

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

ID/X Partners - Data Engineer

Presented by
Ahmad Faishal Akbar



 Bandung City, West Java

 ahmadfaishal9611@gmail.com

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

 <https://github.com/ahmadfaishal9611>

Ahmad Faishal Akbar

Data Enthusiast

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

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**.

Tools

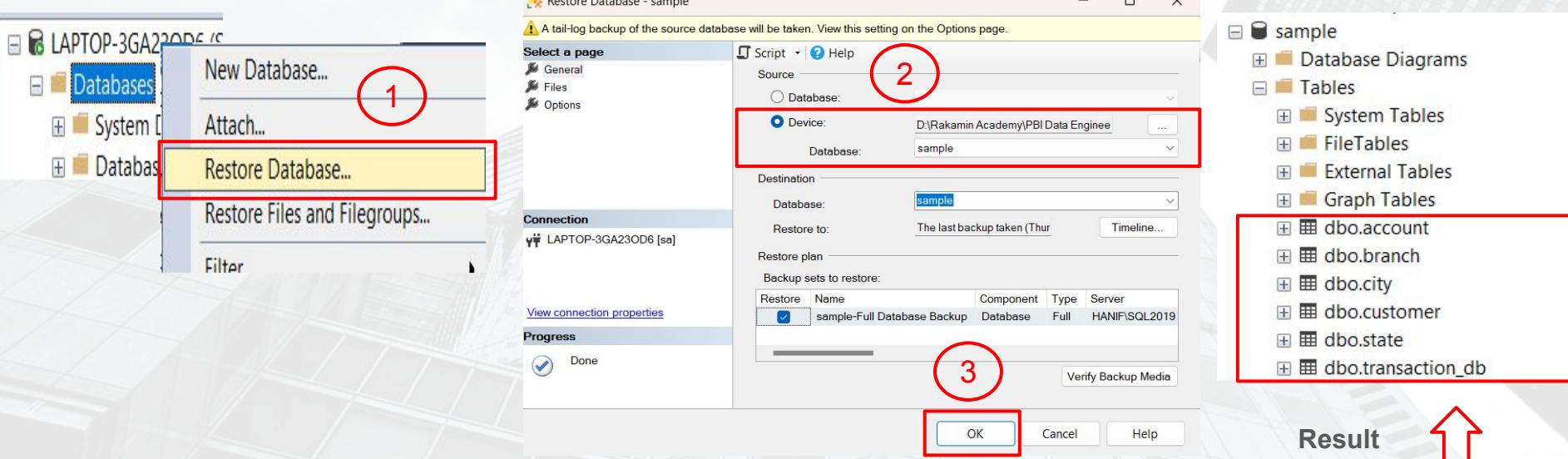


talend

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.



The screenshot illustrates the steps to restore the 'sample' database:

- Left Panel:** Shows the 'Databases' node selected in the Object Explorer. The 'Restore Database...' option is highlighted with a red box and circled with a red number 1.
- Center Panel - Restore Database - sample Dialog:**
 - Source:** The 'Device' radio button is selected, and the path 'D:\Rakamin Academy\PBI Data Enginee' is specified.
 - Database:** The target database is set to 'sample'.
 - Destination:** The destination database is also set to 'sample'.
 - Progress:** The progress bar shows the restore is 'Done'.
 - Buttons:** 'OK' is highlighted with a red box and circled with a red number 3.
- Right Panel - Object Explorer:** Shows the restored database structure for 'sample'.
 - Tables:** Includes 'System Tables', 'FileTables', 'External Tables', and 'Graph Tables'.
 - dbo Schema:** Contains tables like 'account', 'branch', 'city', 'customer', 'state', and 'transaction_db'.

A large red box labeled 'Result' with an upward arrow points to the restored database structure in the Object Explorer.

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

	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

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

dbo.city table

	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. Manga 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

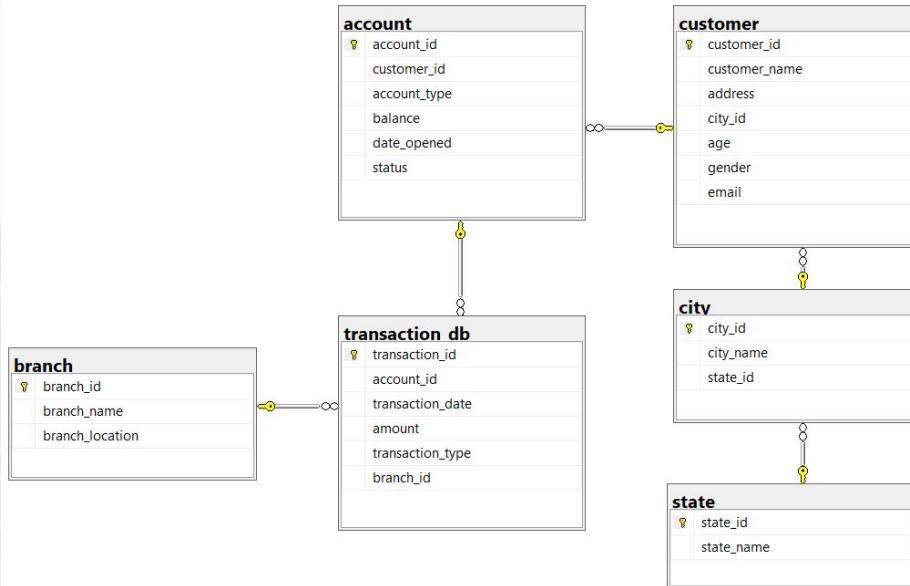
	transaction_id	account_id	transaction_date	amount	transaction_type	branch_id
1	1	1	2024-01-17 09:10:00	100000	Deposit	1
2	2	2	2024-01-17 10:10:00	1000000	Deposit	1
3	3	3	2024-01-18 08:30:00	10000...	Transfer	1

dbo.transaction_db table

1. Data Warehouse Creation

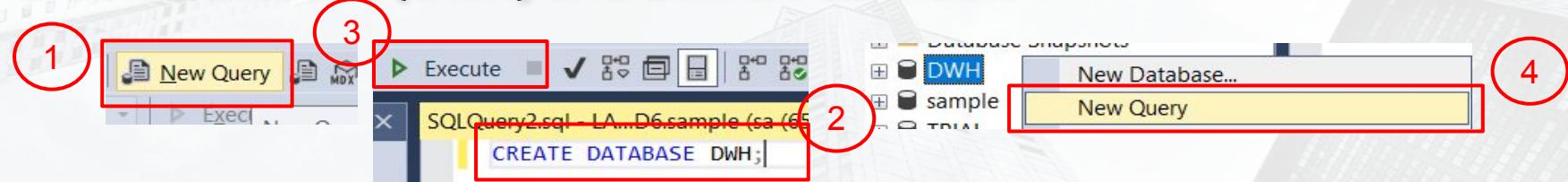
Restore Database

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



1. Data Warehouse Creation

Create Database (DWH) and Dim & Fact Table



```

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)
);

```

```

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)
);

```

□ DWH
+ Database Diagrams
+ Tables
+ System Tables
+ FileTables
+ External Tables
+ Graph Tables
+ dbo.DimAccount
+ dbo.DimBranch
+ dbo.DimCustomer
+ dbo.FactTransaction

Result

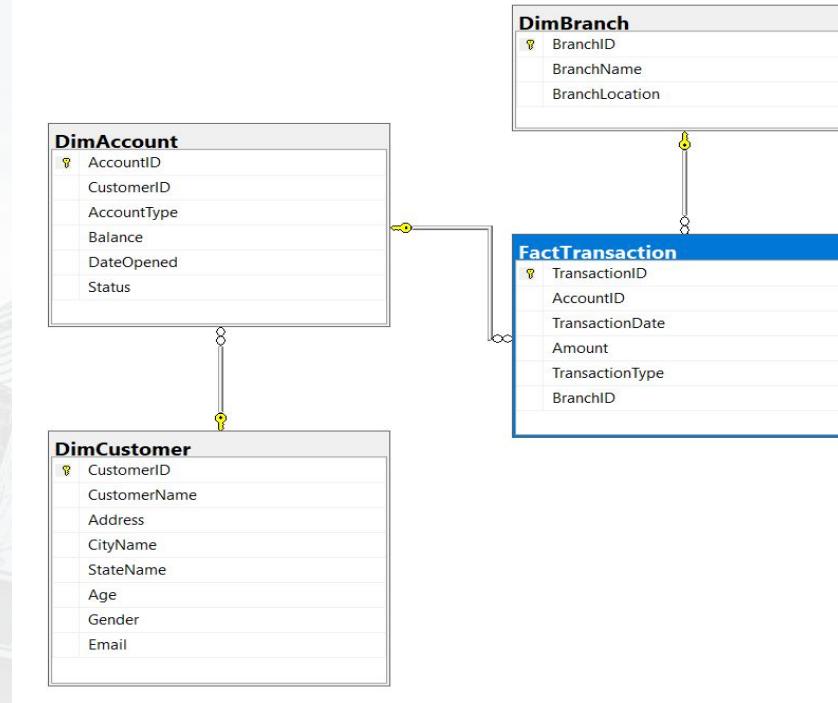


1. Data Warehouse Creation

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

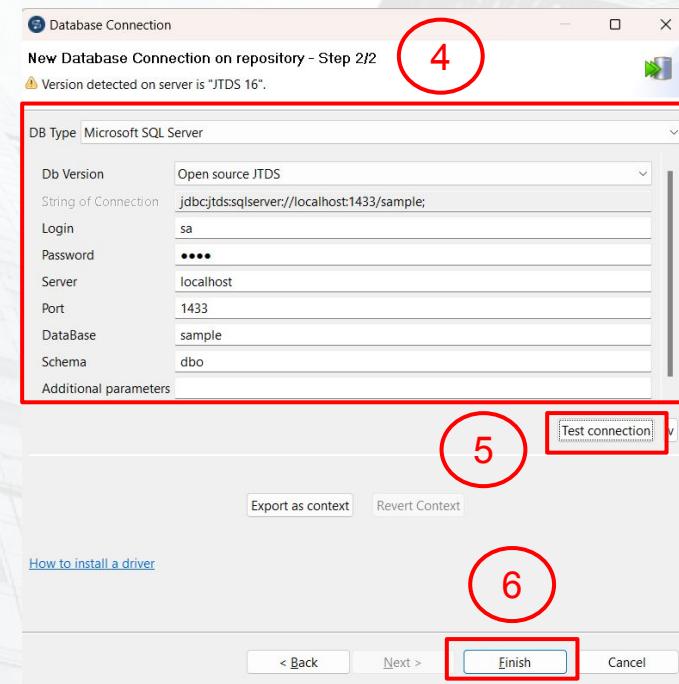
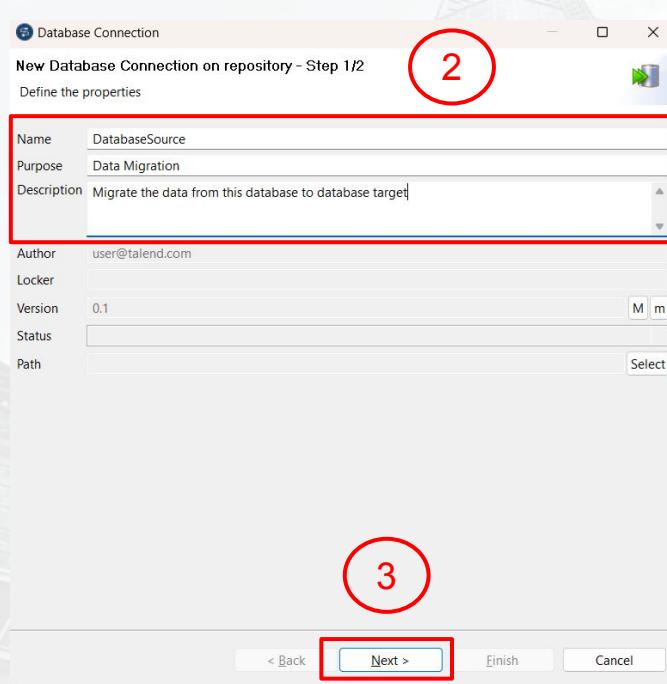
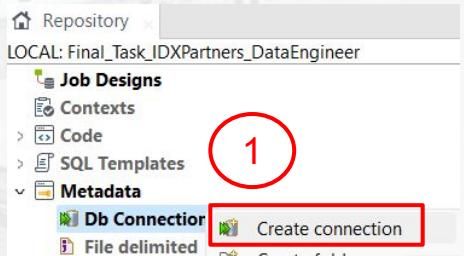
This is a Entity Relationship

Diagram (ERD) of **DWH** database.



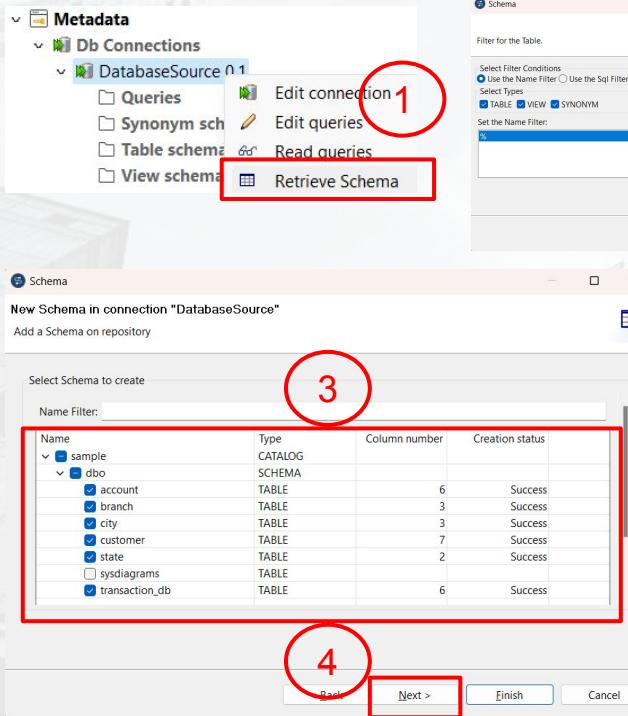
2. Create ETL Job for Dimension Table

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

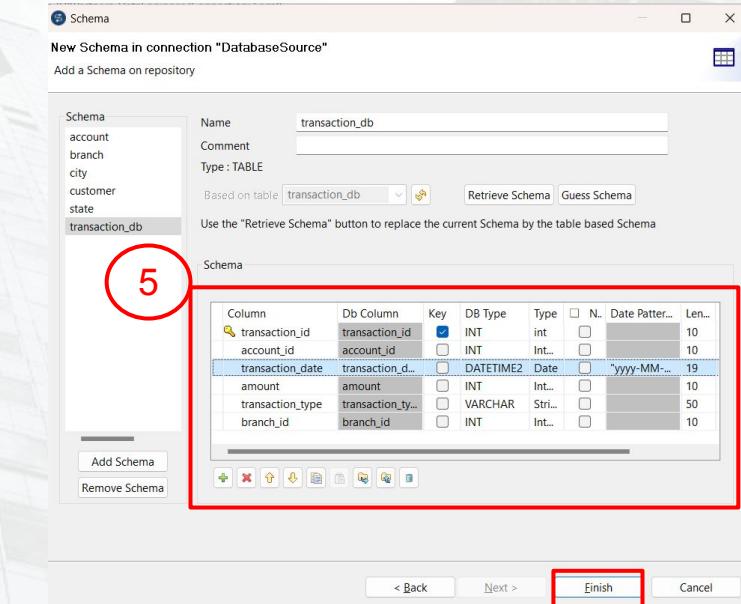
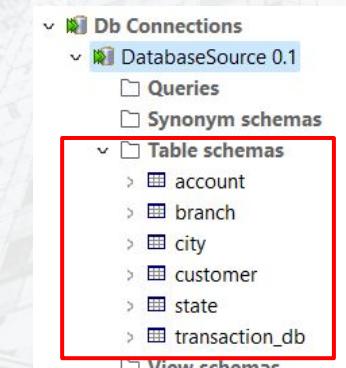


2. Create ETL Job for Dimension Table

Retrieve Schema from DatabaseSource

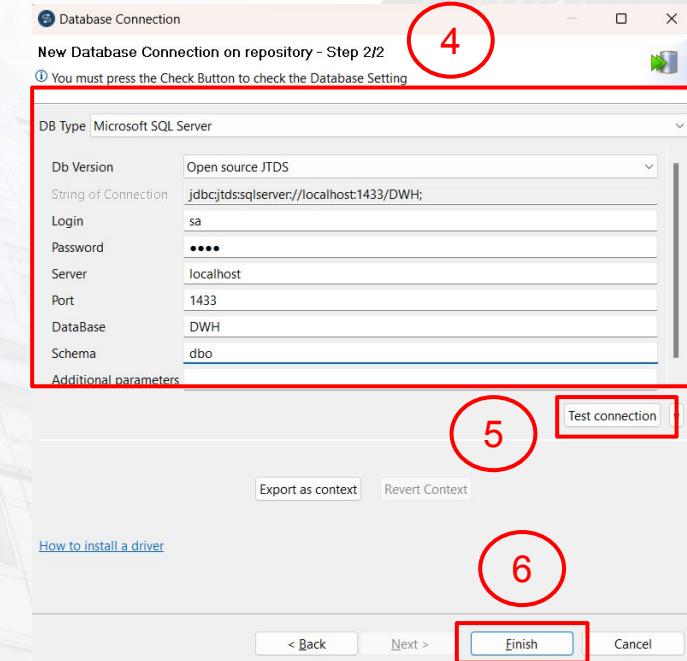
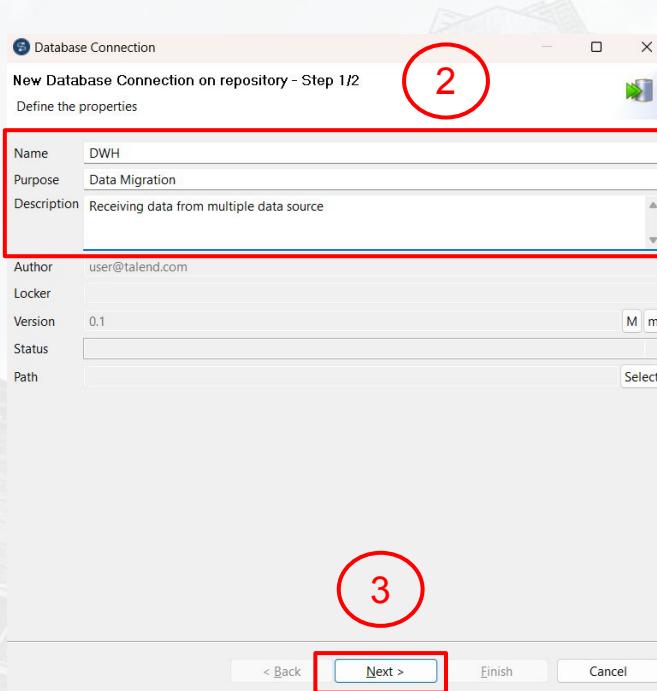
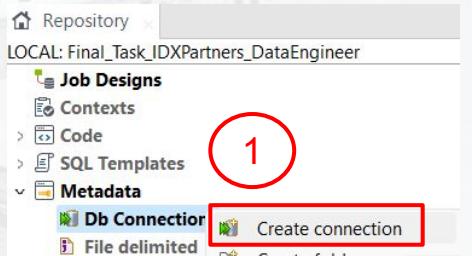


Result 



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

The process involves several steps:

- Open the connection:** Click "Edit connection" (1).
- Retrieve the schema:** Click "Retrieve Schema" (2).
- Select the schema to create:** Choose "FactTransaction" (3).
- Next step:** Click "Next >" (4).
- Schema definition:** Define the schema "FactTransaction" (5). The table structure is as follows:

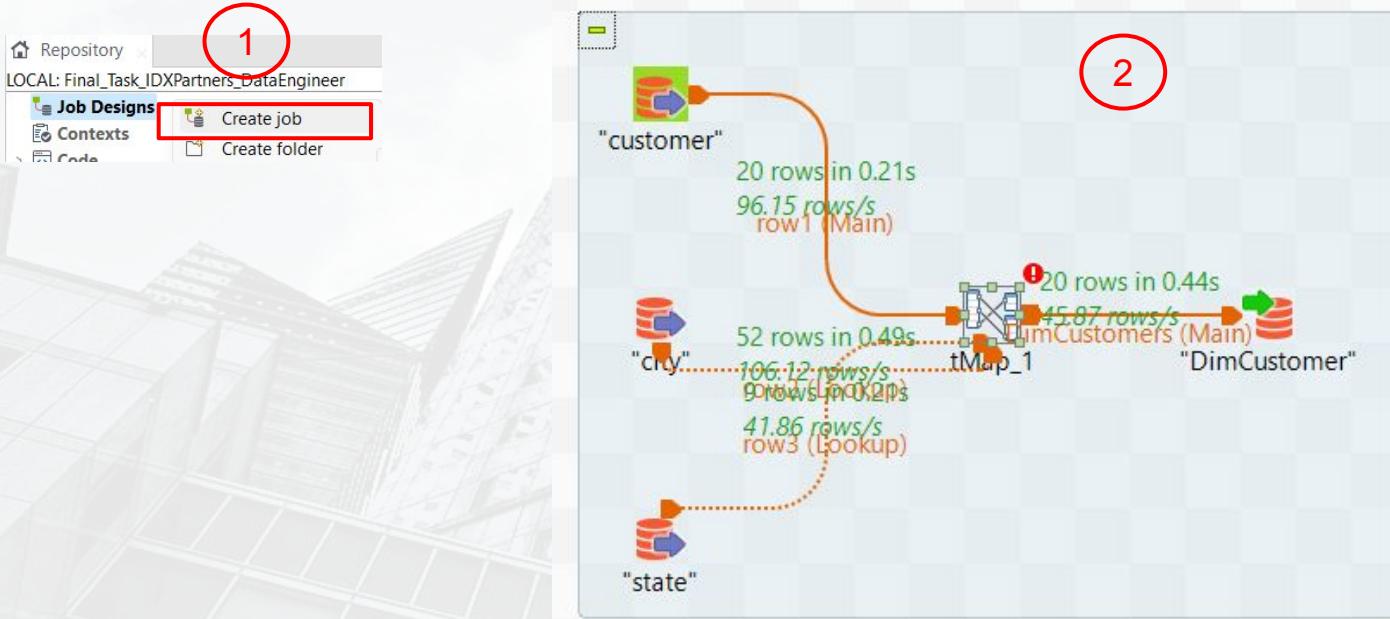
Column	Db Column	K...	DB Ty...	Type	N...	Date P...	Le...	Pre...	D...
TransactionID	TransactionID		INT	int		10	0		
AccountID	AccountID		INT	int		10	0		
Transaction...	Transaction...		DATE...	Da...	"yyyy-...	23	3		
Amount	Amount		INT	int		10	0		
Transacti...	Transacti...		VARCHAR	Str...		50	0		

- Finish:** Click "Finish" (6).

Result: The dimension table "FactTransaction" is successfully created.

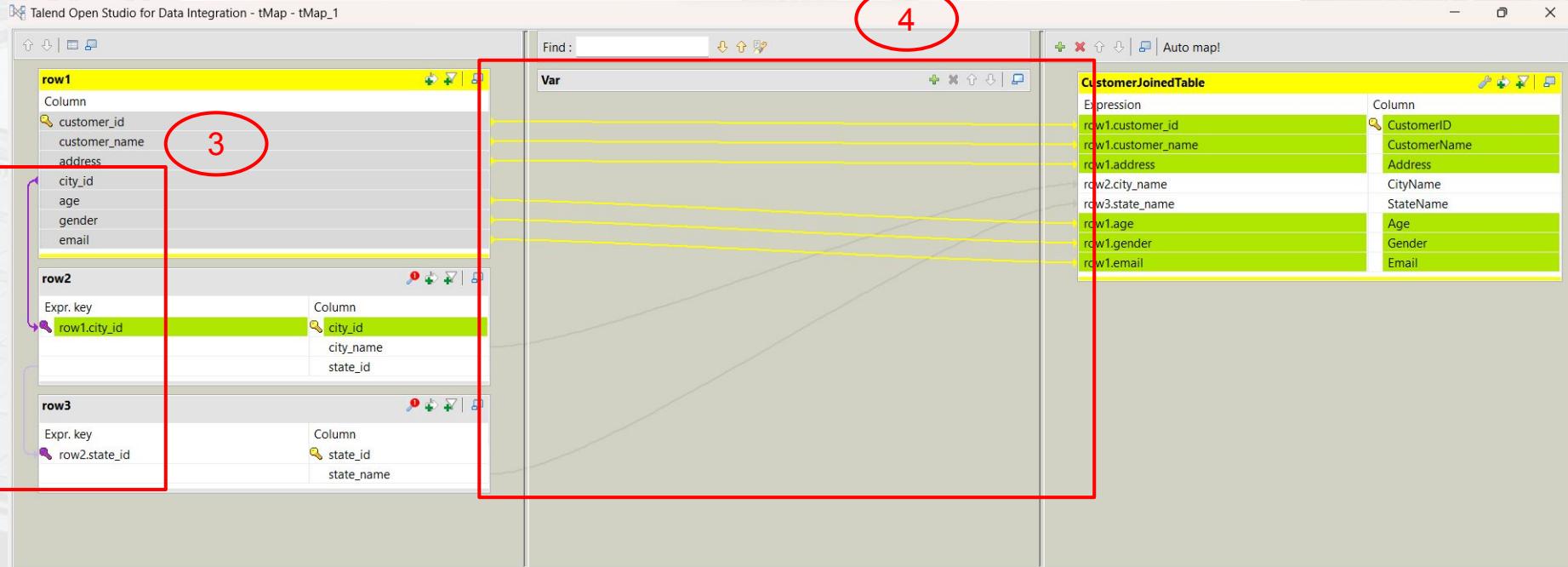
2. Create ETL Job for Dimension Table

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)

CustomerJoinedTable

Expression	Column
row1.customer_id	CustomerID
row1.customer_name	CustomerName
row1.address	Address

Expression Builder

Expression: `StringHandling.UPCASE(row1.customer_name)`

Test:

Var	Value
row1.customer_id	0
row1.customer_name	John Doe
row1.address	123 Main St
row1.city_id	0
row1.age	0
row1.gender	0
row1.email	john.doe@example.com

Categories: User Defined, DataOperation, Mathematical, Numeric, Relational, StringHandling (selected), TalendDataGenerator, TalendDate, TalendString.

Functions: LEFT(String string, int index) : id.String, LEN(String string) : id.Integer - String, RIGHT(String string, int index) : id.String - String, SPACE(int i) : id.String - String, SQQUOTE(String string) : id.String - String, STR(String string, int i) : id.String - String, TRIM(String string) : id.String - String, UPCASE(String string) : id.String - String.

Buttons: Ok (circled 7), Cancel.

CustomerJoinedTable

Expression	Column
row1.customer_id	CustomerID
StringHandling.UPCASE(row1.customer_name)	CustomerName
row1.address	Address
row2.city_name	CityName
row3.state_name	StateName
row1.age	Age
row1.gender	Gender
row1.email	Email

CustomerJoinedTable

Expression	Column
row1.customer_id	CustomerID
StringHandling.UPCASE(row1.customer_name)	CustomerName
StringHandling.UPCASE(row1.address)	Address
StringHandling.UPCASE(row2.city_name)	CityName
StringHandling.UPCASE(row3.state_name)	StateName
row1.age	Age
StringHandling.UPCASE(row1.gender)	Gender
row1.email	Email

Buttons: Apply, Ok (circled 10), Cancel.

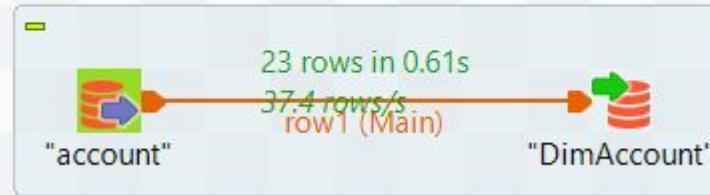
Execution

11

2. Create ETL Job for Dimension Table

Create ETL Job (DimAccount and Dim Branch)

Customer 0.1 *Job DimAccount 0.1 X Job Dim



Account 0.1 *Job DimBranch 0.1 X Job Fact1



2. Create ETL Job for Dimension Table

Result

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

```

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

Results Messages

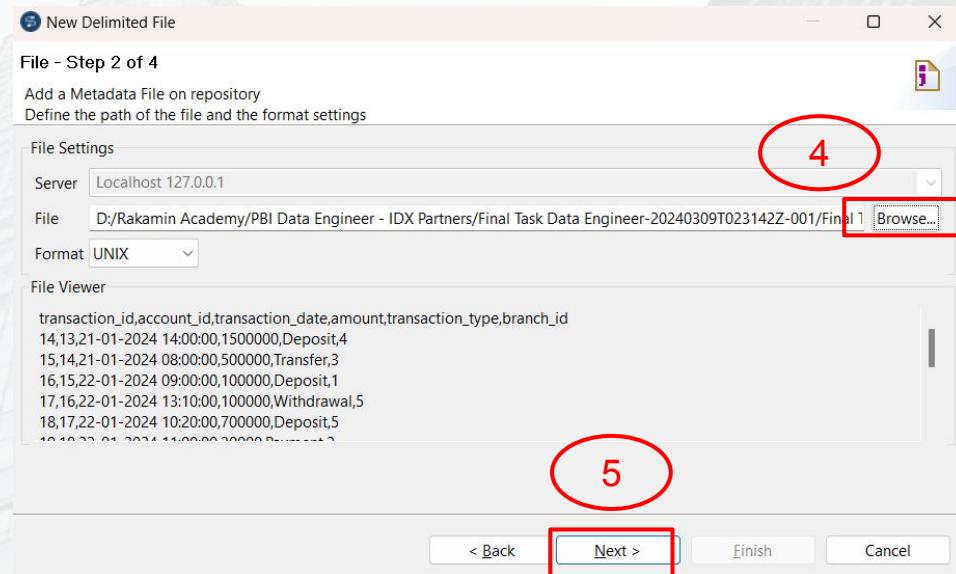
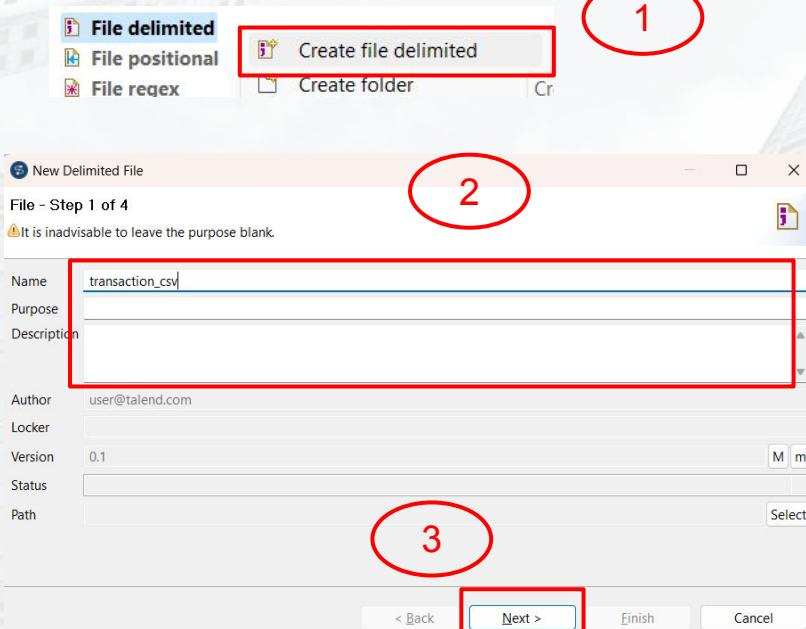
	AccountID	CustomerID	AccountType	Balance	DateOpened	Status
1	1	1	saving	1500000	2020-05-01 09:00:00.000	active
2	2	2	saving	500000	2020-06-01 10:00:00.000	active
3	3	1	checking	25000000	2020-06-21 09:00:00.000	active
4	4	3	checking	4500000	2021-06-24 11:00:00.000	terminated

	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. Ciparay 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	II KENANGA NO. 11	CILANDAK	JAKARTA SELATAN	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**)

File - Step 3 of 4

Add a Metadata File on repository
Define the setting of the parse job

File Settings
Encoding: US-ASCII
Field Separator: Comma (",")
Row Separator: Blank ("\\n")

Escape Char Settings
CSV (radio button selected)
Delimited (radio button selected)
Escape Char: Empty
Text Enclosure: Empty
 Split row before field

Preview | Output
 Set heading row as column names
 Refresh Preview

transaction_id	account_id	transaction_date	amount	transaction_type	branch_id
14	13	21-01-2024 14:00:00	1500000	Deposit	4
15	14	21-01-2024 08:00:00	500000	Transfer	3
16	15	22-01-2024 09:00:00	100000	Deposit	1
17	16	22-01-2024 13:10:00	100000	Withdrawal	5

Export as context Revert Context

< Back Finish Cancel

6 **7** **8** **9** **10**

File - Step 4 of 4

Add a Schema on repository
Define the Schema

Name: metadata
Comment:

Schema
 Click to update schema preview Guess

Description of the Schema

Column	K...	Type	N...	Date Pattern (Ctrl...)	Length	Precision	Default	Comment
transaction_id	<input checked="" type="checkbox"/>	Integer	<input type="checkbox"/>		2	0		
account_id	<input type="checkbox"/>	Integer	<input type="checkbox"/>		2	0		
transaction_date	<input type="checkbox"/>	Date	<input type="checkbox"/>	"dd-MM-yyyy H...	19	0		
amount	<input type="checkbox"/>	Integer	<input type="checkbox"/>		7	0		
transaction_type	<input type="checkbox"/>	String	<input type="checkbox"/>		10	0		
branch_id	<input type="checkbox"/>	Integer	<input type="checkbox"/>		1	0		

< Back Finish Cancel

3. Create ETL Job for Fact Table

Create file Excel (**transaction_excel**)

File Excel **Create file Excel** (1) Create folder

New Excel File (2)

Name: transaction_excel
 Purpose: (red box)
 Description:
 Author: user@talend.com
 Locker:
 Version: 0.1
 Status:
 Path:

< Back Next > (3) Finish Cancel

New Excel File
 File - Step 2 of 4
 Add a Metadata File on repository
 Define the path of the file and the format settings

File Settings (4)
 Server: Localhost 127.0.0.1
 File: D:/Rakamin Academy/PBI Data Engineer - IDX Partners/Final Task Data Engineer-20240309T023142Z.001 (red box)
 Read excel2007 file format(xlsx)
 Generation mode: Memory-consuming(User mode)

File Viewer and Sheets setting
 Set sheets parameters
 All sheets/DSelect sheet (5)
 Sheet1

A	B	C	D	E	F
transa...	acco...	trans...	amo...	trans...	branc...
6.0	6.0	18-Ja...	5000...	With...	1.0
7.0	6.0	19-Ja...	1000...	Paym...	1.0
11.0	10.0	20-Ja...	1000...	Trans...	1.0
12.0	11.0	20-Ja...	5000...	Dепо...	1.0
12.0	12.0	20-Ja...	5000...	Varia...	1.0

< Back Next > (6) Finish Cancel

3. Create ETL Job for Fact Table

Create file Excel (**transaction_excel**)

File - Step 3 of 4
Add a Metadata File on repository
Define the setting of the parse job

File Settings
Encoding: UTF-8
 Advanced separator(for number)
Thousands separator: ;
Decimal separator: ,
Metadata column setting
First column: 1
Last column:
Preview Output
 Set heading row as column names

Rows To Skip
If any rows must be ignored, specify the following parameters
Header: 1
Footer:

Limit Of Rows
If the number of lines must be limited, specify this number
Limit:

transaction_id	account_id	transaction_date	amount	transaction_type	branch_id
6	6	Thu Jan 18 13:10:00 ICT 2024	50000	Withdrawal	1
7	6	Fri Jan 19 14:00:00 ICT 2024	100000	Payment	1
11	10	Sat Jan 20 15:00:00 ICT 2024	1000000	Transfer	1
12	11	Sat Jan 20 10:00:00 ICT 2024	500000	Deposit	1

7

8

File - Step 4 of 4
Add a Schema on repository
Define the Schema

Name: metadata
Comment:

Schema
Click to update schema preview

Description of the Schema

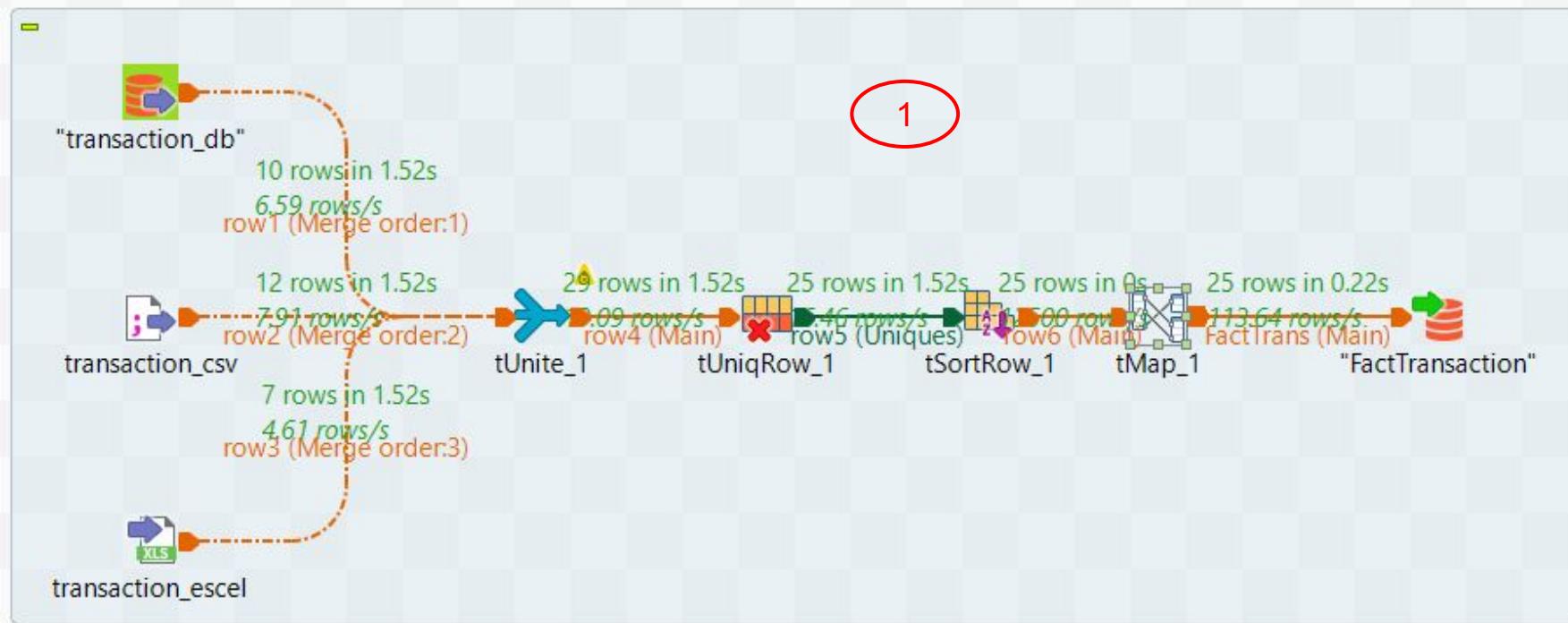
Column	K...	Type	<input type="checkbox"/> N..	Date Pattern (Ctrl...)	Length	Precision	Default	Comment
transaction_id	<input checked="" type="checkbox"/>	Integer	<input type="checkbox"/>	2	0			
account_id	<input type="checkbox"/>	Integer	<input type="checkbox"/>	2	0			
transaction_date	<input type="checkbox"/>	Date	<input type="checkbox"/>	"EEE MMM dd HH:mm:ss zzz yyyy"				
amount	<input type="checkbox"/>	Integer	<input type="checkbox"/>	7	0			
transaction_type	<input type="checkbox"/>	String	<input type="checkbox"/>	10	0			
branch_id	<input type="checkbox"/>	Integer	<input type="checkbox"/>	1	0			

9

10

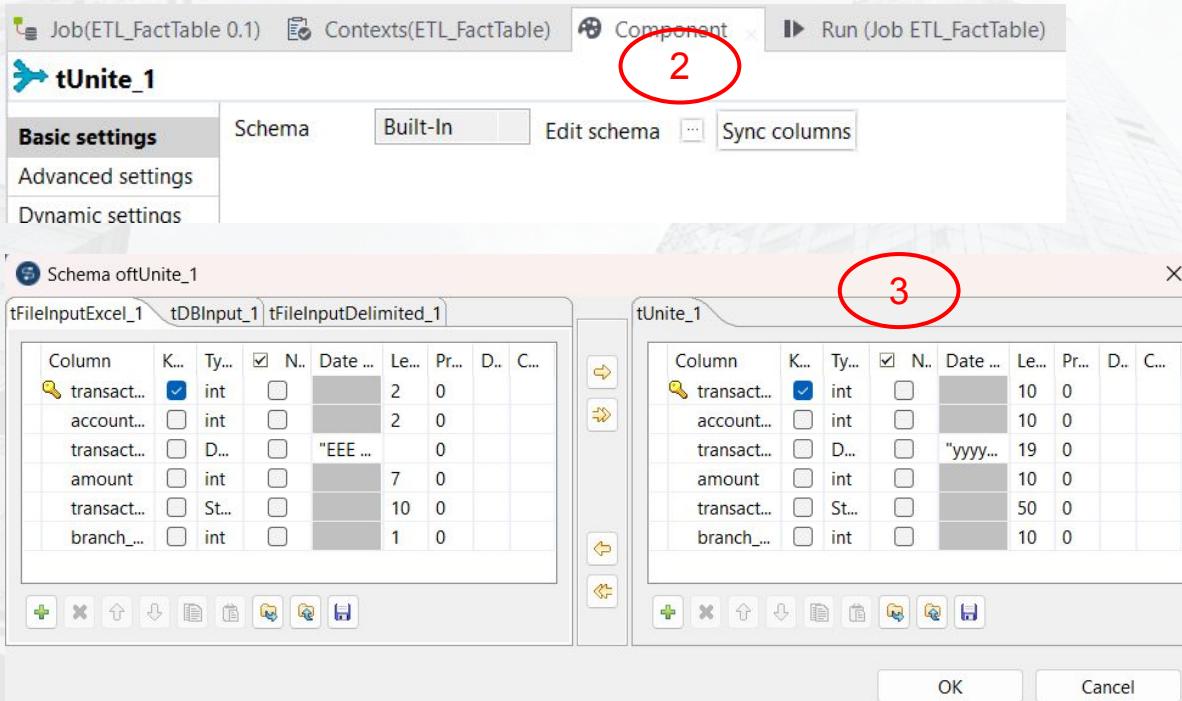
3. Create ETL Job for Fact Table

Create ETL Job



3. Create ETL Job for Fact Table

Create ETL Job



The screenshot shows the Talend Studio interface for creating an ETL job named "Job(ETL_FactTable 0.1)". The "Basic settings" tab is selected. A red circle labeled "2" highlights the "Component" tab, which is currently inactive. A red circle labeled "3" highlights the "Schema" tab, which is active and displays the schema for the "tUnite_1" component.

Basic settings (selected)

Advanced settings

Dynamic settings

tUnite_1

Component (inactive)

Run (Job ETL_FactTable)

Schema (active)

Built-In

Edit schema

Sync columns

Schema oftUnite_1

tFileInputExcel_1 tDBInput_1 tFileInputDelimited_1

Column	K...	Ty...	N...	Date ...	Le...	Pr...	D...	C...
transact...	<input checked="" type="checkbox"/>	int	<input type="checkbox"/>		2	0		
account...	<input type="checkbox"/>	int	<input type="checkbox"/>		2	0		
transact...	<input type="checkbox"/>	D...	<input type="checkbox"/>	"EEE ...		0		
amount	<input type="checkbox"/>	int	<input type="checkbox"/>		7	0		
transact...	<input type="checkbox"/>	St...	<input type="checkbox"/>		10	0		
branch...	<input type="checkbox"/>	int	<input type="checkbox"/>		1	0		

tUnite_1

Column	K...	Ty...	N...	Date ...	Le...	Pr...	D...	C...
transact...	<input checked="" type="checkbox"/>	int	<input type="checkbox"/>		10	0		
account...	<input type="checkbox"/>	int	<input type="checkbox"/>		10	0		
transact...	<input type="checkbox"/>	D...	<input type="checkbox"/>	"yyyy...	19	0		
amount	<input type="checkbox"/>	int	<input type="checkbox"/>		10	0		
transact...	<input type="checkbox"/>	St...	<input type="checkbox"/>		50	0		
branch...	<input type="checkbox"/>	int	<input type="checkbox"/>		10	0		

OK Cancel

3. Create ETL Job for Fact Table

Create ETL Job

Screenshot of the Talend ETL interface showing the configuration of a component named "tUniqRow_1".

The component is set to "Schema" mode. A red circle highlights the "Sync columns" button, which is located at the top right of the component's configuration area.

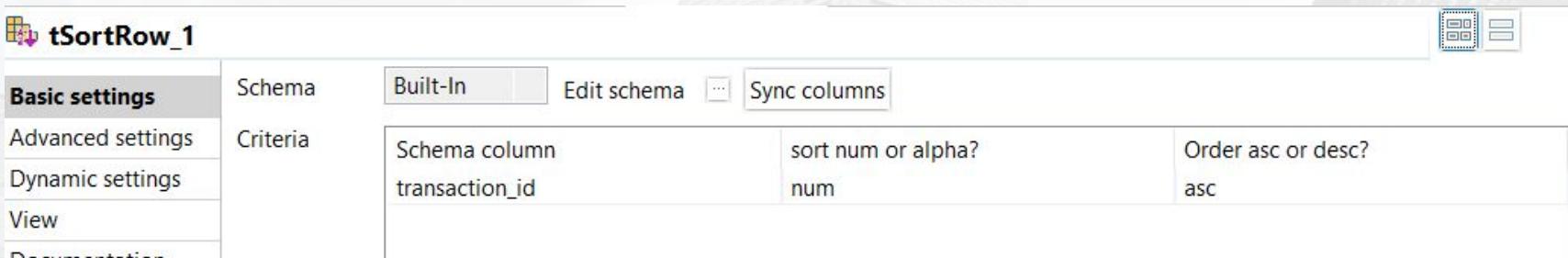
Component Configuration:

Column	Key attribute	Case sensitive
transaction_id	<input checked="" type="checkbox"/>	<input type="checkbox"/>
account_id	<input type="checkbox"/>	<input type="checkbox"/>
transaction_date	<input type="checkbox"/>	<input type="checkbox"/>
amount	<input type="checkbox"/>	<input type="checkbox"/>
transaction_type	<input type="checkbox"/>	<input type="checkbox"/>
branch_id	<input type="checkbox"/>	<input type="checkbox"/>

3. Create ETL Job for Fact Table

Create ETL Job

5



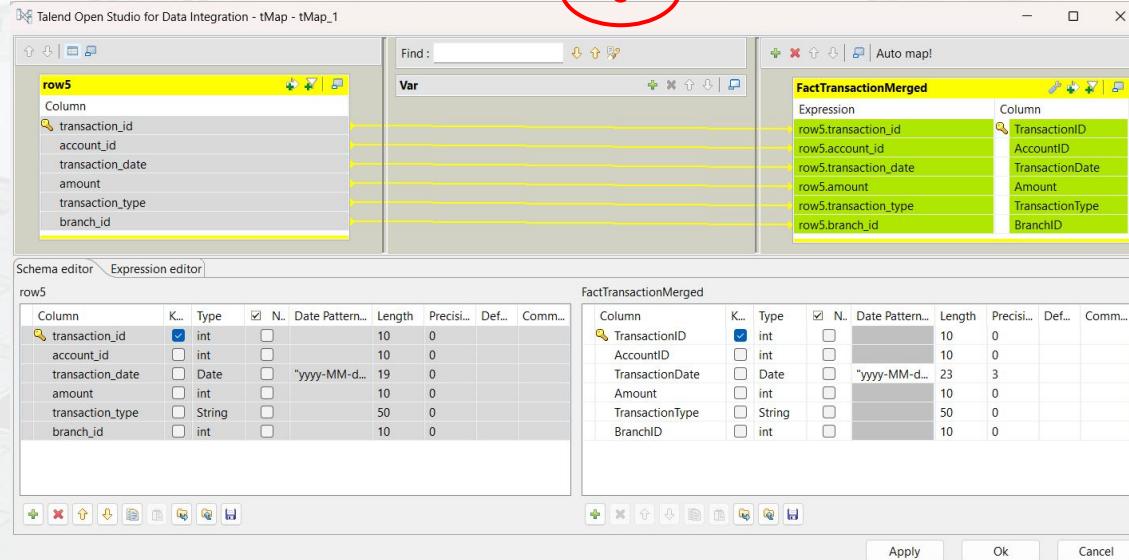
The screenshot shows the Talend Studio interface with the following details:

- Component Name:** tSortRow_1
- Basic settings:** Selected tab.
- Schema:** Built-In
- Criteria:** Transaction ID is set as the schema column for sorting.
- Sort Order:** Ascending (asc).

3. Create ETL Job for Fact Table

Create ETL Job

6



Column	K...	Type	N.	Date Pattern...	Length	Precisi...	Def...	Comm...
transaction_id		int			10	0		
account_id		int			10	0		
transaction_date		Date		"yyyy-MM-d..."	19	0		
amount		int			10	0		
transaction_type		String			50	0		
branch_id		int			10	0		

Column	K...	Type	N.	Date Pattern...	Length	Precisi...	Def...	Comm...
TransactionID		int			10	0		
AccountID		int			10	0		
TransactionDate		Date		"yyyy-MM-d..."	23	3		
Amount		int			10	0		
TransactionType		String			50	0		
BranchID		int			10	0		

7



Execution

Run Kill Clear

```

Starting job ETL_FactTable at 09:30 18/03/2024.
[statistics] connecting to socket on port 3887
[statistics] connected
[statistics] disconnected
Job ETL_FactTable ended at 09:30 18/03/2024. [Exit code = 0]

```

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

```
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

```

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

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



id/x partners