

Explicación lógica del negocio:

Dimensiones:

dim_date:

- **Extract:** Generar calendario (script) hasta el horizonte requerido
- **Transform:** calcular year, month, quarter, week_of_year, is_month_end, is_weekend.
- **Load:** insertar/actualizar solo si hay nuevos días. Es prácticamente estática.
- **Frecuencia:** una vez + anual
- **Tipo de carga:** Generación
- **SQL:** [dim_date.sql](#)

dim_country:

- **Extract:** Desde customers.country, employees.country, transactions.country, expenses.country.
- **Transform:** Deduplicar.
- **Load:** Generar country_key (surrogate key) para el DWH.
- **Frecuencia:** Mensual
- **Tipo de carga:** Full
- **SQL:** [dim_country.sql](#)

dim_acquisition_channel:

- **Extract:** Desde customers.acquisition_channel y/o fuentes de marketing.
- **Transform:** Normalizar nombres de canal
- **Load:** UPSERT por channel_name y generar channel_key.
- **Frecuencia:** Semanal
- **Tipo de carga:** Incremental
- **SQL:** [dim_acquisition_channel.sql](#)

dim_customer_scd:

- **Extract:** Desde customers a STG_CUSTOMERS.
- **Transform:** Limpiar duplicados por customer_id, resolver country_key haciendo lookup a DIM_COUNTRY, resolver registration_date_key via DIM_DATE y adicional se aplica la logica SCD2
 - Lógica SCD2:
 - Comparar registro de staging con la última versión is_current = TRUE por customer_id.
 - Si no hay cambios en atributos relevantes (segment, acquisition_channel, country_key): no se inserta versión nueva.
 - Si hay cambios:
 - Cerrar versión anterior (effective_to_date = hoy, is_current = FALSE).
 - Insertar nueva fila con effective_from_date = hoy, effective_to_date = NULL, is_current = TRUE.
- **Load:** Insert/Update en DIM_CUSTOMER siguiendo la lógica SCD2.
- **Frecuencia:** Diaria

- **Tipo de carga:** SCD 2
- **SQL:** [dim_customer_scd.sql](#)

dim_subscription_scd:

- **Extract:** Desde subscriptions a STG_SUBSCRIPTIONS.
- **Transform:** Resolver start_date_key y end_date_key con DIM_DATE.
 - Lógica SCD2:
 - Mismo patrón que DIM_CUSTOMER, comparando por subscription_id:
 - Cambios en plan, status, monthly_price_usd → nueva versión.
- **Load:** Insert/Update SCD2 en DIM_SUBSCRIPTION.
- **Frecuencia:** Diaria
- **Tipo de carga:** SCD 2
- **SQL:** [dim_subscription_scd.sql](#)

dim_product:

- **Extract:** Desde transactions.product_id.
- **Transform:** N/A
- **Load:** UPSERT por product_id
- **Frecuencia:** Diaria
- **Tipo de carga:** SCD 1
- **SQL:** [dim_product.sql](#)

dim_employee:

- **Extract:** Desde employees a STG_EMPLOYEES.
- **Transform:** Resolver country_key (DIM_COUNTRY) y resolver hire_date_key (DIM_DATE).
- **Load:** SCD1 o SCD2 según si quieras historial de área/salario. Para el ejercicio se dejó como SCD1
- **Frecuencia:** Diaria
- **Tipo de carga:** SCD 1
- **SQL:** [dim_employee.sql](#)

dim_provider:

- **Extract:** expenses.provider
- **Transform:** Normalizar nombres, deduplicar.
- **Load:** UPSERT por provider generando sus keys.
- **Frecuencia:** Semanal
- **Tipo de carga:** Incremental
- **SQL:** [dim_provider.sql](#)

dim_expense_category:

- **Extract:** expenses.category
- **Transform:** Normalizar nombres, deduplicar.
- **Load:** UPSERT por category generando sus keys.
- **Frecuencia:** Mensual
- **Tipo de carga:** Incremental

- **SQL:** [dim_expense_category.sql](#)

dim_payment_method:

- **Extract:** payments.method.
- **Transform:** Normalizar nombres, deduplicar.
- **Load:** UPSERT por method generando sus keys.
- **Frecuencia:** Semanal
- **Tipo de carga:** Incremental
- **SQL:** [dim_payment_method.sql](#)

Hechos:

fact_subscription_mrr:

- **Extract:** Desde DIM_SUBSCRIPTION (vigas actuales) + calendario mensual.
- **Transform:** Para cada mes y suscripción activa en ese mes:
 - mrr_usd = monthly_price_usd.
 - is_active_flag = 1 si la suscripción está vigente ese mes.
 - is_new_flag = 1 en el mes de alta.
 - is_churned_flag = 1 en el mes en que se detecta churn (por ejemplo, cuando el status pasa a canceled y no se reactiva).
 - Resolver customer_key (join DIM_CUSTOMER por customer_id usando la versión vigente en ese mes).
 - Resolver country_key vía DIM_COUNTRY.
- **Load:** Insertar una fila por (subscription_key, date_key_mes).
- **Frecuencia:** Mensual
- **Tipo de carga:** Snapshot
- **SQL:** [fact_subscription_mrr.sql](#)

fact_customer_acquisition:

- **Extract:** Desde DIM_CUSTOMER (primera versión de cada customer_id).
- **Transform:**
 - acquisition_date_key = registration_date_key.
 - Resolver country_key (DIM_COUNTRY).
 - Resolver acquisition_channel_key (lookup DIM_ACQUISITION_CHANNEL usando acquisition_channel).
 - acquired_customers = 1.
- **Load:** Una fila por cliente adquirido.
- **Frecuencia:** Diaria
- **Tipo de carga:** Incremental
- **SQL:** [fact_customer_acquisition.sql](#)

fact_sales:

- **Extract:** Desde transactions a STG_TRANSACTIONS.
- **Transform:**
 - Resolver transaction_date_key (DIM_DATE).
 - Resolver customer_key (DIM_CUSTOMER, versión vigente en fecha transacción).

- Resolver product_key (DIM_PRODUCT).
- Resolver country_key (DIM_COUNTRY, desde transaction.country o customer).
- Calcular total_usd = quantity * unit_price_usd si no viene.
- **Load:**
 - Insertar una fila por transacción (o por línea de detalle si existiera).
 - El transaction_id se almacena como clave de negocio (degenerate) para link con FACT_PAYMENTS.
- **Frecuencia:** Diaria
- **Tipo de carga:** Incremental
- **SQL:** [fact_sales.sql](#)

fact_payments:

- **Extract:** Desde payments a STG_PAYMENTS.
- **Transform:**
 - Resolver payment_date_key (DIM_DATE).
 - Resolver customer_key: via join payments.transaction_id → transactions.customer_id → DIM_CUSTOMER.
 - Resolver transaction_key: buscar en FACT_SALES por transaction_id (o usar el transaction_id como business key referenciado).
 - Resolver payment_method_key (DIM_PAYMENT_METHOD).
 - Resolver country_key (DIM_COUNTRY).
- **Load:** Insertar una fila por pago.
- **Frecuencia:** Diaria
- **Tipo de carga:** Incremental
- **DLL:** [fact_payments.sql](#)

fact_expenses:

- **Extract:** Desde expenses a STG_EXPENSES.
- **Transform:**
 - Resolver date_key (DIM_DATE).
 - Resolver provider_key (DIM_PROVIDER).
 - Resolver expense_category_key (DIM_EXPENSE_CATEGORY).
 - Resolver country_key (DIM_COUNTRY).
 - Resolver channel_key (DIM_ACQUISITION_CHANNEL) si el gasto se puede asociar a un canal (si no, NULL).
- **Load:** Insertar una fila por gasto.
- **Frecuencia:** Diaria
- **Tipo de carga:** Incremental
- **SQL:** [fact_expenses.sql](#)

fact_employee_cost:

- **Extract:** Desde employees + calendario mensual (o sistema de nómina).
- **Transform:**
 - month_date_key = último día de mes en DIM_DATE.
 - Resolver employee_key (DIM_EMPLOYEE).

- Resolver country_key (DIM_COUNTRY).
 - Calcular salary_cost_usd mensual
 - Calcular days_worked (de asistencia, nómina o reglas de negocio).
- **Load:** Una fila por empleado y mes.
- **Frecuencia:** Mensual
- **Tipo de carga:** Snapshot
- **SQL:** [fact_employee_cost.sql](#)

Orquestación:

1. Cargar/actualizar DIM_DATE y catálogos (DIM_COUNTRY, DIM_ACQUISITION_CHANNEL, DIM_PROVIDER, DIM_EXPENSE_CATEGORY, DIM_PAYMENT_METHOD, DIM_PRODUCT).
2. Cargar DIM_CUSTOMER (SCD2).
3. Cargar DIM_SUBSCRIPTION (SCD2).
4. Cargar DIM_EMPLOYEE.
5. Cargar hechos de “base”: FACT_SALES, FACT_EXPENSES, FACT_EMPLOYEE_COST y FACT_PAYMENTS
6. Construir hechos derivados / snapshots: FACT_SUBSCRIPTION_MRR y FACT_CUSTOMER_ACQUISITION