***LoadFactInvoiceLineTransactionInitialLoad***

در این بخش ETL قصد داریم هر سطر از جدول‌ مبدأ InvoiceLine را از طریق زنجیرهٔ روابطش به کلیدهای Surrogate در ابعاد (Dim) برسانیم. به‌طور دقیق:

1. FROM StagingDB.Finance.InvoiceLine AS src
2. INNER JOIN StagingDB.Finance.Invoice AS srcInv
3. ON src.InvoiceID = srcInv.InvoiceID
   * **هدف**: ابتدا رکورد جزئیات فاکتور (InvoiceLine) را با سربرگ فاکتور (Invoice) متناظرش پیوند می‌دهیم.
   * **چرا**: چون InvoiceLine شامل فقط InvoiceID مبدأ است، و ما برای دسترسی به تاریخ صدور (و نیز به ContractID) نیاز داریم به رکورد کامل فاکتور مراجعه کنیم.
4. INNER JOIN Dim.DimInvoice AS inv
5. ON srcInv.InvoiceNumber = inv.InvoiceNumber
   * **هدف**: نگاشت رکورد فاکتور مبدأ به کلید Surrogate ابعاد فاکتور.
   * **چرا**: در Dim.DimInvoice یک کلید Identity (DimInvoiceID) داریم که داخلی است و با InvoiceNumber منبع مبدأ منطبق شده است.
   * **نکته**: اگر در مدل شما DimInvoiceID برابر با InvoiceID مبدأ باشد، می‌توان مستقیماً به آن کلید Surrogate مراجعه کرد. اما چون معمولاً ابعاد از کلید طبیعی (شماره فاکتور) برای پشتیبان‌گیری استفاده می‌کنند، در اینجا InvoiceNumber را مبنا قرار دادیم.
6. INNER JOIN StagingDB.Finance.Contract AS srcCtr
7. ON srcInv.ContractID = srcCtr.ContractID
   * **هدف**: از رکورد فاکتور به قرارداد متناظر برویم.
   * **چرا**: چون ارتباط بین مشتری و فاکتور در جدول قرارداد تعریف شده است و برای یافتن مشتری باید ابتدا قرارداد پیدا شود.
8. INNER JOIN StagingDB.Finance.Customer AS srcCust
9. ON srcCtr.CustomerID = srcCust.CustomerID
   * **هدف**: از قرارداد به رکورد مشتری مبدأ برگردیم.
   * **چرا**: Contract شامل CustomerID طبیعی است و می‌خواهیم اطلاعات‌ مشتری را برای نگاشت بعدی به بعد مشتری (DimCustomer) به‌دست آوریم.
10. INNER JOIN Dim.DimCustomer AS cust
11. ON srcCust.CustomerCode = cust.CustomerCode
    * **هدف**: نگاشت مشتری مبدأ به کلید Surrogate بعد مشتری.
    * **چرا**: DimCustomer روی CustomerCode (کلید طبیعی) ایندکس شده و یک DimCustomerID دارد. بنابراین با مقایسه‌ی CustomerCode مبدأ و ابعاد، به DimCustomerID متناظر می‌رسیم.
12. INNER JOIN Dim.DimServiceType AS svc
13. ON src.ServiceTypeID = svc.SourceServiceTypeID
    * **هدف**: نگاشت نوع خدمت مبدأ به کلید Surrogate بعد خدمت.
    * **چرا**: InvoiceLine شامل ServiceTypeID طبیعی است که در DimServiceType.SourceServiceTypeID نگهداری می‌شود. از آنجا DimServiceTypeID را می‌گیریم.
14. LEFT JOIN Dim.DimTax AS tax
15. ON src.TaxID = tax.TaxID
    * **هدف**: در صورت وجود مالیات در خطوط فاکتور، آن را به کلید Surrogate بعد مالیات نگاشت کنیم.
    * **چرا**: بعضی سطرهای InvoiceLine ممکن است بدون مالیات (NULL) باشند، پس از LEFT JOIN استفاده می‌کنیم تا در نبود TaxID، رکورد حذف نشود و مقدار DimTaxID در خروجی NULL بماند.
16. INNER JOIN Dim.DimDate AS dt
17. ON srcInv.InvoiceDate = dt.FullDate
    * **هدف**: نگاشت تاریخ صدور (InvoiceDate) به جدول تقویم (DimDate).
    * **چرا**: DimDate یک بعد تاریخ کامل است که تمامی روزهای تقویم را پوشش می‌دهد. با نگاشت InvoiceDate به FullDate، به DimDateID متناظر دست می‌یابیم.

**خلاصهٔ جریان داده**

1. از InvoiceLine به Invoice می‌رویم تا تاریخ و شناسه قرارداد را پیدا کنیم.
2. با استفاده از InvoiceNumber (یا InvoiceID→DimInvoiceID) به بعد فاکتور (DimInvoice) می‌رسیم.
3. از فاکتور ابتدا به قرارداد، سپس به مشتری مبدأ می‌رویم.
4. مشتری مبدأ را با CustomerCode به بعد مشتری (DimCustomer) نگاشت می‌کنیم.
5. ServiceTypeID→DimServiceTypeID می‌شود.
6. TaxID→DimTaxID (اختیاری) می‌شود.
7. InvoiceDate→DimDateID می‌شود.
8. مقادیر کمی (Quantity، UnitPrice، NetAmount، TaxAmount و…) را در Fact درج می‌کنیم.

این زنجیرهٔ Join تضمین می‌کند که همهٔ کلیدهای خارجی در Fact به Surrogate کلیدهای ابعاد صحیح اشاره کنند و تمامی رکوردهای معتبر بارگذاری شوند.

***FactCustomerPaymentTransaction***

در این کوئری:

SELECT

pm.DimPaymentMethodID,

inv.DimInvoiceID,

cust.DimCustomerID,

dt.DimDateID,

src.Amount,

DATEDIFF(DAY, invInv.DueDate, src.PaymentDate),

invInv.TotalAmount - src.Amount,

CASE WHEN src.Amount >= invInv.TotalAmount THEN 1 ELSE 0 END,

src.PartialPaymentCount

FROM StagingDB.Finance.Payment AS src

INNER JOIN StagingDB.Finance.Invoice AS invInv

ON src.InvoiceID = invInv.InvoiceID

INNER JOIN Dim.DimInvoice AS inv

ON invInv.InvoiceNumber = inv.InvoiceNumber

INNER JOIN StagingDB.Finance.Contract AS ctr

ON invInv.ContractID = ctr.ContractID

INNER JOIN StagingDB.Finance.Customer AS srcCust

ON ctr.CustomerID = srcCust.CustomerID

INNER JOIN Dim.DimCustomer AS cust

ON srcCust.CustomerCode = cust.CustomerCode

INNER JOIN Dim.DimPaymentMethod AS pm

ON src.PaymentMethod = pm.PaymentMethodName

INNER JOIN Dim.DimDate AS dt

ON src.PaymentDate = dt.FullDate;

اجزای کلیدی کوئری به ترتیب به این صورت عمل می‌کنند:

1. 🔁 ‌در حال استخراج اطلاعات از جدول StagingDB.Finance.Payment هستیم که شامل پرداخت‌های مشتریان است.
2. ⛓️‌ با جدول StagingDB.Finance.Invoice جوین زده می‌شود تا اطلاعات پرداخت‌ها به فاکتور مربوطه وصل شود.
3. 🔄 سپس با جدول Dim.DimInvoice جوین زده می‌شود تا کلید DimInvoiceID را از بُعد فاکتور دریافت کند.
4. ⛓️‌ با جدول Contract جوین زده می‌شود (از طریق Invoice → Contract) تا دسترسی به مشتری حاصل شود.
5. ⛓️‌ سپس با جدول StagingDB.Finance.Customer وصل می‌شود تا به CustomerCode دست یابد.
6. 🧭 بعد با جدول Dim.DimCustomer جوین زده می‌شود تا DimCustomerID استخراج شود.
7. 🧾 برای تعیین نوع پرداخت، فیلد PaymentMethod از جدول Staging با Dim.DimPaymentMethod مچ می‌شود.
8. 📅 برای به‌دست‌آوردن DimDateID، تاریخ پرداخت با FullDate از DimDate مقایسه می‌شود.
9. ⚙️ ستون‌هایی که استخراج می‌شوند:
   * مبلغ پرداخت (Amount)
   * فاصله روز بین تاریخ سررسید فاکتور و تاریخ پرداخت (DaysToPayment)
   * مانده‌طلب (RemainingAmount)
   * تعیین اینکه آیا پرداخت کامل بوده یا نه (IsFullPayment)
   * تعداد پرداخت‌های جزئی (PartialPaymentCount)

این کوئری ساختار مناسبی دارد برای مرحله First Load جدول Fact.FactCustomerPaymentTransaction و به درستی از staging به dimension و سپس به fact متصل می‌شود، مشروط به اینکه:

* تمام رکوردهای مربوطه از staging در dimensions موجود باشند.
* DimPaymentMethod از قبل با داده‌های صحیح پر شده باشد.

اگر داده‌های دایمنشن‌ها قبل از اجرای این کوئری بارگذاری شده‌اند (مثلاً از طریق همان پراسیجرهای FirstLoad که قبلاً ساختیم)، این کوئری درست کار خواهد کرد.