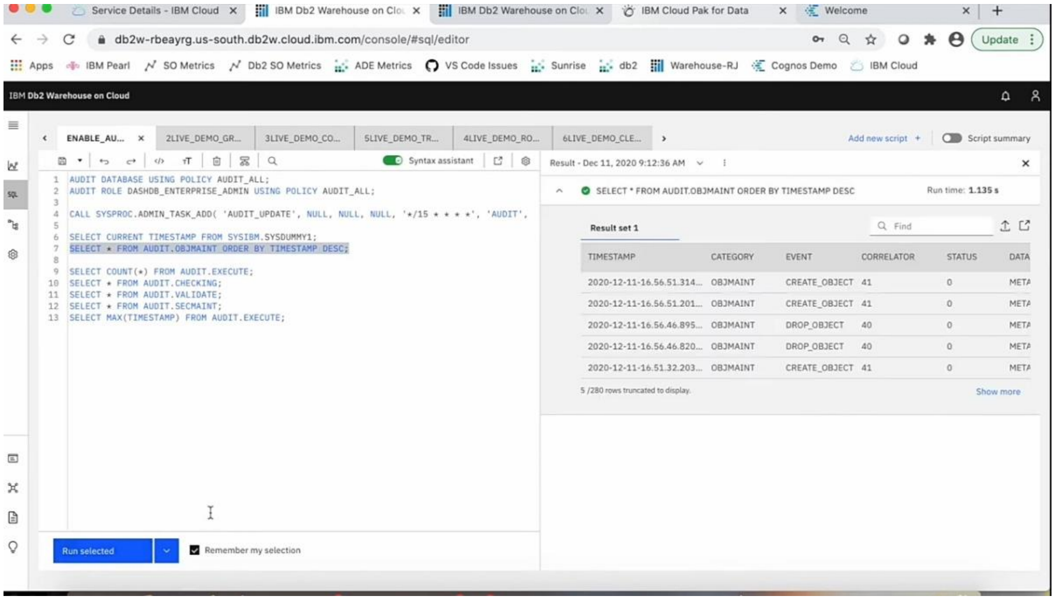
Result - Dec 11, 2020 9:12:25 AM

SELECT CURRENT TIMESTAMP FROM SYSIBM.SYSDUMMY1 Run time: 0.246 s

Result set 1

1
2020-12-11 17:12:25.991...

```
```



```
AUDIT DATABASE USING POLICY AUDIT_ALL;
AUDIT ROLE DASHDB_ENTERPRISE_ADMIN USING POLICY AUDIT_ALL;

CALL SYSPROC.ADMIN_TASK_ADD( 'AUDIT_UPDATE', NULL, NULL, NULL, '*/15 * * * *', 'AUDIT',

SELECT CURRENT TIMESTAMP FROM SYSIBM.SYSDUMMY1;
SELECT * FROM AUDIT.OBJMAINT ORDER BY TIMESTAMP DESC;

SELECT COUNT(*) FROM AUDIT.EXECUTE;
SELECT * FROM AUDIT.CHECKING;
SELECT * FROM AUDIT.VALIDATE;
SELECT * FROM AUDIT.SECMAINT;
SELECT MAX(TIMESTAMP) FROM AUDIT.EXECUTE;
```

Result - Dec 11, 2020 9:12:36 AM

SELECT * FROM AUDIT.OBJMAINT ORDER BY TIMESTAMP DESC Run time: 1.135 s

Result set 1

TIMESTAMP	CATEGORY	EVENT	CORRELATOR	STATUS	DATA
2020-12-11-16.56.51.314...	OBJMAINT	CREATE_OBJECT	41	0	META
2020-12-11-16.56.51.201...	OBJMAINT	CREATE_OBJECT	41	0	META
2020-12-11-16.56.46.895...	OBJMAINT	DROP_OBJECT	40	0	META
2020-12-11-16.56.46.820...	OBJMAINT	DROP_OBJECT	40	0	META
2020-12-11-16.51.32.203...	OBJMAINT	CREATE_OBJECT	41	0	META

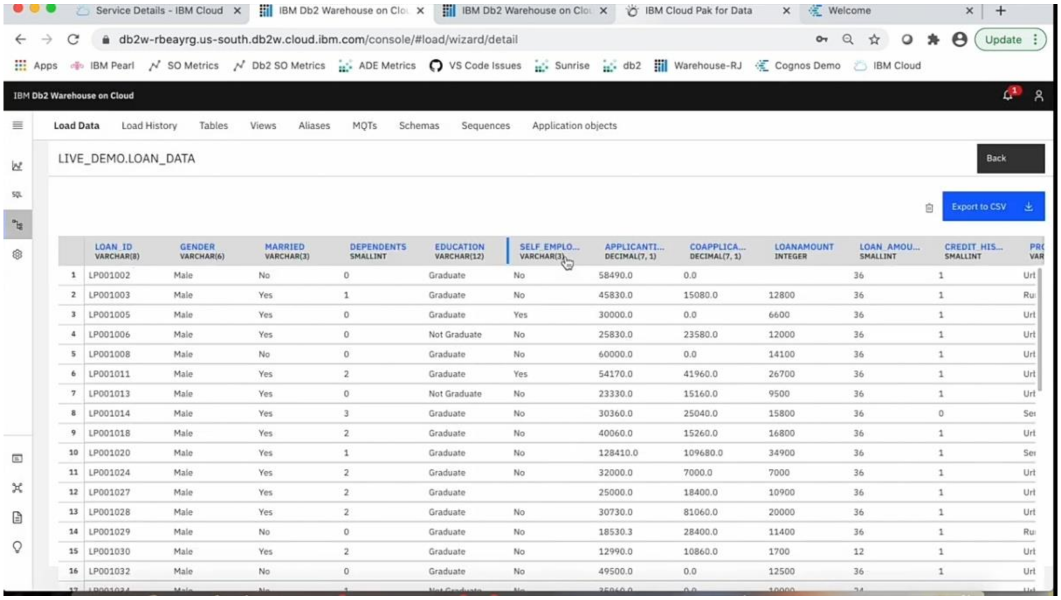
5 /280 rows truncated to display.

Show more

6. Data Sources Identification:

- Identify your data sources, which could include CSV files, other databases, APIs, etc.

.



LIVE_DEMO.LOAN_DATA

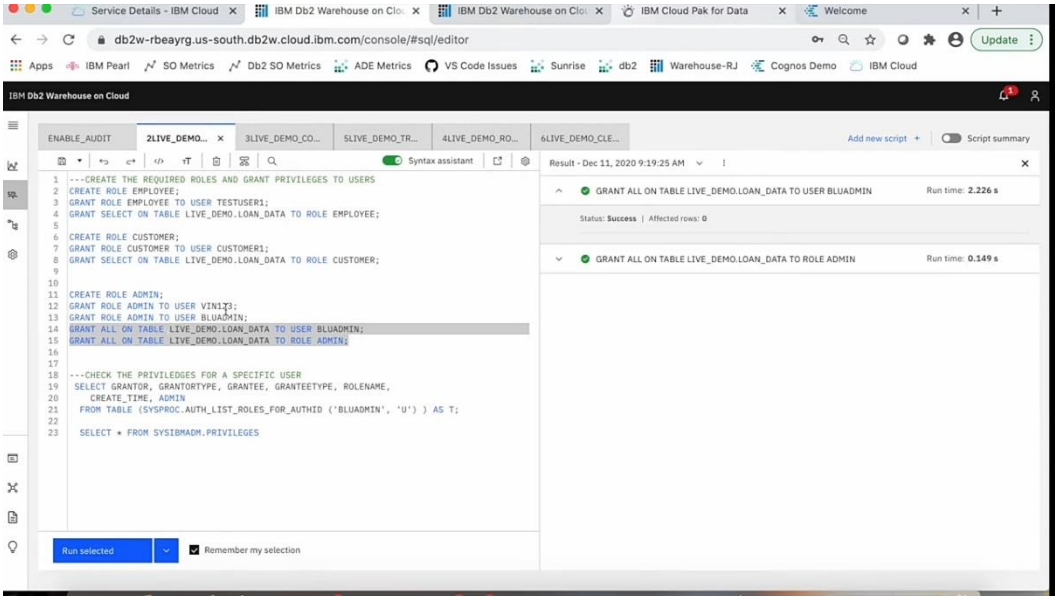
	LOAN_ID VARCHAR(8)	GENDER VARCHAR(6)	MARRIED VARCHAR(3)	DEPENDENTS SMALLINT	EDUCATION VARCHAR(12)	SELF_EMPLO... VARCHAR(3)	APPLICANTI... DECIMAL(7, 1)	COAPPLICA... DECIMAL(7, 1)	LOANAMOUNT INTEGER	LOAN_AMOU... SMALLINT	CREDIT_HIS... SMALLINT	PRC VAR
1	LP001002	Male	No	0	Graduate	No	58490.0	0.0		36	1	Urt
2	LP001003	Male	Yes	1	Graduate	No	45830.0	15080.0	12800	36	1	Ru
3	LP001005	Male	Yes	0	Graduate	Yes	30000.0	0.0	6600	36	1	Urt
4	LP001006	Male	Yes	0	Not Graduate	No	25830.0	23580.0	12000	36	1	Urt
5	LP001008	Male	No	0	Graduate	No	60000.0	0.0	14100	36	1	Urt
6	LP001011	Male	Yes	2	Graduate	Yes	54170.0	41960.0	26700	36	1	Urt
7	LP001013	Male	Yes	0	Not Graduate	No	23330.0	15160.0	9500	36	1	Urt
8	LP001014	Male	Yes	3	Graduate	No	30360.0	25040.0	15800	36	0	Se
9	LP001018	Male	Yes	2	Graduate	No	40060.0	15260.0	16800	36	1	Urt
10	LP001020	Male	Yes	1	Graduate	No	128410.0	109680.0	34900	36	1	Se
11	LP001024	Male	Yes	2	Graduate	No	32000.0	7000.0	7000	36	1	Urt
12	LP001027	Male	Yes	2	Graduate		25000.0	18400.0	10900	36	1	Urt
13	LP001028	Male	Yes	2	Graduate	No	30730.0	81060.0	20000	36	1	Urt
14	LP001029	Male	No	0	Graduate	No	18530.3	28400.0	11400	36	1	Ru
15	LP001030	Male	Yes	2	Graduate	No	12990.0	10860.0	1700	12	1	Urt
16	LP001032	Male	No	0	Graduate	No	49500.0	0.0	12500	36	1	Urt

7. Data Integration Strategy:

- Design a strategy to integrate data sources into your data warehouse. Consider using ETL (Extract, Transform, Load) tools or custom scripts.

- For CSV files, you might use Db2's `IMPORT` command to load data into your tables.

- For external databases, establish connections and design data transfer processes.

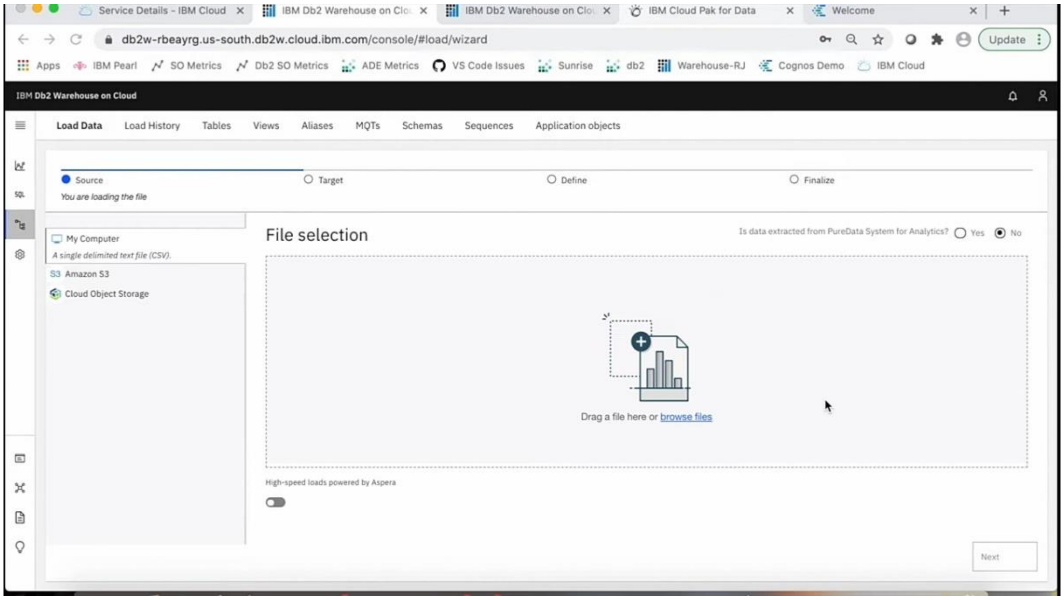


```
---CREATE THE REQUIRED ROLES AND GRANT PRIVILEGES TO USERS
CREATE ROLE EMPLOYEE;
GRANT ROLE EMPLOYEE TO USER TESTUSER1;
GRANT SELECT ON TABLE LIVE_DEMO.LOAN_DATA TO ROLE EMPLOYEE;

CREATE ROLE CUSTOMER;
GRANT ROLE CUSTOMER TO USER CUSTOMER1;
GRANT SELECT ON TABLE LIVE_DEMO.LOAN_DATA TO ROLE CUSTOMER;


CREATE ROLE ADMIN;
GRANT ROLE ADMIN TO USER VIN123;
GRANT ROLE ADMIN TO USER BLUADMIN;
GRANT ALL ON TABLE LIVE_DEMO.LOAN_DATA TO USER BLUADMIN;
GRANT ALL ON TABLE LIVE_DEMO.LOAN_DATA TO ROLE ADMIN;


---CHECK THE PRIVILEDGES FOR A SPECIFIC USER
 SELECT GRANTOR, GRANTORTYPE, GRANTEE, GRANTEETYPE, ROLENAME,
    CREATE_TIME, ADMIN
  FROM TABLE (SYSPROC.AUTH_LIST_ROLES_FOR_AUTHID ('BLUADMIN', 'U') ) AS T;

  SELECT * FROM SYSIBMADM.PRIVILEGES
```

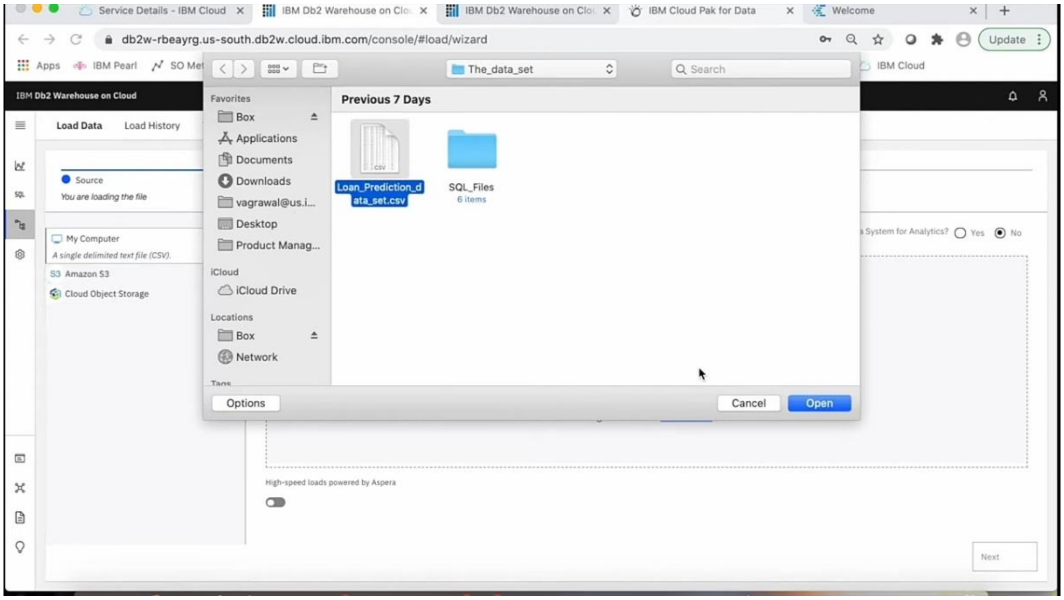
```
Result - Dec 11, 2020 9:19:25 AM
GRANT ALL ON TABLE LIVE_DEMO.LOAN_DATA TO USER BLUADMIN     Run time: 2.226 s
    Status: Success | Affected rows: 0
GRANT ALL ON TABLE LIVE_DEMO.LOAN_DATA TO ROLE ADMIN        Run time: 0.149 s
```

8. Data Loading:

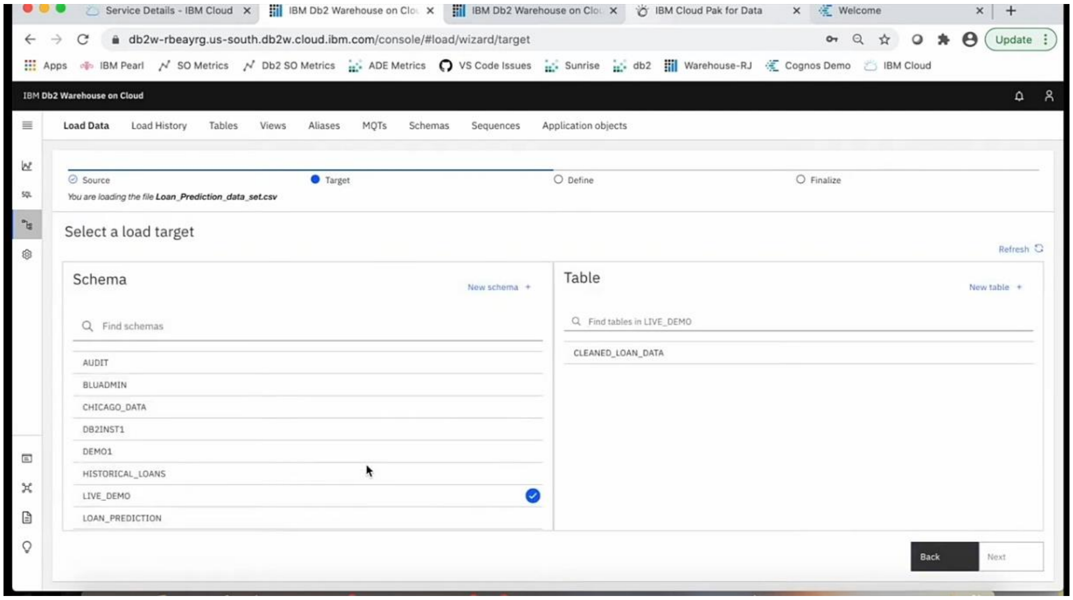
- Implement your data integration strategy to load data into your tables. This may involve running SQL scripts, using ETL tools, or other methods.


db2w-rbeayrg.us-south.db2w.cloud.ibm.com/console/#load/wizard
IBM Db2 Warehouse on Cloud
Load Data
Load History
Tables
Views
Aliases
MQTs
Schemas
Sequences
Application objects
Source
Target
Define
Finalize
You are loading the file
My Computer
A single delimited text file (CSV).
Amazon S3
Cloud Object Storage
File selection
Is data extracted from PureData System for Analytics? ○ Yes ● No
Drag a file here or browse files
High-speed loads powered by Aspera
Next

-add file



-load target file



db2w-rbeayrg.us-south.db2w.cloud.ibm.com/console/#load/wizard/target

*You are loading the file **Loan_Prediction_data_set.csv***

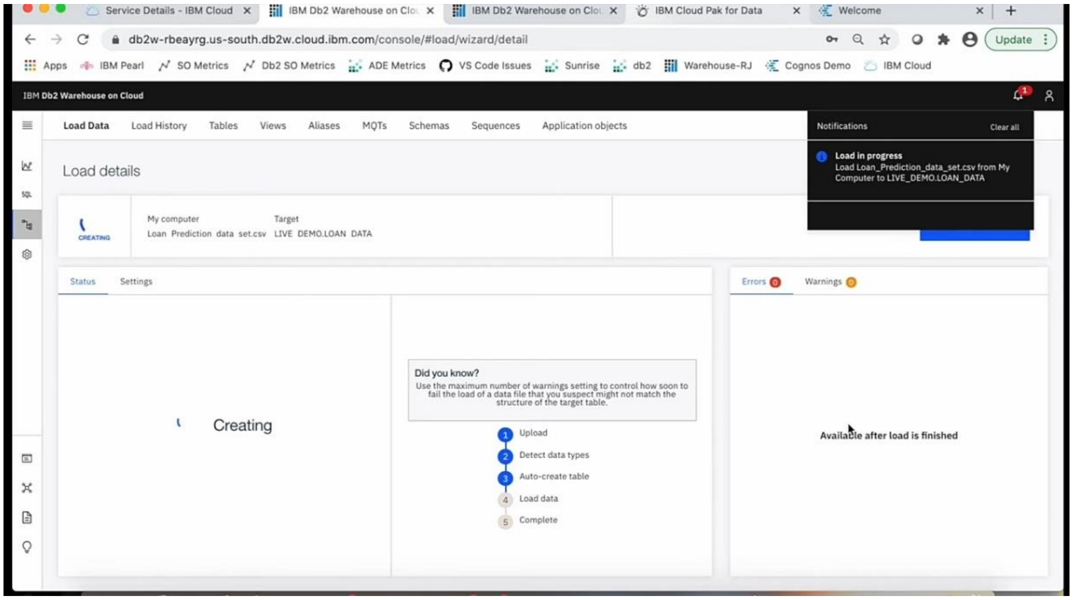
Select a load target

Schema

AUDIT
BLUADMIN
CHICAGO_DATA
DB2INST1
DEMO1
HISTORICAL_LOANS
LIVE_DEMO ✓
LOAN_PREDICTION

Table

CLEANED_LOAN_DATA



db2w-rbeayrg.us-south.db2w.cloud.ibm.com/console/#load/wizard/detail

Load details

CREATING

My computer: Loan Prediction data set.csv
Target: LIVE DEMO.LOAN DATA

Load in progress
Load Loan_Prediction_data_set.csv from My Computer to LIVE_DEMO.LOAN_DATA

Creating

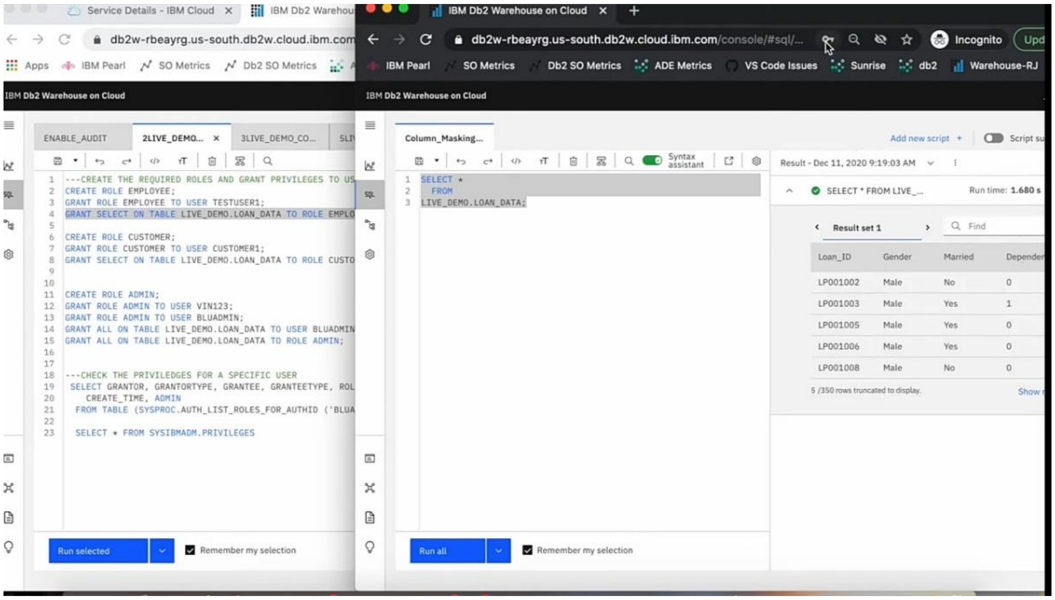
Did you know?
Use the maximum number of warnings setting to control how soon to fail the load of a data file that you suspect might not match the structure of the target table.

1 Upload
2 Detect data types
3 Auto-create table
4 Load data
5 Complete

Available after load is finished

9. Data Validation:

- Verify data integrity by running queries to validate that the data loaded correctly and is consistent with your expectations.



```
---CREATE THE REQUIRED ROLES AND GRANT PRIVILEGES TO US
CREATE ROLE EMPLOYEE;
GRANT ROLE EMPLOYEE TO USER TESTUSER1;
GRANT SELECT ON TABLE LIVE_DEMO.LOAN_DATA TO ROLE EMPLO

CREATE ROLE CUSTOMER;
GRANT ROLE CUSTOMER TO USER CUSTOMER1;
GRANT SELECT ON TABLE LIVE_DEMO.LOAN_DATA TO ROLE CUSTO


CREATE ROLE ADMIN;
GRANT ROLE ADMIN TO USER VIN123;
GRANT ROLE ADMIN TO USER BLUADMIN;
GRANT ALL ON TABLE LIVE_DEMO.LOAN_DATA TO USER BLUADMIN
GRANT ALL ON TABLE LIVE_DEMO.LOAN_DATA TO ROLE ADMIN;


---CHECK THE PRIVILEDGES FOR A SPECIFIC USER
 SELECT GRANTOR, GRANTORTYPE, GRANTEE, GRANTEETYPE, ROL
    CREATE_TIME, ADMIN
  FROM TABLE (SYSPROC.AUTH_LIST_ROLES_FOR_AUTHID ('BLUA

  SELECT * FROM SYSIBMADM.PRIVILEGES
```

```
SELECT *
  FROM
LIVE_DEMO.LOAN_DATA;
```

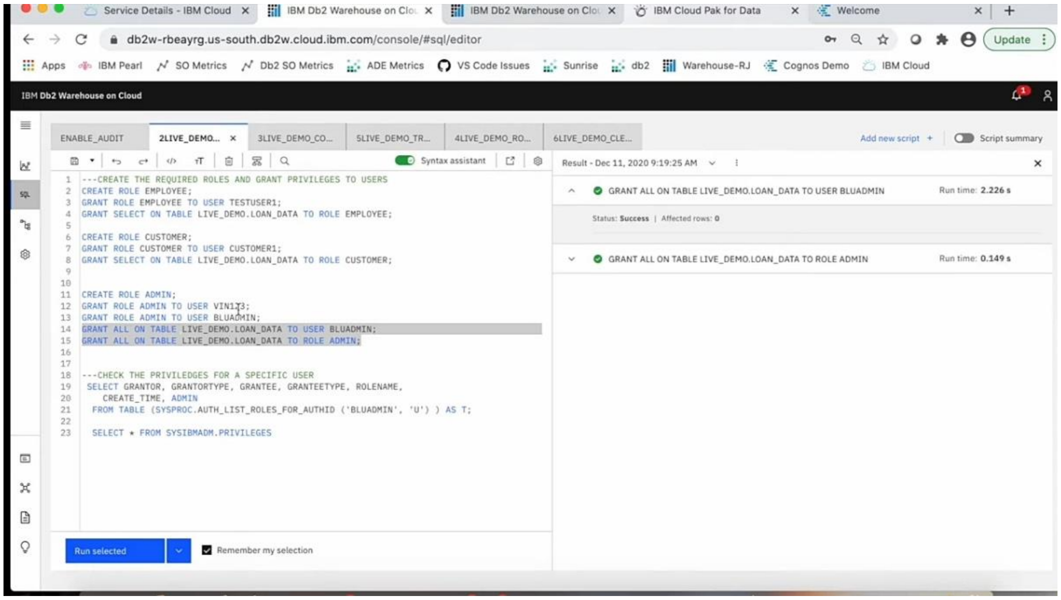
Result - Dec 11, 2020 9:19:03 AM

SELECT * FROM LIVE_... Run time: 1.680 s

Result set 1

Loan_ID	Gender	Married	Depender
LP001002	Male	No	0
LP001003	Male	Yes	1
LP001005	Male	Yes	0
LP001006	Male	Yes	0
LP001008	Male	No	0

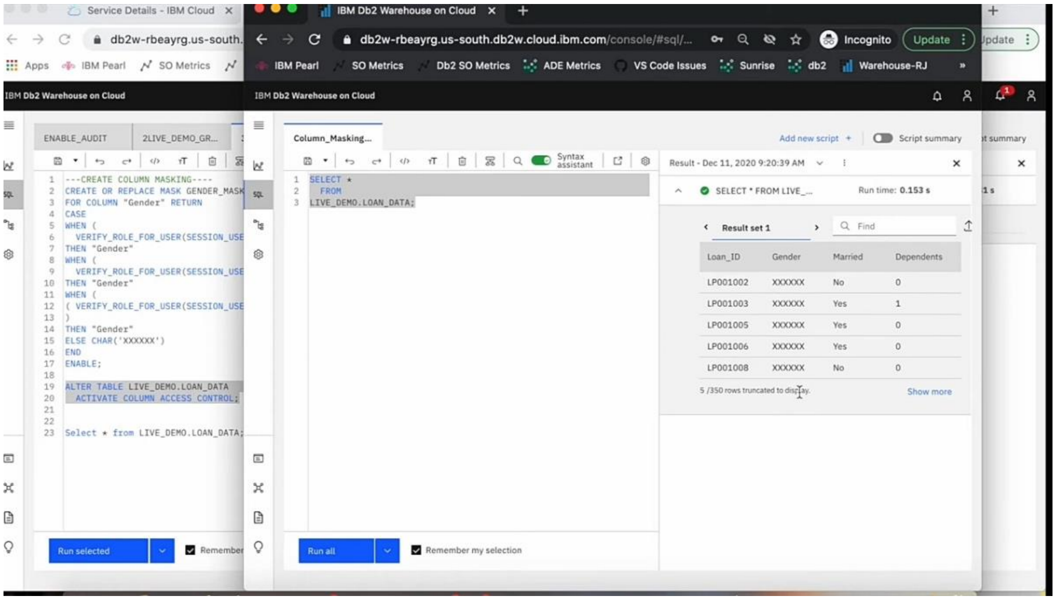
5 /350 rows truncated to display.



```
---CREATE THE REQUIRED ROLES AND GRANT PRIVILEGES TO USERS
CREATE ROLE EMPLOYEE;
GRANT ROLE EMPLOYEE TO USER TESTUSER1;
GRANT SELECT ON TABLE LIVE_DEMO.LOAN_DATA TO ROLE EMPLOYEE;

CREATE ROLE CUSTOMER;
GRANT ROLE CUSTOMER TO USER CUSTOMER1;
GRANT SELECT ON TABLE LIVE_DEMO.LOAN_DATA TO ROLE CUSTOMER;


CREATE ROLE ADMIN;
GRANT ROLE ADMIN TO USER VIN123;
GRANT ROLE ADMIN TO USER BLUADMIN;
GRANT ALL ON TABLE LIVE_DEMO.LOAN_DATA TO USER BLUADMIN;
GRANT ALL ON TABLE LIVE_DEMO.LOAN_DATA TO ROLE ADMIN;


---CHECK THE PRIVILEDGES FOR A SPECIFIC USER
 SELECT GRANTOR, GRANTORTYPE, GRANTEE, GRANTEETYPE, ROLENAME,
    CREATE_TIME, ADMIN
  FROM TABLE (SYSPROC.AUTH_LIST_ROLES_FOR_AUTHID ('BLUADMIN', 'U') ) AS T;

  SELECT * FROM SYSIBMADM.PRIVILEGES
```

```
Result - Dec 11, 2020 9:19:25 AM

GRANT ALL ON TABLE LIVE_DEMO.LOAN_DATA TO USER BLUADMIN    Run time: 2.226 s
    Status: Success | Affected rows: 0

GRANT ALL ON TABLE LIVE_DEMO.LOAN_DATA TO ROLE ADMIN       Run time: 0.149 s
```

-



```
---CREATE COLUMN MASKING----
CREATE OR REPLACE MASK GENDER_MASK
FOR COLUMN "Gender" RETURN
CASE
WHEN (
  VERIFY_ROLE_FOR_USER(SESSION_USE
THEN "Gender"
WHEN (
  VERIFY_ROLE_FOR_USER(SESSION_USE
THEN "Gender"
WHEN (
( VERIFY_ROLE_FOR_USER(SESSION_USE
)
THEN "Gender"
ELSE CHAR('XXXXXX')
END
ENABLE;

ALTER TABLE LIVE_DEMO.LOAN_DATA
  ACTIVATE COLUMN ACCESS CONTROL;


Select * from LIVE_DEMO.LOAN_DATA;
```

```
SELECT *
  FROM
LIVE_DEMO.LOAN_DATA;
```

Result - Dec 11, 2020 9:20:39 AM

SELECT * FROM LIVE_... Run time: 0.153 s

Result set 1

Loan_ID	Gender	Married	Dependents
LP001002	XXXXXX	No	0
LP001003	XXXXXX	Yes	1
LP001005	XXXXXX	Yes	0
LP001006	XXXXXX	Yes	0
LP001008	XXXXXX	No	0

5 /350 rows truncated to display.

Certainly, here's an overview of the data warehouse structure, data integration strategies, ETL processes,

data exploration techniques, and how IBM Db2 Warehouse enables data architects to deliver actionable insights:

Data Warehouse Structure (IBM Db2 Warehouse):

1. **Database Type:**
   IBM Db2 Warehouse is a relational database management system (RDBMS) optimized for analytical workloads, providing support for structured data.

2. **Schema Design:**
   The data warehouse typically uses a star schema or snowflake schema for efficient querying. It includes fact tables for key business metrics and dimension tables for descriptive attributes.

3. **Data Storage and Partitioning:**
   Data is stored in a columnar format for faster analytical queries. Partitioning is used to distribute data across multiple storage devices, optimizing query performance.

4. **Indexing Methods:**
   IBM Db2 Warehouse supports various indexing methods, including B-tree, bitmap, and multidimensional clustering indexes, to accelerate data retrieval.

Data Integration Strategies:

1. **Data Sources:**
   Data can be integrated from various sources, including relational databases, external APIs, flat files, and streaming data sources.

2. **Data Extraction:**
   Data is extracted from source systems using ETL processes, SQL queries, or data connectors provided by IBM Db2 Warehouse.

3. **Transformation Procedures:**
   ETL processes involve data transformation to clean, structure, and format data for consistency and compatibility within the data warehouse.

4. **Loading Mechanisms**:
   Data can be loaded into the warehouse using batch processes or near-real-time streaming to ensure data freshness.

ETL Processes (IBM Db2 Warehouse):

1. **IBM DataStage:**
   IBM DataStage, part of the IBM Information Server, is a popular ETL tool used to extract, transform, and load data into Db2 Warehouse. It provides a visual interface for designing ETL workflows.

2. **SQL Scripts:**
   Data architects can also use SQL scripts to perform ETL operations, leveraging Db2 Warehouse's SQL capabilities for data transformation and loading.

3. **Scheduling and Automation:**
   ETL jobs can be scheduled and automated using IBM DataStage or other scheduling tools, ensuring that data is regularly updated in the warehouse.

In this part you will continue building your project.Continue 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.

ETL

ETL, which stands for extract, transform and load, is a data integration process that combines data from multiple data sources into a single, consistent data store that is loaded into a data warehouse or other target system.

As the databases grew in popularity in the 1970s, ETL was introduced as a process for integrating and loading data for computation and analysis, eventually becoming the primary method to process data for data warehousing projects.

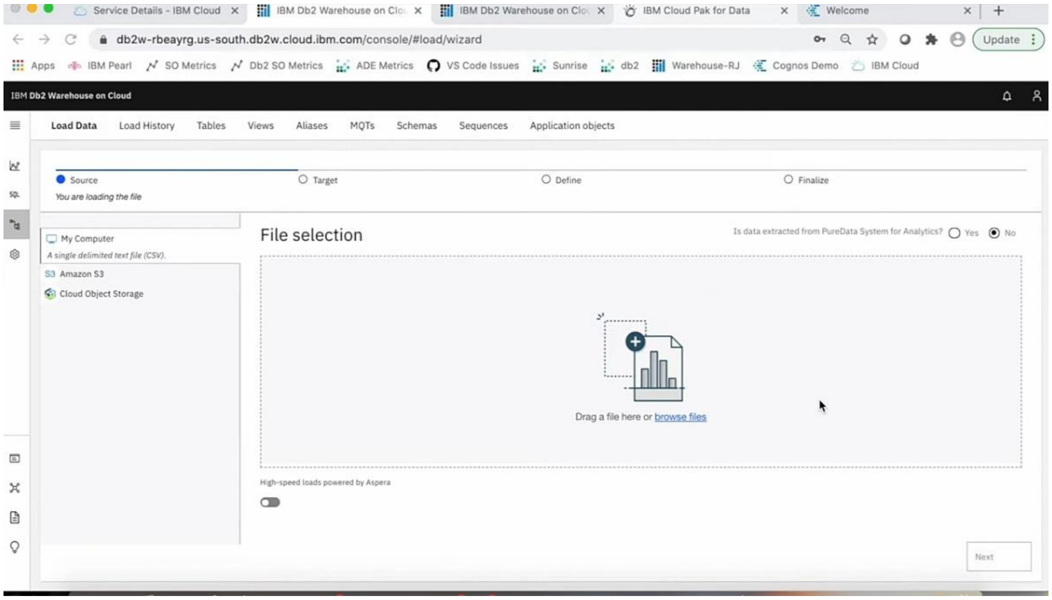
ETL provides the foundation for data analytics and machine learning workstreams. Through a series of business rules, ETL cleanses and organizes data in a way which addresses specific business intelligence needs, like monthly reporting, but it can also tackle more advanced analytics, which can improve back-end processes or end user experiences. ETL is often used by an organization to:

- Extract data from legacy systems
- Cleanse the data to improve data quality and establish consistency
- Load data into a target database

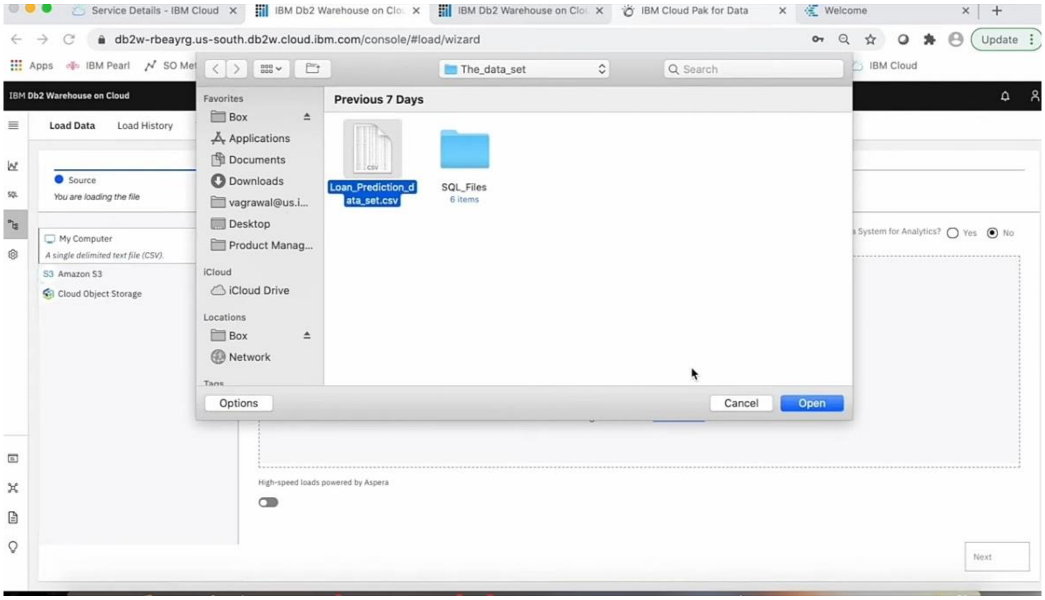
Here are some key steps you might consider:

1. **Data Extraction:**

   Extract data from various source systems. This could involve databases, files, APIs, or other data repositories.


db2w-rbeayrg.us-south.db2w.cloud.ibm.com/console/#load/wizard
IBM Db2 Warehouse on Cloud
Load Data
Load History
Tables
Views
Aliases
MQTs
Schemas
Sequences
Application objects
Source
You are loading the file
Target
Define
Finalize
My Computer
A single delimited text file (CSV).
Amazon S3
Cloud Object Storage
File selection
Drag a file here or browse files
High-speed loads powered by Aspera
Next

Is data extracted from PureData System for Analytics? ○ Yes ● No

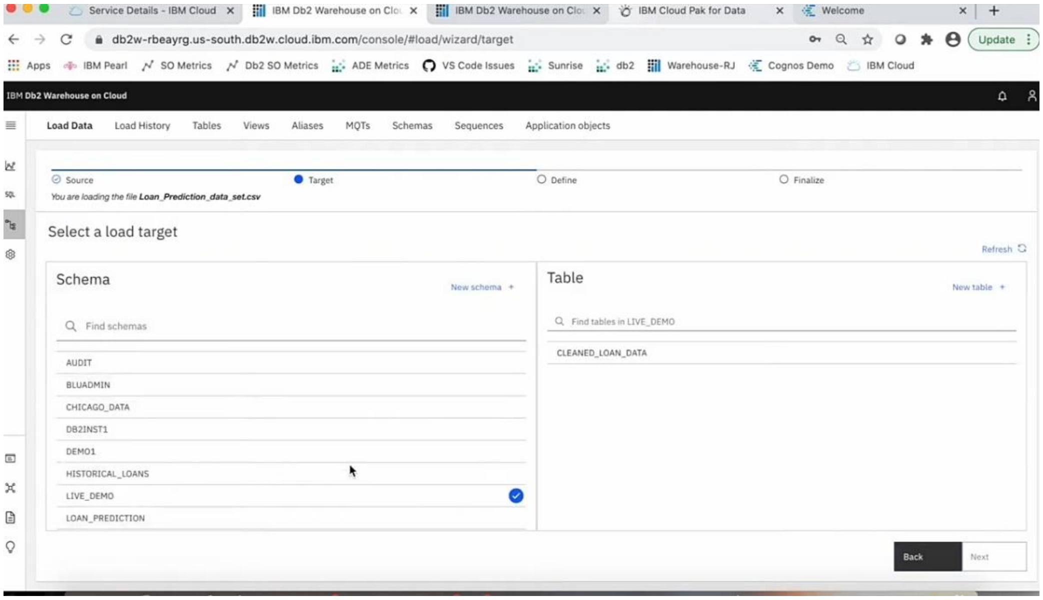


2. **Data Transformation:**

   Clean and preprocess the extracted data. This includes tasks like data cleansing, data enrichment, and data normalization.

3.Data Loading:

Load the transformed data into your IBM Db2 Warehouse. Depending on your needs, you might use bulk loading, real-time streaming, or other loading methods.



You are loading the file *Loan_Prediction_data_set.csv*

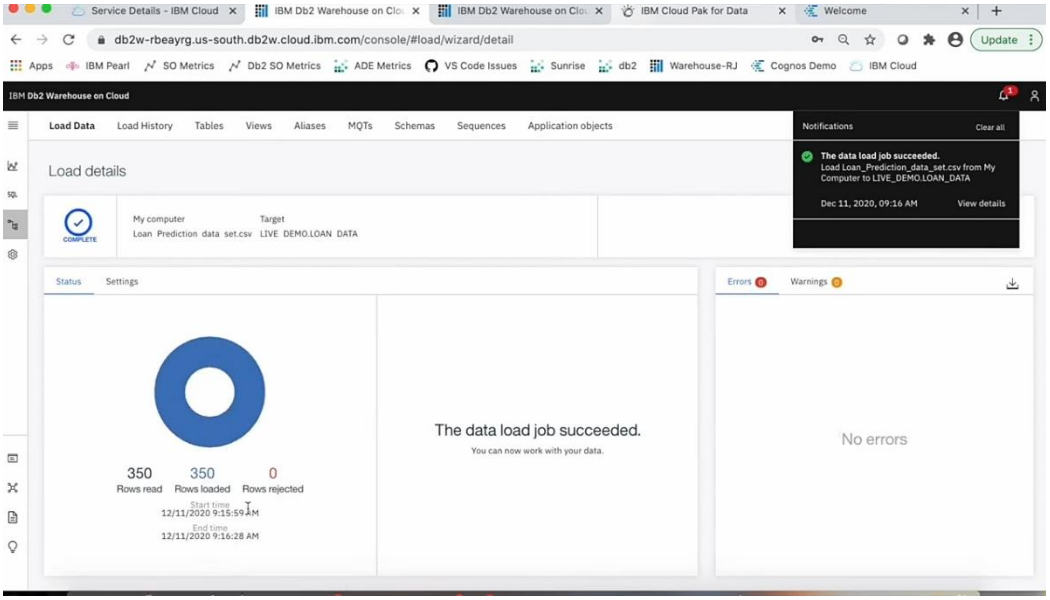
Select a load target

Schema

AUDIT
BLUADMIN
CHICAGO_DATA
DB2INST1
DEMO1
HISTORICAL_LOANS
LIVE_DEMO ✓
LOAN_PREDICTION

Table

CLEANED_LOAN_DATA



Load details

My computer
Loan Prediction data set.csv

Target
LIVE DEMO.LOAN DATA

The data load job succeeded.
Load Loan_Prediction_data_set.csv from My Computer to LIVE_DEMO.LOAN_DATA
Dec 11, 2020, 09:16 AM

350 Rows read
350 Rows loaded
0 Rows rejected

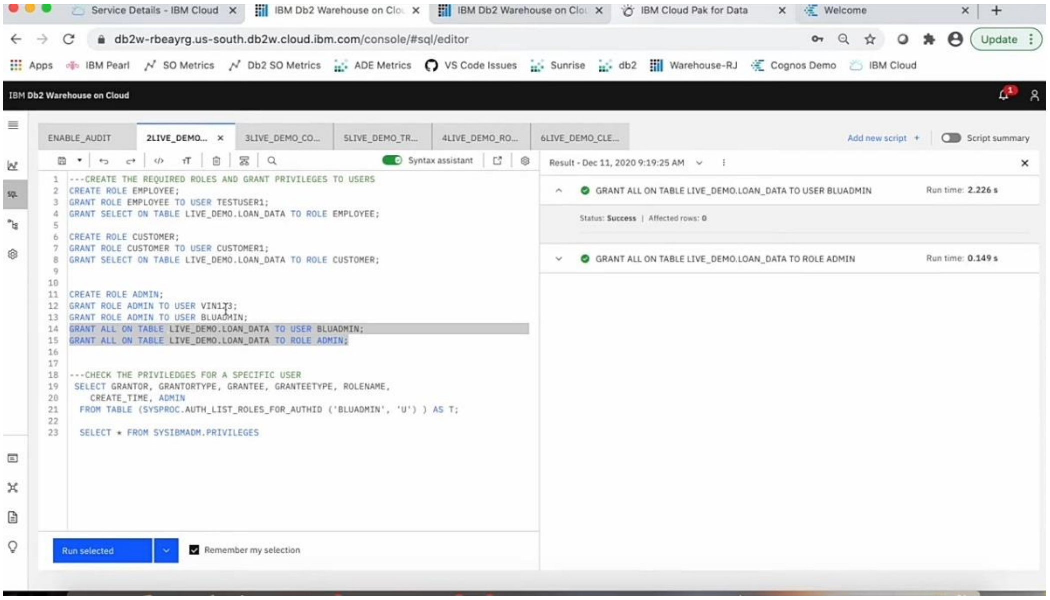
Start time
12/11/2020 9:15:59 AM
End time
12/11/2020 9:16:28 AM

The data load job succeeded.
You can now work with your data.

No errors

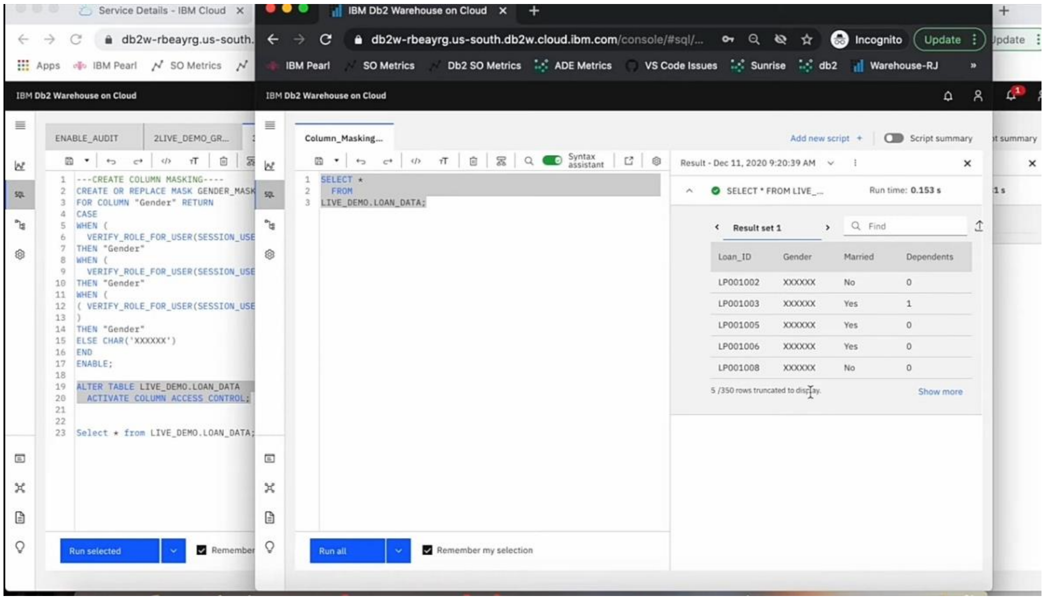
4. **Data Exploration:**

Provide data architects with tools and access to the data in Db2 Warehouse. SQL is a powerful tool for querying and analyzing data.



```
---CREATE THE REQUIRED ROLES AND GRANT PRIVILEGES TO USERS
CREATE ROLE EMPLOYEE;
GRANT ROLE EMPLOYEE TO USER TESTUSER1;
GRANT SELECT ON TABLE LIVE_DEMO.LOAN_DATA TO ROLE EMPLOYEE;

CREATE ROLE CUSTOMER;
GRANT ROLE CUSTOMER TO USER CUSTOMER1;
GRANT SELECT ON TABLE LIVE_DEMO.LOAN_DATA TO ROLE CUSTOMER;


CREATE ROLE ADMIN;
GRANT ROLE ADMIN TO USER VIN123;
GRANT ROLE ADMIN TO USER BLUADMIN;
GRANT ALL ON TABLE LIVE_DEMO.LOAN_DATA TO USER BLUADMIN;
GRANT ALL ON TABLE LIVE_DEMO.LOAN_DATA TO ROLE ADMIN;


---CHECK THE PRIVILEDGES FOR A SPECIFIC USER
 SELECT GRANTOR, GRANTORTYPE, GRANTEE, GRANTEETYPE, ROLENAME,
    CREATE_TIME, ADMIN
  FROM TABLE (SYSPROC.AUTH_LIST_ROLES_FOR_AUTHID ('BLUADMIN', 'U') ) AS T;

  SELECT * FROM SYSIBMADM.PRIVILEGES
```

```
Result - Dec 11, 2020 9:19:25 AM
GRANT ALL ON TABLE LIVE_DEMO.LOAN_DATA TO USER BLUADMIN    Run time: 2.226 s
Status: Success | Affected rows: 0
GRANT ALL ON TABLE LIVE_DEMO.LOAN_DATA TO ROLE ADMIN    Run time: 0.149 s
```



```
---CREATE COLUMN MASKING----
CREATE OR REPLACE MASK GENDER_MASK
FOR COLUMN "Gender" RETURN
CASE
WHEN (
  VERIFY_ROLE_FOR_USER(SESSION_USE
THEN "Gender"
WHEN (
  VERIFY_ROLE_FOR_USER(SESSION_USE
THEN "Gender"
WHEN (
( VERIFY_ROLE_FOR_USER(SESSION_USE
)
THEN "Gender"
ELSE CHAR('XXXXXX')
END
ENABLE;

ALTER TABLE LIVE_DEMO.LOAN_DATA
  ACTIVATE COLUMN ACCESS CONTROL;


Select * from LIVE_DEMO.LOAN_DATA;
```

```
SELECT *
  FROM
LIVE_DEMO.LOAN_DATA;
```

Result - Dec 11, 2020 9:20:39 AM

SELECT * FROM LIVE_... Run time: 0.153 s

Result set 1

Loan_ID	Gender	Married	Dependents
LP001002	XXXXXX	No	0
LP001003	XXXXXX	Yes	1
LP001005	XXXXXX	Yes	0
LP001006	XXXXXX	Yes	0
LP001008	XXXXXX	No	0

5 /350 rows truncated to display.

5. SQL Query Optimization:

Ensure that your SQL queries are optimized for performance. This might involve creating indexes, tuning queries, and monitoring query performance.

6. Data Analysis Techniques:

Support data architects in applying various data analysis techniques such as data mining, machine learning, and statistical analysis to derive insights from the data.

7. Data Governance:

Implement data governance practices to ensure data quality, security, and compliance with relevant regulations.

8. Documentation:

Document the ETL processes, data models, and analysis techniques for future reference and knowledge sharing.

How ETL works

The easiest way to understand how ETL works is to understand what happens in each step of the process.

Extract

During data extraction, raw data is copied or exported from source locations to a staging area. Data management teams can extract data from a variety of data sources, which can be structured or unstructured. Those sources include but are not limited to:

- SQL or NoSQL servers
- CRM and ERP systems
- Flat files
- Email
- Web pages

Transform

In the staging area, the raw data undergoes data processing. Here, the data is transformed and consolidated for its intended analytical use case. This phase can involve the following tasks:

- Filtering, cleansing, de-duplicating, validating, and authenticating the data.

- Performing calculations, translations, or summarizations based on the raw data. This can include changing row and column headers for consistency, converting currencies or other units of measurement, editing text strings, and more.
- Conducting audits to ensure data quality and compliance
- Removing, encrypting, or protecting data governed by industry or governmental regulators
- Formatting the data into tables or joined tables to match the schema of the target data warehouse.

Load

In this last step, the transformed data is moved from the staging area into a target data warehouse. Typically, this involves an initial loading of all data, followed by periodic loading of incremental data changes and, less often, full refreshes to erase and replace data in the warehouse.

For most organizations that use ETL, the process is automated, well-defined, continuous and batch-driven. Typically, ETL takes place during off-hours when traffic on the source systems and the data warehouse is at its lowest.

ETL and other data integration methods

ETL and ELT are just two data integration methods, and there are other approaches that are also used to facilitate data integration workflows. Some of these include:

- **Change Data Capture (CDC)** identifies and captures only the source data that has changed and moves that data to the target system. CDC can be used to reduce the resources required during the ETL "extract" step; it can also be used independently to move data that has been transformed into a data lake or other repository in real time.

- **Data replication** copies changes in data sources in real time or in batches to a central database. Data replication is often listed as a data integration method. In fact, it is most often used to create backups for disaster recovery.

- **Data virtualization** uses a software abstraction layer to create a unified, integrated, fully usable view of data—without physically copying, transforming or loading the source data to a target system. Data virtualization functionality enables an organization to create virtual data warehouses, data lakes and data marts from the same source data for data storage without the expense and complexity of building and managing separate platforms for each. While data virtualization can be used alongside ETL, it is increasingly seen as an alternative to ETL and to other physical data integration methods.

- **Stream Data Integration (SDI)** is just what it sounds like—it continuously consumes data streams in real time, transforms them, and loads them to a target system for analysis. The key word here is continuously. Instead of integrating snapshots of data extracted from sources at a given time, SDI integrates data constantly as it becomes available. SDI enables a data store for powering analytics, machine learning and real-time applications for improving customer experience, fraud detection and more.

Create data model from SQL

In order to demonstrate how you can use the strategy defined in this document to build data model we will be using below mentioned queries.

```
/*

 Query to find all events and occurrences in one year

*/

SELECT eve.eventname,count(*) as total_occurences

FROM Event eve

inner join

Date_cal dat

on

eve.dateid = dat.dateid

where dat.caldate between '2022-01-01' and '2022-12-31'

group by eve.eventname

;/*

Query to find all events on 30th Oct 2022 in Bangalore city

*/

SELECT eve.eventname,

eve.starttime,

ven.venuename,

ven.venuecity,

ven.venuecitypop,

ven.venuestate,
dat.day,
dat.holiday,
dat.caldate
FROM Event eve
inner join
Venue ven
on
eve.venueid = ven.venueid
inner join
Date_cal dat
on
eve.dateid = dat.dateid
where ven.venuecity = 'Bangalore'
and dat.caldate = '2022-10-30'
;/*
Query to find the total number of tickets sold, price paid, and commission paid for each event and then determine profit for each event.
*/
Select  sales_event.eventname , sales_event.tot_Qtysold, sales_event.tot_Pricepaid - sales_event.tot_commission as tot_profit
from(
select
sum(sal.Qtysold) as tot_Qtysold,
sum(sal.Pricepaid) as tot_Pricepaid,
sum(sal.commission) as tot_commission,
eve.eventname
FROM
Sales sal
inner join
Event eve
on
sal.eventid = eve.eventid
group by eve.eventname
)sales_event
order by 2 desc;/*
Query to find least popular events in history
*/SELECT  cat.catname,eve.eventname,sum(sal.Qtysold) as tot_tickets_sold
FROM Category cat
inner join
Event eve on
cat.catid = eve.catid
inner join
Sales sal on sal.eventid = eve.eventid
GROUP BY cat.catname,eve.eventname
ORDER BY 3 asc;
```

SQL into Data Model – The Solution

The solution uses python packages with custom code to parse SQL and then generate **DBML output**. We will use the DBML output generated into the **web app to visualise it** and save as data model.

Step 1:

Install the **sqlparse package**

```
pip install sqlparse
```

Step 2: Install the sql-metadata package

```
pip install sql-metadata
```

Step 3: Run the below python command

```
import sqlparse

from sql_metadata import Parser

strsql = """

/*

 Query to find all events and occurrences in one year

*/

SELECT eve.eventname,count(*) as total_occurences

FROM Event eve

inner join

Date_cal dat

on

eve.dateid = dat.dateid

where dat.caldate between '2022-01-01' and '2022-12-31'

group by eve.eventname;


/*Query to find all events on 30th Oct 2022 in Bangalore city*/

SELECT
eve.eventname,eve.starttime,ven.venuename,ven.venuecity,ven.venuecitypop,ven.venuestate,dat.day,dat.holiday,dat.caldate

FROM Event eve

inner join  Venue ven on

eve.venueid = ven.venueid inner join Date_cal dat

on

eve.dateid = dat.dateid

where ven.venuecity = 'Bangalore'

and dat.caldate = '2022-10-30';

/*
```

```
Query to find the total number of tickets sold, price paid, and commission paid for each event and then determine profit for each event.

*/

Select  sales_event.eventname , sales_event.tot_Qtysold, sales_event.tot_Pricepaid - sales_event.tot_commission as tot_profit

from

(

select

sum(sal.Qtysold) as tot_Qtysold,

sum(sal.Pricepaid) as tot_Pricepaid,

sum(sal.commission) as tot_commission,

eve.eventname

FROM

Sales sal

inner join

Event eve

on

sal.eventid = eve.eventid

group by eve.eventname

)sales_event

order by 2 desc;

/*

Query to find least popular events in history

*/

SELECT

cat.catname,

eve.eventname,

sum(sal.Qtysold) as tot_tickets_sold

FROM Category cat
```

```
inner join
Event eve on
cat.catid = eve.catid
inner join
Sales sal
on
sal.eventid = eve.eventid
GROUP BY cat.catname,eve.eventname
ORDER BY 3 asc;
"""
statements = sqlparse.split(strsql)
all_columns = list()
all_tables = list()
all_joins = ""

for stmt in statements:
    parser = Parser(stmt)
    for key,value in parser.columns_dict.items():
        if value is not None:
            all_columns.append(value)
    for j in parser.tables:
        all_tables.append(j.split())
    i=0
    join_cond=""
    if 'join' in parser.columns_dict:
        for j in parser.columns_dict['join']:
            if i%2 == 0:
                join_cond = join_cond + "" + "Ref: "+j
            else:
                join_cond = join_cond + " - " + j +"|"
            i = i + 1
        all_joins = all_joins + join_cond
all_joins_list = all_joins.split("|")
all_joins_list = list(set(all_joins_list))
all_joins = '|'.join(all_joins_list)
all_columns = sum(all_columns, [])
all_columns = list(set(all_columns))
all_tables = sum(all_tables, [])
all_tables = list(set(all_tables))
i=0
table_def=""
for j in all_tables:
    table_struct=""
    table_struct = "Table "+j+" {\nrowid bigint"
    for i in all_columns:
        if i.startswith(j+'.') :
            table_struct = table_struct +"\n"+ i.split('.')[-1] + " string "
    table_def = table_def + table_struct + "\n}\n|"
for i in table_def[:-1].split("|"):
    print (i)
for i in all_joins[1:].split("|"):
    print (i)
```

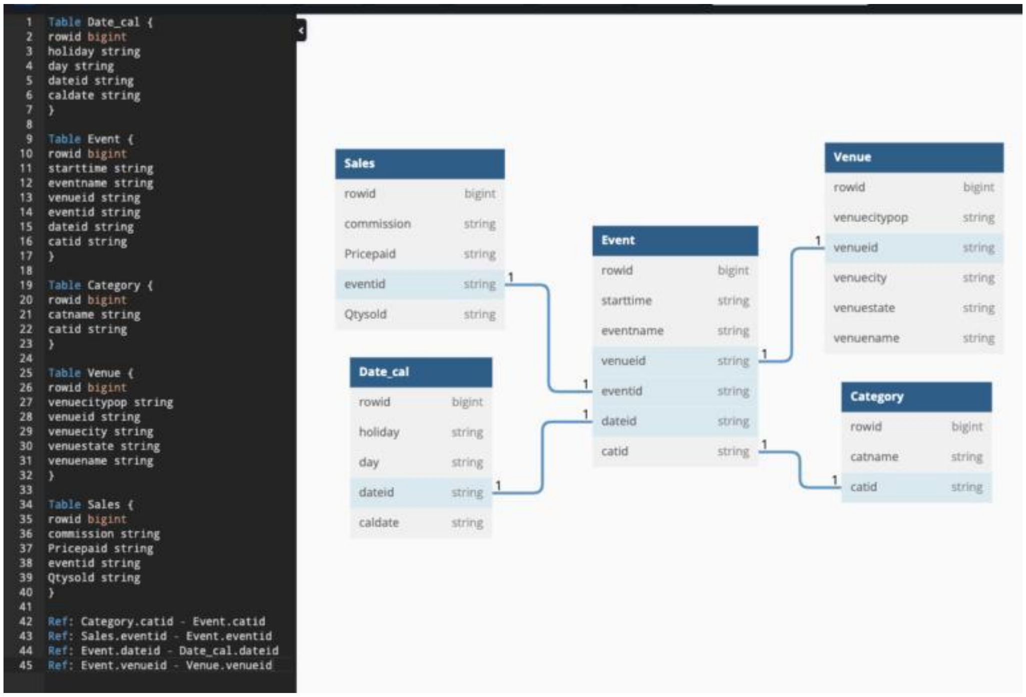
-Change the SQL as per your requirement in the Python code.

Step 4: Copy the DBML output generated

```
Table Date_cal {
rowid bigint
holiday string
day string
dateid string
caldate string
}
Table Event {
rowid bigint
starttime string
eventname string
venueid string
eventid string
dateid string
catid string
}
Table Category {
rowid bigint
catname string
catid string
}

Table Venue {
rowid bigint
venuecitypop string
venueid string
venuecity string
venuestate string
venuename string
}
```

```
Table Sales {
rowid bigint
commission string
Pricepaid string
eventid string
Qtysold string
}
Ref: Category.catid - Event.catid
Ref: Sales.eventid - Event.eventid
Ref: Event.dateid - Date_cal.dateid
Ref: Event.venueid - Venue.venueid
```

Step 5: Paste the DBML output



```
Table Date_cal {
rowid bigint
holiday string
day string
dateid string
caldate string
}

Table Event {
rowid bigint
starttime string
eventname string
venueid string
eventid string
dateid string
catid string
}

Table Category {
rowid bigint
catname string
catid string
}

Table Venue {
rowid bigint
venuecitypop string
venueid string
venuecity string
venuestate string
venuename string
}

Table Sales {
rowid bigint
commission string
Pricepaid string
eventid string
Qtysold string
}

Ref: Category.catid - Event.catid
Ref: Sales.eventid - Event.eventid
Ref: Event.dateid - Date_cal.dateid
Ref: Event.venueid - Venue.venueid
```

Data Exploration Techniques (IBM Db2 Warehouse):

1.SQL Queries:

Data architects and analysts can use SQL queries to explore and analyze data stored in IBM Db2 Warehouse. SQL's powerful analytical functions enable complex data exploration.

2.IBM Cognos Analytics:

IBM Cognos Analytics is a business intelligence and data visualization tool that integrates seamlessly with Db2 Warehouse. It allows for the creation of interactive dashboards and reports.

3.Jupyter Notebooks:

Data scientists and analysts can use Jupyter notebooks with Python or R to perform in-depth data analysis and visualization using data from Db2 Warehouse.

Actionable Insights (IBM Db2 Warehouse):

1. **Real-time Analytics:**

   Db2 Warehouse supports near-real-time data updates, enabling data architects to provide real-time or near-real-time analytics to stakeholders.

2. **Advanced Analytics:**
   With Db2 Warehouse, data architects can incorporate advanced analytics, machine learning models, and statistical analysis into the data exploration process, delivering deeper insights.

3. **Scalability and Performance:**
   Db2 Warehouse's scalable architecture ensures that it can handle large volumes of data and complex queries, facilitating the delivery of actionable insights without performance bottlenecks.

4. **Integration with IBM Watson:**
   For organizations using IBM Watson, integration with Db2 Warehouse allows for AI-powered insights and predictions, enhancing the value of the data warehouse.

In summary, IBM Db2 Warehouse offers a robust data warehousing solution with a structured data warehouse, efficient data integration, ETL capabilities, and powerful data exploration techniques. It empowers data architects to deliver actionable insights by providing real-time analytics, advanced analytics, and seamless integration with BI and AI tools.