Titulación: GRADO INGENIERIA INFORMATICA
Año Académico: 2019/2020 – 2º cuatrimestre
Asignatura: Ficheros y bases de datos (2º curso)
Memoria de Prácticas 1: Diseño Relacional y Carga de Datos

uc3m | Universidad Carlos III de Madrid

Profesor:	FRANCISCO JAVIER CALLE GOMEZ	**Grupo**	81
Alumno/a:	JORGE RODRÍGUEZ FRAILE	**NIA:**	100405951
Alumno/a:	CARLOS RUBIO OLIVARES	**NIA:**	100405834

1. Introducción

El objetivo principal de este trabajo es crear un nuevo diseño de base de datos para traspasar todos los datos, para ello, nos hemos fijado pequeñas metas para ir avanzando poco a poco en nuestro proyecto. El primer paso será conseguir un diseño relacional óptimo, es decir, que podamos implementar el mayor porcentaje de nuestras ideas en SQL sin problema, y que las que no podamos, que no formen parte del 'esqueleto' de nuestro diseño. El siguiente paso consistirá en la creación de todas las relaciones en SQL, aquí podremos introducir diversos puntos que no hemos podido insertar en nuestro diseño como restricciones, es muy importante mantener la integridad del diseño creando en un orden razonable todas las tablas. Por último, pasaremos al volcado de datos a partir de la base antigua, en esta parte deberemos ser muy cuidadosos, ya que puede ser que algunos datos estén mal introducidos, o que incluso nuestro diseño no soporte algunos datos de la base antigua, esto requerirá diversas modificaciones.

2. Diseño Relacional

- Esquema relacional:

- Semántica implícita:

Sup_id	Mecanismo	Descripción
I_1	Tabla de validación	Los posibles valores de color son un grupo cerrado, Blanco y negro o Color, y no varían con el tiempo.
I_2	Tabla de validación	Las relaciones de aspecto son limitadas y estandarizadas por lo que no cambian.
I_3	Tabla de validación	Es un grupo reducido y no volátil de valores los tipos de calificación por edad.
I_4	Tabla de protagonistas con atributo y apellido actor	Para reconocer dentro del reparto de una película a un actor especifico necesitamos su nombre y apellido, lo que le identificara en el reparto junto los datos de la película.
I_5	Tabla con actores y sus likes	Los actores que protagonizan la película, los consideramos protagonistas de lo que queremos conocer los likes.
I_6	Atributo en membresía, media actividad	Cada miembro de un club tiene su propia media de actividad
I_7	Opinión tiene el atributo usuario.	Al almacenar una opinión va ligada al usuario que la dio.
I_8	En las invitación y solución el atributo aceptado como opcional	Un usuario envía una solicitud o un miembro envía una invitación, en esa solitud o invitación hay atributo que hasta que no sea respondida o es ignorada no tendrá valor, ese atributo indicara si ha sido aceptada o no.
I_9	Tabla de validación	Los posibles valores de idioma son un grupo cerrado y no muy volátil.
I_{10}	Diseño de las relaciones históricos	Los datos que almacenaremos de los cubes cerrados serán los propios datos del club y aquellos miembros que formaban parte del mismo.
I_{11}	Concepto de director	Como no se especifica, el atributo director vendrá definido por el apodo o apellido por el que se conozca a dicha persona, sin necesidad de crear una relación nueva.
I_{12}	Nuevo atributo aceptado en propuestas	Hemos supuesto que para ver si una propuesta de visualización está vigente o no mediante el atributo aceptado que es booleano; si es false, dicha propuesta todavía no se ha visualizado, y si es true, la película está vigente para visualizarla en el club.
I_{13}	Definición de reglas de integridad referencial	En este tipo de reglas de las cuales solo algunas se han sido definidas nos hemos tomado la libertad de etiquetar a cada regla de manera que mejor se adapte a nuestro diseño, por ejemplo, nunca borramos en cascada una película, pero si las tablas que se relacionan con ella.

Tabla 1: Semántica implícita

- Semántica explícita no contemplada en el diseño:

Sup_id	Descripción
S_1	La password tenga como mínimo 8 caracteres.
S_2	La edad del usuario, está directamente relacionado con la fecha de nacimiento, se contemplará más adelante.
S_3	Restringir que los usuarios que contraten servicios tengan que tener teléfono.
S_4	Vigilar la coherencia de las fechas.
S_5	No podemos controlar que cuando un usuario es invitado y acepta dicha invitación, pase a formar parte de club.
S_6	No nos es posible limitar que las solicitudes solo se puedan enviar en caso de que club este cerrado.
S_7	Que un usuario al crear un club pase a ser miembro del mismo.
S_8	La valoración tendrá valores entre 0 y 10.
S_9	Cuando un club se va a eliminar se mantendrán sus datos como registro histórico, pero sin actividad. No lo hemos podido añadir a nuestro diseño.
S_{10}	No podemos controlar que el color de la película sea solo Black & White o Color

Tabla 2: Semántica explícita no contemplada

3. Implementación de la Estática Relacional en SQL (LDD)

Semántica explícita re-incorporada:

Sup_id	Descripción de la solución
S_1	Solucionado el control de que una password sea mayor a 8 caracteres con CHECK (LENGTH(password) >8
S_2	Solucionado el problema de volcado de datos solo de fecha_nacimiento y usarlo para calcular edad mediante (SYSDATE-(TO_DATE(birthdate, 'YYYY-MM-DD')))/365.2422
S_3	Para solucionar esto, al insertar en la tabla CONTRATOS hemos puesto la restricción phoneN IS NOT NULL
S_4	Lo hemos resuelto mediante una comparación aritmética en el WHERE al insertar: (TO_DATE(enddate, 'YYYY-MM-DD') > TO_DATE(startdate, 'YYYY-MM-DD'))
S_5	No hemos podido solucionar este problema ya que se tendría que resolver con un ‘trigger’
S_6	Este problema lo hemos resuelto mediante un JOIN entre clubes y old_events para poder sacar el estado del club y que solo añada filas si coincide con ‘Open’
S_7	No se ha resuelto, de nuevo, se puede utilizar un ‘trigger’ para eliminar este problema
S_8	Este problema se ha resuelto con el siguiente código: (SUBSTR(Etype,0,9)='opinion' OR ltrim(Etype,'opinion:') BETWEEN '0' AND '9’

S_9	Esto lo hemos resuelto haciendo un JOIN para poder tomar
S_{10}	Al haber pocos tipos de valores, hemos creado esta restricción dentro de la propia creación de la tabla películas de la siguiente manera: CHECK (color IN ('Color','Black and White')

Tabla 3: Semántica explícita re-incorporada

Semántica implícita:

Sup_id	Mecanismo	Descripción
I_{13}	Tabla Visualizaciones	Hemos supuesto que la media_actividad de un usuario serán sus visualizaciones en un año, y para almacenarlas hemos creado esta tabla.
I_{14}	Fecha Completa	Para relaciones que requerían una fecha completa (como invitación o solicitud) hemos hecho una concatenación de ev_date con ev_hour.
I_{15}	Longitud de DNI	Hemos supuesto una longitud 9 para DNI
I_{16}	Longitud de duración	Hemos supuesto que la duración de las películas estará expresada en minutos, por tanto, la hemos introducido como un NUMBER(4).
I_{17}	TO_NUMBER	Para diversos atributos, como por ejemplo cantidad de likes en Facebook que estaban almacenados como un VARCHAR2 hemos utilizado un cast a un entero para optimizar el guardado de datos.
I_{18}	Cambio de Primary Key en opiniones	Una opinión vendrá identificada por el usuario que la hace a qué hora se realiza, y la película a la que se refiere. Suponemos que un usuario no puede hacer más de una opinión en una película.

Tabla 1(cont.): Semántica implícita

Semántica excluida:

Sup_id	Descripción semántica	Motivo	Explícita/ Implícita
E_1	No hemos podido establecer las relaciones de modificación entre tablas	Esta versión de SQL no respalda esta regla.	Implícita

Tabla 4: Semántica excluida en la creación de tablas

4. Carga de datos (LMD)

Las decisiones que hemos tomado en el volcado de datos y los problemas encontrados han sido:

- El orden que hemos elegido para volcar los datos es el mismo que el de creación, es decir, primero creamos las tuplas que reciben relaciones y después de las que sale la clave ajena, para así evitar errores de atributos y tuplas no creadas.
- Uno de los primeros problemas que nos hemos encontrado ha sido al volcar los datos en la tabla 'USUARIOS', donde no controlábamos bien que las contraseñas fueran de una longitud mayor a 8 caracteres, que arreglamos con un WHERE (LENGTH (passw) > 8)
- Otra de las decisiones que hemos tomado es a la hora de almacenar el atributo edad en la tabla 'PERFILES', usar fecha de nacimiento para calcular la edad y así no tener que leer toda la base para almacenar 2 atributos que nos otorgan la misma información.
- Para almacenar todos los géneros, hemos ido volcando todos los datos de las columnas genre1-5, con un SELECT DISTINCT para no repetir datos, y un WHERE (genreX NOT IN (SELECT tipo FROM GENEROS) para evitar que se repitan los mismos géneros al hacer consultas en columnas que pueden tener los mismos datos almacenados.
- En la tabla 'IMBDS' hemos cambiado imbd_score de string a numero mediante un TO_NUMBER, al igual que con num_user_for_reviews y num_critic_for_reviews.
- En la tabla PALABRAS_PELICULA hemos hecho palabras clave principal junto con película_titulo y película_director
- En PROTAGONISTAS hemos eliminado apellido y hemos combinado todo el nombre en un solo atributo.
- En CLUBES se ha cambiado el formato de abierto a VARCHAR2(100) y se han eliminado cod_postal, ciudad y país ya que no se contemplaban en el diseño relacional, por otro lado, hemos añadido el atributo slogan, ya que el volcado de datos lo requería. En CLUBES_HISTORICOS hemos hecho las mismas modificaciones, solo que sin añadir slogan.
- En INVITACIONES hemos eliminado aceptado, ya que no podemos usarlo a la hora de pasar los datos.
- Respecto a SOLICITUDES hemos eliminado el atributo aceptado, al igual que en INVITACIONES, pero hemos añadido frase_aceptacion, para hacerlo coincidir con el formato de datos antiguo.
- En diversas tablas había datos mal introducidos, como, por ejemplo, géneros que no existían, películas y usuarios no guardadas en la base de datos, para ello hemos usado la expresión (NOT) IN, y hacer coherente el volcado de código. Las tablas donde hemos encontrado problemas han sido: GENEROS, PELICULAS, PALABRAS_PELICULA, GENEROS_PELICULA, PROTAGONISTA, MEMBRESIA, PROPUESTAS, INVITACIONES, SOLICITUDES, CREACIONES _CLUB, CLUBES_HISTORICOS, VISUALIZACIONES Y OPINIONES.
- En PELICULAS, hemos hecho cast de diversos atributos como duración o numero de likes de Facebook para que se ajustara mejor al diseño de la tabla.
- En PROPUESTAS, hemos supuesto que el asunto y el mensaje sean obligatorios, por la integridad del volcado de datos.

- En CLUBES HISTORICOS hemos hecho un INNER JOIN para poder sacar fecha de creación y slogan de clubes.
- En VISUALIZACIONES hemos realizado otro JOIN para poder sacar el usuario y a que club pertenece la persona que ha realizado los viewings.
- No hemos usado el atributo num_voted_users de la tabla old_movies ya que hemos usado la cantidad de votaciones de usuarios y críticos, cuya suma es la del atributo en cuestión, lo que resultaría en guardar el mismo tipo de información dos veces.
- Se había planeado crear una tabla REGISTROS con todos los eventos de tipo 'registration' pero no se nos es posible porque este evento no nos proporciona la contraseña de un usuario, que es clave para crearlo.
- En cuanto a añadir eventos de aceptados o rechazados, no podemos tratar este problema ya que se usaría un 'trigger' para hacer que las invitaciones desaparecieran ocurrir este tipo de evento, por tanto, no usamos estos registros de datos.