



Project-Based Internship

Data Engineer Test

Membuat Data Warehouse dan Stored Procedure untuk salah satu client ID/X Partners



Table of Content

- 1 Challenge
- 2 Prerequisite
- 3 Hints
- 4 Submission

Challenge

Salah satu *client* dari perusahaan ID/X Partners yang bergerak di industri perbankan, memiliki kebutuhan untuk membuat sebuah *Data Warehouse* dari beberapa sumber data yang berbeda yang tersimpan di dalam sistem mereka. Beberapa sumber data tersebut antara lain :

1. transaction_excel (file excel)
2. transaction_csv (file csv)
3. transaction_db (Database SQL Server)
4. account (Database SQL Server)
5. customer (Database SQL Server)
6. branch (Database SQL Server)
7. city (Database SQL Server)
8. state (Database SQL Server)

Permasalahan yang mereka hadapi saat ini adalah mereka kesulitan untuk mengekstrak data dari berbagai sumber (excel, csv, database) secara bersamaan sehingga pelaporan dan analisis data mereka selalu mengalami keterlambatan. Sumber data tersebut dapat *download* disini : [Final Task Data Engineer](#)

Challenge

Sebagai seorang Data Engineer, ada beberapa tugas yang perlu anda lakukan untuk optimalisasi proses ETL di perusahaan tersebut yaitu :

1. Membuat sebuah *database* baru yang akan kita anggap sebagai *Data Warehouse* baru bernama **DWH**. Setelah itu buat tiga tabel dimension yaitu **DimAccount**, **DimCustomer**, **DimBranch** dan satu tabel fact yaitu **FactTransaction**. (***jangan lupa untuk memberikan primary key dan foreign key di setiap tabel*)
2. Membuat job ETL di aplikasi talend untuk memindahkan data dari source(ke seluruh tabel Dimension. Khusus untuk tabel DimCustomer, format kolom yang disimpan adalah CustomerID, CustomerName, Address, CityName, StateName, Age, Gender, Email. Semua data dari kolom tersebut diubah menjadi huruf kapital kecuali untuk kolom CustomerID, Age dan Email. (***Untuk penamaan kolom-kolom diharapkan mengikuti kaidah PascalCase, contoh `account_id = AccountID`*).
3. Membuat job ETL untuk menggabungkan data transaksi (transaksi_excel, transaksi_csv, transaksi_db) menjadi satu di tabel **FactTransaction**. Perlu diperhatikan karena ketiga file tersebut berasal dari source yang berbeda-beda, maka pastikan tidak ada row yang duplikat di dalam tabel FactTransaction.

Challenge

4. Membuat dua *Stored Procedure* (SP) dengan parameter, untuk membantu mereka mendapatkan ringkasan data dengan cepat. *Stored Procedure* yang diminta yaitu :
 - **DailyTransaction** (untuk menghitung banyaknya transaksi beserta total nominalnya setiap harinya). Kolom yang ditampilkan yaitu Date, TotalTransactions, TotalAmount. Kolom TotalAmount didapat dengan menjumlahkan Amount setiap harinya. Lalu buatlah dua parameter yaitu start_date dan end_date jadi ketika menjalankan SP ini dengan memasukkan parameter tersebut, maka akan menampilkan data sesuai rentang tanggal yang kita masukkan. Berikut contoh output yang diharapkan :

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

Challenge

- **BalancePerCustomer** (untuk mengetahui sisa balance per customer). Kolom yang ditampilkan yaitu CustomerName, AccountType, Balance, CurrentBalance. Kolom CurrentBalance didapat dari kolom Balance di tabel account dikurang total amount yang ditransaksikan di tabel transaction untuk setiap account_id. Untuk setiap transaction_type = Deposit, maka balance akan bertambah, selain itu maka Balance akan berkurang. Buatlah parameter bernama name sehingga ketika menjalankan SP ini dengan memasukkan nama salah satu customer tersebut, maka akan menampilkan data sesuai dengan yang kita input. Lalu pastikan untuk filter yang accountnya berstatus active. Berikut adalah contoh transaksi salah satu customer bernama Shelly dan hasil output yang diharapkan :

TransactionID	CustomerName	TransactionDate	AccountType	Balance	Amount	TransactionType
1	SHELLY JUWITA	2024-01-17 09:10:00	saving	1500000	100000	Deposit
3	SHELLY JUWITA	2024-01-18 08:30:00	checking	25000000	10000000	Transfer
4	SHELLY JUWITA	2024-01-18 10:45:00	checking	25000000	1000000	Withdrawal

50 | EXEC BalancePerCustomer @name = 'Shelly'

%

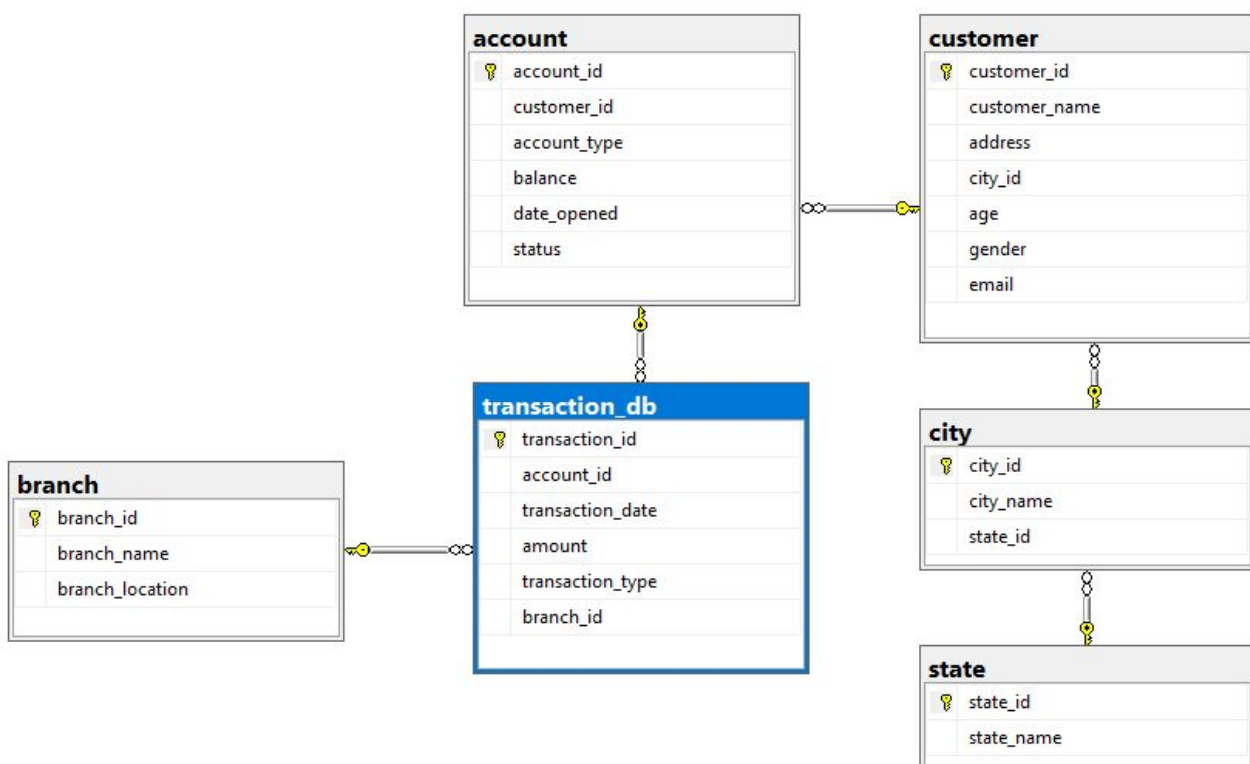
Results Messages

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

Prerequisite

Sebelum mengerjakan challenges, silakan untuk menyelesaikan pre-requisite di bawah ini:

1. Pastikan Database **SQL Server** dan **SSMS** (SQL Server Management Studio) sudah terinstall pada laptop anda, Lalu restore database **sample.bak**. Database ini akan kita anggap sebagai **source** yang berisi tabel transaction_db, account, customer, branch,city,state.
2. Pastikan aplikasi **Talend** sudah terinstall
3. Ada tiga file yang tersedia yaitu **transaction_excel.xlsx**, **transaction_csv.csv** dan backup file database **sample.bak**. Untuk memahami relasi antar tabel yang ada di sistem. Perhatikan gambar di bawah ini :



Hints

1. Lakukan restore database sample.bak sebagai sumber data yang akan dipakai, dalam database tersebut tersedia tabel **transaction_db**, **account**, **customer**, **branch**, **city** dan **state**.
2. Semua kolom tabel Fact dan Dimension mengikuti kolom dari tabel yang ada dari source, khusus untuk tabel DimCustomer dibentuk dari hasil kombinasi tabel customer, city dan state untuk mengambil kolom CityName dan StateName.
3. Gunakan komponen **tUnite**, **tUniq** dan **tMap** untuk melakukan penggabungan data, memisahkan data yang duplikat dan ingin melakukan transformasi data lainnya.
4. Untuk soal membuat store procedure **BalancePerCustomer** bisa mengkombinasikan CASE WHEN dengan SUM untuk mendapatkan kolom CurrentBalance.
5. Berikut adalah deskripsi kolom dari setiap tabel yang tersedia :

Tabel transaction_db (berisi data transaksi) .File yang excel dan csv juga memiliki kolom yang sama seperti ini.

- transaction_id = ID transaksi
- account_id = ID account (rekening)
- transaction_date = tanggal transaksi
- amount = nominal yang ditransaksikan
- transaction_type = jenis transaksi.
- branch_id = ID bank cabang

Hints

Tabel account (berisi data account atau rekening)

- account_id = ID account
- customer_id = ID Customer
- account_type = jenis rekening, saving untuk menyimpan uang jangka panjang seperti deposito, checking untuk rekening sehari-hari
- balance = saldo rekening
- date_opened = tanggal rekening dibuat
- status = status rekening

Tabel branch (berisi data kantor cabang bank)

- branch_id = ID kantor cabang
- branch_name = nama cabang
- branch_location = alamat lokasi cabang

Tabel customer (berisi data customer)

- customer_id= ID customer
- customer_name = nama customer
- address = alamat customer
- city_id = ID kecamatan
- age = umur
- gender = jenis kelamin
- email = email customer

Hints

Tabel city (berisi data kelurahan)

- city_id= ID Kelurahan
- city_name = nama kelurahan
- state_id= ID Kota

Tabel state (berisi data kota)

- state_id= ID Kota
- state_name = nama kota

Submission

Setelah Anda menyelesaikan Challenge yang telah diberikan, Anda **DIWAJIBKAN** untuk membuat **File Final Submission** dalam format PPT dengan template <https://bit.ly/template-final-task-vix-de> yang berisi:

- Biodata Diri
- Hasil Pengerjaan
- Link Folder/File Hasil Pengerjaan (optional)
- Link Github Hasil Pengerjaan
- Link Video Presentasi Hasil Pekerjaan / Penjelasan Project yang Telah Anda Kerjakan

Video presentasi yang telah dibuat, harus Anda upload pada **YouTube** atau **Google Drive** untuk mendapatkan link yang akan dicantumkan pada file final submission nantinya.

Kemudian, **kumpulkan** *File Final Submission* tersebut dengan format **FinalTask_IDX Partners_DE_Nama Lengkap**

File tersebut yang akan **diberikan kepada company** sebagai hasil akhir dari program VIX ini. Diharapkan Anda dapat membuat *File Final Submission* ini dengan sungguh-sungguh. Terima kasih.



Thank You!



id/x partners