Rakamin Academy X id/x partners

Creating a Data Warehouse and Stored Procedures for One of ID/X Partners Clients

ID/X Partners - Data Engineer

Presented by
Ahmad Faishal Akbar

Rakamin Academy

Ahmad Faishal Akbar

Data Enthusiast

Bandung City, West Java

ahmadfaishal9611@gmail.com

https://www.linkedin.com/in/ahmad-faishal-akbar-546a55194/

https://github.com/ahmadfaishal9611

Hi, I'm **Ahmad**. I graduated from **Institut Teknologi Bandung** majoring **Geodesy and Geomatics Engineering**. I have a strong interest in **big data analytics** and problem-solving in business. I have experience working with **SQL**, **Python**, **Google Data Studio**, and **Tableau**. Feel free to see my **portfolio** here: https://bit.ly/42S4Gwx

Courses and Certification

Rakamin Academy

Data Analyst with Python | certificate link	**<Jan, 2023>**
Business Intelligence Analyst | certificate link	**<Nov, 2022>**
Data Analysis in Excel | certificate link	**<Des, 2022>**
Data Analysis in Spreadsheets | certificate link	**<Des, 2022>**

About Company

ID/X Partners is a company that **provides consulting services** that specializes in **utilizing data analytic and decisioning (DAD) solutions** combined with an integrated risk management and marketing discipline to help clients optimize the portfolio profitability and business process.

Project Description

One of the clients of ID/X Partners company, operating in the banking industry, has a need to **build a Data Warehouse from various different data sources** stored within their system. The current issue they are facing is difficulty in **extracting data from various sources (Excel, CSV, databases) simultaneously**, leading to delays in their data reporting and analysis processes.

As a Data Engineer, there are several tasks you need to perform to **optimize** the **ETL process** in the company:

1. **Create a new database named DWH**, which will serve as the new Data Warehouse. Then, create three **dimension tables: DimAccount, DimCustomer, DimBranch**, and **one fact table: FactTransaction**. (Make sure to provide **primary keys** and **foreign keys** in each table.)
2. **Create an ETL job** in Talend to **move data from sources to all Dimension tables**. Specifically for the DimCustomer table, the column format to be stored is CustomerID, CustomerName, Address, CityName, StateName, Age, Gender, Email. All data from these columns are converted to **uppercase** except for the CustomerID, Age, and Email columns. (Ensure that **column naming follows PascalCase** conventions, for example, account_id = AccountID).

Project Description

3. **Create an ETL job to combine transaction data** (from Excel, CSV, and database sources) into **one table, FactTransaction**. It is important to note that since the three files come from different sources, **ensure** that there are **no duplicate rows** in the FactTransaction table.
4. **Create two Stored Procedures (SP)** with parameters to help them quickly obtain data summaries:
    - **DailyTransaction** (to calculate the number of transactions along with their total amounts each day). The columns to be displayed are **Date, TotalTransactions, TotalAmount**. Then create **two parameters: start_date** and **end_date** so that when running this SP by entering these parameters, it will display data according to the date range we input.
    - **BalancePerCustomer** (to determine the remaining balance per customer). The columns to be displayed are **CustomerName, AccountType, Balance, CurrentBalance**. Create **a parameter named name** so that when running this SP by entering the name of one of the customers, it will display data according to the input. Then make sure to **filter accounts** with an **active status.**

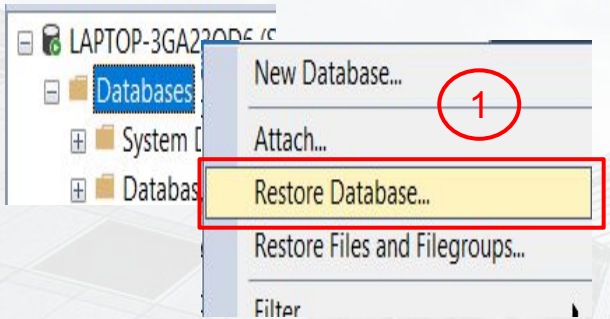
Rakamin Academy

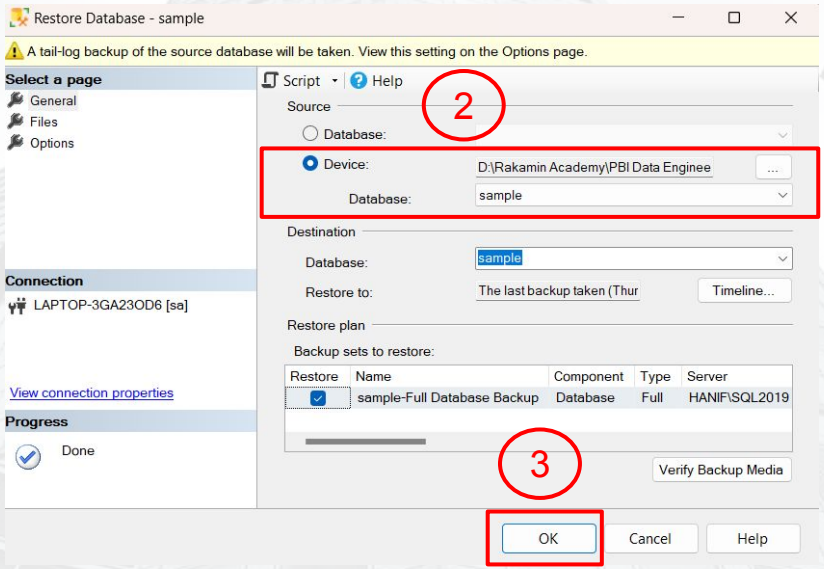
Tools

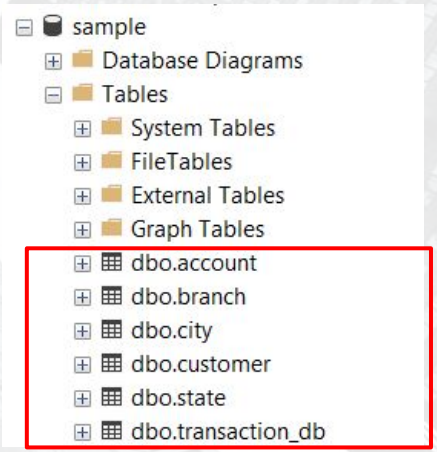
1. Data Warehouse Creation

Restore Database

Before We create a Data Warehouse for storing the ETL result, We should restore the database source (**sample.bak**) first.







Result

1. Data Warehouse Creation

Restore Database

These are the few first rows of each table in **sample** database (database source).

	account_id	customer_id	account_type	balance	date_opened	status
1	1	1	saving	1500000	2020-05-01 09:00:00	active
2	2	2	saving	500000	2020-06-01 10:00:00	active
3	3	1	checking	25000000	2020-06-21 09:00:00	active
4	4	3	checking	4500000	2021-06-24 11:00:00	terminated

dbo.account table

dbo.branch table

	branch_id	branch_name	branch_location
1	1	KC Jakarta	Jl. Gatot Subroto No 13
2	2	KC Bogor	Jl. Padjajaran No 43
3	3	KC Depok	Jl. Raya Sawangan No 34

dbo.city table

	city_id	city_name	state_id
1	1	Cilincing	1
2	2	Kelapa Gading	1
3	3	Tanjung Priok	1

dbo.state table

	state_id	state_name
1	1	Jakarta Utara
2	2	Jakarta Selatan
3	3	Jakarta Pusat

	customer_id	customer_name	address	city_id	age	gender	email
1	1	Shelly Juwita	Jl. Boulevard No. 31	2	25	female	shelly@gmail.com
2	2	Bobi Rinaldo	Jl. Mangga No. 1	3	31	male	Bobi@gmail.com
3	3	Adam Malik	Jl. Kincir Angin No...	5	23	male	Adam@gmail.com

dbo.customer table

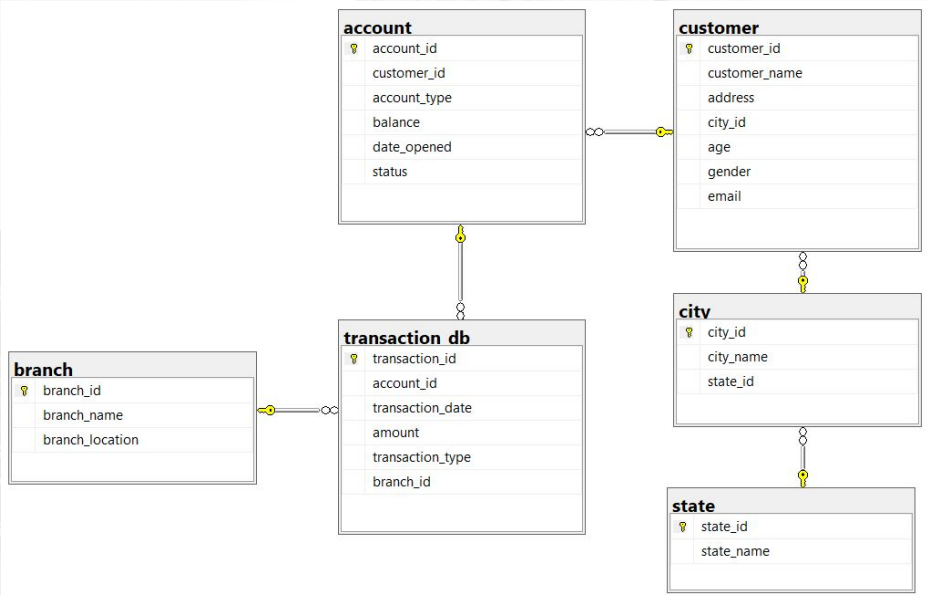
	transaction_id	account_id	transaction_date	amount	transaction_type	branch_id
1	1	1	2024-01-17 09:10:00	100000	Deposit	1
2	2	2	2024-01-17 10:10:00	1000000	Deposit	1
3	3	3	2024-01-18 08:30:00	10000...	Transfer	1

dbo.transaction_db table

1. Data Warehouse Creation

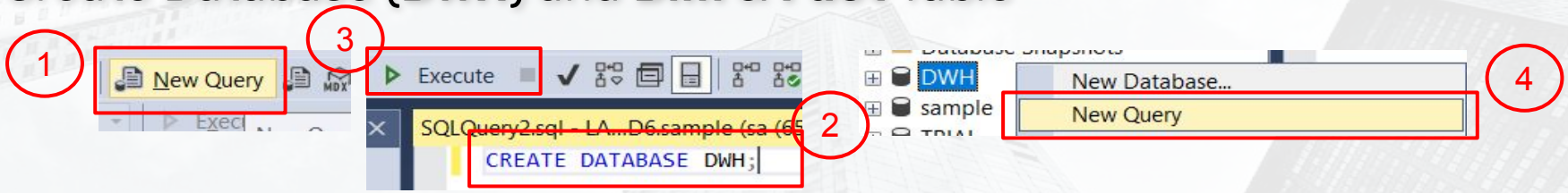
Restore Database

This is a Entity Relationship Diagram (ERD) of **sample** database.



1. Data Warehouse Creation

Create Database (**DWH**) and **Dim** & **Fact** Table




```sql
CREATE TABLE DimCustomer (
    CustomerID INT PRIMARY KEY NOT NULL,
    CustomerName VARCHAR (50) NOT NULL,
    Address VARCHAR (50) NOT NULL,
    CityName VARCHAR (50) NOT NULL,
    StateName VARCHAR (50) NOT NULL,
    Age INT NOT NULL,
    Gender VARCHAR (50) NOT NULL,
    Email VARCHAR (50) NOT NULL
);

CREATE TABLE DimAccount (
    AccountID INT PRIMARY KEY NOT NULL,
    CustomerID INT NOT NULL,
    AccountType VARCHAR (50) NOT NULL,
    Balance INT NOT NULL,
    DateOpened DATETIME NOT NULL,
    Status VARCHAR (50) NOT NULL,
    FOREIGN KEY (CustomerID) REFERENCES DimCustomer(CustomerID)
);
```

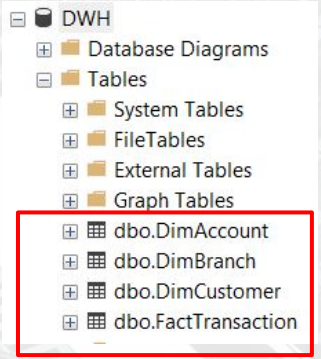
5

```sql
CREATE TABLE DimBranch (
    BranchID INT PRIMARY KEY NOT NULL,
    BranchName VARCHAR (50) NOT NULL,
    BranchLocation VARCHAR (50) NOT NULL
);

CREATE TABLE FactTransaction (
    TransactionID INT PRIMARY KEY NOT NULL,
    AccountID INT NOT NULL,
    TransactionDate DATETIME NOT NULL,
    Amount INT NOT NULL,
    TransactionType VARCHAR (50) NOT NULL,
    BranchID INT NOT NULL,
    FOREIGN KEY (AccountID) REFERENCES DimAccount(AccountID),
    FOREIGN KEY (BranchID) REFERENCES DimBranch(BranchID)
);
```

6

Execute

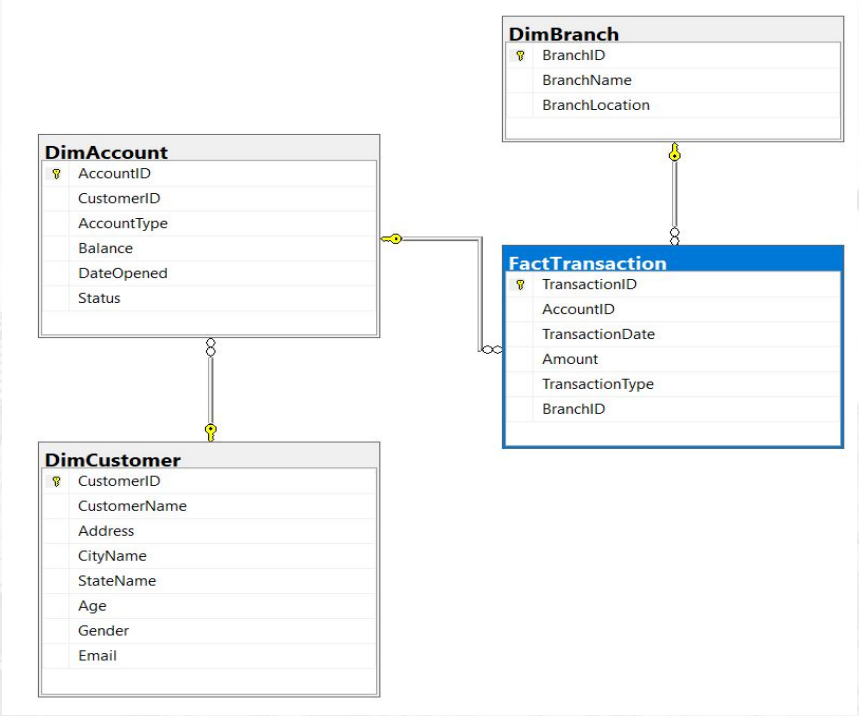


Result

1. Data Warehouse Creation

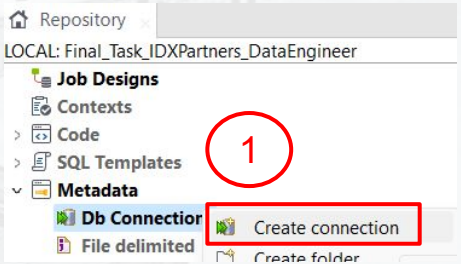
Create Database (**DWH**) and **Dim** & **Fact** Table

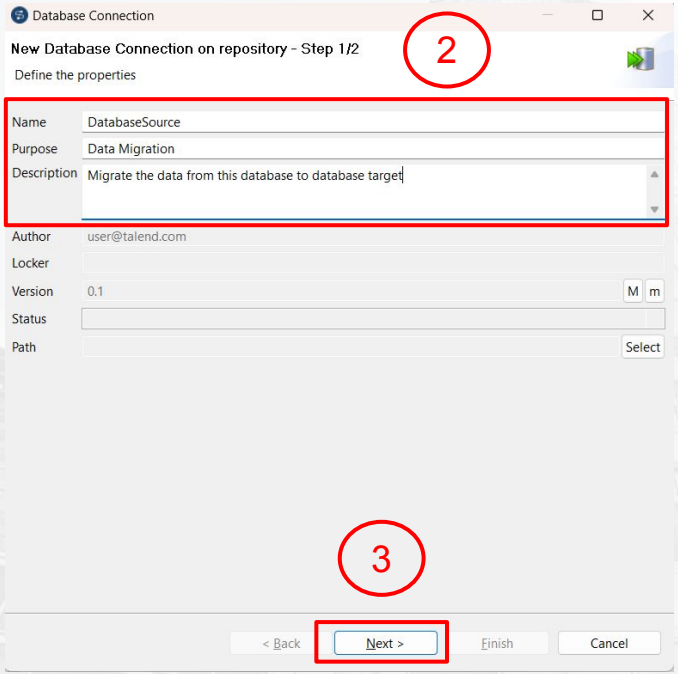
This is a Entity Relationship Diagram (ERD) of **DWH** database.

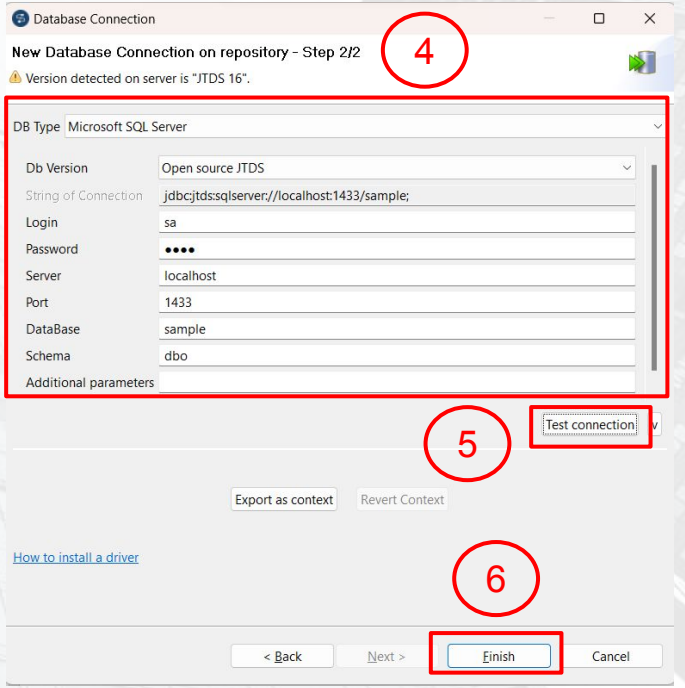


2. Create ETL Job for Dimension Table

Create Database Connection to **sample** database (database source)

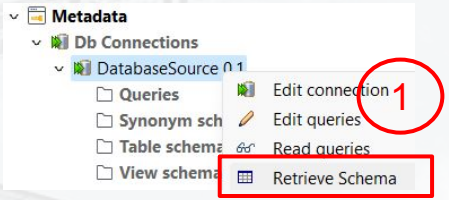


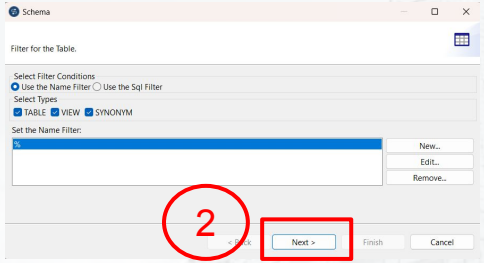


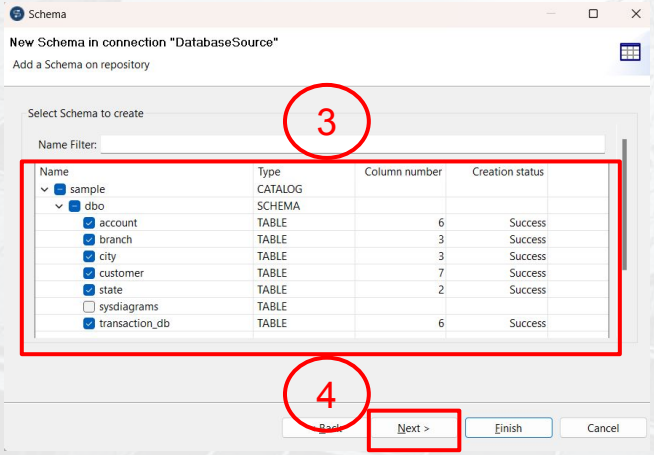


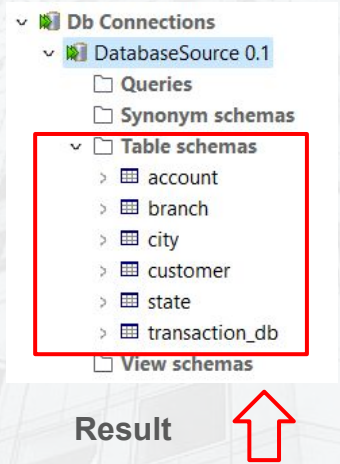
2. Create ETL Job for Dimension Table

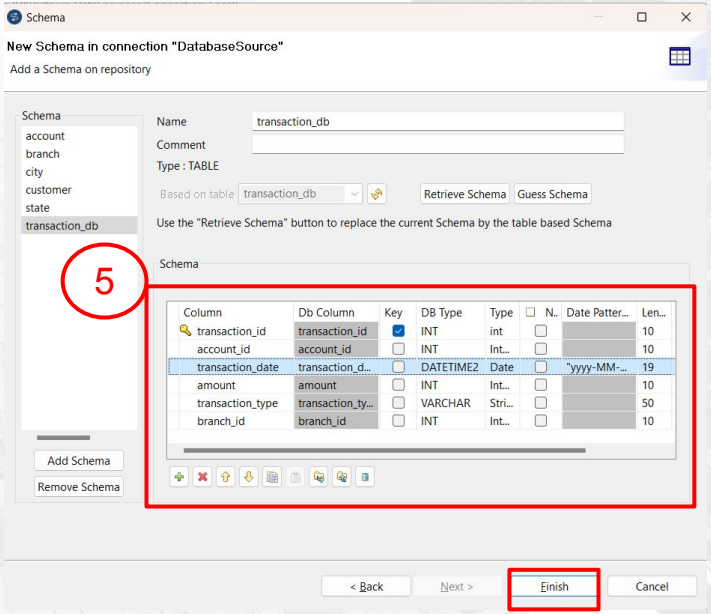
Retrieve Schema from **DatabaseSource**





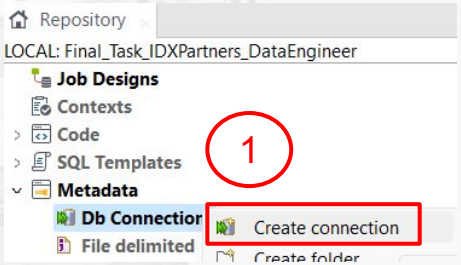


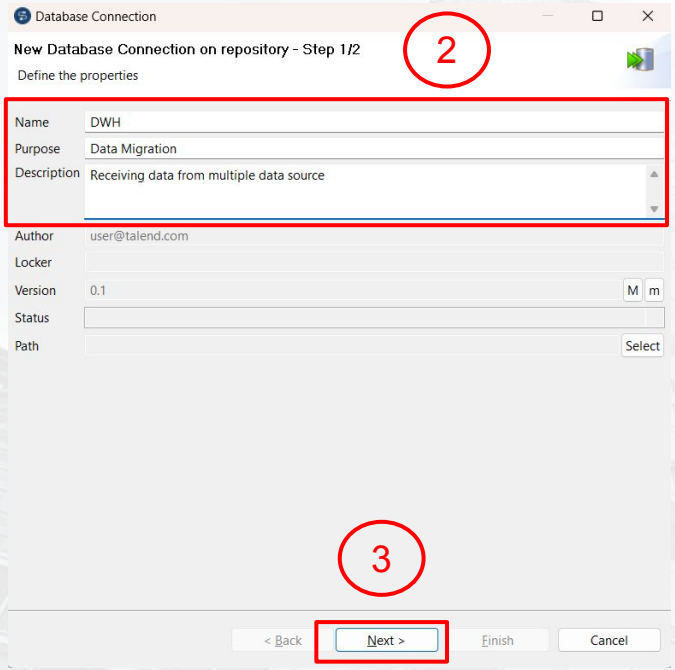


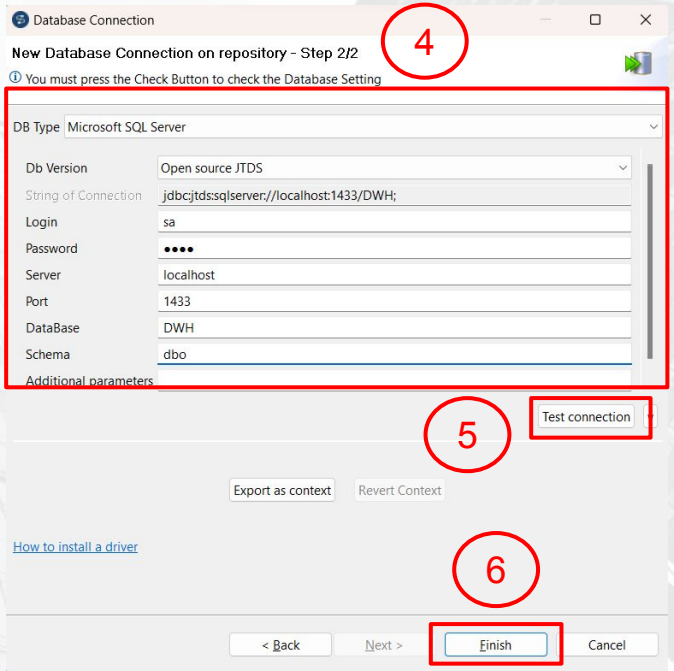


2. Create ETL Job for Dimension Table

Create Database Connection to **DWH** database (database target)

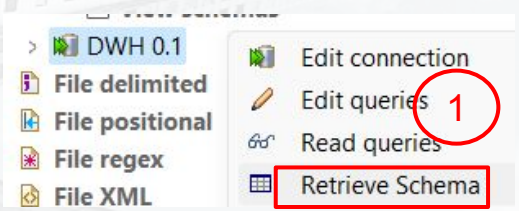


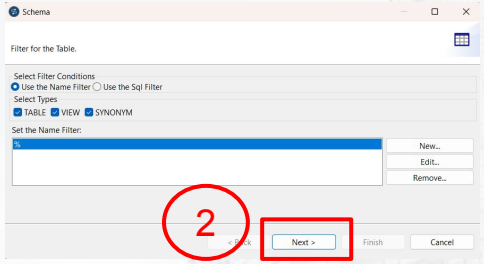


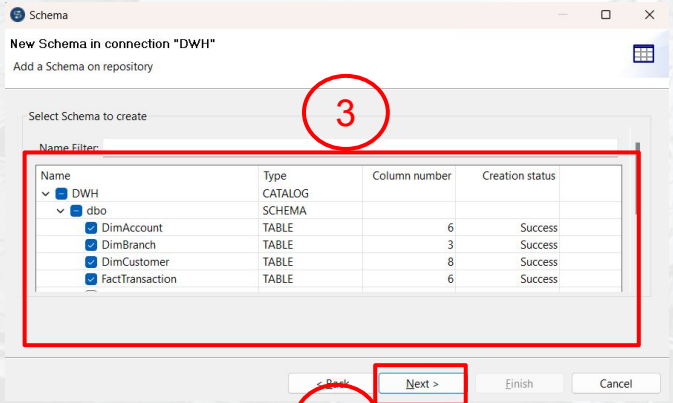


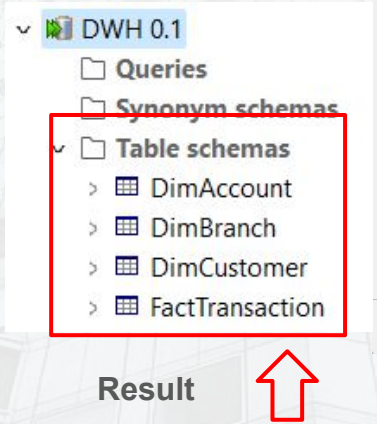
2. Create ETL Job for Dimension Table

Retrieve Schema from **DWH**

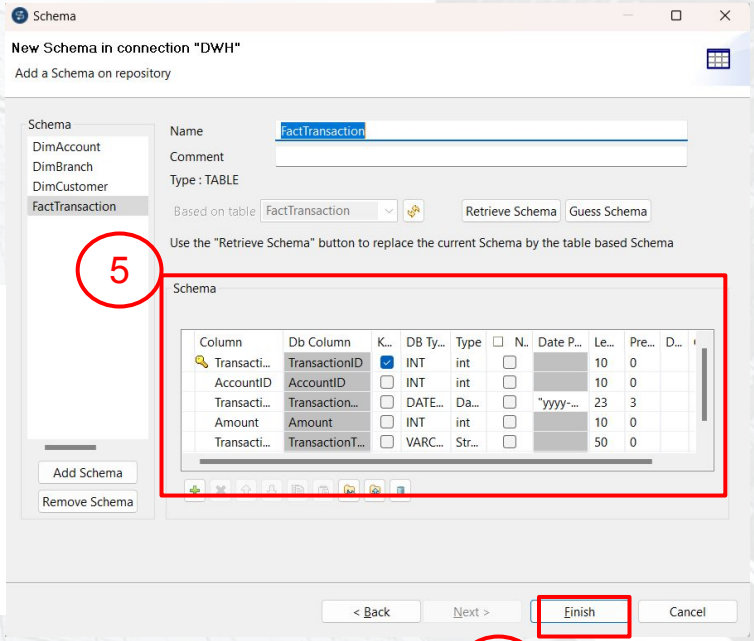






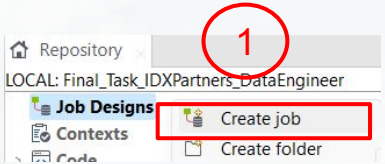


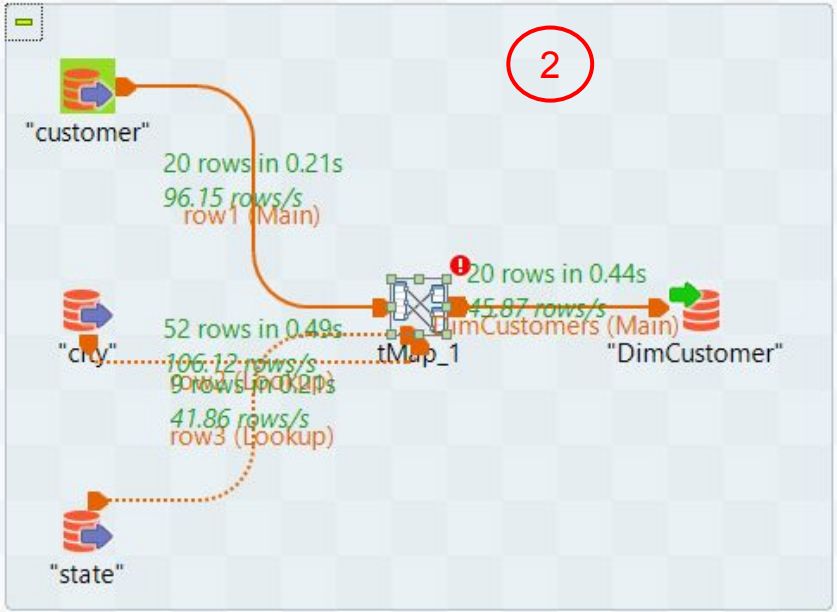
Result



2. Create ETL Job for Dimension Table

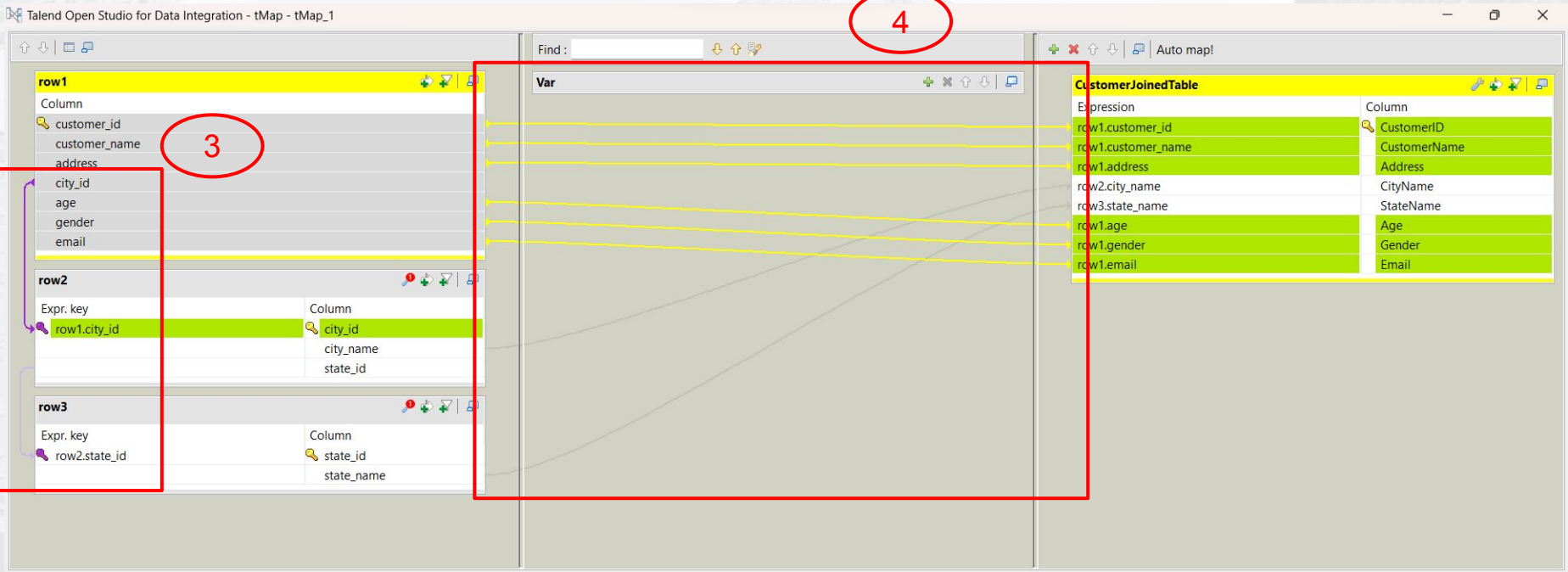
Create ETL Job (DimCustomer)





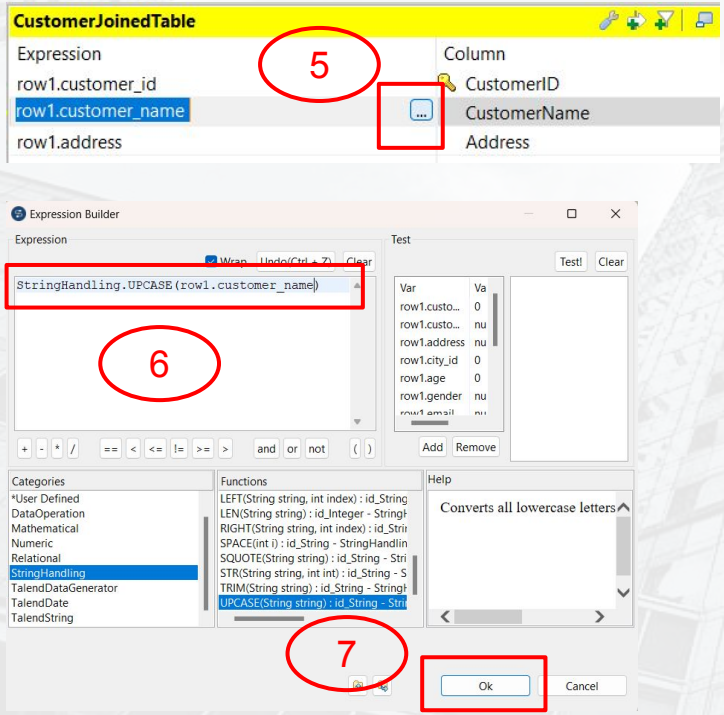
2. Create ETL Job for Dimension Table

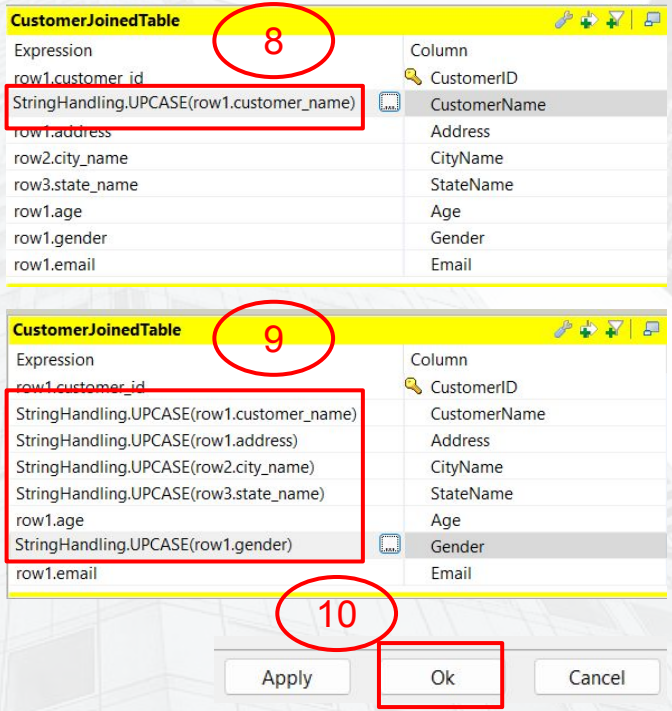
Create ETL Job (DimCustomer)



2. Create ETL Job for Dimension Table

Create ETL Job (DimCustomer)





2. Create ETL Job for Dimension Table

Create ETL Job (DimAccount and Dim Branch)

2. Create ETL Job for Dimension Table

Result

SQLQuery1.sql - L...3OD6.DWH (sa (95))*

```
SELECT * FROM DimAccount;
SELECT * FROM DimBranch;
SELECT * FROM DimCustomer;
```

Results | Messages

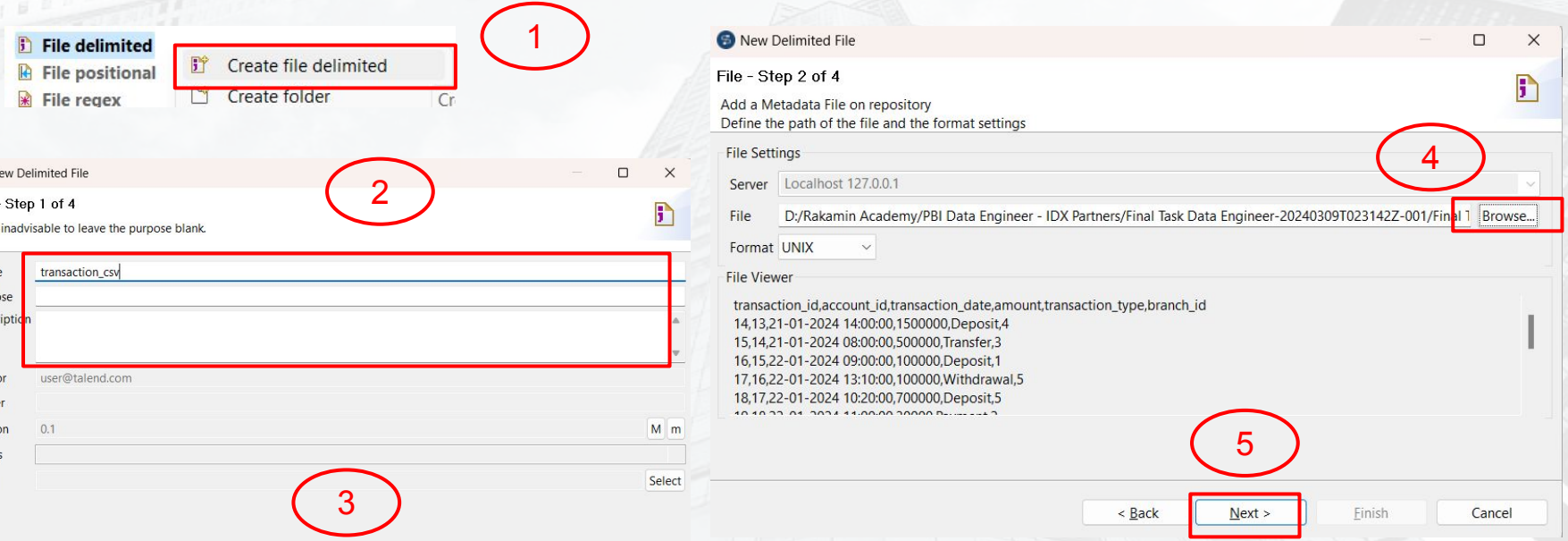
	AccountID	CustomerID	AccountType	Balance	DateOpened	Status
1	1	1	saving	1500000	2020-05-01 09:00:00.000	active
2	2	2	saving	500000	2020-06-01 10:00:00.000	active
3	3	1	checking	25000000	2020-06-21 09:00:00.000	active
4	4	3	checking	4500000	2021-06-24 11:00:00.000	terminated

	BranchID	BranchName	BranchLocation
1	1	KC Jakarta	Jl. Gatot Subroto No 13
2	2	KC Bogor	Jl. Padjajaran No 43
3	3	KC Depok	Jl. Raya Sawangan No 34
4	4	KC Tangerang	Jl. Cisauk No 50

	CustomerID	CustomerName	Address	CityName	StateName	Age	Gender	Email
1	1	SHELLY JUWITA	JL. BOULEVARD NO. 31	KELAPA GADING	JAKARTA UTARA	25	FEMALE	shelly@gmail.com
2	2	BOBI RINALDO	JL. MANGGA NO. 1	TANJUNG PRIOK	JAKARTA UTARA	31	MALE	Bobi@gmail.com
3	3	ADAM MALIK	JL. KINCIR ANGIN NO....	PADEMANGAN	JAKARTA UTARA	23	MALE	Adam@gmail.com
4	4	SUSI RAHMA	JL. KENANGA NO. 11	CILANDAK	JAKARTA SELA	45	FEMALE	Susi@gmail.com

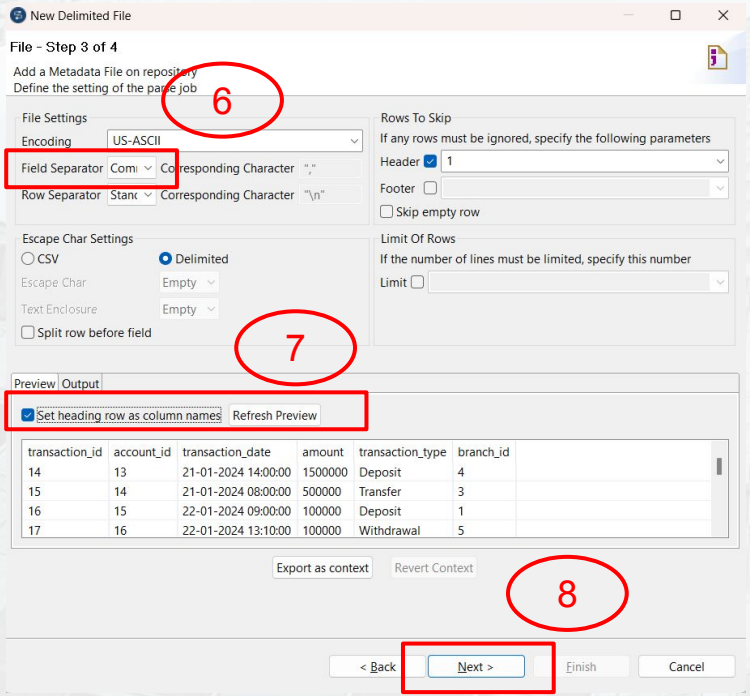
3. Create ETL Job for Fact Table

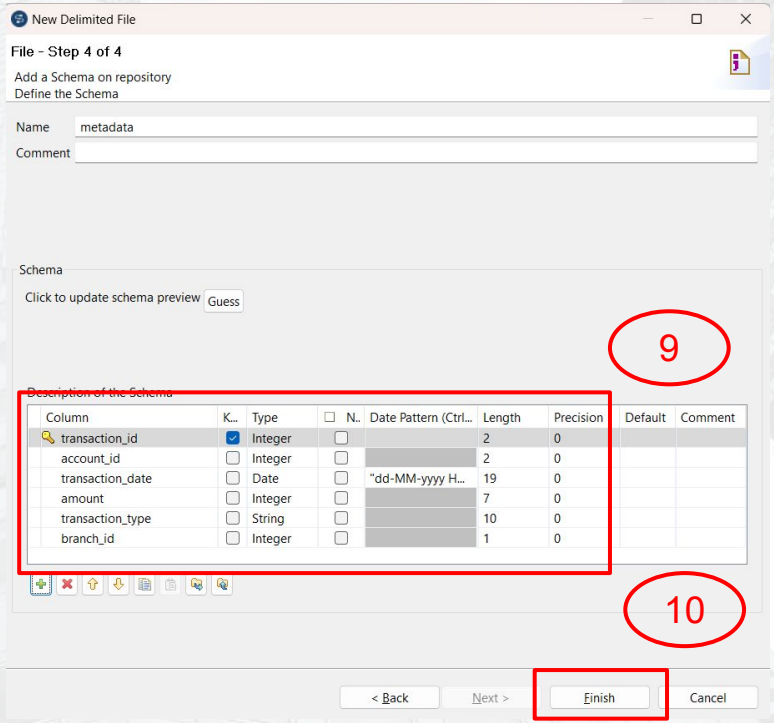
Create file delimited (**transaction_csv**)



3. Create ETL Job for Fact Table

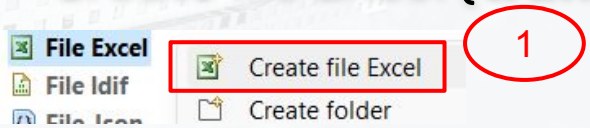
Create file delimited (**transaction_csv**)

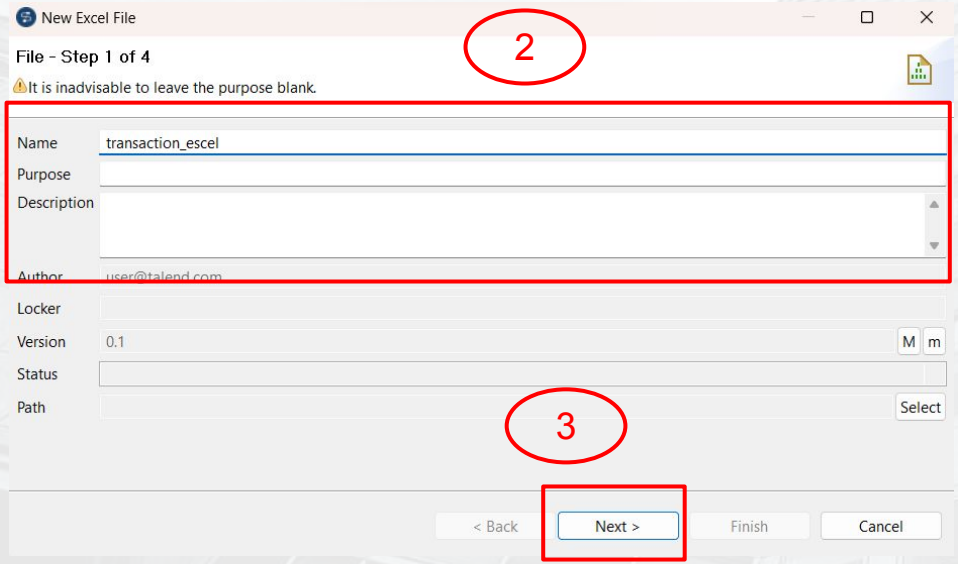


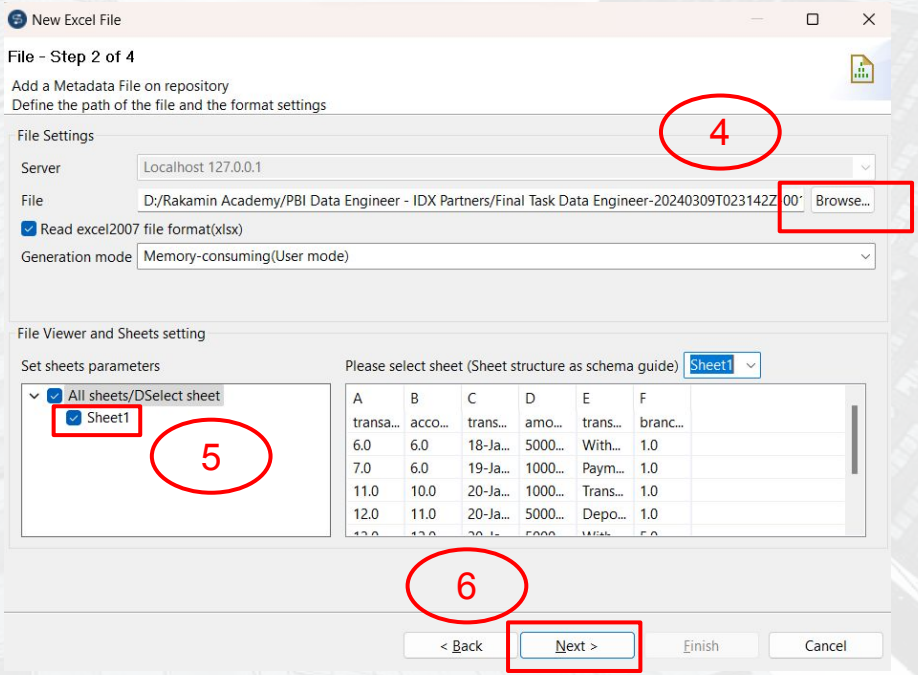


3. Create ETL Job for Fact Table

Create file Excel (**transaction_excel**)

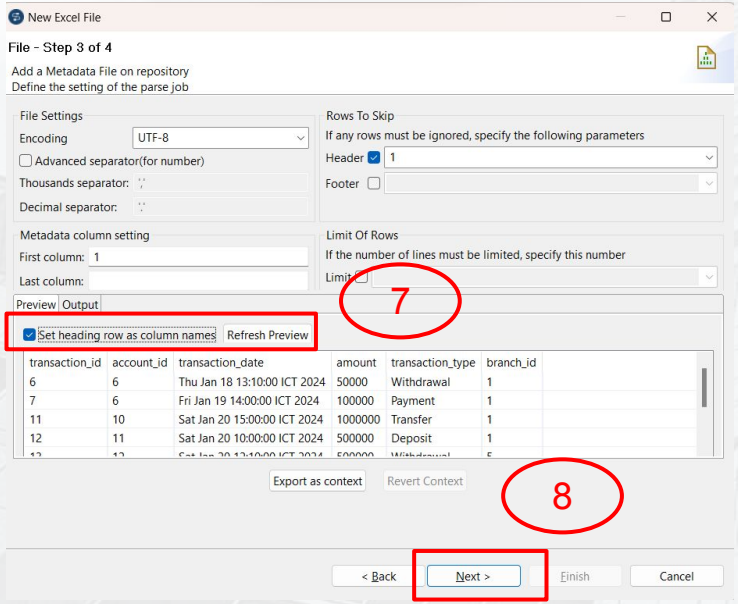


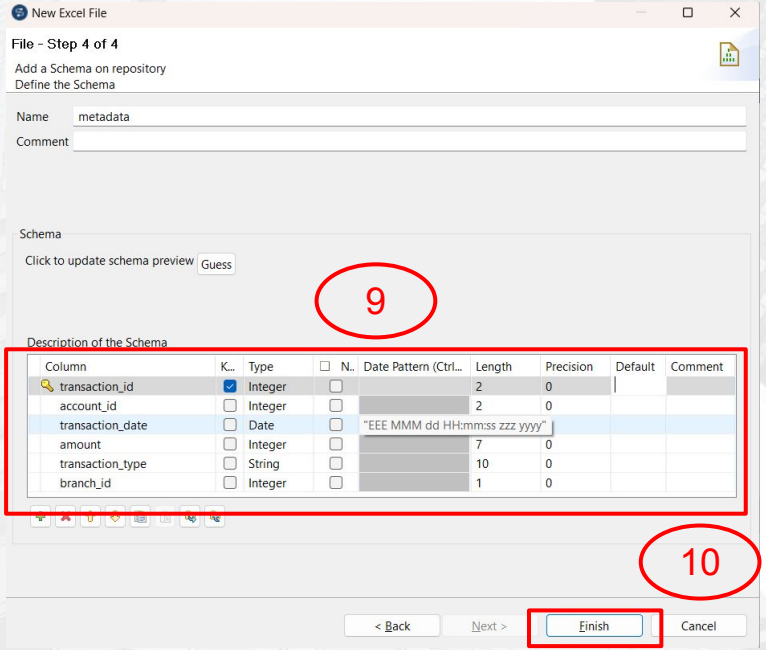




3. Create ETL Job for Fact Table

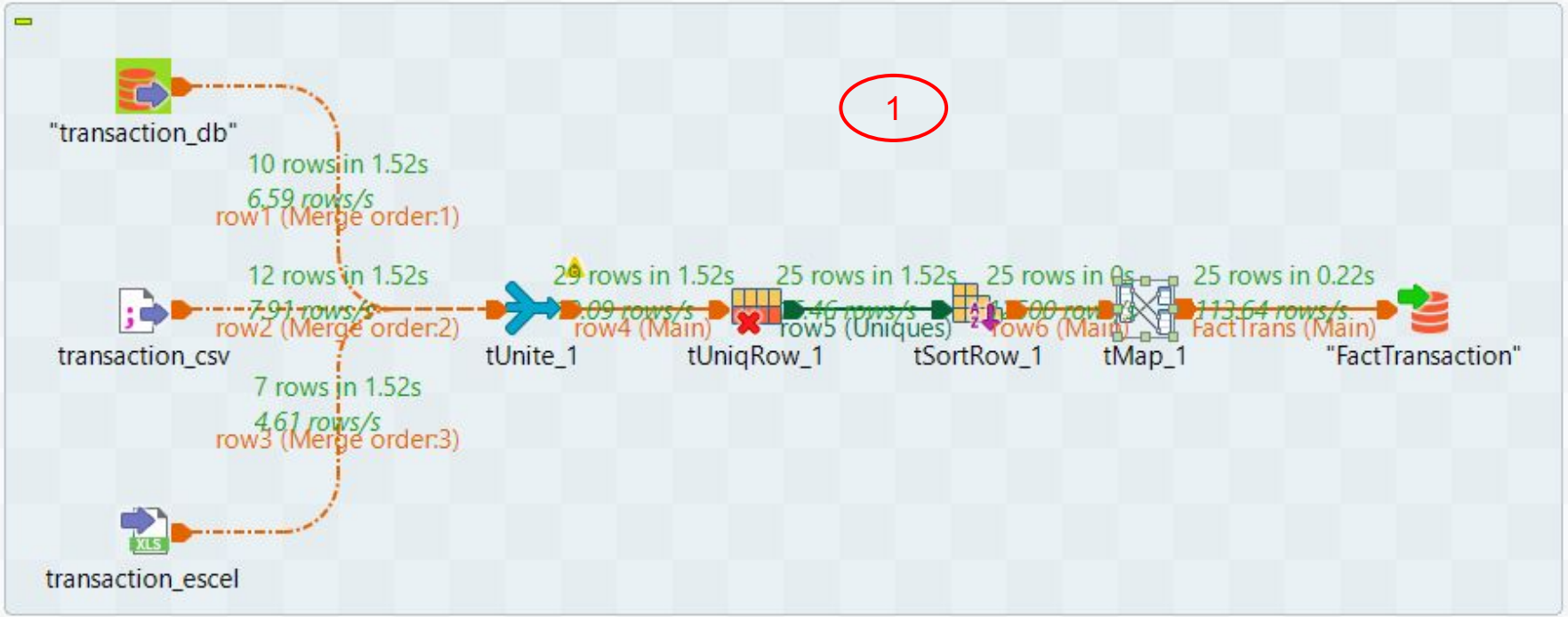
Create file Excel (**transaction_excel**)





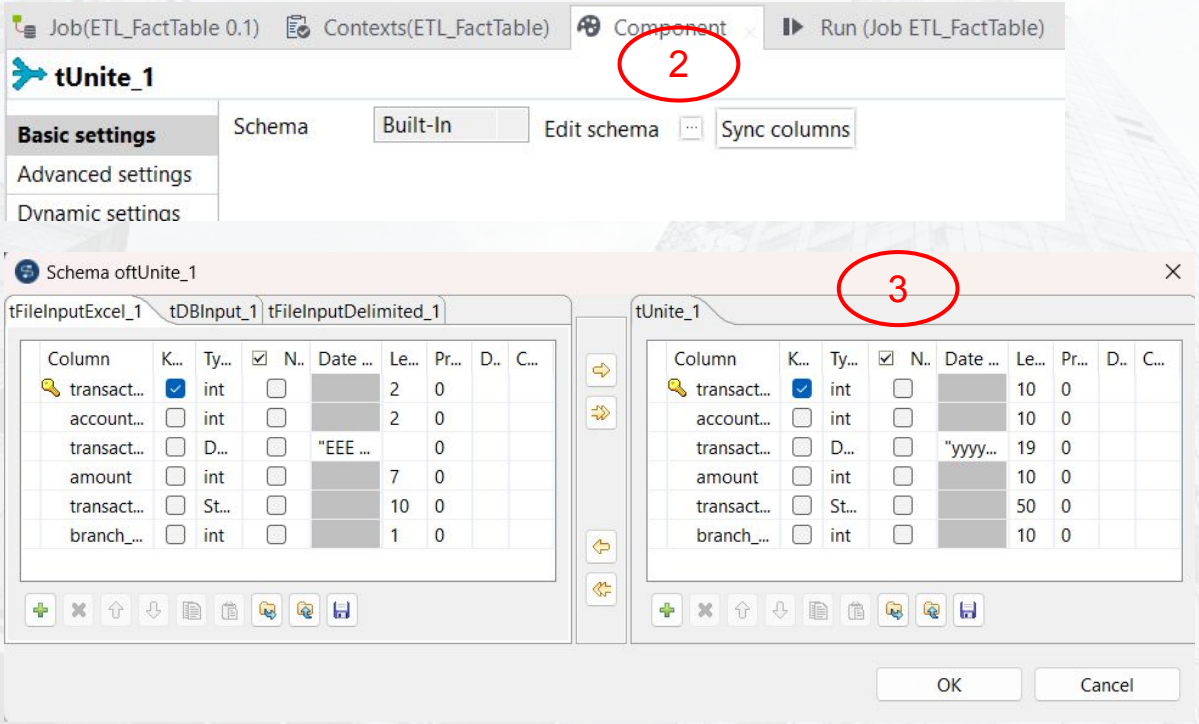
3. Create ETL Job for Fact Table

Create ETL Job



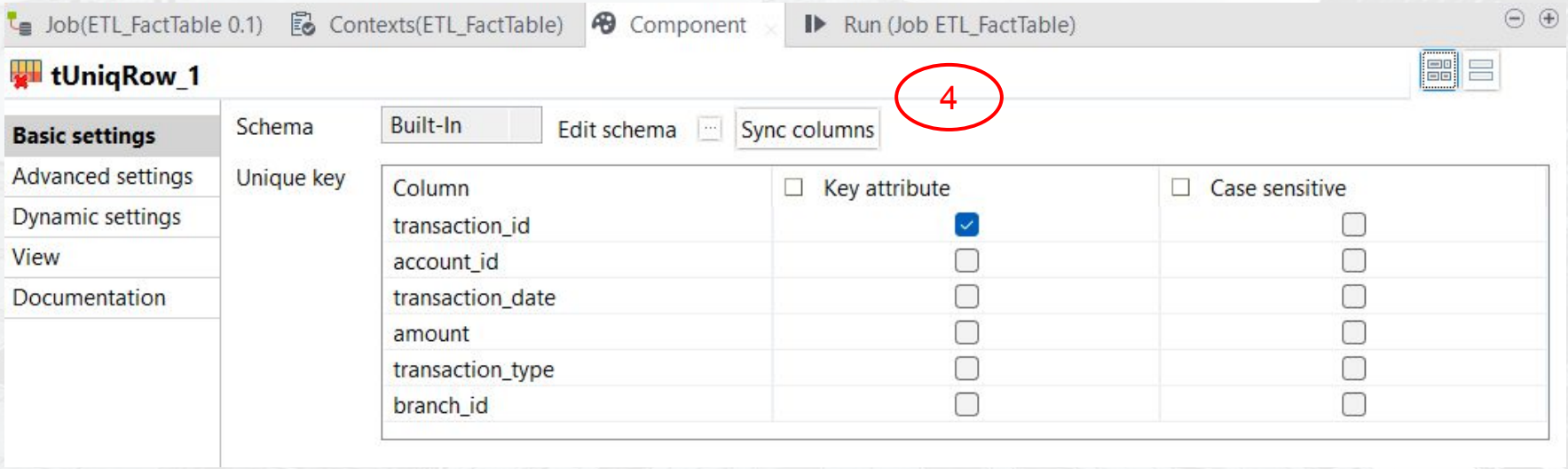
3. Create ETL Job for Fact Table

Create ETL Job



3. Create ETL Job for Fact Table

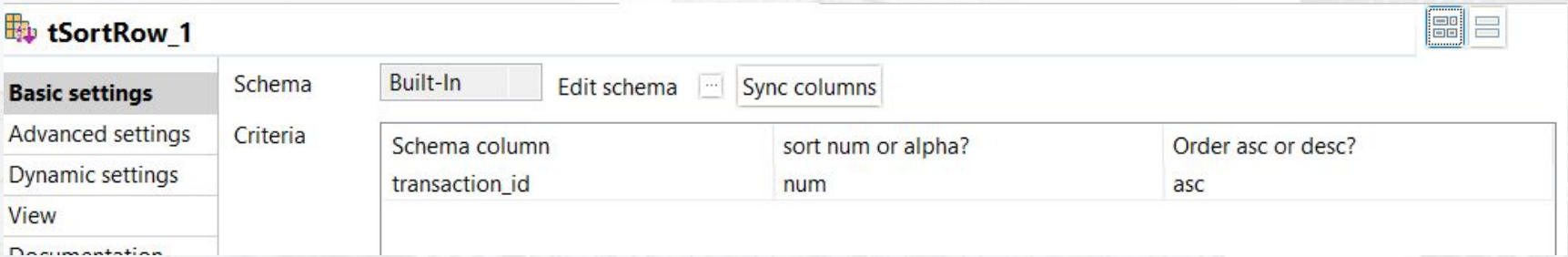
Create ETL Job



3. Create ETL Job for Fact Table

Create ETL Job

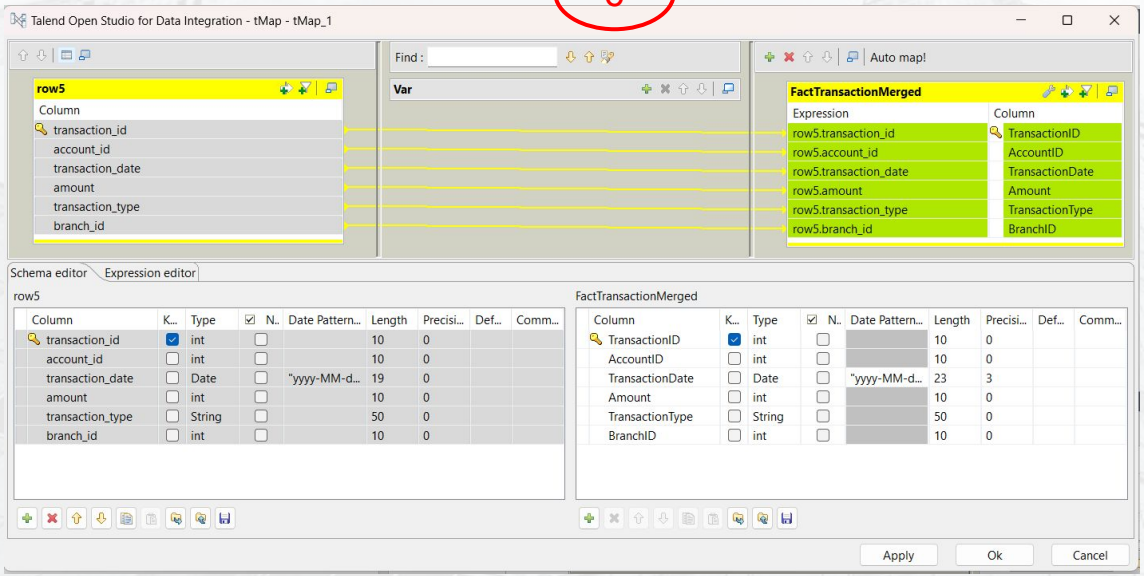
5



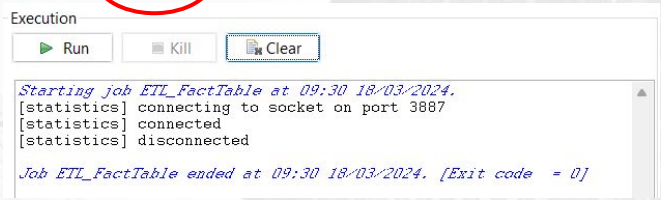
3. Create ETL Job for Fact Table

Create ETL Job

6



7



3. Create ETL Job for Fact Table

Result

SQLQuery1.sql - L...3OD6.DWH (sa (95))*

```
SELECT * FROM FactTransaction;
```

Results | Messages

	TransactionID	AccountID	TransactionDate	Amount	TransactionType	BranchID
1	1	1	2024-01-17 09:10:00.000	100000	Deposit	1
2	2	2	2024-01-17 10:10:00.000	1000000	Deposit	1
3	3	3	2024-01-18 08:30:00.000	10000000	Transfer	1
4	4	3	2024-01-18 10:45:00.000	1000000	Withdrawal	1
5	5	5	2024-01-18 11:10:00.000	200000	Deposit	1
6	6	6	2024-01-18 13:10:00.000	50000	Withdrawal	1
7	7	6	2024-01-19 14:00:00.000	100000	Payment	1
8	8	7	2024-01-19 09:10:00.000	5000000	Deposit	1
9	9	8	2024-01-19 10:40:00.000	300000	Withdrawal	2
10	10	9	2024-01-20 12:10:00.000	2000000	Deposit	1
11	11	10	2024-01-20 15:00:00.000	1000000	Transfer	1
12	12	11	2024-01-20 10:00:00.000	500000	Deposit	1
13	13	12	2024-01-20 12:10:00.000	500000	Withdrawal	5
14	14	13	2024-01-21 14:00:00.000	1500000	Deposit	4
15	15	14	2024-01-21 08:00:00.000	500000	Transfer	3
16	16	15	2024-01-22 09:00:00.000	100000	Deposit	1
17	17	16	2024-01-22 13:10:00.000	100000	Withdrawal	5
18	18	17	2024-01-22 10:20:00.000	700000	Deposit	5
19	19	18	2024-01-22 11:00:00.000	30000	Payment	2
20	20	19	2024-01-22 15:00:00.000	2500000	Deposit	2
21	21	20	2024-01-22 11:30:00.000	150000	Payment	4
22	22	21	2024-01-22 10:45:00.000	800000	Withdrawal	5
23	23	22	2024-01-22 10:50:00.000	100000	Withdrawal	1
24	24	23	2024-01-22 11:10:00.000	300000	Payment	1
25	25	23	2024-01-22 14:30:00.000	400000	Deposit	1

4. Create Stored Procedure

DailyTransaction

```sql
CREATE PROCEDURE dbo.DailyTransaction
(
    @start_date DATETIME,
    @end_date DATETIME
)
AS
BEGIN
    SELECT
        CAST (TransactionDate AS DATE) AS Date,
        COUNT(TransactionID) AS TotalTransactions,
        SUM(Amount) AS TotalAmount
    FROM dbo.FactTransaction
    WHERE CAST (TransactionDate AS DATE) BETWEEN @start_date AND @end_date
    GROUP BY CAST (TransactionDate AS DATE);
END;

EXEC dbo.DailyTransaction @start_date='2024-01-18', @end_date='2024-01-20';
```

Results | Messages

	Date	TotalTransactions	TotalAmount
1	2024-01-18	4	11250000
2	2024-01-19	3	5400000
3	2024-01-20	4	4000000

4. Create Stored Procedure

BalancePerCustomer

```sql
CREATE PROCEDURE dbo.BalancePerCustomer
(
    @name VARCHAR(50)
)
AS
BEGIN
    WITH a AS
    (
        SELECT
            cus.CustomerName,
            acc.AccountType,
            acc.Balance,
            SUM(CASE WHEN trans.TransactionType = 'Deposit' THEN trans.Amount ELSE ((-1) * trans.Amount) END) AS TotalAmount
        FROM
            FactTransaction AS trans
        LEFT JOIN
            DimAccount AS acc ON trans.AccountID = acc.AccountID
        LEFT JOIN
            DimCustomer AS cus ON acc.CustomerID = cus.CustomerID
        WHERE acc.Status = 'active'
        GROUP BY cus.CustomerName, acc.AccountType, acc.Balance
    )
    SELECT
        CustomerName,
        AccountType,
        Balance,
        (Balance + TotalAmount) AS CurrentBalance
    FROM a
    WHERE CustomerName LIKE '%' + @name + '%';
END;

EXEC dbo.BalancePerCustomer @name = 'Shelly';
```

Results | Messages

	CustomerName	AccountType	Balance	CurrentBalance
1	SHELLY JUWITA	checking	25000000	14000000
2	SHELLY JUWITA	saving	1500000	1600000

Rakamin Academy

GitHub

https://github.com/ahmadfaishal9611/Final-Task_ID-X-Partners_Data-Engineer

Presentation Video

https://drive.google.com/file/d/1j_G_GZS30Z0frT6Y6JpbmokmCs-aY4GP/view?usp=sharing

Thank You