

Mi pregunta...

Según Strengolt hay dos patrones comunes para el consumo de datos: El consumo directo (sin copia) y el consumo a través de DDS (con copia).

Expícame estos dos conceptos y contame cómo sería un flujo de trabajo con ellos en la vida real, por favor.

Respuesta

¡Excelente pregunta! Esta es la distinción más importante y práctica de todo el capítulo 11, y es fundamental para entender cómo funciona la arquitectura que propone Strengolt.

Has llegado al punto donde se decide cómo se “cocinan” los datos para generar valor. Te lo explicaré de forma sencilla, con una analogía, los conceptos clave y un flujo de trabajo real para cada uno.

La Analogía: El Restaurante y la Cocina Profesional

Imagina que los **Productos de Datos** son ingredientes de alta calidad en un mercado de agricultores. Están limpios, etiquetados y listos para ser usados. Ahora, como consumidor, tienes dos formas de “comer”:

1. **Consumo Directo (Ir al Restaurante):** Vas a un restaurante que usa esos ingredientes. Pides un plato del menú (ej: “ensalada de tomates”). El chef (el motor de consultas) va al puesto de tomates, los corta, les pone un poco de aderezo (transformación ligera al vuelo) y te lo sirve. Tú te lo comes y te vas. **No te llevas los tomates a casa.** El resultado es para consumo inmediato.
2. **Consumo a través de DDS (Comprar los Ingredientes y Cocinar en Casa):** Eres un chef profesional y quieres crear un plato complejo y único (ej: una *sopa desconstruida de 7 vegetales con espuma de albahaca*). Vas al mercado, compras los mejores tomates, cebollas, pimientos, etc. (varios *data products*). Te los llevas a **tu propia cocina profesional** (el **DDS**), donde los combinás, los transformas con técnicas complejas y creas tu plato. **Guardas la receta y el resultado** en tu cocina para usarlo más tarde.

Explicación de los Conceptos

Antes de seguir, está bueno clarificar el significado de “transformación”:

Una “transformación” es cualquier operación que modifica los datos originales de un producto de datos para hacerlos adecuados y valiosos para un caso de uso específico del consumidor.

1. Consumo Directo (sin copia)

- **¿Qué es?:** Es el patrón donde la herramienta del consumidor (como un *dashboard* de BI o un *notebook* de un analista) lee los datos **directamente del producto de datos original cada vez que los necesita**. Las transformaciones que se realizan son simples, se hacen “al vuelo” y los resultados no se guardan en un nuevo lugar permanente.

- **¿Cuándo se usa?:** Es ideal para casos de uso **simples y de solo lectura**:

- **Reporting y Dashboards:** Un *dashboard* en Tableau o Power BI que muestra las ventas diarias.
 - **Análisis Ad-hoc:** Un analista que necesita responder una pregunta rápida con una consulta SQL.
 - Sistemas operacionales que necesitan consultar información de referencia (ej: una lista de productos).

- **Ventajas:**

- **Simpleza:** No hay que construir ni mantener *pipelines* de datos complejos.
 - **Sin duplicación:** Evita los costos y riesgos de seguridad de tener múltiples copias de los datos.
 - **Datos frescos:** Siempre estás leyendo la última versión del producto de datos.

- **Desventajas:**

- **Rendimiento limitado:** No funciona si las consultas son muy pesadas o si necesitas combinar muchos productos de datos. La experiencia del usuario sería muy lenta.

2. Consumo a través de Domain Data Stores (DDS) (con copia)

- **¿Qué es?:** Es el patrón donde el consumidor **extrae los datos** de uno o varios productos de datos, los **copia a su propio entorno controlado** (el DDS), y allí los combina y transforma para crear nuevos conjuntos de datos, optimizados para su caso de uso específico.

- **¿Cuándo se usa?:** Es necesario para casos de uso **complejos**:

- **Entrenamiento de Modelos de Machine Learning:** Se necesita una tabla “plana” y ancha con todas las características (*features*).
- **Análisis Avanzado:** Cuando necesitas combinar datos de muchos dominios (Ventas, Marketing, Logística) para obtener una visión de 360°.
- **Transformaciones de Negocio Complejas:** Cuando la lógica para generar los *insights* es demasiado pesada para ejecutarla “al vuelo”.
- **Preservar Resultados:** Cuando los nuevos *insights* generados son valiosos y necesitan ser guardados para análisis futuros.

- **Ventajas:**

- **Potencia y Flexibilidad:** Permite realizar cualquier tipo de transformación y análisis, por muy complejo que sea.
- **Alto Rendimiento:** Como los datos ya están pre-procesados y optimizados en el DDS, las consultas finales son muy rápidas.

- **Desventajas:**

- **Complejidad:** Requiere construir y mantener un *pipeline* de ETL/ELT para mover y transformar los datos.
 - **Duplicación de datos:** Se crea una copia, con los costos y riesgos que ello implica.
 - **Latencia de datos:** Los datos en el DDS tendrán un retraso respecto a los productos de datos originales (el tiempo que tarde en ejecutarse el *pipeline*).
-

Flujo de Trabajo en la Vida Real

Imaginemos una empresa de comercio electrónico.

Flujo de Trabajo: Consumo Directo

Caso de Uso: El equipo de Ventas necesita un *dashboard* en tiempo real que muestre las ventas totales por categoría de producto de la última hora.

1. **Descubrimiento:** Un analista de Ventas va al catálogo de datos y encuentra un *data product* llamado “Transacciones de Venta en Tiempo Real”, que es servido por el dominio de “Pedidos”. El catálogo indica que es una tabla en Snowflake que se actualiza cada pocos minutos.
2. **Conexión:** El analista abre su herramienta de BI (Tableau) y se conecta directamente a esa tabla en Snowflake.
3. **Análisis “al vuelo”:** Dentro de Tableau, arrastra el campo “monto_venta” y el campo “categoría_producto”. Aplica un filtro para la última hora. Tableau envía una consulta SQL (GROUP BY categoría_producto) a Snowflake.
4. **Resultado:** Snowflake procesa la consulta y devuelve el resultado agregado. El *dashboard* se muestra al gerente de ventas. Cada vez que actualiza el *dashboard*, se envía una nueva consulta a Snowflake.
 - **Conclusión:** No se creó ninguna copia permanente de los datos. El *dashboard* es solo una “ventana” que lee directamente del producto de datos.

Flujo de Trabajo: Consumo a través de DDS

Caso de Uso: El equipo de Ciencia de Datos quiere construir un modelo para predecir qué clientes tienen un alto riesgo de abandonar la empresa (*churn*).

1. **Descubrimiento:** Los científicos de datos van al catálogo y se dan cuenta de que necesitan combinar tres *data products*:
 - “Historial de Compras” (del dominio de Pedidos).
 - “Tickets de Soporte” (del dominio de Atención al Cliente).
 - “Actividad en la App” (del dominio de Producto).
2. **Creación del DDS:** El equipo usa la plataforma de autoservicio para crear su propio **DDS**. Esto podría ser un nuevo esquema en Snowflake llamado `datascience_churn_project`.
3. **Construcción del Pipeline (la copia):** Usando una herramienta como dbt, crean un *pipeline* que:
 - Se conecta a los tres *data products*.

- Extrae los datos relevantes de cada uno.
 - Los **copia** en tablas “brutas” dentro de su DDS (`datascience_churn_project.raw_compras`, etc.).
4. **Transformación y Creación de Valor:** Dentro de su DDS, ejecutan transformaciones complejas para combinar las tres tablas y crear nuevas características (*features*), como “días_desde_ultima_compra” o “numero_tickets_abiertos”. El resultado es una única tabla final llamada `datascience_churn_project.training_dataset`.
5. **Resultado:** Los científicos de datos usan esta tabla final para entrenar su modelo de *churn*. La tabla es **persistente** y se guarda dentro de su DDS.
- **Conclusión:** El consumo directo era imposible. Necesitaban copiar y transformar los datos en su propio entorno controlado (el DDS) para poder realizar una tarea tan compleja.