Concepción de Sistemas de Información

Instituto de Computación – Facultad de Ingeniería – Universidad de la República

SEMINARIO

Estudio de modelos y técnicas de workflow en vista a la definición de un proceso para la carga y mantenimiento de data warehouses

Presentación día 18 con fecha 28/12/2001

Artículo: **Conceptualización de DW de Enseñanza**

Expositor: **Ignacio Larrañaga**

Índice

- Contexto
  - Objetivos Generales y Específicos.
- Arquitectura del DW de Enseñanza
  - Data Warehouse, Repositorio de Fuente de Datos, Datos Auxiliares y Correspondencias
- Proceso de Carga
  - Data Warehouse, Repositorio de Fuente de Datos, Datos Auxiliares y Correspondencias
- Posibles Líneas de Trabajo
- Conclusiones
- Bibliografía

Informacion extraida de [SIAEE]

Objetivos Generales

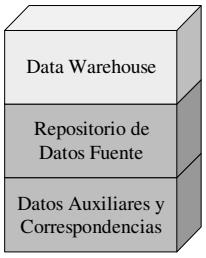
- El objetivo principal de este proyecto es construir un Sistema Informático al servicio de la Investigación Educativa, que permita el acceso flexible a información sobre actividades curriculares para la construcción de perfiles numéricos y evaluación de parámetros en la relación educación-plan de estudios.
- Este sistema debe servir como apoyo a la toma de decisiones tanto en la asignación optima de los recursos educativos como en la evolución de las curriculas.
- Este proyecto apunta también al fortalecimiento de los sistemas de información de apoyo a la toma de decisiones en la Facultad de Ingeniería, especialmente en el área de la enseñanza.

Objetivos Específicos

- Los objetivos específicos del proyecto se enmarcan en los siguientes ítems:
  - Fortalecimiento de las UAE y mejoramiento de la formación y evaluación docente vinculadas con el asesoramiento a los docentes en materia de gestión, de metodologías y de evaluación de enseñanza.
  - Construcción de una base de datos con datos extraídos del Sistema de Bedelia, e implementación de su actualización de forma que sea transparente para los usuarios que la exploten.
  - Implementación de reportes y consultas pre-definidas sobre valores de indicadores significativos a ser evaluados en forma permanente en la facultad.

Arquitectura del DW

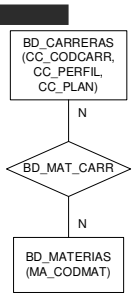
- Esta estructurado en tres capas.
- La tercer capa corresponde al Data Warehouse propiamente dicho .



Los campos dentro de parentesis son los que identifican las entidades, por simplicidad los otros no fueron agregados.

MER del DW

- Para entender que es lo que hay en este comentare brevemente la información que hay en el DW.
- Existen carreras, que pertenecen a un plan y sobre las cuales se manejan perfiles.
- Estas carreras tienen materias que a su vez pueden estar en varias carreras.

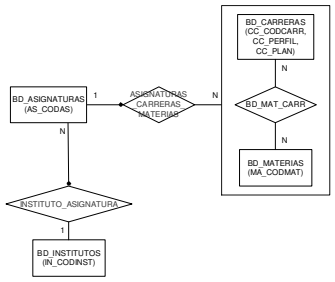


Data Warehouse
Repositorio de Datos Fuente
Datos Auxiliares y Correspondencias

MER del DW (II)

En rojo se encuentran las relaciones que no tenían correspondiente en la base de datos.

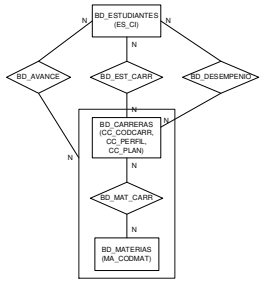
- Estas materias a su vez agrupan asignaturas que son dictadas por un instituto.



Data Warehouse
Repositorio de Datos Fuente
Datos Auxiliares y Correspondencias

MER del DW (III)

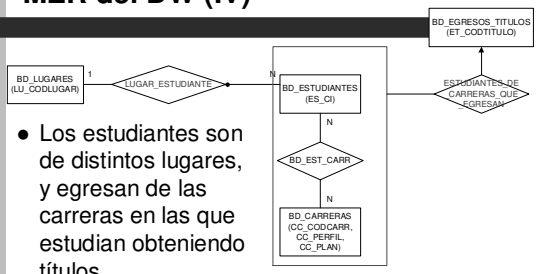
- Por otro lado están los estudiantes, que estudian en diferentes carreras, y tienen cierto desempeño; al mismo tiempo que avanzan en las materias de la carrera.



Data Warehouse
Repositorio de Datos Fuente
Datos Auxiliares y Correspondencias

MER del DW (IV)

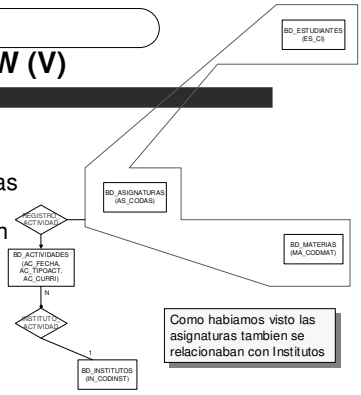
- Los estudiantes son de distintos lugares, y egresan de las carreras en las que estudian obteniendo títulos.



Data Warehouse
Repositorio de Datos Fuente
Datos Auxiliares y Correspondencias

MER del DW (V)

- Estos estudiantes, de asignaturas que corresponden a materias, registran actividades, asociadas a institutos.

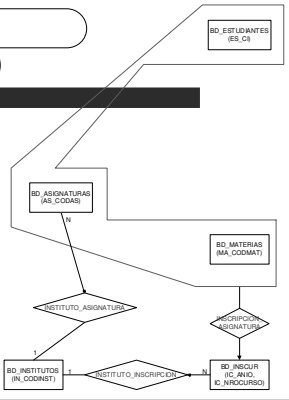


Como habiamos visto las asignaturas tambien se relacionaban con Institutos

Data Warehouse
Repositorio de Datos Fuente
Datos Auxiliares y Correspondencias

MER del DW (VI)

- Finalmente, los estudiantes de asignaturas (correspondientes con materias) se inscriben a cursos dictados por estos institutos (claramente el instituto debe dictar la asignatura, etc., etc.).



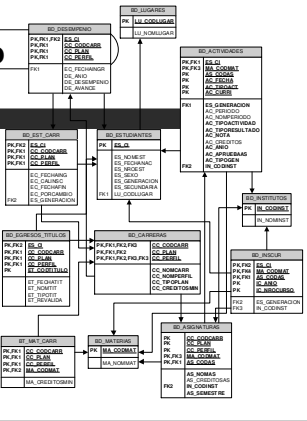
Data Warehouse
Repositorio de Datos Fuente
Datos Auxiliares y Correspondencias

Diseño Físico del DW

- Básicamente solo para mostrar la complejidad del diseño de este.

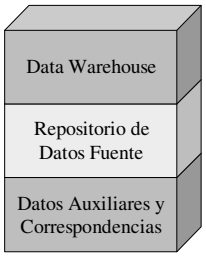
Las claves foraneas se deducieron, dado que no estaban en el manual [MANDWE] porque lo que tome fue un manual de usuario.

Tampoco se si en realidad se establecieron.

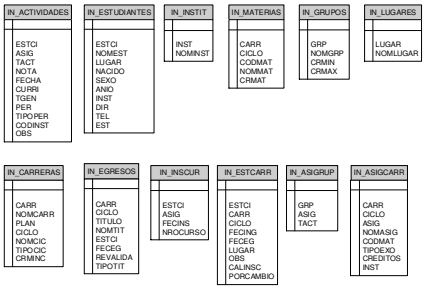


Arquitectura del DW (II)

- La segunda capa o capa intermedia corresponde a aquellas tablas que son cargadas con los datos extraídos mediante los scripts de la base de datos de Bedelía.



Diseño Físico de la Capa Intermedia

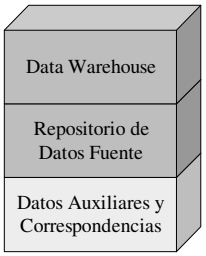


Problemas vistos al revisar la Capa Intermedia

- Problemas vistos al revisar la documentación:
  - Las tablas de datos para esta capa no tienen claves.
  - Problemas con tipos de datos (p.e. IN_ACTIVIDADES.FECHA Numeric(18,0))

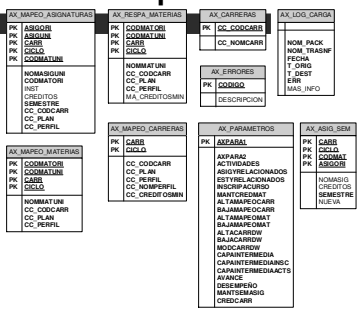
Arquitectura del DW (III)

- La primer capa está formada por tablas auxiliares y tablas que se utilizan para establecer correspondencias entre los datos de Bedelía y los datos del Data Warehouse.



Diseño Físico de la Capa Auxiliar

- Estas tablas, básicamente de mapeos, catalogos internos y logs forman la capa intermedia.

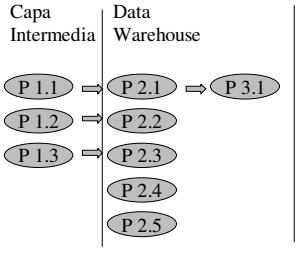


Datos de la Capa Auxiliar

- Catalogos:
  - AX_CARRERAS, AX_ERRORES, AX_ASIG_SEM
- Mapeos:
  - AX_MAPEO_ASIGNATURAS, AX_MAPEO_CARRERAS, AX_MAPEO_MATERIAS,
- Logs:
  - AX_PARAMETROS, AX_LOG_CARGA

Proceso de Carga

- Procesos:
  1. Capa Intermedia
     1. Base de Datos
     2. Inscripciones a Curso
     3. Actividades
  2. Data Warehouse
     1. Estudiantes, Carreras, Inscripción a Carreras, Egresos
     2. Inscripciones a Curso
     3. Actividades
     4. Avance por Materia
     5. Desempeño por Carrera
  3. Data Warehouse (II)
     1. Asignaturas, Materias, Materias de Carreras, Asignaturas de Carreras



P1.1 (Capa Intermedia – Base de Datos)

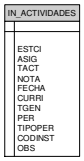
- En este proceso se realiza la carga de la capa intermedia (con datos de estudiantes, carreras, asignaturas, etc.)
- Por lo que entiendo de este proceso básicamente se trata de levantar datos de archivos ASCII (porque por alguna razón no se dio acceso a los datos de Bedelia directamente). Por lo que aparentemente no reviste ninguna complejidad y podría ser realizado con cualquier utilitario para importar archivos de texto en tablas.

P1.2 (Capa Intermedia – Inscripciones a Curso)

- Con este proceso se realiza la carga de la capa intermedia con datos de inscripciones a curso.
- Merece la misma observación que en el caso anterior, o sea aparentemente no reviste complejidad excesiva.

P1.3 (Capa Intermedia – Actividades)

- Con este proceso se realiza la carga de la capa intermedia con datos de las actividades de los estudiantes.
- Nuevamente lo mismo que en el caso anterior.



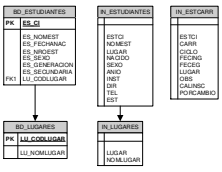
P2.1 (DW – Estudiantes, Carreras, Estudiantes insc. Carreras, Egresos)

- En esta etapa se cargan en el DW a partir de lo ya tomado en la capa intermedia todo lo que se lista en el titulo de esta diapositiva.
- Como se mostrara con dos ejemplos (Estudiantes y Carreras) tomando "conceptualmente" lo que hay que hacer se podría realizar usando SQL.
- Es de notar que no se llego al detalle de los procesos, por lo que podría haber aspectos que invaliden esta suposición.

Mapeo de Estudiantes

- Conjeturando la siguiente SQL podría realizar la carga:

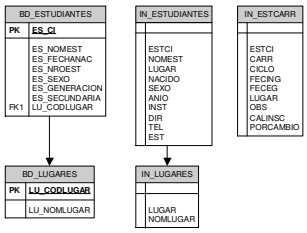
```
select ESTCI as ES_CI, NOMEST as
    ES_NOMEST, NACIDO as
    ES_FECHANAC, EST as ES_NROEST,
    SEXO as ES_SEXO, "Generación" +
    FECING as ES_GENERACION, INST as
    ES_SECUNDARIA, LUGAR as
    LU_CODLUGAR
from IN_ESTUDIANTES, IN_LUGARES,
    IN_ESTCARR
where IN_ESTUDIANTES.ESTCI =
    IN_ESTCARR.ESTCI and
    IN_ESTCARR.FECING = (
    select min (FECING)
    from IN_ESTCARR
    where ESTCI = IN_ESTUDIANTES.ESTCI)
```



Mapeo de Estudiantes (II)

- **Finalmente nos falto:**

```
select LUGAR as
   LU_CODLUGAR
   , NOMLUGAR
   as
   LU_NOMLUGAR
from
   IN_LUGARES
```



Mapeo de Carreras

- Carreras
  - Se fusionan varios códigos de carrera en uno único.
  - Se ofrecen valores por defecto.
- Mapeo de carreras
  - *<Carrera, Ciclo, Nombre Carrera, Nombre Ciclo, Plan, Créditos> => <Carrera, Nombre Carrera>*
  - Mapeo 1:1.
  - Si no hay créditos se solicitan en el momento de establecer el Mapeo.
- Créditos de Carreras
  - En base al mapeo establecido y que cada mapeo tiene créditos se permiten modificar.

Mapeo de Carreras (II)

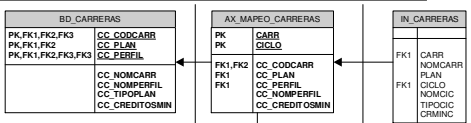


Tabla final en el Data Warehouse

Tablas Auxiliares de Mapeo y Definicion.

Tabla Original

Mapeo de Carreras (III)

- En la diapositiva anterior se puede ver como todos los datos en el data warehouse se obtienen directamente de las dos capas inferiores.
- Conjeturando se podría obtener con una SQL como esta (o una variación operando con los campos para formar nuevos valores concatenando cosas):

```
select CC_CODCARR, CC_PLAN, CC_PERFIL, CC_NOMCARR,
   CC_NOMPERFIL, TIPOCIC AS CC_TIPOPLAN, CC_CREDITOSMIN
from AX_MAPEO_CARRERAS, AX_CARRERAS, IN_CARRERAS
where AX_MAPEO_CARRERAS.CC_CODCARR =
   AX_CARRERAS.CC_CODCARR and
   AX_MAPEO_CARRERAS.CARR = IN_CARRERAS.CARR and
   AX_MAPEO_CARRERAS.CICLO = IN_CARRERAS.CICLO
```

Otros Datos Auxiliares

- Además de los procesos anteriores en esta etapa se procesan datos ingresados por el usuario para agregarlos al Data Warehouse. Como por ejemplo:
  - Mapeo de Materias
    - Debido a que en la base de datos de Bedelía no se maneja el concepto de materia.
    - Mapeo N:1.
  - Semestres de Asignaturas

Procedimientos Almacenados Relacionados

- Obviamente hay otro tipo de acciones que son tomadas en este momento, por ejemplo el calculo de los creditos minimos de una carrera.
- Para esto se encontro un procedimiento almacenado que hacia el procesamiento, el cual se muestra a continuacion.

Procedimientos Almacenados Relacionados (II)

```
create procedure ActualMATERIAS
as
    -- variables locales al procedimiento
    declare @CARR int
    declare @PLAN varchar(30)
    declare @PERFIL int
    declare @CODMAT varchar(6)
    declare @CNTCRED int

    declare Materias cursor for
        select C.CC_CODCARR, C.CC_PLAN, C.CC_PERFIL, MC.MA_CODMAT
        from BD_MAT_CARR MC, BD_CARRERAS C
        where upper(C.CC_TIPOPLAN) <> 'C' and C.CC_CODCARR = MC.CC_CODCARR
and C.CC_PLAN = MC.CC_PLAN  and C.CC_PERFIL = MC.CC_PERFIL

    OPEN Materias
    fetch next from Materias INTO @CARR, @PLAN, @PERFIL, @CODMAT
```

Procedimientos Almacenados Relacionados (III)

```
    WHILE (@@FETCH_STATUS = 0) BEGIN
        begin transaction
        set @CNTCRED = 0
        set @CNTCRED = (
                        select count(*) from BD_ASIGNATURAS A
                        where A.CC_CODCARR = @CARR and A.CC_PLAN = @PLAN and
A.CC_PERFIL = @PERFIL and A.MA_CODMAT = @CODMAT
                    )
        if (@CNTCRED is null) begin
                set @CNTCRED = 0
        end

        update BD_MAT_CARR set MA_CREDITOSMIN = @CNTCRED
        where CC_CODCARR = @CARR and CC_PLAN = @PLAN and CC_PERFIL = @PERFIL and
MA_CODMAT = @CODMAT

        commit transaction
        fetch next from Materias INTO @CARR, @PLAN, @PERFIL, @CODMAT
    END

    CLOSE Materias
    DEALLOCATE Materias
GO
```

Procedimientos Almacenados Relacionados (IV)

- La duda que me queda es porque esto no se realizo cuando se cargaba por primera vez la tabla, tal vez algo escape a mi vision pero no podria haberse hecho de esta forma:

```
select ......,
        (select count(*) from BD_ASIGNATURAS A
        where A.CC_CODCARR = @CARR and A.CC_PLAN = @PLAN and A.CC_PERFIL = @PERFIL and
A.MA_CODMAT = @CODMAT) as CREDITOSMIN
        ......
from ....
```

Cuando se cargaro la tabla BD_MAT_CARR, o como otra opcion colocando los datos en una vista para consultarlos en el momento de cargar BD_MAT_CARR??

P2.2 (DW – Inscripciones a Curso)

- Con este proceso se realiza la carga del Data Warehouse con datos de inscripciones a curso.
- Lo que se realiza a mi entender podría ser comparable con lo mostrado anteriormente, es decir armar consultas para reunir información y así tenerla disponible para el DW. Claramente estoy hablando muy superficialmente del proceso dado que no llegue a estudiarlo.

P2.3 (DW – Actividades)

- De la misma manera que antes este proceso me parece comparable con los anteriores (misma observacion).
- Un aspecto a tener en cuenta que pudo haber guiado la implementacion podria haber sido la performance obtenida al hacerlo con consultas.
  - Optimizacion de Consultas vs Programacion Estructurada, para resolver el mismo problema.

P2.4 (DW – Avance por Materia)

- Este procedimiento parece claramente mas complicado.
- No voy a mostrar el procedimiento almacenado que realiza esto (porque llevaria muchas transparencias y no aportaria a mi entender) pero conceptualmente creo se entiende lo que hace para calcular el avance.
- Sin embargo si mostrare algunas secciones que me parecen interesantes, mas por lo que se puede ver a simple vista que por el codigo mismo.

P2.4 (DW – Avance por Materia) (II)

```
IF (Upper(@CCTIPOPLAN) = 'C') BEGIN
    SET @TOT =(select case when SUM(AC_CREDITOS) is null then 0 else SUM(AC_CREDITOS)
    end
                from BD_ACTIVIDADES
                where   Upper(AC_APRUEBAAS) = 'S' and
                        Upper(AC_TIPORESULTADO) = 'APROBADO' and
                        ES_CI = @ESTCI and
                        MA_CODMAT = @MAT)
    END
ELSE BEGIN
    SET @TOT = (select COUNT(*) from BD_ACTIVIDADES
                where   Upper(AC_APRUEBAAS) = 'S' and
                        Upper(AC_TIPORESULTADO) = 'APROBADO' and
                        ES_CI = @ESTCI and
                        MA_CODMAT = @MAT)
    END
```

P2.4 (DW – Avance por Materia) (III)

```
IF (not (@CREDMIN is null)) BEGIN
    IF (@CREDMIN <> 0) BEGIN
        IF ( not (@TOT is null)) BEGIN
            SET @VALOR = ((@TOT * 100) / @CREDMIN)
            IF (@VALOR >=0 and @VALOR <= 10) BEGIN
                SET @AV = '0 a 10%'
            END
            ELSE BEGIN
                IF (@VALOR > 10 and @VALOR <= 70) BEGIN
                    SET @AV = '11 a 70%'
                END
                ELSE BEGIN
                    IF (@VALOR > 70 and @VALOR <= 99) BEGIN
                        SET @AV = '71 a 99%'
                    END
                    ELSE BEGIN
                        SET @AV = '100%'
                    END
                END
            END
        END
```

P2.4 (DW – Avance por Materia) (IV)

```
        ELSE BEGIN
            SET @VALOR = 0
            SET @AV = 'NO CALC. SUM NULL'
        END
    END
    ELSE BEGIN
        SET @VALOR = 0
        SET @AV = 'NO CALC. CRED 0'
    END
END
ELSE BEGIN
    SET @VALOR = 0
    SET @AV = 'NO CALC. CRED NULL'
END
insert into BD_AVANCE values
    (@ESTCI,@CCCODCARR,@CCPLAN,@CCPERFIL,@MAT,@FECHA_ACT,@VALOR,@AV)
```

P2.5 (DW – Desempeño por Carrera)

- Nuevamente lo mismo que en el caso anterior el procedimiento parece claramente complicado.
- Claramente es del mismo corte que el anterior.
- En ambos procedimientos me quedo la duda que no se podria hacer sin usar SQL (porque no lo pude mirar en detalle).

P3.1 (DW – Asignaturas, Materias, Materias de Carreras, Asignaturas de Carreras)

- Aquí por lo que pude entender se realiza un procesamiento similar al realizado en 2.1 (DW – Estudiantes, Carreras, Estudiantes insc. Carreras, Egresos).
- Para mi la razon de la separacion es que es necesario tener cargado lo anterior al procesar este paso.

Proceso de Carga (II)

- Casi finalizando:
  - El sistema mantiene el estado de la carga (utilizando una tabla por lo que pude deducir).
  - Permite borrar el contenido del Warehouse excepto la información para históricos.
  - No vi nada que controlara la recarga de datos (entiendo por esto volver a cargar los mismos datos), no se si seria importante tampoco.
  - Hay ciertos parámetros que son pedidos en tiempo de ejecución de la carga (p.e. Año de referencia para calcular el desempeño).

Sobre Log's

- Por actividad del proceso de carga (es decir por nodo que se represento en el grafo).
- Filtrado por fecha de ejecución.
- Código de error general reúne todas las posibles fallas.
- Información disponible (referente a DTS):
  - Nombre del paquete.
  - Nombre de la transformación.
  - Fecha.
  - Tabla Original.
  - Los demás datos no aparecían en el Manual, vi algunos en las tablas pero son los que uno imaginaria, código de error, etc.

Posibles lineas de Trabajo

- Completar la conceptualizacion de todo el trabajo.
- Tratar de encontrar versiones alternativas del proceso utilizando SQL puro de ser posible y marcando que cosas no son realizables.
- Tratar de expresar la transformacion mediante primitivas.
- No encontre vi nada que indicara cuanto tiempo lleva una carga. Seria interesante saber cuanto demora y si se pudieran sustituir partes por p.e. SQL cuanto se penzaliza o no el proceso.

Conclusiones

- Por la poca profundidad que se pudo alcanzar no se pudo detectar porque el proceso no podria ser exclusivamente SQL. Pero aparentemente esta puerta aun esta abierta.
- Cual es el aporte de DTS finalmente ??
  - Pareceria que el proceso es controlado por la aplicación en el sentido en que se marcan expresamente los procesos ya realizados y la interfase sugiere y guia la secuencia de ejecucion.

Conclusiones (II)

- El problema global no parece excesivamente complejo (mirandolo desde mi punto de vista).

Bibliografía

- [SIAEE] Sistema informático de apoyo a la evaluación de la enseñanza.
- [MANDWE]: Manual de Usuario, Data Warehouse de Enseñanza.