توضیحات نحوه پیاده سازی Etl های Data Warehouse

1. DimDate

هدف: ایجاد یک جدول تقویم کامل که تمام تاریخ‌ها را از «تاریخ شروع» (مثلاً ۲۲ ژوئن ۲۰۲۵) تا امروز در اختیار قرار می‌دهد.  
نحوه بارگذاری اولیه

1. رکوردهای پیشین را پاک می‌کند و شناسه (سروگیت) دوباره از ۱ شروع می‌شود.
2. با استفاده از یک CTE عددی، به تعداد روزهای بین تاریخ شروع و امروز، یک ردیف ایجاد می‌کند؛ برای هر ردیف، ستون‌های سال، ماه، روز، شماره هفته در سال، نام ماه و نام روز هفته و پرچم تعطیل/غیرتعطیل (Weekend) محاسبه و درج می‌شود.  
   از این جدول برای ‏Join‏ کردن تمام جدول‌های فکت بر مبنای تاریخ (تاریخ فاکتور، تاریخ پرداخت، …) استفاده می‌شود.

2. DimCustomer (SCD Type 2)

هدف نگهداری تاریخچه تغییرات اطلاعات مشتری (نام، نوع، کشور و …) به‌صورت SCD2  
بارگذاری اولیه

1. تمامی رکوردها را خالی می‌کند.
2. برای هر مشتری از استیجینگ، یک ردیف جدید با StartDate = Today ، EndDate = NULL و IsCurrent = 1 درج می‌کند.  
   نگهداری تغییرات

* در به‌روزرسانی‌های بعدی وقتی اطلاعات مشتری (مثلاً آدرس یا نوع) تغییر می‌کند، ردیف جاری با IsCurrent = 1 به‌روزرسانی می‌شود EndDate = Today و IsCurrent = 0، سپس یک ردیف جدید با مقادیر به‌روز و IsCurrent = 1 درج می‌شود.
* امکان گزارش‌گیری «در یک تاریخ مشخص چه اطلاعاتی برای مشتری وجود داشته» فراهم می‌شود.

3. DimServiceType (SCD Type 1)

هدف فهرست انواع خدمات فعلی؛ تاریخچه نگهداری نمی‌شود (SCD Type 1).  
بارگذاری اولیه

1. جدول را truncate کرده و تمام ردیف‌های ServiceType از استیجینگ را درج می‌کند.  
   نگهداری تغییرات

* در هر بار به‌روزرسانی، سطرهای موجود را با داده‌های جدید جایگزین (update/overwrite) می‌کند یا سرویس جدید را insert می‌کند.
* این دایمنش برای Lookup کردن در فکت InvoiceLine استفاده می‌شود.

4. DimTax (SCD Type 2)

هدف نگهداری اطلاعات مالیات با تاریخچه – نرخ، نوع و بازه اثرگذاری (EffectiveFrom/To).  
بارگذاری اولیه

1. تمام رکوردها پاک می‌شوند و هر ردیف با StartDate = Today و IsCurrent = 1 درج می‌شود.  
   نگهداری تغییرات

* اگر نرخ یا تاریخ اثرگذاری مالیات جدیدی در استیج تغییر کند، ردیف جاری به EndDate = Today و IsCurrent = 0 بروزرسانی می‌شود و یک ردیف جدید جاری درج می‌گردد.
* از این دایمنش برای محاسبه مالیات ردیف‌های فاکتور بر اساس تاریخ صدور استفاده می‌شود.

5. DimBillingCycle (SCD Type 1)

هدف تعریف دوره صدور فاکتور (ماهانه، فصلی، …).  
بارگذاری اولیه

1. Truncate و سپس درج همه دوره‌ها از استیجینگ.  
   نگهداری تغییرات

* دوره‌های جدید یا تغییر یافته را update/insert می‌کند؛ تاریخچه نگهداری نمی‌شود.
* به قراردادها (Contracts) پیوسته و در snapshot فکت‌ ماهانه استفاده می‌شود.

6. DimPaymentMethod (SCD Type 1)

هدف لیست روش‌های پرداخت (نقد، کارت، انتقال و …).  
بارگذاری اولیه

1. Truncate و insert مقادیر متمایز از ستون PaymentMethod در جدول پرداخت‌های استیجینگ.

نگهداری تغییرات

* فقط روش‌های جدید را insert می‌کند؛ تاریخچه نگهداری نمی‌شود.
* برای پیوستن به فکت FactCustomerPaymentTransaction کاربرد دارد.

7. DimContract (SCD Type 3)

نگهداری «وضعیت فعلی» یک قرارداد و تا دو مرتبه «وضعیت قبلی» آن.  
بارگذاری اولیه

1. truncate و insert همه قراردادها با PrevContractStatus1/2 = NULL.  
   نگهداری تغییرات

* هنگام تغییر وضعیت قرارداد، وضعیت قبلی اول به PrevContractStatus1 و آن‌یکی به PrevContractStatus2 منتقل می‌شود و وضعیت فعلی به ستون ContractStatus نوشته می‌شود.

Active → Suspended  
مثلاً اگر پرداختی به‌موقع انجام نشود یا اختلاف حقوقی پیش بیاید، ممکن است وضعیت به «معلق» (Suspended) تغییر کند.

Suspended → Active  
بعد از رفع مشکل (پرداخت بدهی، حل اختلاف)، قرارداد دوباره به حالت «فعال» برمی‌گردد.

Active → Expired → Renewed  
بعد از پایان دوره‌ی اولیه (مثلاً یک‌ساله)، وضعیت به «Expired» می‌رسد. اگر طرفین تصمیم به تمدید بگیرند، ممکن است وضعیت به «Active» یا «Renewed» تغییر کند.

Terminated  
در موارد خاص، ممکن است قرارداد زودتر از موعد به «Terminated» (قطع‌شده) تغییر کند.

در fact انباشته (accumulating snapshot) برای تحلیل چرخه عمر فاکتور استفاده می‌شود.

8. DimInvoice (SCD Type 1)

هدف ذخیره مشخصات ثابت سربرگ فاکتور (شماره، وضعیت، تاریخ ایجاد، …).  
بارگذاری اولیه

1. truncate و insert تمام فاکتورها.  
   نگهداری تغییرات

* فقط فاکتورهای جدید اضافه می‌شوند؛ آپدیت صورت نمی‌گیرد.
* کلید مرجع برای همه فکت‌های مرتبط با فاکتور.

(Fact Procedures)

1. LoadFactInvoiceLineTransactionInitialLoad

هدف: ایجاد فکت تراکنشی جزئیات سطرهای فاکتور برای گزارش‌گیری بر اساس ابعاد فاکتور، مشتری، نوع خدمت، مالیات و تاریخ.

بارگذاری اولیه  
 حذف تمام رکوردهای قبلی از FactInvoiceLineTransaction  
 اعتبارسنجی اینکه ستون‌های کلیدی staging.InvoiceLine ,InvoiceID، ServiceTypeID، Quantity، UnitPrice، TaxAmount، NetAmount خالی نباشند.  
درج همه رکوردها با JOIN به DimInvoice (InvoiceNumber)، DimCustomer (CustomerCode)، DimServiceType و DimTax و DimDate (InvoiceDate)  
نگهداری مقادیر Quantity، UnitPrice، DiscountPercent، NetAmount و TaxAmount

به‌روزرسانی‌های بعدی  
 استخراج رکوردهای جدید یا تغییر یافته از staging.InvoiceLine  
 INSERT شرطی با WHERE NOT EXISTS برای جلوگیری از درج مضاعف  
ثبت تعداد رکوردهای درج‌شده در ETLLog

1. LoadFactCustomerPaymentTransactionInitialLoad

هدف : ایجاد فکت تراکنشی پرداخت مشتریان برای تحلیل جریان نقدی و رفتار پرداخت شامل مبلغ، تأخیر، مانده و روش پرداخت.

بارگذاری اولیه  
حذف تمام رکوردهای قبلی از FactCustomerPaymentTransaction  
اعتبارسنجی NULL در ستون‌های InvoiceID، PaymentDate، Amount و PaymentMethod در staging.Payment  
 JOIN به DimInvoice، DimCustomer، DimPaymentMethod و DimDate  
 محاسبه DaysToPayment، RemainingAmount، IsFullPayment و درج PartialPaymentCount صفر

به‌روزرسانی‌های بعدی  
استخراج رکوردهای جدید از staging.Payment  
 INSERT شرطی با WHERE NOT EXISTS   
ثبت تعداد رکوردهای درج‌شده در ETLLog

1. LoadFactCustomerBillingMonthlySnapshotInitialLoad

هدف : ایجاد Periodic Snapshot ماهانه از وضعیت مالی مشتری شامل تعداد فاکتورها، مبلغ خالص و مالیات، مجموع تخفیف، مجموع پرداخت، تأخیر متوسط و بیشینه مانده.

بارگذاری اولیه  
حذف همه رکوردها از FactCustomerBillingMonthlySnapshot  
اعتبارسنجی NULL در InvoiceDate و ContractID از staging.Invoice  
 CTE اول: تجمیع InvoiceLine بر اساس مشتری و تاریخ اولین روز ماه TotalInvoiceCount، TotalNetAmount، TotalTaxAmount، TotalDiscount  
 CTE دوم: تجمیع Payment بر اساس کلید TotalPaid، AveragePaymentDelay  
 CTE سوم: محاسبه بیشینه مانده Outstanding  
 JOIN همه CTE ها با ابعاد DimCustomer، DimDate، DimBillingCycle و INSERT نهایی

به‌روزرسانی‌های بعدی  
تکرار aggregation برای داده‌های جدید (ماه‌های اضافه‌شده)  
, INSERT شرطی با WHERE NOT EXISTS روی کلید مرکب DimCustomerID، DimDateID  
 ثبت لاگ تعداد درج‌ها

1. LoadFactInvoiceLifecycleAccumulatingInitialLoad

هدف: ساخت یک Accumulating Snapshot برای دنبال کردن چرخه حیات هر فاکتور تا آخرین پرداخت: تعداد پرداخت، تأخیر اول و آخر، روز تا بسته‌شدن.

بارگذاری اولیه  
حذف رکوردها از FactInvoiceLifecycleAccumulating  
اعتبارسنجی NULL در InvoiceDate و DueDate  
 CTE برای تجمیع پرداخت‌ها SUM Amount، COUNT، MIN و MAX تاریخ پرداخت   
با ابعاد DimInvoice، DimCustomer، DimContract، DimDate برای InvoiceDateID، DueDateID، FirstPaymentDateID، FinalPaymentDateID  
داده‌های محاسبه‌شده

PaidAmount، OutstandingAmount، PaymentCount، FirstPaymentDelay، FinalPaymentDelay، DaysToClose

به‌روزرسانی‌های بعدی  
CTE PayAgg برای یک رکورد پرداخت   
 CTE SrcRaw با ROW\_NUMBER برای اطمینان از یک‌به‌یک بودن بر DimInvoiceID  
 MERGE بین Src و جدول فکت  
– WHEN MATCHED → UPDATE ستون‌های متغیر  
– WHEN NOT MATCHED → INSERT رکورد جدید   
 OUTPUT به یک جدول موقت MergeLog و شمارش INSERT/UPDATE  
ثبت لاگ نهایی

1. LoadFactCustomerContractActivationInitialLoad

هدف: ثبت تاریخ فعال‌سازی هر قرارداد برای هر مشتری (factless) تا بتوان تعداد فعال‌سازی‌ها بر اساس تاریخ را تحلیل کرد.

بارگذاری اولیه  
پاک کردن FactCustomerContractActivationFactless

JOIN staging.Contract → staging.Customer → DimCustomer و DimContract و DimDate (بر اساس StartDate)  
INSERT سه‌تایی (DimDateID، DimCustomerID، DimContractID)

به‌روزرسانی‌های بعدی  
 INSERTرکوردهای جدید با WHERE NOT EXISTS   
ثبت تعداد رکوردهای جدید در ETLLog