

LAPORAN PRAKTIKUM DATA WAREHOUSE
ETL ERP FINANCE

Dosen Pengampu : Edi Satriyanto S.Si., M.Si



DISUSUN UNTUK MEMENUHI TUGAS MATA KULIAH :
PRAKTIKUM DATA WAREHOUSE

OLEH : KELOMPOK 7

SANIA NUR MIRZA– 3323600014

NADILA IMAARAH– 3323600015

PROGRAM STUDI SAINS DATA TERAPAN A
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

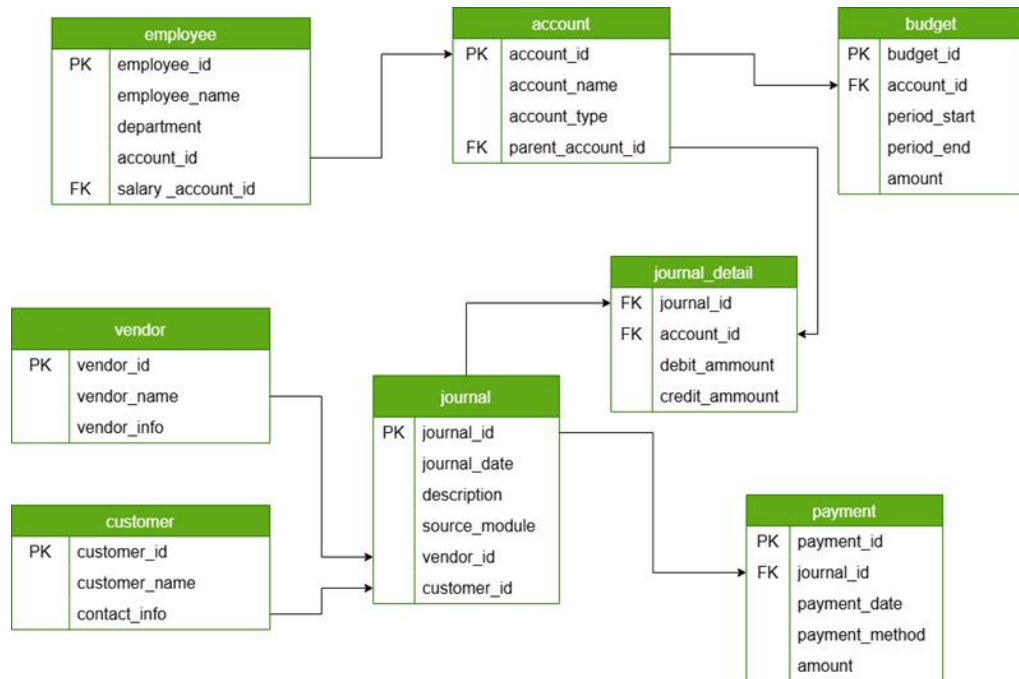
2025

1. ERP

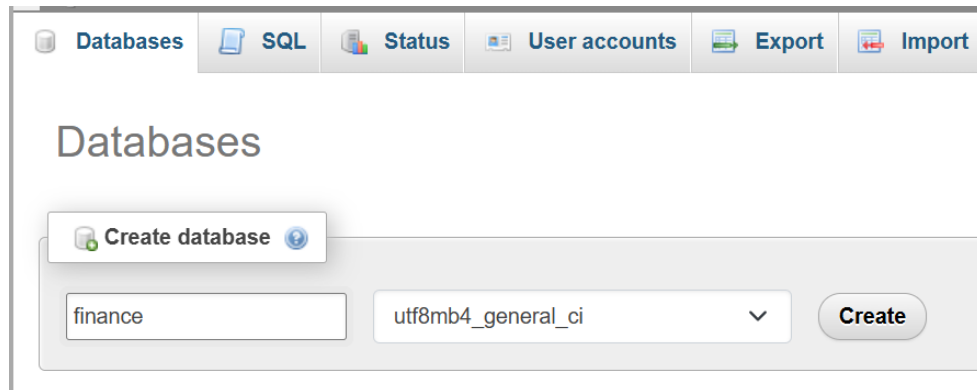
Modul ERP finance yang akan dikembangkan untuk Data Mart akan fokus pada pengelolaan data transaksi keuangan perusahaan, yang melibatkan berbagai elemen seperti employee, account, budget, vendor, customer, journal, dan payment. Modul ini bertujuan untuk mengotomatisasi dan mengintegrasikan proses keuangan yang ada, memberikan manajemen kontrol yang lebih baik, serta mempermudah pelaporan dan analisis keuangan. Semua data yang dihasilkan akan diolah dan dimasukkan ke dalam Data Mart untuk analisis lebih lanjut. Untuk sistem kerja modul ERP yakni sebagai berikut :

- a. employee : Modul ini menyimpan segala informasi mengenai staff yang terlibat dalam pencatatan keuangan yang ada dalam perusahaan. Informasi yang dimiliki meliputi beberapa atribut, yakni employee_name, employee_id, department, account_id, dan salary_account_id. Informasi dari modul ini dapat digunakan dalam melacak siapa yang melakukan transaksi pada buku besar hingga menyimpan informasi akun terkait penggajian.
- b. account : Modul ini bertujuan untuk menyimpan daftar akun dalam chart of accounts perusahaan. Atribut yang dimiliki antara lain : account_id, account_name, account_type, parent_account_id. Atribut - atribut tersebut dapat digunakan untuk menyusun laporan keuangan berdasarkan tipe akun dan mendukung pembuatan struktur akun bertingkat.
- c. budget : Modul ini digunakan untuk mencatat alokasi anggaran terhadap akun - akun keuangan tertentu dan periode tertentu. Atribut yang digunakan budget_id, account_id, period_start, period_end, dan ammount.
- d. vendor : Modul ini mencakup informasi mengenai penyedia barang/jasa yang digunakan untuk transaksi. Modul ini mencakup informasi mengenai vendor_id, vendor_name, dan vendor_info.
- e. journal_detail : Modul ini digunakan untuk mencatat seluruh rincian dari transaksi keuangan yang dilakukan, mencatat rincian akun, hingga analisis dan laporan keuangan. Atribut yang dimiliki journal_id, account_id, debit_ammount, dan credit_ammount.
- f. customer : Modul ini menyimpan data pelanggan yang melakukan transaksi dengan perusahaan, sehingga perusahaan dapat melakukan pencatatan invoice dan penerimaan pembayaran dan memonitor pembayaran dari pelanggan. Atribut yang dimiliki antara lain customer_id, customer_name, dan contact_info.
- g. journal : Modul journal adalah catatan dari seluruh transaksi di perusahaan. Informasi yang dimiliki mencakup journal_id, journal_date, description, source_module, vendor_id, dan customer_id. Dari informasi tersebut perusahaan dapat mengawasi keuangan yang ada di perusahaan, dapat menjadikan modul ini sebagai referensi utama untuk pencatatan transaksi keuangan, serta memudahkan dalam pelacakan asal transaksi.
- h. payment : Modul ini berfungsi untuk mencatat pembayaran yang dilakukan atau diterima. Modul ini memiliki atribut payment_id, journal_id, payment_date, payment_method, dan ammount yang berguna untuk mengelola arus uang masuk/keluar, mendukung adanya pelaporan keuangan dan rekonsiliasi.

2. ERD



Membuat database finance pada mysql untuk dijadikan sumber data etl pada table input. Untuk table output akan di simpan pada database postgres



1. Memasukkan data pada database finance

A. Memasukkan data tabel account pada database Mysql dengan memasukkan query seperti di gambar, kemudian pilih import dan masukkan data yang ingin di masukkan

```

1 CREATE TABLE account (
2     account_id VARCHAR(20) PRIMARY KEY,
3     account_name VARCHAR(100),
4     account_type VARCHAR(50),
5     parent_account_id INT
6 );
7

```

Hasil nya akan menjadi seperti ini

account_id	account_name	account_type	parent_account_id
ACC-301	Kewajiban Jangka Panjang	Liability	8
ACC-302	Modal Disetor	Equity	2
ACC-303	Biaya Marketing	Expense	5
ACC-304	Pendapatan Penjualan	Revenue	18
ACC-305	Pembayaran Supplier	Liability	5
ACC-306	Kas	Asset	4
ACC-307	Biaya Gaji	Expense	3
ACC-308	Inventaris	Asset	12
ACC-309	Peralatan Kantor	Asset	19
ACC-310	Laba Ditahan	Equity	19
ACC-311	Kewajiban Jangka Panjang	Liability	10
ACC-312	Pendapatan Lain-Lain	Revenue	16
ACC-313	Pembayaran Supplier	Liability	14
ACC-314	Modal Disetor	Equity	12
ACC-315	Laba Ditahan	Equity	13
ACC-316	Pendapatan Penjualan	Revenue	19
ACC-317	Kas	Asset	7
ACC-318	Inventaris	Asset	8

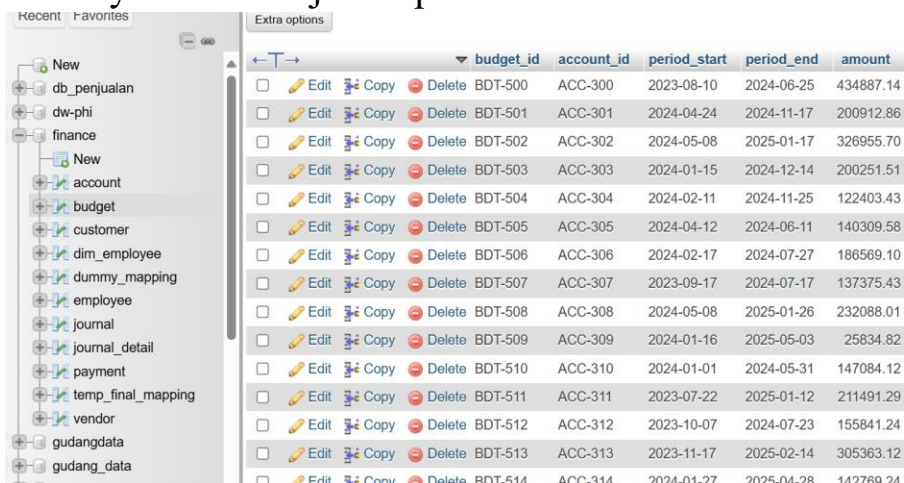
B. Memasukkan data tabel budget pada database Mysql dengan memasukkan query seperti di gambar, kemudian pilih import dan masukkan data yang ingin di masukkan

```

1 CREATE TABLE budget (
2     budget_id VARCHAR(20) PRIMARY KEY,
3     account_id VARCHAR(20),
4     period_start DATE,
5     period_end DATE,
6     amount DECIMAL(15,2)
7 );
8

```

Hasilnya akan menjadi seperti ini



	budget_id	account_id	period_start	period_end	amount
<input type="checkbox"/>	BDT-500	ACC-300	2023-08-10	2024-06-25	434887.14
<input type="checkbox"/>	BDT-501	ACC-301	2024-04-24	2024-11-17	200912.86
<input type="checkbox"/>	BDT-502	ACC-302	2024-05-08	2025-01-17	326955.70
<input type="checkbox"/>	BDT-503	ACC-303	2024-01-15	2024-12-14	200251.51
<input type="checkbox"/>	BDT-504	ACC-304	2024-02-11	2024-11-25	122403.43
<input type="checkbox"/>	BDT-505	ACC-305	2024-04-12	2024-06-11	140309.58
<input type="checkbox"/>	BDT-506	ACC-306	2024-02-17	2024-07-27	186569.10
<input type="checkbox"/>	BDT-507	ACC-307	2023-09-17	2024-07-17	137375.43
<input type="checkbox"/>	BDT-508	ACC-308	2024-05-08	2025-01-26	232088.01
<input type="checkbox"/>	BDT-509	ACC-309	2024-01-16	2025-05-03	25834.82
<input type="checkbox"/>	BDT-510	ACC-310	2024-01-01	2024-05-31	147084.12
<input type="checkbox"/>	BDT-511	ACC-311	2023-07-22	2025-01-12	211491.29
<input type="checkbox"/>	BDT-512	ACC-312	2023-10-07	2024-07-23	155841.24
<input type="checkbox"/>	BDT-513	ACC-313	2023-11-17	2025-02-14	305363.12
<input type="checkbox"/>	BDT-514	ACC-314	2024-01-27	2025-04-28	142769.24

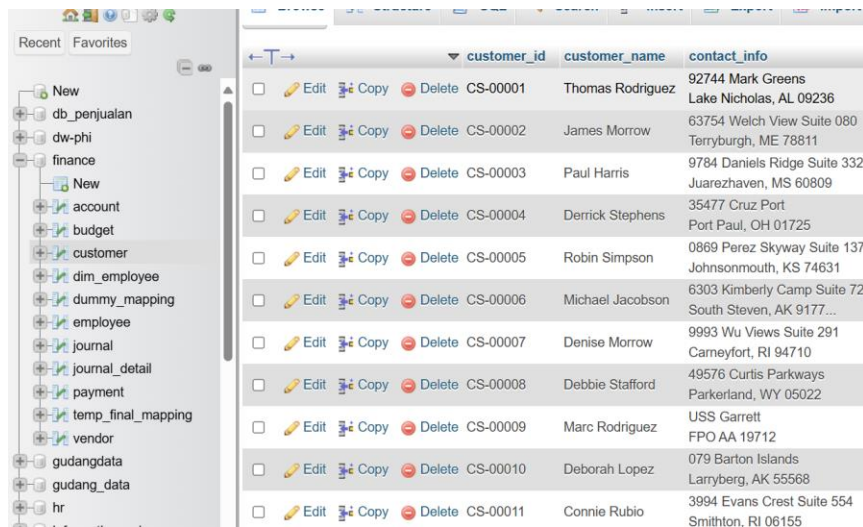
C. Memasukkan data tabel customer pada database Mysql dengan memasukkan query seperti di gambar, kemudian pilih import dan masukkan data yang ingin di masukkan

```

1 CREATE TABLE customer (
2     customer_id VARCHAR(20) PRIMARY KEY,
3     customer_name VARCHAR(100),
4     contact_info TEXT
5 );
6

```

Hasilnya akan menjadi seperti ini



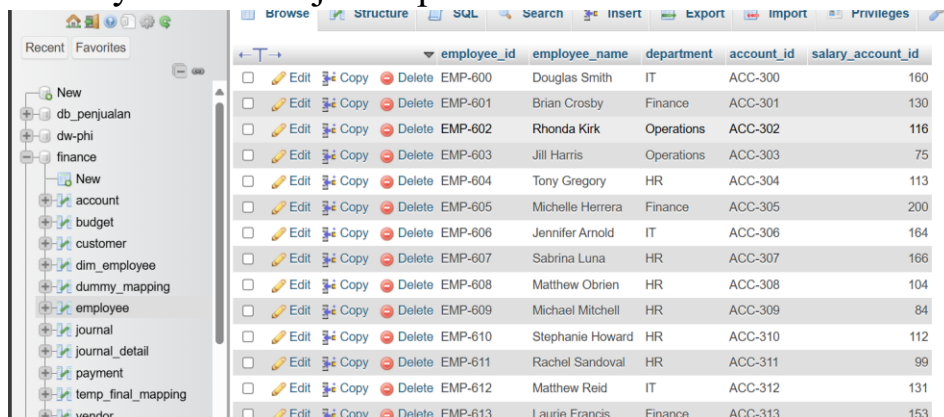
D. Memasukkan data tabel employee pada database Mysql dengan memasukkan query seperti di gambar, kemudian pilih import dan masukkan data yang ingin di masukkan

```

1 CREATE TABLE employee (
2     employee_id VARCHAR(20) PRIMARY KEY,
3     employee_name VARCHAR(100),
4     department VARCHAR(50),
5     account_id VARCHAR(20),
6     salary_account_id INT
7 );
8

```

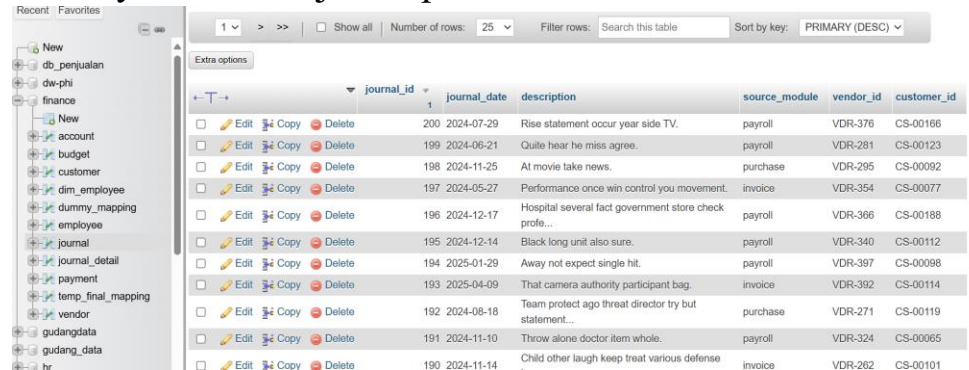
Hasilnya akan menjadi seperti ini



- E. Memasukkan data tabel journal pada database Mysql dengan memasukkan query seperti di gambar, kemudian pilih import dan masukkan data yang ingin di masukkan

```
1 CREATE TABLE journal (  
2     journal_id INT PRIMARY KEY,  
3     journal_date DATE,  
4     description TEXT,  
5     source_module VARCHAR(50),  
6     vendor_id VARCHAR(20),  
7     customer_id VARCHAR(20)  
8 );  
9 |
```

Hasilnya akan menjadi seperti ini

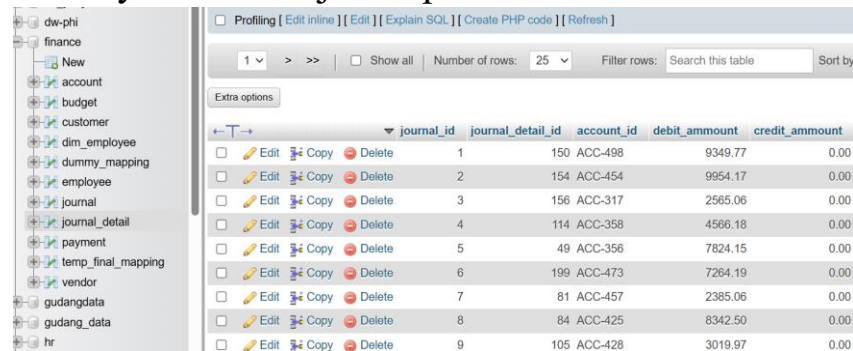


	journal_id	journal_date	description	source_module	vendor_id	customer_id
<input type="checkbox"/>	200	2024-07-29	Rise statement occur year side TV.	payroll	VDR-376	CS-00166
<input type="checkbox"/>	199	2024-06-21	Quite hear he miss agree.	payroll	VDR-281	CS-00123
<input type="checkbox"/>	198	2024-11-25	At movie take news.	purchase	VDR-295	CS-00092
<input type="checkbox"/>	197	2024-05-27	Performance once win control you movement.	invoice	VDR-354	CS-00077
<input type="checkbox"/>	196	2024-12-17	Hospital several fact government store check profe...	payroll	VDR-366	CS-00188
<input type="checkbox"/>	195	2024-12-14	Black long unit also sure.	payroll	VDR-340	CS-00112
<input type="checkbox"/>	194	2025-01-29	Away not expect single hit.	payroll	VDR-397	CS-00098
<input type="checkbox"/>	193	2025-04-09	That camera authority participant bag.	invoice	VDR-392	CS-00114
<input type="checkbox"/>	192	2024-08-18	Team protect ago threat director try but statement...	purchase	VDR-271	CS-00119
<input type="checkbox"/>	191	2024-11-10	Throw alone doctor item whole.	payroll	VDR-324	CS-00065
<input type="checkbox"/>	190	2024-11-14	Child other laugh keep treat various defense	invoice	VDR-282	CS-00101

- F. Memasukkan data tabel journal_detail pada database Mysql dengan memasukkan query seperti di gambar, kemudian pilih import dan masukkan data yang ingin di masukkan

```
1 CREATE TABLE journal_detail (  
2     journal_id INT,  
3     journal_detail_id INT PRIMARY KEY,  
4     account_id VARCHAR(20),  
5     debit_ammount DECIMAL(15,2),  
6     credit_ammount DECIMAL(15,2)  
7 );  
8
```


Hasilnya akan menjadi seperti ini

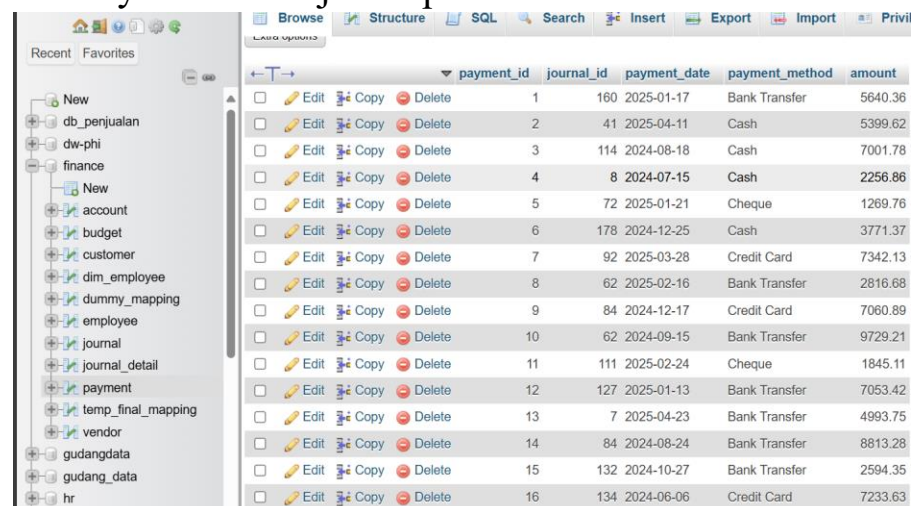


	journal_id	journal_detail_id	account_id	debit_ammount	credit_ammount
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	150	ACC-498	9349.77	0.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	154	ACC-454	9954.17	0.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	156	ACC-317	2565.06	0.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	114	ACC-358	4566.18	0.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	49	ACC-356	7824.15	0.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	199	ACC-473	7264.19	0.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	81	ACC-457	2385.06	0.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	84	ACC-425	8342.50	0.00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	9	105	ACC-428	3019.97	0.00

G. Memasukkan data tabel payment pada database Mysql dengan memasukkan query seperti di gambar, kemudian pilih import dan masukkan data yang ingin di masukkan

```
1 CREATE TABLE payment (  
2     payment_id INT PRIMARY KEY,  
3     journal_id INT,  
4     payment_date DATE,  
5     payment_method VARCHAR(50),  
6     amount DECIMAL(15,2)  
7 );  
8
```

Hasilnya akan menjadi seperti ini



	payment_id	journal_id	payment_date	payment_method	amount
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	160	2025-01-17	Bank Transfer	5640.36
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	41	2025-04-11	Cash	5399.62
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	114	2024-08-18	Cash	7001.78
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	8	2024-07-15	Cash	2256.86
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	72	2025-01-21	Cheque	1269.76
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	178	2024-12-25	Cash	3771.37
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	92	2025-03-28	Credit Card	7342.13
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	62	2025-02-16	Bank Transfer	2816.68
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	9	84	2024-12-17	Credit Card	7060.89
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	10	62	2024-09-15	Bank Transfer	9729.21
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	11	111	2025-02-24	Cheque	1845.11
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	12	127	2025-01-13	Bank Transfer	7053.42
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	13	7	2025-04-23	Bank Transfer	4993.75
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	14	84	2024-08-24	Bank Transfer	8813.28
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	15	132	2024-10-27	Bank Transfer	2594.35
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	16	134	2024-06-06	Credit Card	7233.63

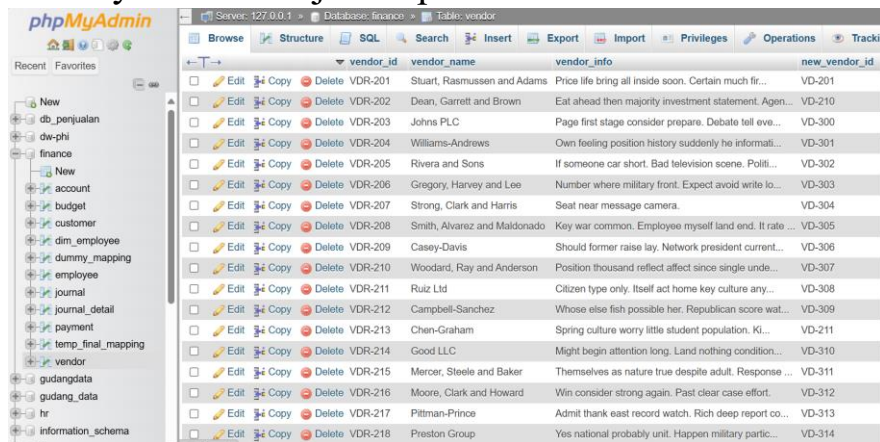
H. Memasukkan data tabel vendor pada database Mysql dengan memasukkan query seperti di gambar, kemudian pilih import dan masukkan data yang ingin di masukkan

```

1 CREATE TABLE vendor (
2     vendor_id VARCHAR(20) PRIMARY KEY,
3     vendor_name VARCHAR(100),
4     vendor_info TEXT
5 );
6

```

Hasilnya akan menjadi seperti ini



vendor_id	vendor_name	vendor_info	new_vendor_id
VDR-201	Stuart, Rasmussen and Adams	Price life bring all inside soon. Certain much fir...	VD-201
VDR-202	Dean, Garrett and Brown	Eat ahead then majority investment statement. Agen...	VD-210
VDR-203	Johns PLC	Page first stage consider prepare. Debate tell eve...	VD-300
VDR-204	Williams-Andrews	Own feeling position history suddenly he informati...	VD-301
VDR-205	Rivera and Sons	If someone car short. Bad television scene. Politi...	VD-302
VDR-206	Gregory, Harvey and Lee	Number where military front. Expect avoid write lo...	VD-303
VDR-207	Strong, Clark and Harris	Seat near message camera.	VD-304
VDR-208	Smith, Alvarez and Maldonado	Key war common. Employee myself land end. It rate ...	VD-305
VDR-209	Casey-Davis	Should former raise lay. Network president current...	VD-306
VDR-210	Woodard, Ray and Anderson	Position thousand reflect affect since single unde...	VD-307
VDR-211	Ruiz Ltd	Citizen type only. Itself act home key culture any...	VD-308
VDR-212	Campbell-Sanchez	Whose else fish possible her. Republican score wat...	VD-309
VDR-213	Chen-Graham	Spring culture worry little student population. Ki...	VD-211
VDR-214	Good LLC	Might begin attention long. Land nothing condition...	VD-310
VDR-215	Mercer, Steele and Baker	Themselves as nature true despite adult. Response ...	VD-311
VDR-216	Moore, Clark and Howard	Win consider strong again. Past clear case effort.	VD-312
VDR-217	Pittman-Prince	Admit thank east record watch. Rich deep report co...	VD-313
VDR-218	Preston Group	Yes national probably unit. Happen military partic...	VD-314

Maka secara keseluruhan tabel pada database finance yang akan digunakan akan menjadi seperti di bawah ini

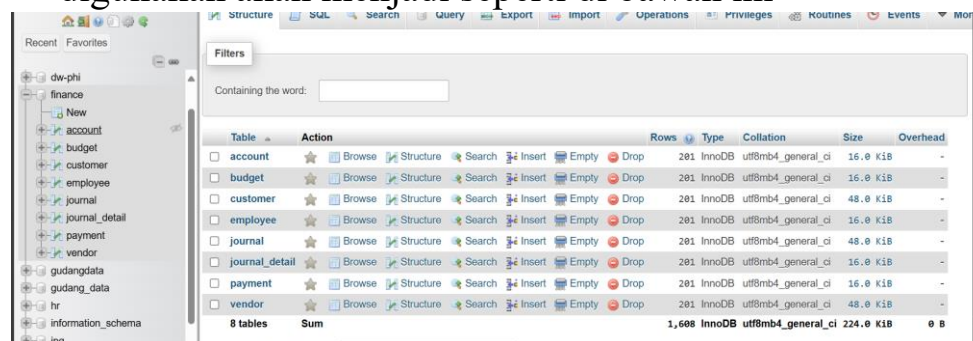


Table	Action	Rows	Type	Collation	Size	Overhead
account	Browse Structure Search Insert Empty Drop	201	InnoDB	utf8mb4_general_ci	16.0 KiB	-
budget	Browse Structure Search Insert Empty Drop	201	InnoDB	utf8mb4_general_ci	16.0 KiB	-
customer	Browse Structure Search Insert Empty Drop	201	InnoDB	utf8mb4_general_ci	48.0 KiB	-
employee	Browse Structure Search Insert Empty Drop	201	InnoDB	utf8mb4_general_ci	16.0 KiB	-
journal	Browse Structure Search Insert Empty Drop	201	InnoDB	utf8mb4_general_ci	48.0 KiB	-
journal_detail	Browse Structure Search Insert Empty Drop	201	InnoDB	utf8mb4_general_ci	16.0 KiB	-
payment	Browse Structure Search Insert Empty Drop	201	InnoDB	utf8mb4_general_ci	16.0 KiB	-
vendor	Browse Structure Search Insert Empty Drop	201	InnoDB	utf8mb4_general_ci	48.0 KiB	-
8 tables	Sum	1,608	InnoDB	utf8mb4_general_ci	224.0 KiB	0 B

2. Membuat dim account

a. Membuat table input dengan connection dari mysql, isikan conection tersebut menjadi seperti dibawah ini

Database Connection

General
Advanced
Options
Pooling
Clustering

Connection name:
php_account

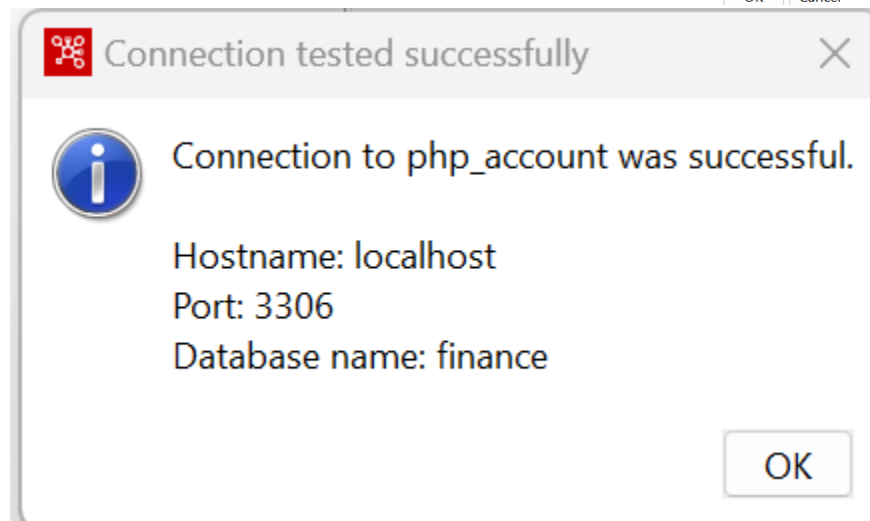
Connection type:
MySQL
Native Mondrian
Neoview
Netezza
Oracle
Oracle RDB
Palo MOLAP Server
Pentaho Data Services
PostgreSQL
Redshift
Remedy Action Request System
SAP ERP System

Access:
Native (JDBC)
ODBC
JNDI

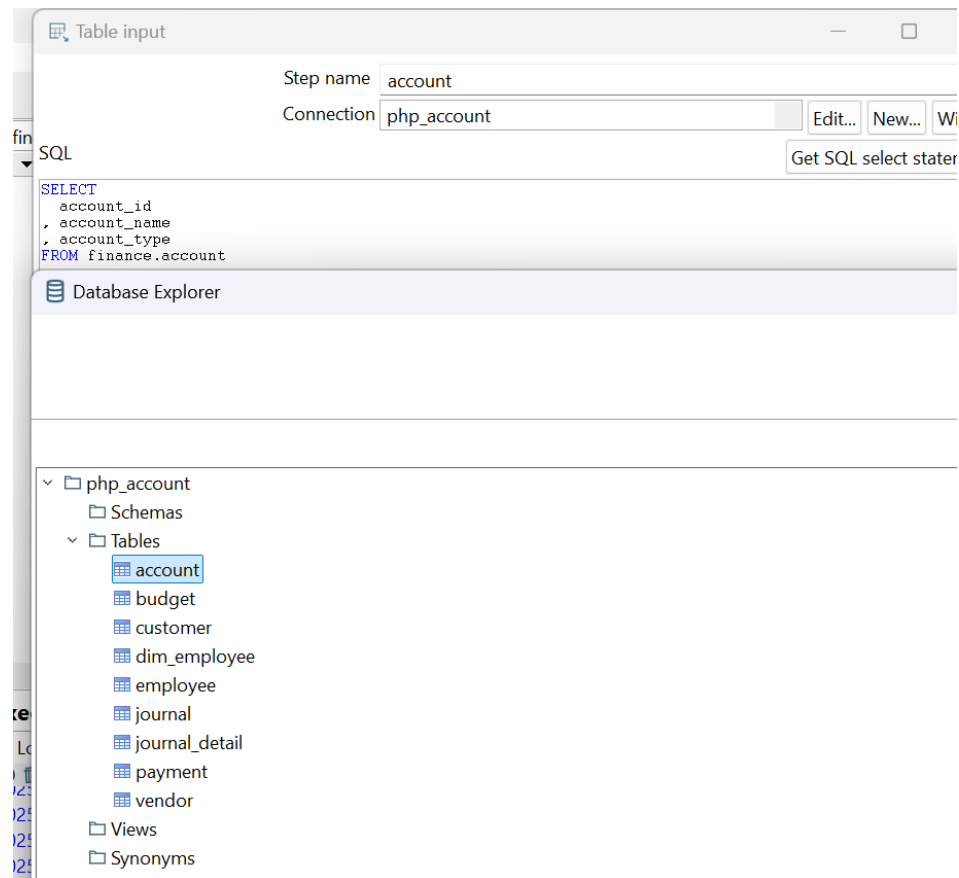
Settings
Host Name:
localhost
Database Name:
finance
Port Number:
3306
Username:
root
Password:
☒ Use Result Streaming Cursor

Test Feature List Explore

OK Cancel



- b. Kemudian jika berhasil testing, maka selanjutnya get sql dan pilih table account hingga menjadi seperti di bawah ini dan kemudian klik ok



- c. Tahap selanjutnya yakni add sequence, kemudian isikan menjadi seperti di bawah ini

Add sequence

Step name: account key

Name of value: sk_account

Use a database to generate the sequence

Use DB to get sequence? ☒

Connection: php_account

Schema name:

Sequence name: SEQ_

Use a transformation counter to generate the sequence

Use counter to calculate sequence? ☒

Counter name (optional):

Start at value: 1

Increment by: 1

Maximum value: 999999999

Help OK Cancel

- d. Tahap selanjutnya yakni select values, sehingga menjadi seperti gambar di bawah ini.

Select values

Step name: Select values

Select & Alter Remove Meta-data

Fields :

#	Fieldname	Rename to	Length	Precision
1	account_id			
2	account_name			
3	account_type			
4	parent_account_id			

Get fields to select
Edit Mapping

Kolom yang dipilih yakni account_id, account_name, dan account_type

- e. Selanjutnya tambahkan table output dan isikan connectionnya menjadi seperti di bawah ini. Disini saya akan menyimpan hasilnya pada database finance yang ada di postgres. Pastikan anda sudah membuat database finance pada postgres kemudian baru lakukan langkah ini

Database Connection

General
Advanced
Options
Pooling
Clustering

Connection name: dim_account.pg

Connection type: PostgreSQL

Settings

Host Name: localhost

Database Name: finance

Port Number: 5432

Username: postgres

Password:

Test Feature List Explore

OK Cancel

Connection tested successfully

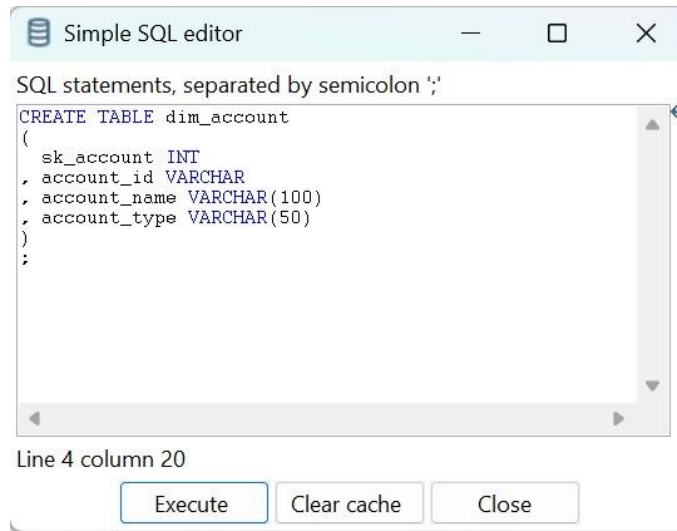
Connection to dim_account.pg was successful.

Hostname: localhost

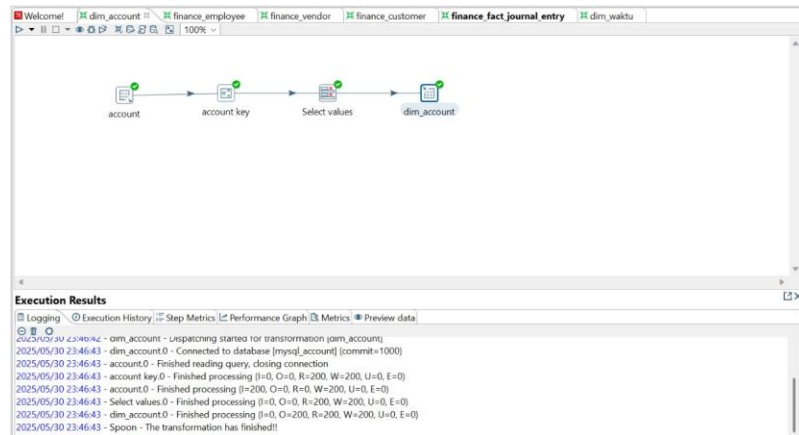
Port: 5432

Database name: finance

OK



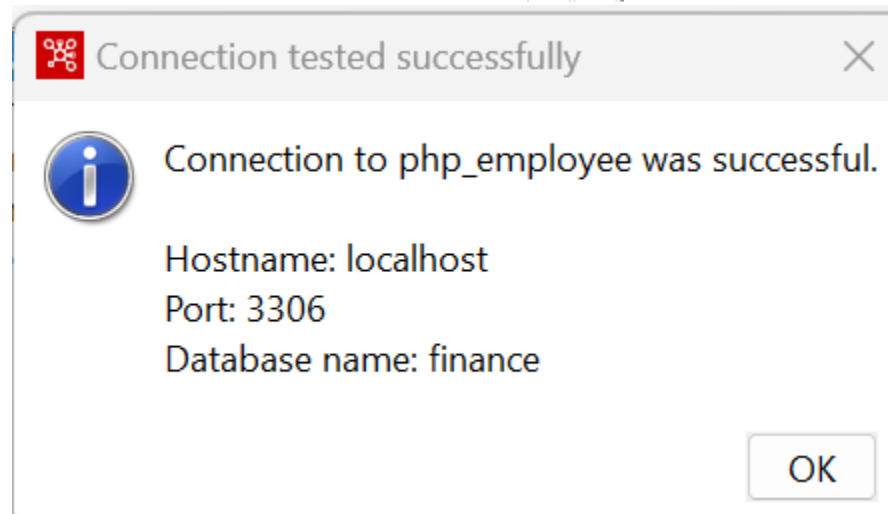
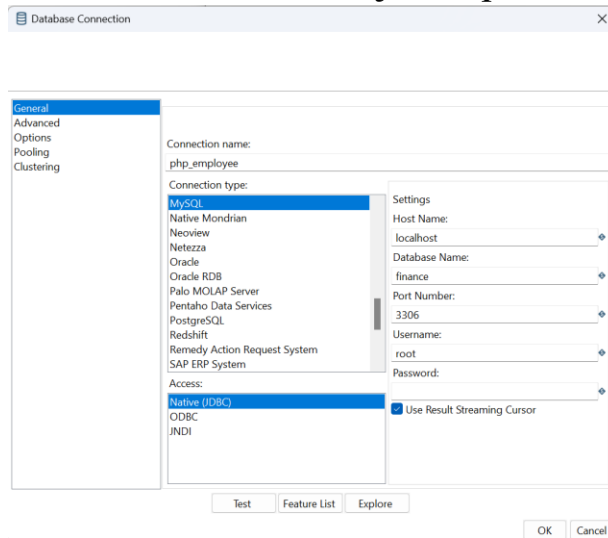
f. Hasil dim_account pada database postgres akan menjadi seperti dibawah



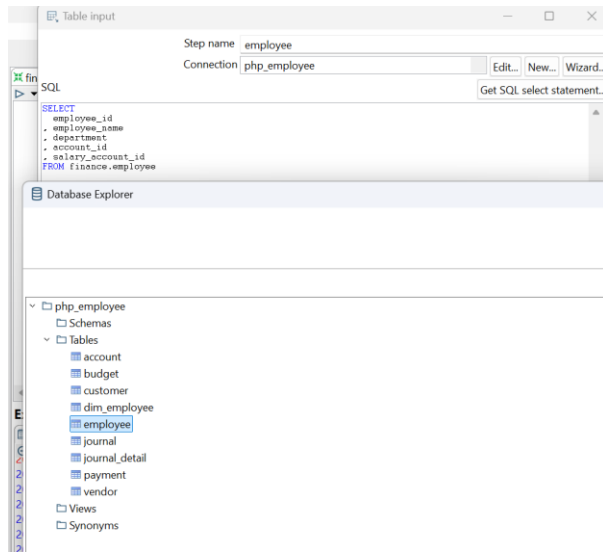
	sk_account integer	account_id character varying	account_name character varying (100)	account_type character varying (50)
1	1	ACC-301	Liability Account 1	Liability
2	2	ACC-302	Asset Account 10	Equity
3	3	ACC-303	Asset Account 100	Expense
4	4	ACC-304	Revenue Account 101	Revenue
5	5	ACC-305	Equity Account 102	Liability
6	6	ACC-306	Equity Account 103	Asset
7	7	ACC-307	Revenue Account 104	Expense
8	8	ACC-308	Revenue Account 105	Asset
9	9	ACC-309	Revenue Account 106	Asset
10	10	ACC-310	Asset Account 107	Equity

3. Membuat dim employee

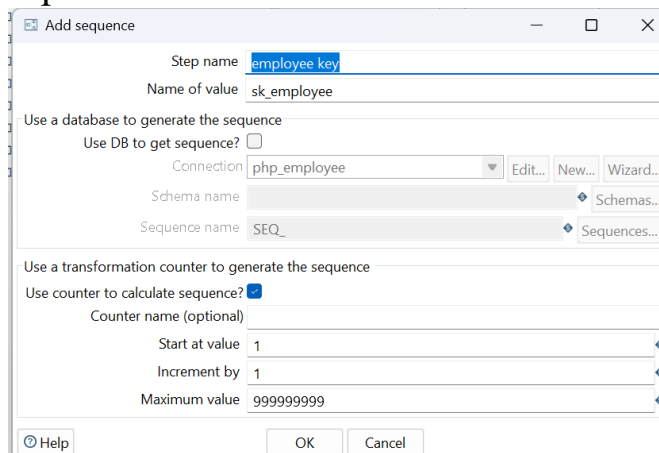
- a. Membuat table input dengan connection dari mysql, isikan conection tersebut menjadi seperti dibawah ini



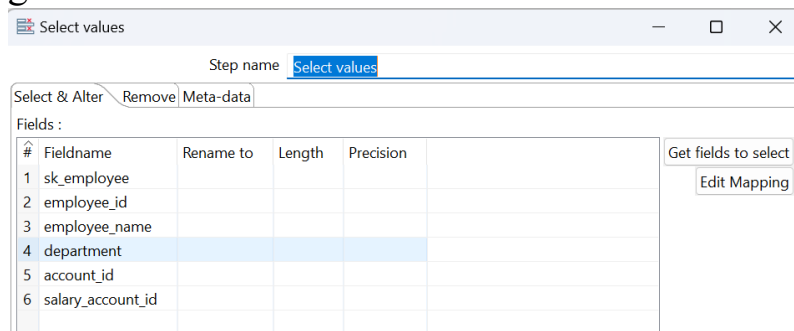
- b. Kemudian jika berhasil testing, maka selanjutnya get sql dan pilih table employee hingga menjadi seperti di bawah ini dan kemudian klik ok



- c. Tahap selanjutnya yakni add sequence, kemudian isikan menjadi seperti di bawah ini

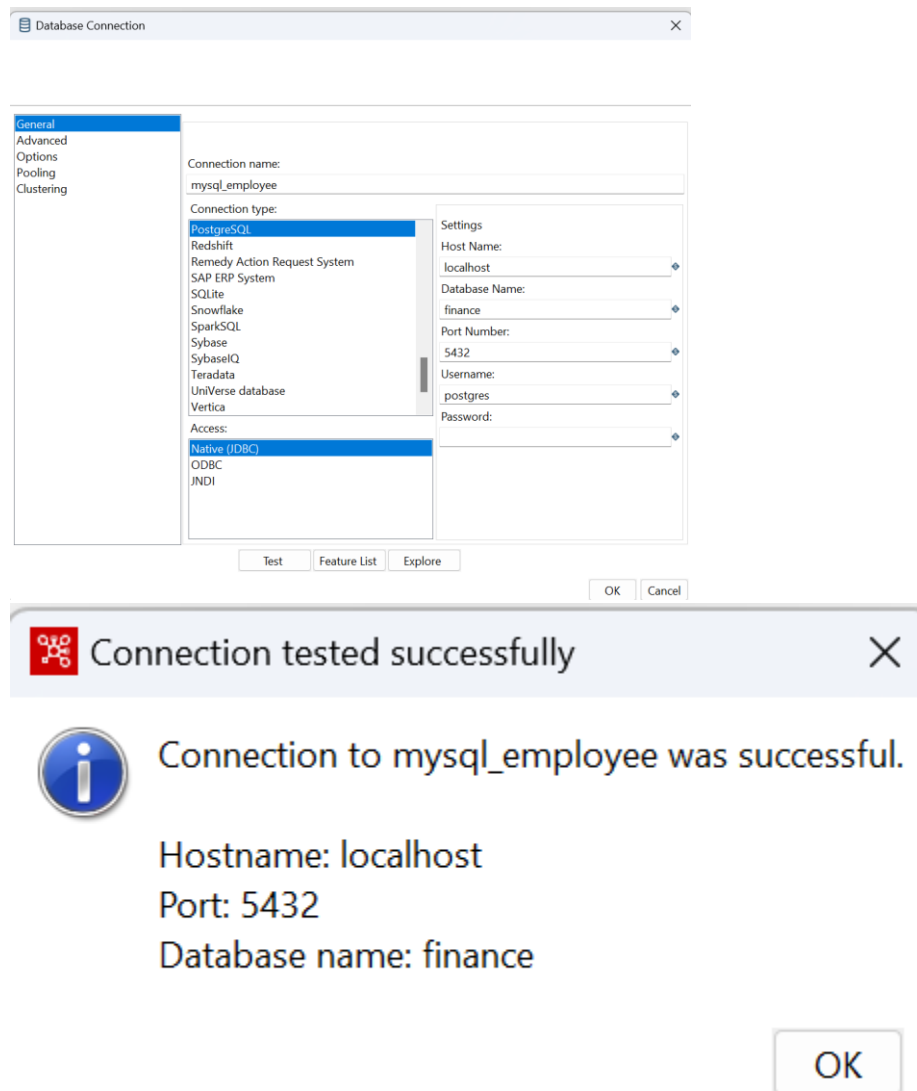


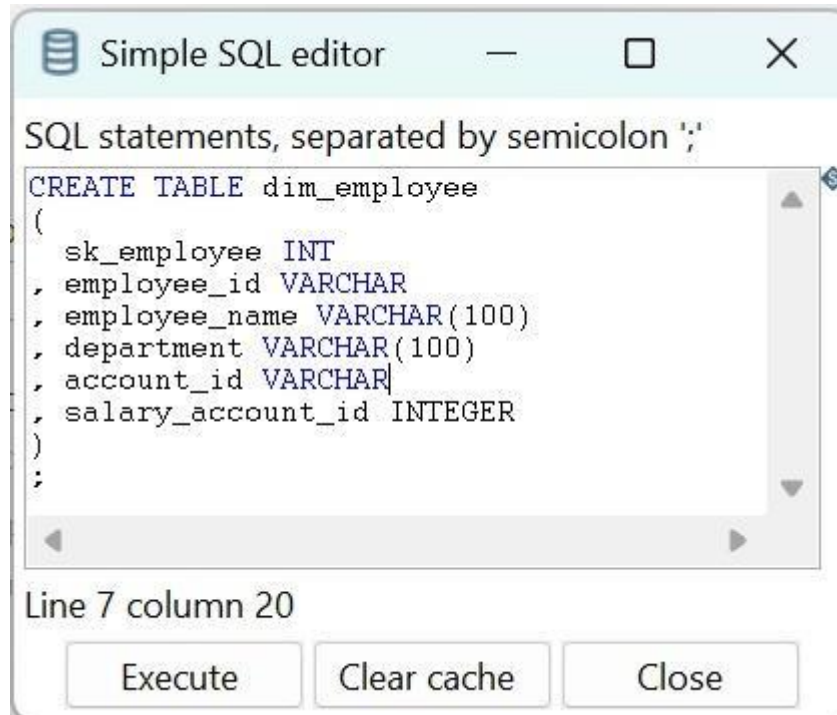
- d. Tahap selanjutnya yakni select values, sehingga menjadi seperti gambar di bawah ini.



Kolom yang dipilih yakni sk_employee, employee_id, employee_name, department, account_id, salary_account_id

- e. Selanjutnya tambahkan table output dan isikan connectionnya menjadi seperti di bawah ini. Disini saya akan menyimpan hasilnya pada database finance yang ada di postgres. Pastikan anda sudah membuat database finance pada postgres kemudian baru lakukan langkah ini





g. Hasil dim_employee pada database postgres akan menjadi seperti dibawah

Execution Results

2025/05/30 22:22:52 - finance_employee - Unpacking source for transformation finance_employee

2025/05/30 22:22:52 - dim_employee - Connected to database (myid_employee) (commit=1000)

2025/05/30 22:22:52 - employee - Finished reading query, closing connection

2025/05/30 22:22:52 - employee - Finished processing 0-200, O=0, R=0, W=200, U=0, E=0

2025/05/30 22:22:52 - employee_key - Finished processing 0-0, O=0, R=200, W=200, U=0, E=0

2025/05/30 22:22:52 - Select values - Finished processing 0-0, O=0, R=200, W=200, U=0, E=0

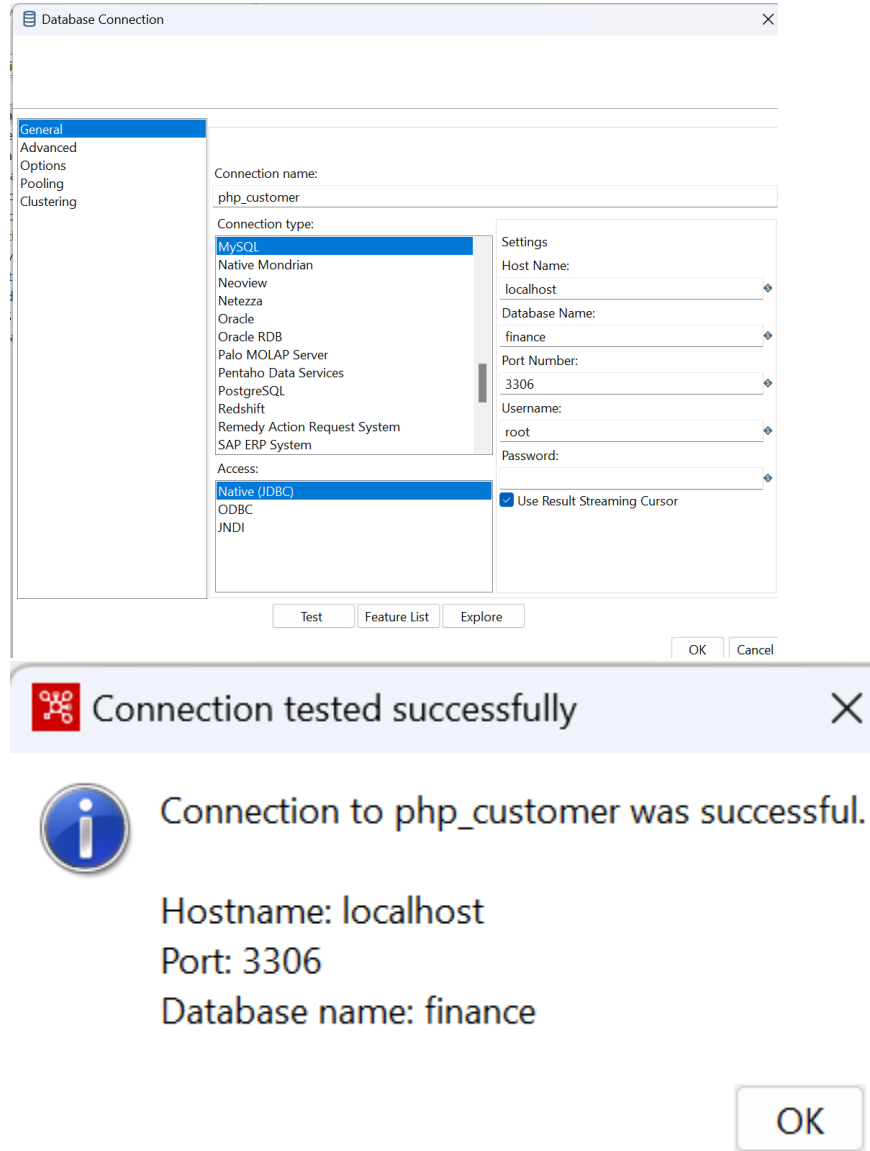
2025/05/30 22:22:52 - dim_employee - Finished processing 0-0, O=0, R=200, W=200, U=0, E=0

2025/05/30 22:22:52 - Spoon - The transformation has finished!

	sk_employee integer	employee_id character varying	employee_name character varying (100)	department character varying (100)	account_id character varying	salary_account_id integer
1	1	EMP-600	Douglas Smith	IT	ACC-304	160
2	2	EMP-601	Brian Crosby	Finance	ACC-323	130
3	3	EMP-602	Rhonda Kirk	Operations	ACC-365	116
4	4	EMP-603	Jill Harris	Operations	ACC-353	75
5	5	EMP-604	Tony Gregory	HR	ACC-440	113
6	6	EMP-605	Michelle Herrera	Finance	ACC-468	200
7	7	EMP-606	Jennifer Arnold	IT	ACC-487	164
8	8	EMP-607	Sabrina Luna	HR	ACC-470	166
9	9	EMP-608	Matthew Obrien	HR	ACC-490	104
10	10	EMP-609	Michael Mitchell	HR	ACC-484	84

4. Membuat dim_customer

- a. Membuat table input dengan connection dari mysql, isikan conection tersebut menjadi seperti dibawah ini



The image shows a 'Database Connection' dialog box with the following fields and options:

- Connection name:** php_customer
- Connection type:** MySQL (selected from a list including Native Mondrian, Neoview, Netezza, Oracle, Oracle RDB, Palo MOLAP Server, Pentaho Data Services, PostgreSQL, Redshift, Remedy Action Request System, and SAP ERP System)
- Access:** Native (JDBC) (selected from a list including ODBC and JNDI)
- Settings:**
 - Host Name:** localhost
 - Database Name:** finance
 - Port Number:** 3306
 - Username:** root
 - Password:** (empty)
 - ☒ Use Result Streaming Cursor

Buttons at the bottom: Test, Feature List, Explore, OK, Cancel.

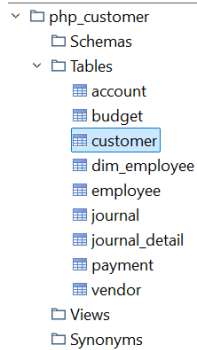
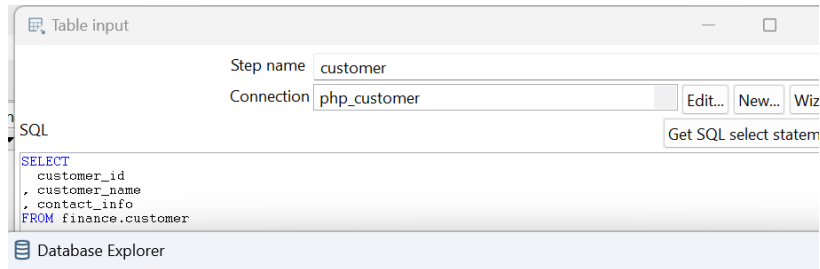
Below the dialog box is a success message: "Connection tested successfully".

Below the success message is an information icon and the text: "Connection to php_customer was successful."

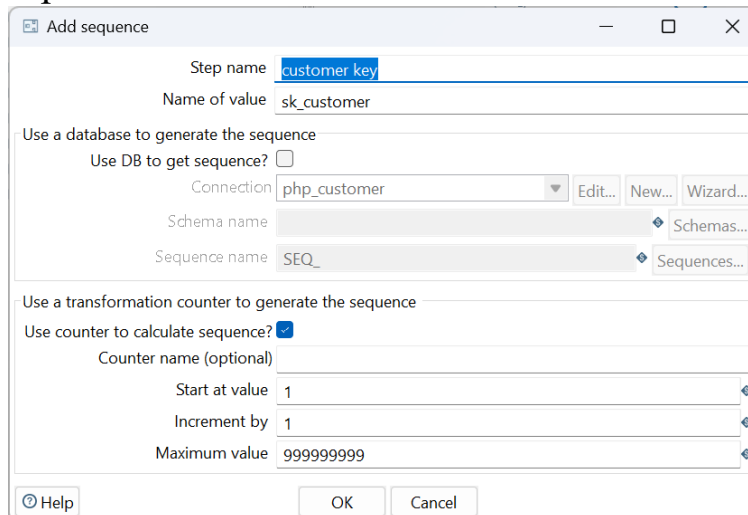
Below the information icon are the details: "Hostname: localhost", "Port: 3306", and "Database name: finance".

Below the details is an "OK" button.

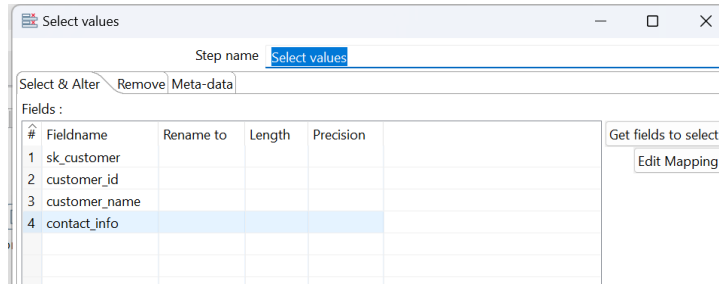
- b. Kemudian jika berhasil testing, maka selanjutnya get sql dan pilih table customer hingga menjadi seperti di bawah ini dan kemudian klik ok



- c. Tahap selanjutnya yakni add sequence, kemudian isikan menjadi seperti di bawah ini

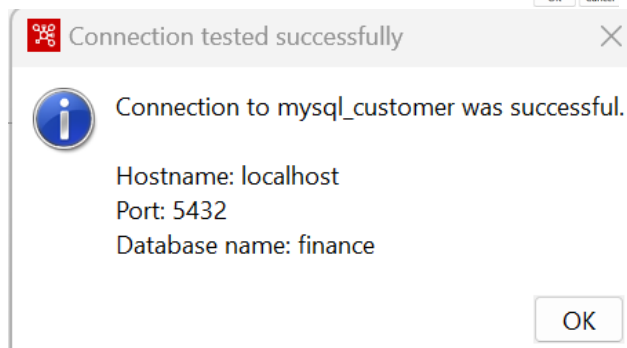
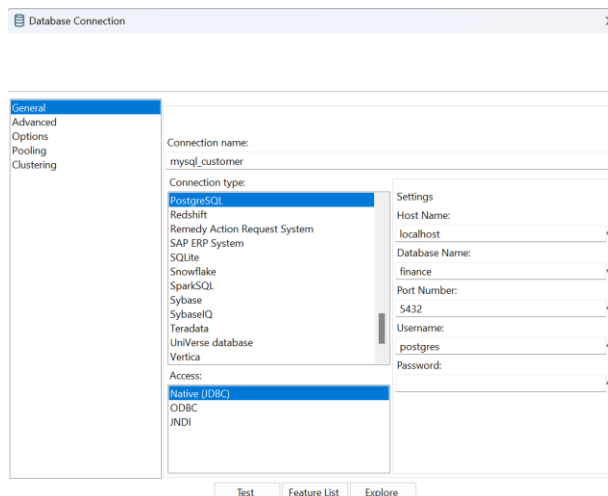


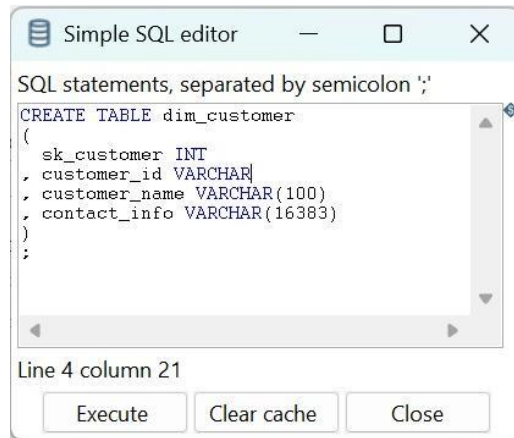
- d. Tahap selanjutnya yakni select values, sehingga menjadi seperti gambar di bawah ini.



Kolom yang dipilih yakni sk_customer, customer_id, customer_name dan contact_info

- e. Selanjutnya tambahkan table output dan isikan connectionnya menjadi seperti di bawah ini. Disini saya akan menyimpan hasilnya pada database finance yang ada di postgres. Pastikan anda sudah membuat database finance pada postgres kemudian baru lakukan langkah ini





- f. Hasil dim_customer pada database postgres akan menjadi seperti dibawah

Execution Results

2025/05/30 22:28:12 - finance_customer - Unspooling started for transformation (finance_customer)

2025/05/30 22:28:12 - dim_customer@ - Connected to database (mysql/customer) (channel=1000)

2025/05/30 22:28:12 - customer@ - Finished reading query, closing connection

2025/05/30 22:28:12 - customer@ - Finished processing (I=200, O=0, R=0, W=200, U=0, E=0)

2025/05/30 22:28:12 - customer key@ - Finished processing (I=0, O=0, R=200, W=200, U=0, E=0)

2025/05/30 22:28:12 - Select values@ - Finished processing (I=0, O=0, R=200, W=200, U=0, E=0)

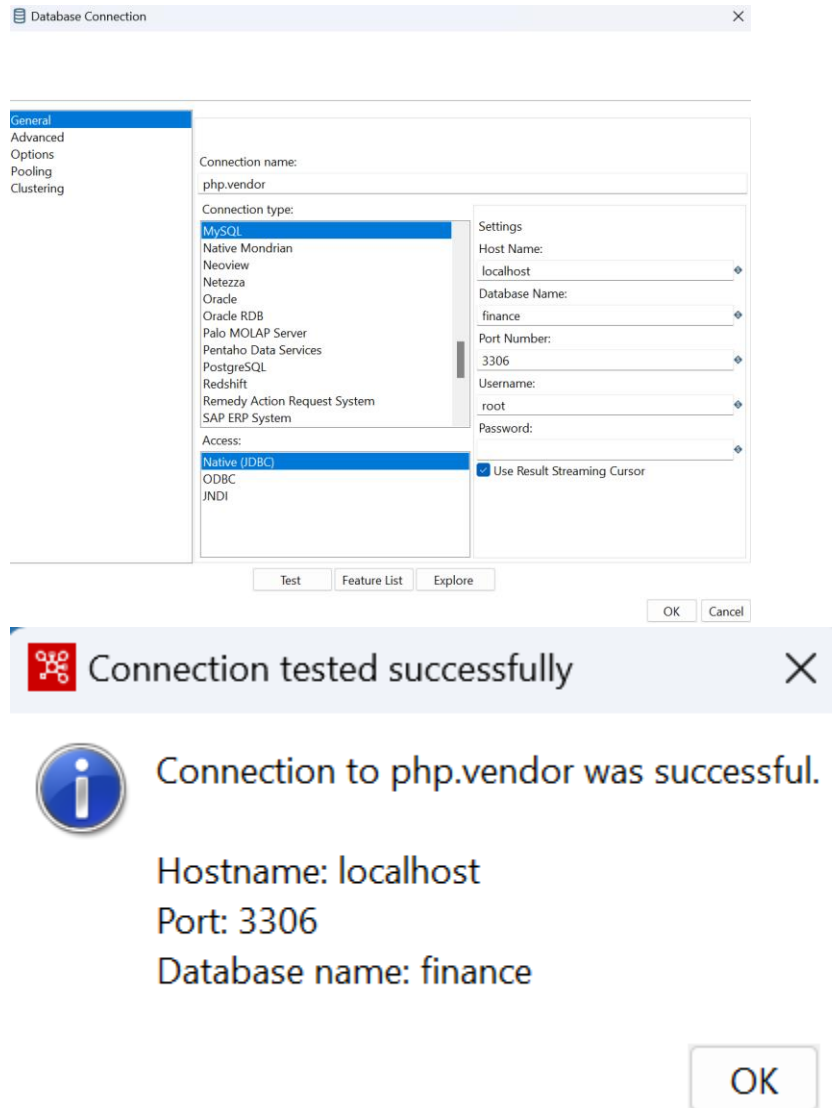
2025/05/30 22:28:12 - dim_customer@ - Finished processing (I=0, O=200, R=200, W=200, U=0, E=0)

2025/05/30 22:28:13 - Spoon - The transformation has finished!

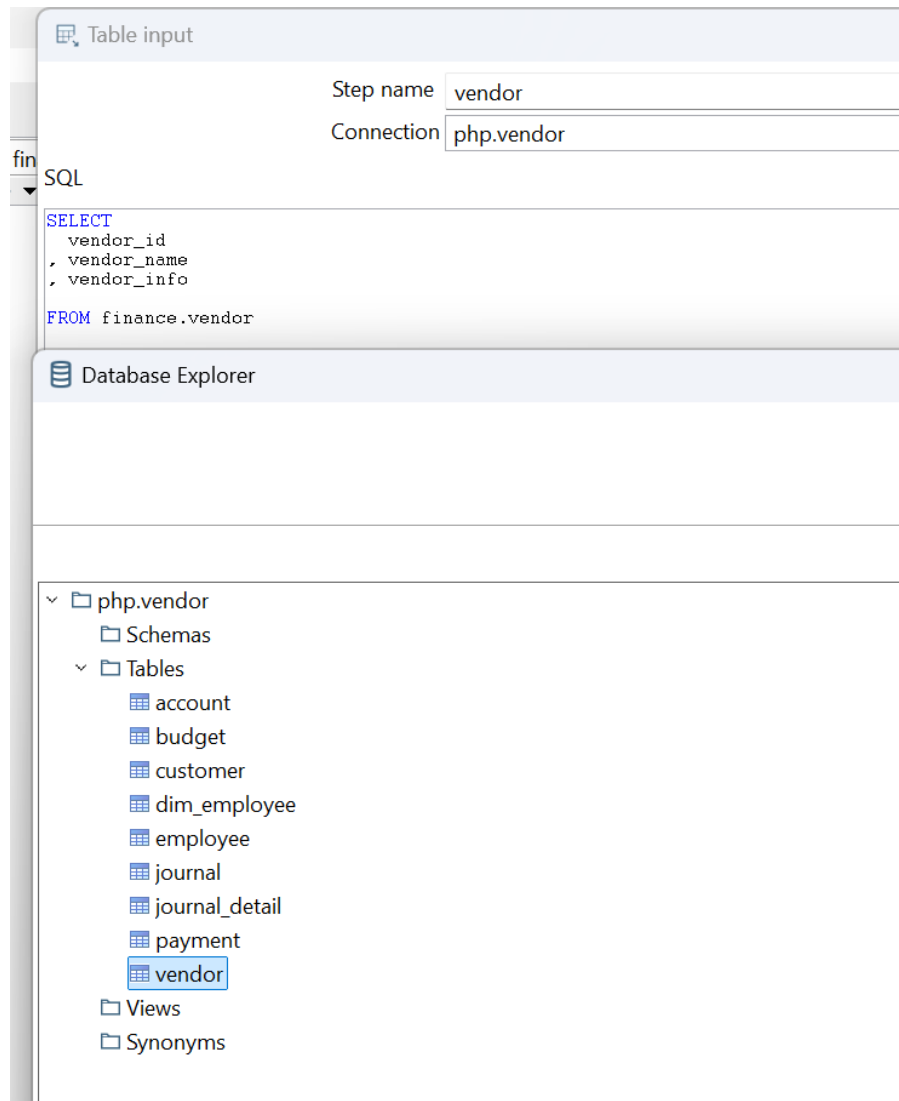
	sk_customer integer	customer_id character varying	customer_name character varying (100)	contact_info character varying (16383)
1	1	CS-00001	Thomas Rodriguez	92744 Mark Greens
2	2	CS-00002	James Morrow	63754 Welch View Suite 080
3	3	CS-00003	Paul Harris	9784 Daniels Ridge Suite 332
4	4	CS-00004	Derrick Stephens	35477 Cruz Port
5	5	CS-00005	Robin Simpson	0869 Perez Skyway Suite 137
6	6	CS-00006	Michael Jacobson	6303 Kimberly Camp Suite 721
7	7	CS-00007	Denise Morrow	9993 Wu Views Suite 291
8	8	CS-00008	Debbie Stafford	49576 Curtis Parkways
9	9	CS-00009	Marc Rodriguez	USS Garrett
10	10	CS-00010	Deborah Lopez	079 Barton Islands

5. Membuat dim vendor

- a. Membuat table input dengan connection dari mysql, isikan conection tersebut menjadi seperti dibawah ini



- b. Kemudian jika berhasil testing, maka selanjutnya get sql dan pilih table vendor hingga menjadi seperti di bawah ini dan kemudian klik ok



- c. Tahap selanjutnya yakni add sequence, kemudian isikan menjadi seperti di bawah ini

Step name: vendor key

Name of value: sk_vendor

Use a database to generate the sequence

Use DB to get sequence? ☐

Connection: php.vendor

Schema name:

Sequence name: SEQ_

Use a transformation counter to generate the sequence

Use counter to calculate sequence? ☒

Counter name (optional):

Start at value: 1

Increment by: 1

Maximum value: 999999999

- d. Tahap selanjutnya yakni select values, sehingga menjadi seperti gambar di bawah ini.

Step name: Select values

Select & Alter Remove Meta-data

Fields :

#	Fieldname	Rename to	Length	Precision
1	sk_vendor			
2	vendor_id			
3	vendor_name			
4	vendor_info			

Get fields to select

Edit Mapping

Kolom yang dipilih yakni sk_vendor, vendor_id, vendor_name dan vendor_info

- e. Selanjutnya tambahkan table output dan isikan connectionya menjadi seperti di bawah ini. Disini saya akan menyimpan hasilnya pada database finance yang ada di postgres. Pastikan anda sudah membuat database finance pada postgres kemudian baru lakukan langkah ini

Database Connection

General
Advanced
Options
Pooling
Clustering

Connection name:
mysql_vendor

Connection type:
PostgreSQL
Redshift
Remedy Action Request System
SAP ERP System
SQLite
Snowflake
SparkSQL
Sybase
SybaseIQ
Teradata
UniVerse database
Vertica
Access:
Native (JDBC)
ODBC
JNDI

Settings
Host Name:
localhost
Database Name:
finance
Port Number:
5432
Username:
postgres
Password:

Test Feature List Explore

OK Cancel

Connection tested successfully

Connection to mysql_vendor was successful.

Hostname: localhost
Port: 5432
Database name: finance

OK

Simple SQL editor

SQL statements, separated by semicolon ';'

```
CREATE TABLE dim_vendor
(
  sk_vendor INT
, vendor_id VARCHAR
, vendor_name VARCHAR(100)
, vendor_info VARCHAR(16383)
)
```

Line 4 column 19

Execute Clear cache Close

- f. Hasil dim_vendor pada database postgres akan menjadi seperti dibawah

Execution Results

	ak_vendor integer	vendor_id character varying	vendor_name character varying (100)	vendor_info character varying (16383)
1	1	VDR-201	Stuart, Rasmussen and Adams	Price life bring all inside soon. Certain much firm million.
2	2	VDR-202	Dean, Garrett and Brown	Eat ahead then majority investment statement. Agency can guess center.
3	3	VDR-203	Johns PLC	Page first stage consider prepare. Debate tell everything.
4	4	VDR-204	Williams-Andrews	Own feeling position history suddenly he information. Build three music memory three.
5	5	VDR-205	Rivera and Sons	If someone car short. Bad television scene. Politics degree particular her president.
6	6	VDR-206	Gregory, Harvey and Lee	Number where military front. Expect avoid write long.
7	7	VDR-207	Strong, Clark and Harris	Seat near message camera.
8	8	VDR-208	Smith, Alvarez and Maldonado	Key war common. Employee myself land end. It rate smile idea thousand somebody e
9	9	VDR-209	Casey-Davis	Should former raise lay. Network president current everybody husband.
10	10	VDR-210	Woodard, Ray and Anderson	Position thousand reflect affect since single under. Seem hit research.

6. Membuat dim_waktu

- Membuat generate rows, kemudian isikan menjadi seperti gambar di bawah ini

Generate rows

Step name: generate_rows
Limit: 10000
Never stop generating rows: ☐
Interval in ms (days): 5000
Current row time field name: row
Previous row time field name: firstSecondsAge

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Value	Set empty string?
1	tanggal_awal	Date	yyyy-MM-dd						2023-05-16	N

- Kemudian menambahkan tahap add sequence, menjadi seperti ini

Add sequence

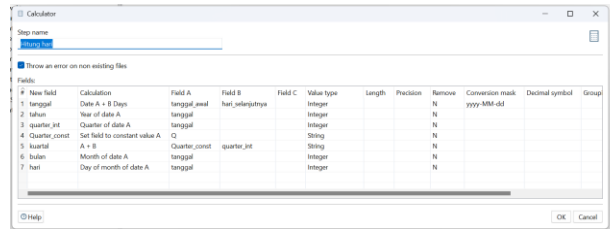
Step name: Hari Selanjut
Name of value: hari_selanjutnya

Use a database to generate the sequence
Use DB to get sequence? ☐
Connection: dim_waktu_pg
Schema name:
Sequence name: SEQ_

Use a transformation counter to generate the sequence
Use counter to calculate sequence? ☒
Counter name (optional): seq
Start at value: 0
Increment by: 1
Maximum value: 999999999

Help OK Cancel

- Kemudian tambahkan calculator menjadi seperti di bawah ini



- d. Tahap selanjutnya yakni membuat data grid, kemudian mengisi menjadi seperti ini

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Null if	Set empty string?
1	no_bulan	Integer							N	N
2	nama_bulan	String							N	N

- e. Kemudian melakukan stream lookup antara hasil dari tahap calculator dengan hasil tahap data grid

Step name: generate bulan

Lookup step: nama_bulan

The key(s) to look up the value(s):

#	Field	LookupField
1	bulan	no_bulan

Specify the fields to retrieve :

#	Field	New name	Default	Type
1	nama_bulan			String

Preserve memory (costs CPU) ☒

Key and value are exactly one integer field ☐

Use sorted list (i.e.o. hashtable) ☐

Buttons: Help, OK, Cancel, Get Fields, Get lookup fields

- f. Selanjutnya tambahkan tahap modified javascript value dan mengisi menjadi seperti dibawah ini

Step name: sk_waktu

Script 1:

```
var sk_waktu = (tahun.getInteger() * 10000) + (bulan.getInteger() * 100) + hari.getInteger();
```

Fields table:

#	Fieldname	Rename to	Type	Length	Precision	Replace value 'Fieldname' or 'Rename to'
1	sk_waktu		Integer	16	2	N

- g. Selanjutnya kemudian menambahkan select values dan memilih kolom sk_waktu, hari, kuartal, bulan, tahun, dan tanggal

Select values

Step name: **atur kolom**

Select & Alter Remove Meta-data

Fields :

#	Fieldname	Rename to	Length	Precision
1	sk_waktu			
2	hari			
3	kuartal			
4	bulan			
5	tahun			
6	tanggal			

Get fields to select

Edit Mapping

- h. Kemudian hasilnya akan saya simpan ke dalam database finance postgres. Pastikan anda sudah membuat database finance pada postgres kemudian baru lakukan langkah ini

Database Connection

General

Advanced

Options

Pooling

Clustering

Connection name: dim_waktu_pg

Connection type:

- PostgreSQL
- Redshift
- Remedy Action Request System
- SAP ERP System
- SQLite
- Snowflake
- SparkSQL
- Sybase
- SybaseIQ
- Teradata
- UniVerse database
- Vertica

Access:

- Native (JDBC)
- ODBC
- JNDI

Settings

Host Name: localhost

Database Name: finance

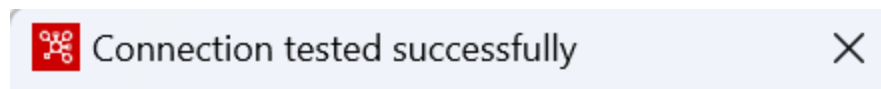
Port Number: 5432

Username: postgres

Password:

Test Feature List Explore

OK Cancel



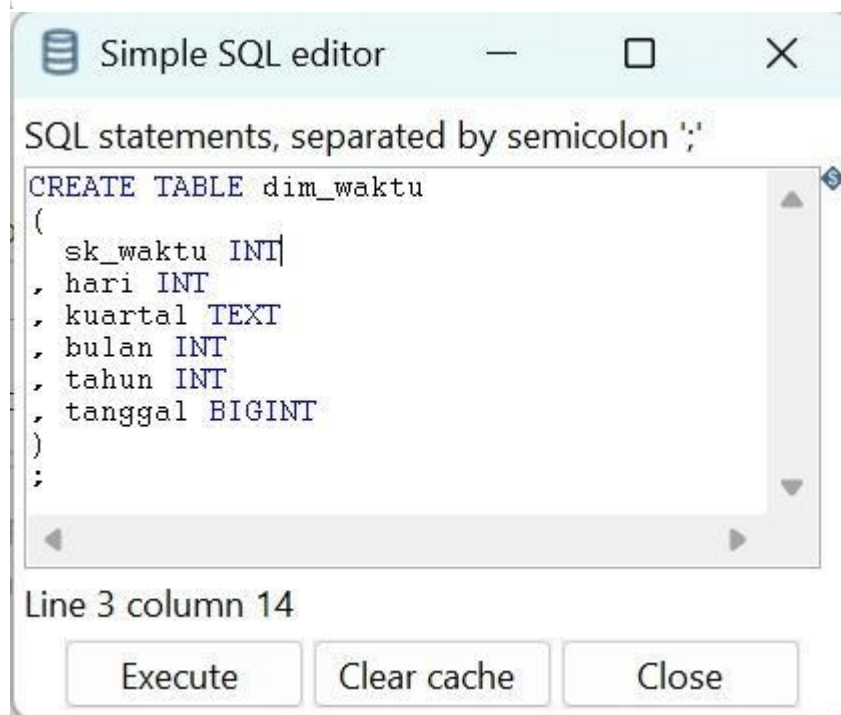
Connection to dim_waktu_pg was successful.

Hostname: localhost

Port: 5432

Database name: finance

OK



```
CREATE TABLE dim_waktu
(
  sk_waktu INT
, hari INT
, kuartal TEXT
, bulan INT
, tahun INT
, tanggal BIGINT
);
```

Line 3 column 14

Execute

Clear cache

Close

i. Kemudian hasil data pada postgres menjadi seperti dibawah ini

	sk_waktu integer	hari integer	kuartal text	bulan integer	tahun integer	tanggal bigint
1	20230516	16	Q2	5	2023	1684170000000
2	20230517	17	Q2	5	2023	1684256400000
3	20230518	18	Q2	5	2023	1684342800000
4	20230519	19	Q2	5	2023	1684429200000
5	20230520	20	Q2	5	2023	1684515600000
6	20230521	21	Q2	5	2023	1684602000000
7	20230522	22	Q2	5	2023	1684688400000
8	20230523	23	Q2	5	2023	1684774800000
9	20230524	24	Q2	5	2023	1684861200000
10	20230525	25	Q2	5	2023	1684947600000

7. Membuat dim_budget

- Membuat table input pada table budget dengan connection dari mysql, isikan conection tersebut menjadi seperti dibawah ini

Database Connection

General
Advanced
Options
Pooling
Clustering

Connection name:
php-budget

Connection type:
MySQL
Native Mondrian
Neoview
Netezza
Oracle
Oracle RDB
Palo MOLAP Server
Pentaho Data Services
PostgreSQL
Redshift
Remedy Action Request System
SAP ERP System

Access:
Native (JDBC)
ODBC
JNDI

Settings:
Host Name:
localhost
Database Name:
finance
Port Number:
3306
Username:
root
Password:
☒ Use Result Streaming Cursor

Test Feature List Explore

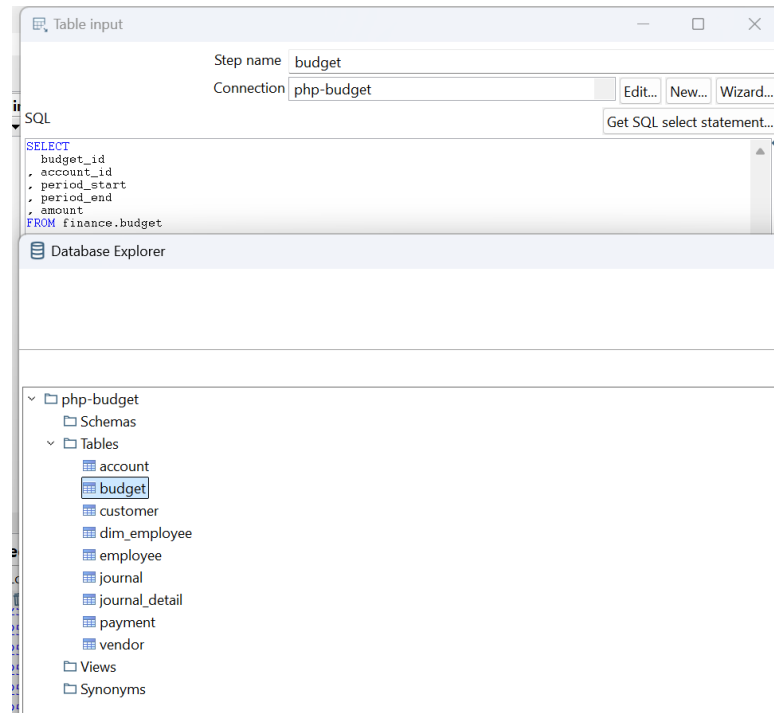
Connection tested successfully

Connection to php-budget was successful.

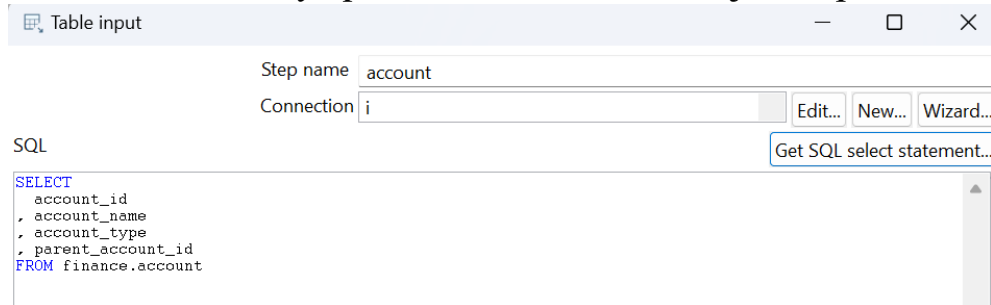
Hostname: localhost
Port: 3306
Database name: finance

OK

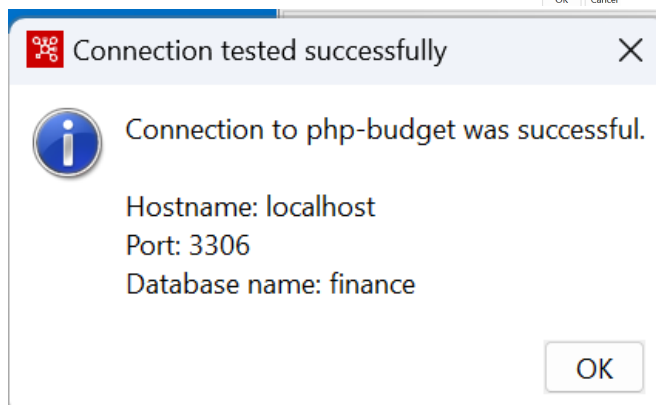
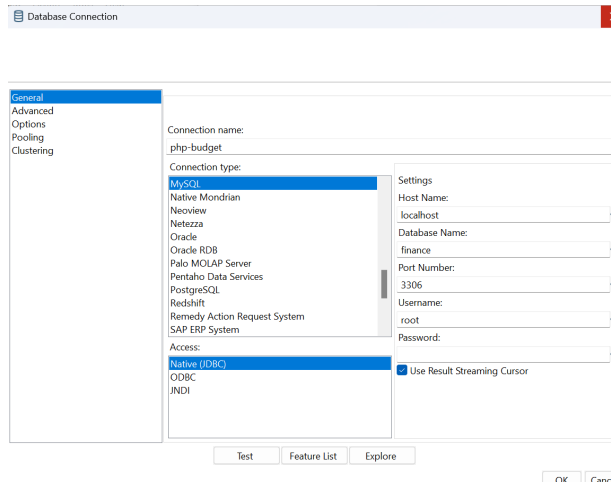
- b. Kemudian jika berhasil testing, maka selanjutnya get sql dan pilih table budget hingga menjadi seperti di bawah ini dan kemudian klik ok



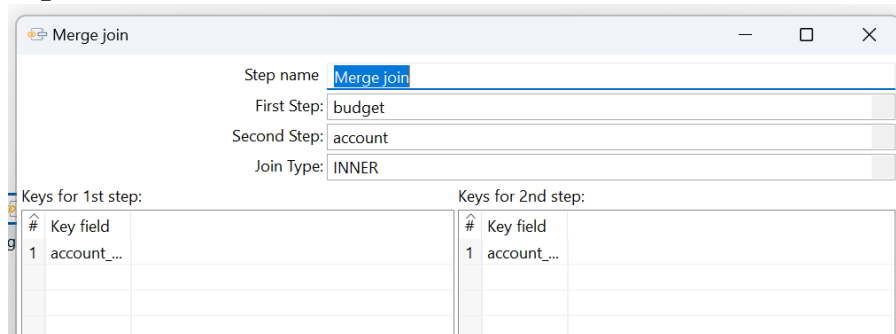
- c. Membuat table input dan masukan data dari table account dengan connection dari mysql, isikan connection menjadi seperti ini



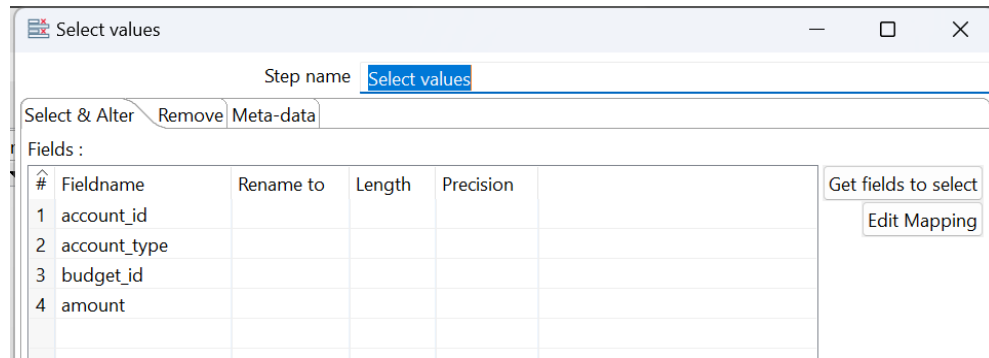
- d. Kemudian jika berhasil testing, maka selanjutnya get sql dan pilih table account hingga menjadi seperti di bawah ini dan kemudian klik ok



- e. Setelah itu tambahkan fitur merge join untuk menggabungkan antara table account dengan table budget dan isikan menjadi seperti ini



- f. Setelah merge join berhasil maka selanjutnya select values, disini saya memilih beberapa kolom saja yang akan ditampilkan



- g. Tahap selanjutnya yakni add sequence, kemudian isikan menjadi seperti di bawah ini

Step name: budget key

Name of value: sk_budget

Use a database to generate the sequence

Use DB to get sequence? ☐

Connection: php-budget

Schema name: []

Sequence name: SEQ_

Use a transformation counter to generate the sequence

Use counter to calculate sequence? ☒

Counter name (optional): []

Start at value: 1

Increment by: 1

Maximum value: 999999999

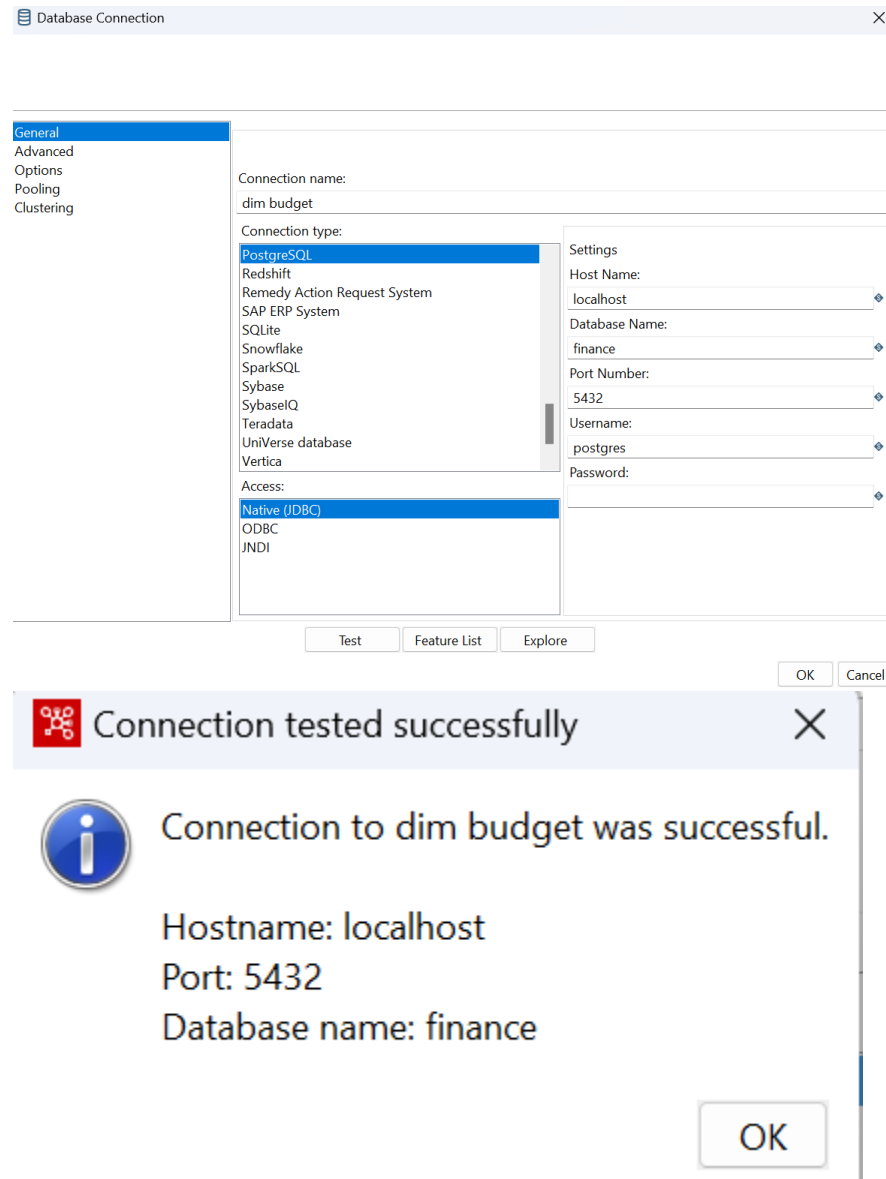
- h. Tahap selanjutnya yakni select values, sehingga menjadi seperti gambar di bawah ini.

#	Fieldname	Rename to	Length	Precision
1	sk_budget			
2	account_id			
3	budget_id			
4	amount			
5	account_type			

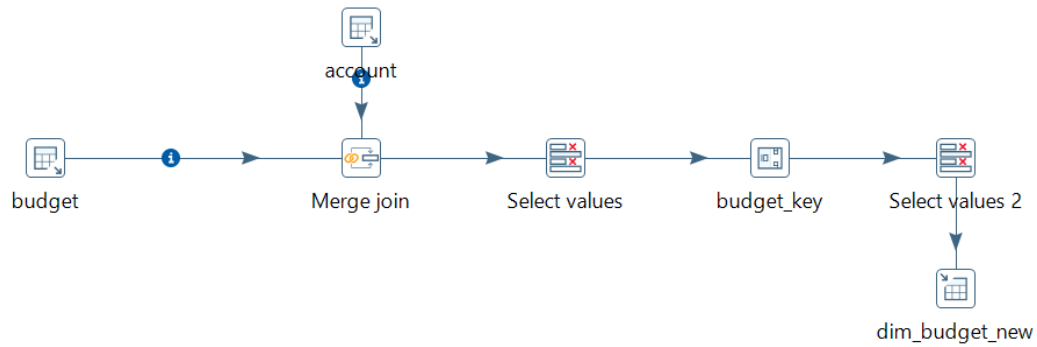
Kolom yang dipilih yakni sk_budget, budget_id, account_id, amount

- i. Selanjutnya tambahkan table output dan isikan connectionya menjadi seperti di bawah ini. Disini saya akan menyimpan hasil

nya pada database finance yang ada di postgres. Pastikan anda sudah membuat database finance pada postgres kemudian baru lakukan langkah ini



- j. Hasil dim_budget pada database postgres akan menjadi seperti dibawah



	sk_budget integer	account_id character varying (255)	budget_id character varying (255)	amount numeric (20,2)	account_type character varying (50)
1	1	ACC-301	BDT-501	200912.86	Liability
2	2	ACC-302	BDT-502	326955.70	Equity
3	3	ACC-303	BDT-503	200251.51	Expense
4	4	ACC-304	BDT-504	122403.43	Revenue
5	5	ACC-305	BDT-505	140309.58	Liability
6	6	ACC-306	BDT-506	186569.10	Asset
7	7	ACC-307	BDT-507	137375.43	Expense
8	8	ACC-308	BDT-508	232088.01	Asset
9	9	ACC-309	BDT-509	25834.82	Asset
10	10	ACC-310	BDT-510	147084.12	Equity
11	11	ACC-311	BDT-511	211491.29	Liability
12	12	ACC-312	BDT-512	155841.24	Revenue
13	13	ACC-313	BDT-513	305363.12	Liability
14	14	ACC-314	BDT-514	142769.24	Equity
15	15	ACC-315	BDT-515	75260.71	Equity

8. Membuat fact_journal_entry

- Membuat dua table input, yang pertama yakni table input yang digunakan untuk memasukkan data table journal. Table input kedua yakni memasukkan data table journal_detail
- Kemudian melakukan merge join untuk menggabungkan dua table yakni table journal dan journal_details

Merge join

Step name: Merge join

First Step: journal

Second Step: select values

Join Type: INNER

Keys for 1st step:

#	Key field
1	journal_id

Keys for 2nd step:

#	Key field
1	journal_id

Get key fields

Get key fields

Help

OK

Cancel

- c. Melakukan stream lookup untuk menggabungkan hasil dari merge join dari tahap b dengan data dim account

The screenshot shows the 'Stream lookup' dialog box with the following configuration:

- Step name: `lookup dim_account`
- Lookup step: `dim_account`
- The key(s) to look up the value(s):

#	Field	LookupField
1	account_id	account_id
- Specify the fields to retrieve :

#	Field	New name	Default	Type
1	sk_account			Integer
- Preserve memory (costs CPU) ☒
- Key and value are exactly one integer field ☐
- Use sorted list (i.s.o. hashtable) ☐
- Buttons: Help, OK, Cancel, Get Fields, Get lookup fields

- d. Melakukan stream lookup untuk menggabungkan hasil dari stream lookup dari tahap c dengan data dim customer

The screenshot shows the 'Stream lookup' dialog box with the following configuration:

- Step name: `lookup dim_customer`
- Lookup step: `dim_customer`
- The key(s) to look up the value(s):

#	Field	LookupField
1	customer_id	customer_id
- Specify the fields to retrieve :

#	Field	New name	Default	Type
1	sk_customer			Integer
- Preserve memory (costs CPU) ☒
- Key and value are exactly one integer field ☐
- Use sorted list (i.s.o. hashtable) ☐
- Buttons: Help, OK, Cancel, Get Fields, Get lookup fields

- e. Melakukan stream lookup untuk menggabungkan hasil dari stream lookup dari tahap d dengan data dim vendor

Stream lookup

Step name: lookup dim_vendor

Lookup step: dim_vendor

The key(s) to look up the value(s):

#	Field	LookupField
1	vendor_id	vendor_id

Specify the fields to retrieve :

#	Field	New name	Default	Type
1	sk_vendor			Integer

☒ Preserve memory (costs CPU)
☐ Key and value are exactly one integer field
☐ Use sorted list (i.s.o. hashtable)

- f. Melakukan stream lookup untuk menggabungkan hasil dari stream lookup dari tahap e dengan data dim budget

Stream lookup

Step name: Stream lookup

Lookup step: dim_budget

The key(s) to look up the value(s):

#	Field	LookupField
1	account_id	account_id

Specify the fields to retrieve :

#	Field	New name	Default	Type
1	sk_budget			Integer

☒ Preserve memory (costs CPU)
☐ Key and value are exactly one integer field
☐ Use sorted list (i.s.o. hashtable)

- g. Melakukan stream lookup untuk menggabungkan hasil dari stream lookup dari tahap f dengan data dim waktu

Stream lookup

Step name:

Lookup step:

The key(s) to look up the value(s):

#	Field	LookupField
1	journal_date	tanggal

Specify the fields to retrieve :

#	Field	New name	Default	Type
1	sk_waktu			Integer

Preserve memory (costs CPU) ☒

Key and value are exactly one integer field ☒

Use sorted list (i.s.o. hashtable) ☐

- h. Selanjutnya melakukan add sequence dari hasil stream lookup di tahap sebelumnya, untuk add sequence menjadi seperti gambar di bawah ini

Add sequence

Step name:

Name of value:

Use a database to generate the sequence

Use DB to get sequence? ☐

Connection:

Schema name:

Sequence name:

Use a transformation counter to generate the sequence

Use counter to calculate sequence? ☒

Counter name (optional):

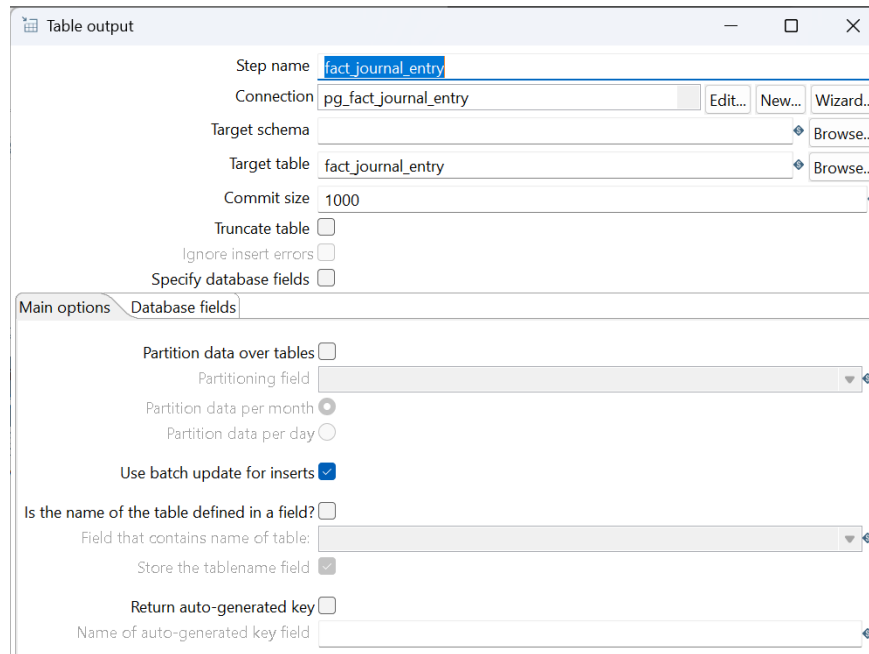
Start at value:

Increment by:

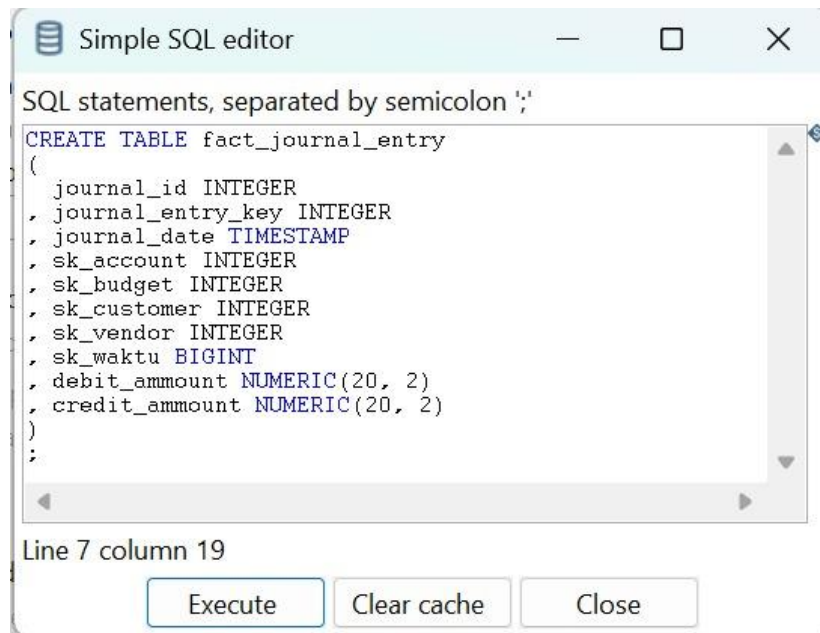
Maximum value:

- i. Selanjutnya melakukan select values, untuk memilih kolom yang penting saja untuk di tampilkan
Kolom yang dipilih pada tahap ini yakni journal_id, journal_entry_key, journal_date, sk_account, sk_customer, sk_vendor, sk_budget, sk_waktu, debit_amount, dan credit_amount
- j. Selanjutnya tambahkan table output dan isikan connectionya menjadi seperti di bawah ini. Disini saya akan menyimpan hasil nya pada database finance yang ada di postgres. Pastikan anda

sudah membuat database finance pada postgres kemudian baru lakukan langkah ini



The 'Table output' window is used to configure the output of a query. It includes fields for 'Step name' (fact_journal_entry), 'Connection' (pg_fact_journal_entry), 'Target schema' (empty), 'Target table' (fact_journal_entry), and 'Commit size' (1000). There are checkboxes for 'Truncate table', 'Ignore insert errors', and 'Specify database fields'. The 'Main options' tab is selected, showing options for 'Partition data over tables', 'Partitioning field', 'Partition data per month', 'Partition data per day', 'Use batch update for inserts' (checked), 'Is the name of the table defined in a field?', 'Field that contains name of table', 'Store the tablename field' (checked), 'Return auto-generated key', and 'Name of auto-generated key field'.



The 'Simple SQL editor' window displays SQL statements separated by semicolons. The code shown is a CREATE TABLE statement for fact_journal_entry with columns: journal_id INTEGER, journal_entry_key INTEGER, journal_date TIMESTAMP, sk_account INTEGER, sk_budget INTEGER, sk_customer INTEGER, sk_vendor INTEGER, sk_waktu BIGINT, debit_ammount NUMERIC(20, 2), and credit_ammount NUMERIC(20, 2). The status bar indicates 'Line 7 column 19'. Buttons for 'Execute', 'Clear cache', and 'Close' are at the bottom.

```
CREATE TABLE fact_journal_entry
(
  journal_id INTEGER
, journal_entry_key INTEGER
, journal_date TIMESTAMP
, sk_account INTEGER
, sk_budget INTEGER
, sk_customer INTEGER
, sk_vendor INTEGER
, sk_waktu BIGINT
, debit_ammount NUMERIC(20, 2)
, credit_ammount NUMERIC(20, 2)
)
;
```

k. Maka hasil fact pada database postgres saya yakni menjadi seperti di bawah

finance_employee finance_vendor finance_customer finance_fact_journal dim_waktu finance_budget Transformation 1 DIM_DIM_BUDGET

Execution Results

Logging Execution History Step Metrics Performance Graph Metrics Preview data

2025/05/31 16:39:15 - 100000 puoget.U - finished processing (I=0, O=0, R=4000, W=0, U=0, E=0)

2025/05/31 16:39:15 - dim_waktu.0 - Finished reading query, closing connection

2025/05/31 16:39:15 - dim_waktu.0 - Finished processing (I=10000, O=0, R=0, W=10000, U=0, E=0)

2025/05/31 16:39:15 - lookup dim_waktu.0 - Finished processing (I=0, O=0, R=10200, W=200, U=0, E=0)

2025/05/31 16:39:15 - journal_entry_key.0 - Finished processing (I=0, O=0, R=200, W=200, U=0, E=0)

2025/05/31 16:39:15 - Select values 2.0 - Finished processing (I=0, O=0, R=200, W=200, U=0, E=0)

2025/05/31 16:39:16 - fact_journal_entry.0 - Finished processing (I=0, O=0, R=200, W=200, U=0, E=0)

2025/05/31 16:39:16 - Spoon - The transformation has finished!!

	journal_id integer	journal_entry_key integer	journal_date timestamp without time zone	sk_account integer	sk_customer integer	sk_vendor integer	sk_waktu bigint	sk_budget integer	debit_ammount numeric (20,2)
1	1	1	2024-11-25 00:00:00	198	26	199	20241125	198	9349.77
2	2	2	2024-11-27 00:00:00	154	29	100	20241127	154	9954.17
3	3	3	2024-10-07 00:00:00	17	46	67	20241007	17	2565.06
4	4	4	2025-01-31 00:00:00	58	78	191	20250131	58	4566.18
5	5	5	2025-01-27 00:00:00	56	183	69	20250127	56	7824.15
6	6	6	2024-11-12 00:00:00	173	130	81	20241112	173	7264.19
7	7	7	2025-05-06 00:00:00	157	1	151	20250506	157	2385.06
8	8	8	2024-10-10 00:00:00	125	141	30	20241010	125	8342.50
9	9	9	2024-07-13 00:00:00	128	170	143	20240713	128	3019.97
10	10	10	2024-12-22 00:00:00	52	63	56	20241222	52	3014.36

9. Membuat view di postgres

Masukkan query di bawah ini untuk membuat view

```

Query Query History
1  CREATE VIEW view_financial_activity AS
2  SELECT
3      f.journal_id,
4      f.journal_entry_key,
5      f.journal_date,
6
7      da.account_name,
8      da.account_type,
9
10     dc.customer_name,
11     dv.vendor_name,
12     de.department,
13
14     dw.bulan,
15     dw.tahun,
16
17     db.amount AS budget_amount
18 FROM fact_journal_entry f
19 LEFT JOIN dim_account da ON f.sk_account = da.sk_account
20 LEFT JOIN dim_budget db ON f.sk_budget = db.sk_budget
21 LEFT JOIN dim_customer dc ON f.sk_customer = dc.sk_customer
22 LEFT JOIN dim_vendor dv ON f.sk_vendor = dv.sk_vendor
23 LEFT JOIN dim_employee de ON da.account_id = de.account_id
24 LEFT JOIN dim_waktu dw ON f.sk_waktu = dw.sk_waktu;

```

Maka tampilan viewnya akan menjadi seperti di bawah ini

	journal_id integer	journal_entry_key integer	journal_date timestamp without time zone	account_name character varying (100)	account_type character varying (50)	customer_name character varying (100)	vendor_name character varying (100)	department character varying (100)	bulan integer	tahun integer	budget_amount numeric (20,2)
1	67	67	2024-05-18 00:00:00	Asset Account 200	Equity	Travis Duarte	Parker, Williams and Phillips	IT	5	2024	211714.00
2	102	102	2024-05-21 00:00:00	Asset Account 37	Liability	Travis Duarte	Norris, Wall and Mueller	Finance	5	2024	347134.70
3	59	59	2024-05-21 00:00:00	Revenue Account 62	Equity	Jay Russell	Parker, Williams and Phillips	IT	5	2024	134304.27
4	35	35	2024-05-21 00:00:00	Asset Account 100	Expense	Joseph Waters	Smith, Alvarez and Maldonado	Operations	5	2024	326955.70
5	160	160	2024-05-23 00:00:00	Asset Account 199	Liability	Ernest Foster	Strong-Leonard	Operations	5	2024	351285.72
6	145	145	2024-05-24 00:00:00	Equity Account 121	Asset	Christopher Castillo	Arnold, Marsh and Ward	IT	5	2024	90768.79
7	113	113	2024-05-24 00:00:00	Equity Account 128	Equity	Isaac Johnson	Johnson Inc	IT	5	2024	202564.83
8	197	197	2024-05-27 00:00:00	Asset Account 55	Asset	Jacob Baulista	Collins, Rodriguez and Hudson	IT	5	2024	357008.79
9	165	165	2024-05-31 00:00:00	Asset Account 173	Expense	Tina Munoz	Castro, McGuire and Cooper	Sales	5	2024	281859.65
10	120	120	2024-06-02 00:00:00	Expense Account 138	Revenue	Scott Lin	Burton Inc	Operations	6	2024	377456.27

10. Membuat forecasting jumlah pengeluaran berdasarkan departemen, di sini saya memilih departemen it dan hr selama 12 bulan kedepan. Disini saya menggunakan coding python untuk membuat forecasting dengan metode Eksponensial Smoothing.

Coding Forecasting Departemen IT

```
# 1. Exponential smoothing pada 'budget_amount'
df_it = df_it.copy()
df_it['budget_amount_smooth'] = df_it['budget_amount'].ewm(alpha=0.3).mean()

# 2. Buat fitur tanggal tambahan
df_it['month'] = df_it.index.month
df_it['day'] = df_it.index.day
df_it['dayofweek'] = df_it.index.dayofweek

# 3. Buat fitur lag dari data yang telah di-smoothing
df_it['lag_1'] = df_it['budget_amount_smooth'].shift(1)
df_it['lag_2'] = df_it['budget_amount_smooth'].shift(2)
df_it['lag_3'] = df_it['budget_amount_smooth'].shift(3)
df_it['lag_4'] = df_it['budget_amount_smooth'].shift(4)
df_it['lag_5'] = df_it['budget_amount_smooth'].shift(5)

# Hapus baris dengan NaN akibat shift
df_it.dropna(inplace=True)

# 4. Tentukan fitur dan target
features = ['month', 'day', 'dayofweek', 'lag_1', 'lag_2', 'lag_3']
X = df_it[features]
y = df_it['budget_amount_smooth']

# 5. Split data train dan test (misal 80% train, 20% test)
split_idx = int(len(df_it) * 0.8)
X_train, X_test = X.iloc[:split_idx], X.iloc[split_idx:]
y_train, y_test = y.iloc[:split_idx], y.iloc[split_idx:]

# 6. Buat dan latih model Random Forest
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# 7. Prediksi di test set
y_pred = model.predict(X_test)

# 8. Evaluasi hasil prediksi
def mape(y_true, y_pred):
    mask = y_true != 0
    return np.mean(np.abs((y_true[mask] - y_pred[mask]) / y_true[mask])) * 100

print(f"MAPE: {mape(y_test.values, y_pred):.2f}%")
print(f"MAE: {mean_absolute_error(y_test, y_pred):.2f}")
print(f"RMSE: {np.sqrt(mean_squared_error(y_test, y_pred)):.2f}")

# Prediksi 3 bulan ke depan (misal data harian, 90 hari ke depan)
n_days_ahead = 90
last_date = df_it.index[-1]
```

```

future_dates = [last_date + pd.Timedelta(days=i) for i in range(1,
n_days_ahead+1)]

# Simpan semua prediksi
future_preds = []
# Gunakan salinan terakhir dari data untuk lag tracking
df_future = df_it.copy()

for date in future_dates:
    lag_1 = df_future['budget_amount_smooth'].iloc[-1]
    lag_2 = df_future['lag_1'].iloc[-1]
    lag_3 = df_future['lag_2'].iloc[-1]
    lag_4 = df_future['lag_3'].iloc[-1]
    lag_5 = df_future['lag_4'].iloc[-1]

    month = date.month
    day = date.day
    dayofweek = date.dayofweek

    input_features = pd.DataFrame({
        'month': [month],
        'day': [day],
        'dayofweek': [dayofweek],
        'lag_1': [lag_1],
        'lag_2': [lag_2],
        'lag_3': [lag_3],
    })

    pred = model.predict(input_features)[0]
    future_preds.append(pred)

    new_row = {
        'budget_amount_smooth': pred,
        'month': month,
        'day': day,
        'dayofweek': dayofweek,
        'lag_1': lag_1,
        'lag_2': lag_2,
        'lag_3': lag_3,
        'lag_4': lag_4,
        'lag_5': lag_5
    }
    df_future.loc[date] = new_row

# Gabungkan hasil prediksi test dan forecasting
test_results = pd.DataFrame({

```

```

        'actual': y_test.values,
        'predicted': y_pred
    }, index=y_test.index)

future_df = pd.DataFrame({
    'predicted': future_preds
}, index=future_dates)

forecast_results = pd.concat([test_results, future_df])

# Siapkan data akhir untuk disimpan
final_data = []

final_data = []

# 1. Data aktual - ambil dari DataFrame asli (sebelum processing)
# Konversi ke list untuk menghindari masalah Series
for idx in df_it.index:
    # Ambil nilai individual untuk menghindari Series
    account_type_val = df_it.loc[idx, 'account_type'] if 'account_type' in
df_it.columns else 'Equity'
    department_val = df_it.loc[idx, 'department'] if 'department' in
df_it.columns else 'HR'
    budget_amount_val = df_it.loc[idx, 'budget_amount']

    # Pastikan semua nilai adalah scalar, bukan Series
    if hasattr(account_type_val, 'iloc'):
        account_type_val = account_type_val.iloc[0] if
len(account_type_val) > 0 else 'Equity'
    if hasattr(department_val, 'iloc'):
        department_val = department_val.iloc[0] if len(department_val) > 0
else 'HR'
    if hasattr(budget_amount_val, 'iloc'):
        budget_amount_val = budget_amount_val.iloc[0] if
len(budget_amount_val) > 0 else 0.0

    final_data.append({
        'account_type': str(account_type_val),
        'department': str(department_val),
        'budget_amount': float(budget_amount_val),
        'journal_date': idx.strftime('%Y-%m-%d'),
        'data_type': 'actual'
    })

# 2. Data forecast - gunakan hasil prediksi future_preds
# Tentukan account_type dan department untuk forecast

```



```

if 'account_type' in df_it.columns:
    most_common_account_type = str(df_it['account_type'].mode()[0])
else:
    most_common_account_type = 'Equity'

if 'department' in df_it.columns:
    most_common_department = str(df_it['department'].mode()[0])
else:
    most_common_department = 'IT'

# Tambahkan data forecast
for i, pred_value in enumerate(future_preds):
    forecast_date = future_dates[i]
    final_data.append({
        'account_type': most_common_account_type,
        'department': most_common_department,
        'budget_amount': float(round(pred_value, 2)),
        'journal_date': forecast_date.strftime('%Y-%m-%d'),
        'data_type': 'forecast'
    })

# Buat DataFrame final
df_final = pd.DataFrame(final_data)

# Tampilkan contoh data yang akan disimpan
print("\n=== DATA YANG AKAN DISIMPAN KE POSTGRESQL ===")
print("Data Actual (10 terakhir):")
actual_data = df_final[df_final['data_type'] == 'actual'].tail(10)
print(actual_data.to_string(index=False))

print("\nData Forecast (10 pertama):")
forecast_data = df_final[df_final['data_type'] == 'forecast'].head(10)
print(forecast_data.to_string(index=False))

# Ganti dengan informasi database-mu
username = 'postgres'
password = ''
host = 'localhost'
port = '5433'
database = 'finance'

# Buat koneksi
engine =
create_engine(f'postgresql+psycopg2://{username}:{password}@{host}:{port}/{database}')

```

```
# Simpan df_hr ke tabel 'hr_data' di PostgreSQL
df_final.to_sql('fc_department_IT', engine, if_exists='replace',
index=False)

print("Data berhasil disimpan ke PostgreSQL.")
```

Hasil forecasting yang ditampilkan pada view

	account_type character varying	department character varying	budget_amount numeric	journal_date date	data_type character varying
1	Liability	IT	294028.47	2024-06-20	actual
2	Liability	IT	434887.14	2024-06-25	actual
3	Revenue	IT	181606.94	2024-06-26	actual
4	Liability	IT	46925.94	2024-07-03	actual
5	Liability	IT	155841.24	2024-07-07	actual
6	Equity	IT	134304.27	2024-07-09	actual
7	Liability	IT	434887.14	2024-07-10	actual
8	Equity	IT	142769.24	2024-07-12	actual
9	Liability	IT	371147.42	2024-07-13	actual
10	Equity	IT	75587.54	2024-07-20	actual
11	Liability	IT	111541.55	2024-07-26	actual
12	Asset	IT	441376.38	2024-07-29	actual
13	Asset	IT	357008.79	2024-07-30	actual
14	Equity	IT	211714	2024-08-03	actual
15	Equity	IT	119062.44	2024-08-10	actual
16	Liability	IT	371179.35	2024-09-09	actual
17	Liability	IT	371179.35	2024-09-09	actual
18	Equity	IT	410029.01	2024-09-15	actual
19	Expense	IT	374663.71	2024-09-28	actual
20	Equity	IT	134304.27	2024-10-02	actual
21	Asset	IT	47338.19	2024-10-07	actual
22	Liability	IT	487381.62	2024-10-10	actual
23	Equity	IT	278248.69	2024-10-16	actual
24	Asset	IT	497957.37	2024-10-28	actual
25	Equity	IT	422624.79	2024-10-31	actual
26	Asset	IT	452544.02	2024-11-10	actual
27	Equity	IT	479685.33	2024-11-24	actual

	account_type character varying 🔒	department character varying 🔒	budget_amount numeric 🔒	journal_date date 🔒	data_type character varying 🔒
59	Equity	IT	251210.07	2025-05-21	forecast
60	Equity	IT	246862.45	2025-05-22	forecast
61	Equity	IT	256772.89	2025-05-23	forecast
62	Equity	IT	259238.43	2025-05-24	forecast
63	Equity	IT	270229.89	2025-05-25	forecast
64	Equity	IT	270667.04	2025-05-26	forecast
65	Equity	IT	274591.26	2025-05-27	forecast
66	Equity	IT	297682.38	2025-05-28	forecast
67	Equity	IT	315898.16	2025-05-29	forecast
68	Equity	IT	319515.42	2025-05-30	forecast
69	Equity	IT	322168.8	2025-05-31	forecast
70	Equity	IT	309741.3	2025-06-01	forecast
71	Equity	IT	268254.63	2025-06-02	forecast
72	Equity	IT	246743.15	2025-06-03	forecast
73	Equity	IT	243478.64	2025-06-04	forecast
74	Equity	IT	236176.11	2025-06-05	forecast
75	Equity	IT	236674.68	2025-06-06	forecast
76	Equity	IT	237132.42	2025-06-07	forecast
77	Equity	IT	237829.74	2025-06-08	forecast
78	Equity	IT	234736.86	2025-06-09	forecast
79	Equity	IT	244822.5	2025-06-10	forecast
80	Equity	IT	246057.9	2025-06-11	forecast
81	Equity	IT	247035.72	2025-06-12	forecast
82	Equity	IT	246699.87	2025-06-13	forecast
83	Equity	IT	246486.67	2025-06-14	forecast
84	Equity	IT	247046.03	2025-06-15	forecast

Coding Forecasting Departemen HR

```
# 1. Exponential smoothing pada 'budget_amount'
df_hr = df_hr.copy()
df_hr['budget_amount_smooth'] =
df_hr['budget_amount'].ewm(alpha=0.3).mean()

df_hr.head()
# 2. Buat fitur tanggal tambahan
df_hr['month'] = df_hr.index.month
df_hr['day'] = df_hr.index.day
df_hr['dayofweek'] = df_hr.index.dayofweek
```

```

df_hr.head()
# 3. Buat fitur lag dari data yang telah di-smoothing
df_hr['lag_1'] = df_hr['budget_amount_smooth'].shift(1)
df_hr['lag_2'] = df_hr['budget_amount_smooth'].shift(2)
df_hr['lag_3'] = df_hr['budget_amount_smooth'].shift(3)
df_hr['lag_4'] = df_hr['budget_amount_smooth'].shift(4)
df_hr['lag_5'] = df_hr['budget_amount_smooth'].shift(5)

# Hapus baris dengan NaN akibat shift
df_hr.dropna(inplace=True)

df_hr.isnull().sum()
# 4. Tentukan fitur dan target
features = ['month', 'day', 'dayofweek', 'lag_1', 'lag_2', 'lag_3']
X = df_hr[features]
y = df_hr['budget_amount_smooth'] # target adalah versi smooth
# 5. Split data train dan test (misal 80% train, 20% test)
split_idx = int(len(df_hr) * 0.8)
X_train, X_test = X.iloc[:split_idx], X.iloc[split_idx:]
y_train, y_test = y.iloc[:split_idx], y.iloc[split_idx:]
# 6. Buat dan latih model Random Forest
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
# 7. Prediksi di test set
y_pred = model.predict(X_test)
# 8. Evaluasi hasil prediksi
def mape(y_true, y_pred):
    mask = y_true != 0
    return np.mean(np.abs((y_true[mask] - y_pred[mask]) /
y_true[mask])) * 100

print(f"MAPE: {mape(y_test.values, y_pred):.2f}%")
print(f"MAE: {mean_absolute_error(y_test, y_pred):.2f}")
print(f"RMSE: {np.sqrt(mean_squared_error(y_test, y_pred)):.2f}")
# Prediksi 3 bulan ke depan (misal data harian, 90 hari ke depan)
n_days_ahead = 90
last_date = df_hr.index[-1]
future_dates = [last_date + pd.Timedelta(days=i) for i in range(1,
n_days_ahead+1)]

# Simpan semua prediksi
future_preds = []
# Gunakan salinan terakhir dari data untuk lag tracking
df_future = df_hr.copy()

for date in future_dates:
    # Ambil nilai lag terakhir dari hasil sebelumnya
    lag_1 = df_future['budget_amount_smooth'].iloc[-1]
    lag_2 = df_future['lag_1'].iloc[-1]
    lag_3 = df_future['lag_2'].iloc[-1]
    lag_4 = df_future['lag_3'].iloc[-1]
    lag_5 = df_future['lag_4'].iloc[-1]

```

```

# Buat fitur tanggal
month = date.month
day = date.day
dayofweek = date.dayofweek

# Buat DataFrame 1 baris untuk prediksi
input_features = pd.DataFrame({
    'month': [month],
    'day': [day],
    'dayofweek': [dayofweek],
    'lag_1': [lag_1],
    'lag_2': [lag_2],
    'lag_3': [lag_3],
})

# Prediksi
pred = model.predict(input_features)[0]

# Simpan hasil
future_preds.append(pred)

# Tambahkan baris baru ke df_future untuk update lag di iterasi selanjutnya
new_row = {
    'budget_amount_smooth': pred,
    'month': month,
    'day': day,
    'dayofweek': dayofweek,
    'lag_1': lag_1,
    'lag_2': lag_2,
    'lag_3': lag_3,
    'lag_4': lag_4,
    'lag_5': lag_5
}
df_future.loc[date] = new_row

# Buat DataFrame hasil prediksi
future_df = pd.DataFrame({
    'forecast_date': future_dates,
    'predicted_budget_amount': future_preds
})
future_df.set_index('forecast_date', inplace=True)

future_df.head()

# 8. Evaluasi hasil prediksi
def mape(y_true, y_pred):
    mask = y_true != 0
    return np.mean(np.abs((y_true[mask] - y_pred[mask]) /
y_true[mask])) * 100

print(f"MAPE: {mape(y_test.values, y_pred):.2f}%")
print(f"MAE: {mean_absolute_error(y_test, y_pred):.2f}")
print(f"RMSE: {np.sqrt(mean_squared_error(y_test, y_pred)):.2f}")

```

```

# Prediksi 3 bulan ke depan (misal data harian, 90 hari ke depan)
n_days_ahead = 90
last_date = df_hr.index[-1]
future_dates = [last_date + pd.Timedelta(days=i) for i in range(1,
n_days_ahead+1)]

# Simpan semua prediksi
future_preds = []

# Gunakan salinan terakhir dari data untuk lag tracking
df_future = df_hr.copy()

for date in future_dates:
    # Ambil nilai lag terakhir dari hasil sebelumnya
    lag_1 = df_future['budget_amount_smooth'].iloc[-1]
    lag_2 = df_future['lag_1'].iloc[-1]
    lag_3 = df_future['lag_2'].iloc[-1]

    # Buat fitur tanggal
    month = date.month
    day = date.day
    dayofweek = date.dayofweek

    # Buat DataFrame 1 baris untuk prediksi
    input_features = pd.DataFrame({
        'month': [month],
        'day': [day],
        'dayofweek': [dayofweek],
        'lag_1': [lag_1],
        'lag_2': [lag_2],
        'lag_3': [lag_3],
    })

    # Prediksi
    pred = model.predict(input_features)[0]

    # Simpan hasil
    future_preds.append(pred)

    # Tambahkan baris baru ke df_future untuk update lag di iterasi
    selanjutnya
    new_row = {
        'budget_amount_smooth': pred,
        'month': month,
        'day': day,
        'dayofweek': dayofweek,
        'lag_1': lag_1,
        'lag_2': lag_2,
        'lag_3': lag_3
    }
    df_future.loc[date] = new_row

# Buat DataFrame hasil forecasting dengan data aktual dan prediksi
# Gabungkan data test (aktual) dengan prediksi test

```

```

test_results = pd.DataFrame({
    'actual': y_test.values,
    'predicted': y_pred
}, index=y_test.index)

# Buat DataFrame untuk forecasting masa depan
future_df = pd.DataFrame({
    'predicted': future_preds
}, index=future_dates)

# Gabungkan hasil test dan forecasting
forecast_results = pd.concat([
    test_results,
    future_df
])

# Siapkan data untuk disimpan ke PostgreSQL
final_data = []
final_data = []

# 1. Data aktual - ambil dari DataFrame asli (sebelum processing)
# Konversi ke list untuk menghindari masalah Series
for idx in df_hr.index:
    # Ambil nilai individual untuk menghindari Series
    account_type_val = df_hr.loc[idx, 'account_type'] if 'account_type'
in df_hr.columns else 'Equity'
    department_val = df_hr.loc[idx, 'department'] if 'department' in
df_hr.columns else 'HR'
    budget_amount_val = df_hr.loc[idx, 'budget_amount']

    # Pastikan semua nilai adalah scalar, bukan Series
    if hasattr(account_type_val, 'iloc'):
        account_type_val = account_type_val.iloc[0] if
len(account_type_val) > 0 else 'Equity'
    if hasattr(department_val, 'iloc'):
        department_val = department_val.iloc[0] if len(department_val)
> 0 else 'HR'
    if hasattr(budget_amount_val, 'iloc'):
        budget_amount_val = budget_amount_val.iloc[0] if
len(budget_amount_val) > 0 else 0.0

    final_data.append({
        'account_type': str(account_type_val),
        'department': str(department_val),
        'budget_amount': float(budget_amount_val),
        'journal_date': idx.strftime('%Y-%m-%d'),
        'data_type': 'actual'
    })

# 2. Data forecast - gunakan hasil prediksi future_preds
# Tentukan account_type dan department untuk forecast
if 'account_type' in df_hr.columns:
    most_common_account_type = str(df_hr['account_type'].mode()[0])
else:

```

```

        most_common_account_type = 'Equity'

if 'department' in df_hr.columns:
    most_common_department = str(df_hr['department'].mode()[0])
else:
    most_common_department = 'HR'

# Tambahkan data forecast
for i, pred_value in enumerate(future_preds):
    forecast_date = future_dates[i]
    final_data.append({
        'account_type': most_common_account_type,
        'department': most_common_department,
        'budget_amount': float(round(pred_value, 2)),
        'journal_date': forecast_date.strftime('%Y-%m-%d'),
        'data_type': 'forecast'
    })

# Buat DataFrame final
df_final = pd.DataFrame(final_data)

# Tampilkan contoh data yang akan disimpan
print("\n=== DATA YANG AKAN DISIMPAN KE POSTGRESQL ===")
print("Data Actual (10 terakhir):")
actual_data = df_final[df_final['data_type'] == 'actual'].tail(10)
print(actual_data.to_string(index=False))

print("\nData Forecast (10 pertama):")
forecast_data = df_final[df_final['data_type'] == 'forecast'].head(10)
print(forecast_data.to_string(index=False))

# Ganti dengan informasi database-mu
username = 'postgres'
password = ''
host = 'localhost'
port = '5433'
database = 'finance'






# Buat koneksi
engine =
create_engine(f'postgresql+psycopg2://{username}:{password}@{host}:{port}/{database}')

# Simpan df_hr ke tabel 'hr_data' di PostgreSQL
df_final.to_sql('fc_department_HR', engine, if_exists='replace',
index=False)

print("Data berhasil disimpan ke PostgreSQL.")

```


	account_type character varying	department character varying	budget_amount numeric	journal_date date	data_type character varying
143	Liability	HR	214153.14	2024-10-29	actual
144	Asset	HR	277498.73	2024-11-02	actual
145	Expense	HR	417372.03	2024-11-09	actual
146	Liability	HR	171823.74	2024-11-14	actual
147	Asset	HR	232088.01	2024-11-22	actual
148	Asset	HR	232088.01	2024-12-06	actual
149	Revenue	HR	323287.06	2024-12-19	actual
150	Equity	HR	425100.14	2024-12-24	actual
151	Liability	HR	171823.74	2025-01-10	actual
152	Asset	HR	232088.01	2025-01-23	actual
153	Expense	HR	204359.28	2025-01-24	actual
154	Revenue	HR	271829.67	2025-02-19	actual
155	Expense	HR	417372.03	2025-02-21	actual
156	Expense	HR	261529.82	2025-02-25	actual
157	Liability	HR	392835.06	2025-02-28	actual
158	Liability	HR	122403.43	2025-03-09	actual
159	Equity	HR	460366.7	2025-03-19	actual
160	Equity	HR	223804.56	2025-04-14	actual
161	Expense	HR	32472.02	2025-04-24	actual
162	Asset	HR	232088.01	2025-04-27	actual
163	Liability	HR	285314.68	2025-04-30	actual
164	Expense	HR	261529.82	2025-05-03	actual
165	Expense	HR	456130.93	2025-05-06	actual
166	Expense	HR	302154.84	2025-05-07	forecast
167	Expense	HR	291630.08	2025-05-08	forecast
168	Expense	HR	267817.6	2025-05-09	forecast

	account_type character varying 	department character varying 	budget_amount numeric 	journal_date date 	data_type character varying 
214	Expense	HR	264200.29	2025-06-24	forecast
215	Expense	HR	265070.88	2025-06-25	forecast
216	Expense	HR	272359.49	2025-06-26	forecast
217	Expense	HR	273488.71	2025-06-27	forecast
218	Expense	HR	275626.63	2025-06-28	forecast
219	Expense	HR	273176.77	2025-06-29	forecast
220	Expense	HR	267832.92	2025-06-30	forecast
221	Expense	HR	263517.24	2025-07-01	forecast
222	Expense	HR	264421.81	2025-07-02	forecast
223	Expense	HR	265743.19	2025-07-03	forecast
224	Expense	HR	269394.87	2025-07-04	forecast
225	Expense	HR	271780	2025-07-05	forecast
226	Expense	HR	276616.49	2025-07-06	forecast
227	Expense	HR	266100.82	2025-07-07	forecast
228	Expense	HR	265236.58	2025-07-08	forecast
229	Expense	HR	262172.22	2025-07-09	forecast
230	Expense	HR	268307.35	2025-07-10	forecast
231	Expense	HR	268024.53	2025-07-11	forecast
232	Expense	HR	273376.55	2025-07-12	forecast
233	Expense	HR	270277.05	2025-07-13	forecast
234	Expense	HR	266166.38	2025-07-14	forecast
235	Expense	HR	262651.16	2025-07-15	forecast
236	Expense	HR	265703.91	2025-07-16	forecast
237	Expense	HR	266988.3	2025-07-17	forecast

12. Dashboard Forecasting

