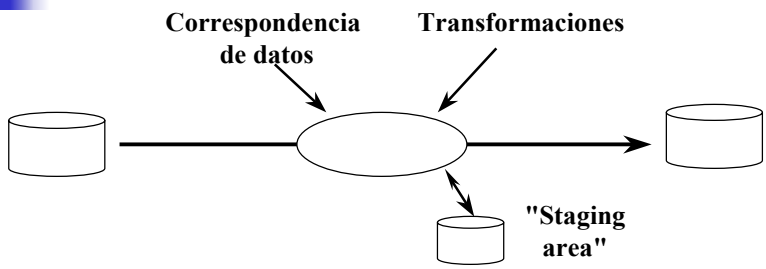
Carga inicial



- **Extracción de datos**
- **Transformación (Limpieza)**
- **Herramientas de ETL**

Extracción de datos

- **Involucra *técnicas* para la extracción de información en las fuentes.**
  - Programas específicos (ej.: C, Cobol, PL/SQL)
  - Herramientas ETL.
- **Desde el punto de vista de *arquitectura*, el enfoque utilizado consiste en asociar *una componente por c/ fuente*.**
  - Se le suele llamar *wrapper*.
  - Función:
    - básica: Proveer una descripción de los datos almacenados en la fuente en un modelo de datos común.
    - en contexto DW: (básica) + detectar y extraer cambios de interés ocurridos en las fuentes y propagarlos.

Descripción en un MD común

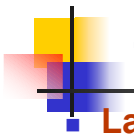
- **Empaquetar la fuente de datos ofreciendo el mismo formato y modelo de datos que el usado en el sistema de DW.**

- **Caso 1:**
  - Fuente de datos: conjunto de docs XML
  - Modelo del DW: relacional
  - Se han propuesto generadores de wrappers [jedi, w4f...]

Descripción en un MD común

- **Caso 2:**
  - Fuente de datos y sistema de DW el mismo modelo de datos.
  - Función del wrapper:
    - transformar formato de datos
    - soporte para la comunicación
- **Caso 2 típico:**
  - Fuente y sistema DW ambos relacionales
  - Wrapper = componentes de "middleware"

    Ej.: ODBC/OleDB (Microsoft), IDAPI (Borland), OCI (Oracle)...



Transformación

- **La limpieza de datos constituye *uno* de los procesos dentro de la *transformación de datos* para la construcción de un DW.**
  - La transformación de datos involucra:
    - cambios en las estructuras de representación de los datos
    - limpieza
    - integración de diferentes valores y estructuras de datos
    - resumen y agrupamiento de datos
  - **El laboratorio consistirá en experimentar la programación de *transformaciones* usando una herramienta específica.**

Limpieza de datos

- ***"Data cleaning*" ("*data cleansing*")**
- **Presente en la mayoría de los procesos de migración de datos.**
- **Su objetivo es la *calidad de los datos* obtenidos al final de la migración.**
  - Calidad de datos como juicio sobre la condición o el estado de los datos a examinar.
  - El nivel de calidad es definido según los requerimientos de las aplicaciones.

Ejemplos de datos "sucios"

- **Diferentes formatos de datos para el mismo atributo.**
  - Ej.: la información sobre el departamento en un atributo dirección puede aparecer bajo las siguientes formas:
    - abreviación
    - nombre
    - un código
- **Conflicto entre la descripción del atributo y los valores.**
  - Ej.:
    - Un atributo nombre puede contener nombres personales y comerciales.
    - Rangos
    - Escalas

Ejemplos de datos "sucios"

- **Atributos de texto libre pueden ocultar información importante.**
  - Ej.: algunas etiquetas como "C/O" dentro de nombres y direcciones, "Fax: ", ...
- **Valores faltantes que deben ser asignados de acuerdo al esquema destino.**
  - Más que sucios serían incompletos.

Ejemplos de datos "sucios"

- **Valores inconsistentes para la misma entidad.**
  - Ej.: errores de tipografía
- **Información duplicada originada de tener la misma información sobre la misma entidad pero usando una clave diferente.**
  - Esta situación puede ocurrir tanto trabajando con una o varias fuentes origen.

Funcionalidades de ayuda

- **Las herramientas de migración gral y orientadas a DW ofrecen funcionalidades para ayudar a resolver los problemas anteriores:**
  - Funciones de conversión y de normalización
  - Limpieza para casos y dominios específicos
  - Algoritmos de correspondencias entre campos equivalentes de fuentes diferentes.
    - Independientes del dominio
    - Basados en reglas

Conversión y normalización

- **Conversión: se ofrece mediante un wrapper para cada fuente o tipo de fuente.**
- **Normalización: usar un formato común para todos los datos pertenecientes al mismo tipo para permitir la comparación entre campos.**
  - Ej.: Strings a mayúsculas o a minúsculas
    Fechas en formato "dd/mm/yyyy"
- **Otros tipos de normalización pueden ser orientadas a comparar campos equivalentes.**
  - Ej.: Corregir guiones que separan palabras.

Limpieza para casos y dominios específicos

- **Ejemplo: Nombres y direcciones**
- **Las técnicas utilizan metainformación.**
  - Tablas para buscar datos válidos (ej.: códigos postales)
  - Diccionarios para buscar sinónimos y abreviaciones (e.g. "Apto", "Apt.", "Apartamento").
- **Ejemplo de herramienta:**
  - Oracle Pure Integrate

Algo. de correspondencia entre campos

- **"Field Matching Algorithms".**
- **Problema: identificar las *mismas* entidades descriptas por valores diferentes.**
- **Dos conjuntos de métodos:**
  - Métodos independientes del dominio
  - Métodos basados en reglas

Métodos independientes del dominio

- **Ejemplo 1: Algoritmos de [Monge,Elkan 1996]**
  - "Degree of matching" entre dos campos
  - Dos strings están en correspondencia si:
    - son iguales, o
    - uno es prefijo del otro
- **Ejemplo 2: Oracle Pure Integrate**
  - Provee dos métodos para comparar posibles registros "sucios" entre diferentes fuentes:
    - "matching" basado en claves
    - "matching" basado en campos no claves ("fuzzy matching")

Métodos basados en reglas

- **Idea de los métodos:**
  - Toman en cuenta un conjunto de reglas que establecen equivalencias entre registros de diferentes bds.
- **Dos categorías de métodos:**
  - reglas especificadas por el usuario (desarrollador, ...)
  - reglas derivadas automáticamente aplicando técnicas de data mining a las fuentes.

Reglas definidas por el usuario

- **Ejemplo: Oracle Pure Integrate**
  - Permite la especificación de reglas de combinación de registros usando criterios predefinidos.
    - Ej.: elegir el valor de campo que ocurre más frecuentemente
- **Desventaja:**
  - las reglas a escribir es una tarea de mucho tiempo
  - las reglas nunca cubren todas los posibles errores en los datos

Reglas derivadas automáticamente

- **Idea general**
  - Se calculan estadísticas que involucran palabras y relaciones entre ellas.
- **El resultado devuelto por estos métodos es un conjunto de reglas identificadas sobre los datos.**
- **Desventaja**
  - Nivel de incertidumbre sobre las reglas derivadas.

Reglas derivadas automáticamente

- **Ejemplo: Herramienta WizRule**
  - Regla if-then

    if Customer is "Summit" and Item is Computer type X then Salesperson = "Dan Wilson"

    Rule's probability: 0.98

    Rule exists in 103 records

    Error probability < 0.1

Resumen

Por más detalle

- **[JLVV2000]**
  - M. Jarke, M. Lenzerini, Y. Vassiliou, P. Vassiliadis. "Fundamentals of Data Warehouses". Springer-Verlag, 2000.
- **[Monge & Elkan, 1996].**

  "The field matching problem: Algorithms and Applications". Proc. of Knowledge Discovery and Data Mining Conf. (KDD), 1996.
- **Oracle Pure Integrate**

  En home page de Oracle
- **WizRule Tool**
  - http://www.wizsoft.com/

Herramientas ETL

Extraction, Transformation and Loading

- **Características generales**
- **Microsoft DTS**

Características generales

- **Objetivo principal**
  - *facilitar* el desarrollo de aplicaciones que *migran* datos aplicando *transformaciones.*
- **En este tipo de aplicaciones, los objetos típicos a definir:**
  - conexiones
  - estructuras de los depósitos de datos
  - correspondencias y transformaciones entre los depósitos
  - excepciones
  - planificaciones de las transformaciones

Características generales

- **Las herramientas ETL son *ambientes especializados* que permiten la definición y manipulación de objetos típicos en aplicaciones de intercambios de datos.**
  - Facilidades para la modificación y mantenimiento de las aplicaciones.
- **En estas herramientas, el data warehouse y/o los data marts son vistos como depósitos adonde migrar datos transformados.**

Características generales

- **En general, ETLs *NO* ofrecen funcionalidades específicas para:**
  - la captura de cambios en los datos,
  - la integración de esquemas y datos
- **ETLs son "pobres" en cuanto al manejo de excepciones.**
  - No significa que no se puedan manejar sino que su manejo es aún "engorroso".
- **Las herramientas pueden clasificarse en 3 categorias**
  - "Loaders"
  - Generadores de código
  - Ambientes especializados

"Loaders"

- **Importadores/Exportadores convencionales entre archivos ascii y Rdbms.**
  - E.g. SQL*Loader de Oracle.
- **Ofrecen parametrización mediante archivos de control.**
  - E.g. delimitador, formato de fechas, ...
- **Adecuado para cargas sin demasiadas transformaciones en los datos a partir de archivos de texto simples.**
- **No adecuado**
  - diferentes fuentes de datos (no sólo texto)
  - transformaciones complejas
  - planificación de diferentes procesos de carga

Generadores de código

- **Editores gráficos permitiendo definir**
  - conexiones a fuentes de datos
  - transformaciones entre los datos
- **Generan programas en lenguajes como Cobol, C, RPG, ABAP, ...**
  - Pueden ser afinados posteriormente.
- **Orientados particularmente a extracción directa en mainframes.**
- **El inconveniente es la gestión y coordinación de una gran cantidad de programas.**
- **E.g. Passport (Carleton), Warehouse Manager (Prism).**

Ambientes especializados

- **Editores gráficos para definición y planificación de procesos de carga.**
- **Lenguajes de programación para definir las transformaciones.**
  - Proveen el motor de ejecución de los programas escritos en estos lenguajes.
  - Ofrecen funciones predefinidas y permiten el agregado de funciones definidas por el usuario.
- **Mecanismos para el control del flujo de los procesos.**

Microsoft DTS

- **Se trata de una componente predefinida del RDBMS SQL Server de Microsoft.**
- **Como cliente, esta componente se presenta bajo 3 formas:**
  - DTS Designer

    Asistente gráfico para la definición de los procesos (paquetes) encargados de la transformación de datos.
  - DTS Import y Export wizards

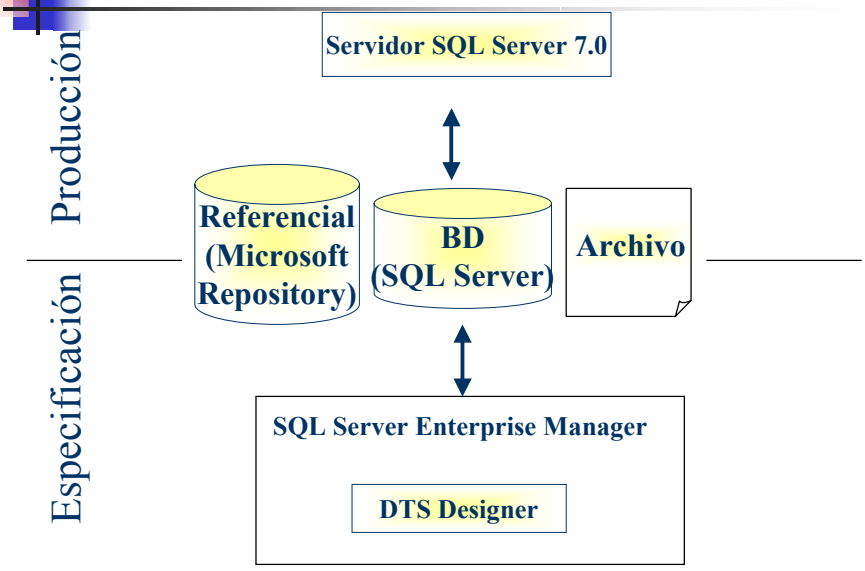
    Asistente gráfico para la definición de paquetes más simples.
  - DTS programming interfaces (API)

    Interfaces para ser usadas desde leng. de programación (VBasic, VC++)
- **Para la ejecución y planificación**
  - Servidor SQL Server 7.0 (incluyendo el servicio Agent)

Microsoft DTS (1)

- **Permite el uso de un referencial para almacenar todas las definiciones.**
  - Necesariamente: Microsoft Repository
  - Otras formas de almacenar las definiciones:
    - en archivo con formato específico
    - en SQL Server (dentro de la bd de nombre *msdb*)

DTS / Arquitectura de la herramienta



DTS / Acceso a los datos

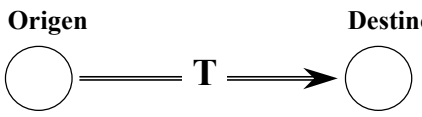
- **DTS se apoya fuertemente en el acceso y almacenamiento a través de OLE DB.**
  - DTS es un consumidor OLE DB
- **Provee conexiones especificas para archivos de texto.**

DTS / Tareas

- **DTS se basa en la definición de tareas y un orden parcial entre ellas.**
- **La tarea básica que permite definir correspondencias y transformaciones entre la fuente de datos origen y la fuente de datos destino:**
  - Transform data (data pump)
- **Transform data**
  - Accede y almacena datos a traves de Ole DB o archivos de texto.
  - Copia y/o transforma datos entre las fuentes.
  - La transformación puede tratarse de (extremos):
    - una simple copia entre columnas
    - una invocación de un script (VB Script, JScript)

DTS / Transform data

Simplificando,



T:
- **copia**
- **copia más transformaciones incluyendo funciones en un lenguaje script (VB Script, JScript).**

se interpreta como:

```
for each o ∈ Origen
        columnas(d) = T ( columnas(o) );
        insert d en Destino;
endfor
```


DTS / Otras tareas

- **Data Driven**
  - Permite realizar actualizaciones y borrados además de inserciones.
- **Execute SQL**
  - Permite definir un conjunto de instrucciones SQL.
- **Execute Process**
  - Permite invocar a un ejecutable (.exe, .bat)
- **Send Mail**

DTS / Otras tareas (1)

- **Bulk Insert**
  - Método rápido para copiar datos en archivos ascii a una bd SQL Server. No permite definición de transformaciones.
- **Active X Script**
  - Permite invocar un Active X script (VB Script, Perl Script, Java Script)

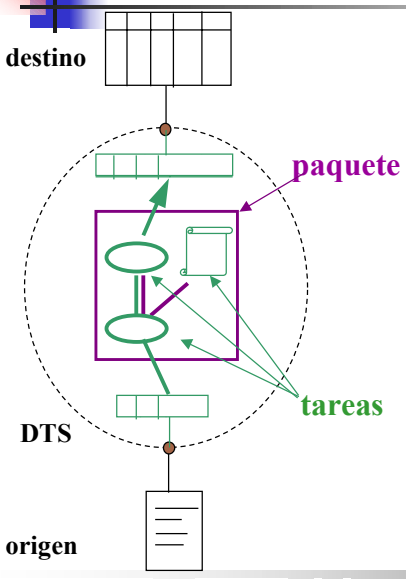
DTS / Paquete

- **Un paquete es un "workflow" que define un proceso de transformación.**
- **Un paquete es un grafo donde:**
  - los nodos son *tareas*, y
  - los arcos representan *pasos* que definen la secuencia en la cual se ejecutarán las tareas.
- **Un paso puede tener asociado una restricción de precedencia definiendo cómo el resultado de una tarea determina la ejecución de la otra.**
  - on success
  - on failure
  - on completion

DTS / Paquete (1)

- **Posible confusión "gráfica":**
  - Transform data es un nodo (tarea) dentro del workflow.

DTS / Etapas en la definición



1. **Conexión**
   (Acceso a las fuentes de datos)

2. **Tareas**
   - **Importación de estructuras**

3. **Paquetes**

4. **Activación de paquetes**
   - **tiempo**

Conclusión

- **A nuestro conocimiento, no hay *una* herramienta que realice o ayude a realizar todas las tareas que requiere instanciar (poblar) un data warehouse relacional.**
- **Variedad enfatizando algunos aspectos más que otros**
  - Análisis del estado de los datos origen
  - Limpieza
  - Extracción, transformación y carga
  - Captura de cambios en los datos