

Práctica 2 BD

Parte 1: Creación de una base de datos

A partir del siguiente enunciado, se creará una base de datos en Oracle que será utilizada en el resto de sesiones.

Enunciado

Te encargan diseñar una base de datos para una web sobre cine. Habrá que almacenar las películas, con su título, año de estreno, género/s, así como la lista de actores, director/es, y demás personal implicado en su realización, destacando el papel de cada uno en la película y, para los actores, una descripción del personaje que han interpretado. También desean guardar información similar sobre series de TV, incluyendo el periodo de emisión y los distintos capítulos de cada temporada. Es destacar que algunas de las películas son *remakes*¹, secuelas², o *precuelas*³ de otras películas.

Se pide:

- 1) **Diseñar el esquema E/R de la base de datos**, especificando las restricciones que existan de la manera más precisa posible. Justificar aquellas partes que se consideren discutibles. Se deberán incluir soluciones alternativas a la elegida en aquellas partes del esquema que sean discutibles.
- 2) **Traducir el esquema E/R al modelo relacional**, explicando cualquier decisión que se haya tomado en este paso. Normalizar el esquema indicando las comprobaciones realizadas, esté o no normalizado. Finalmente, traducir el esquema relacional ya normalizado a tablas SQL.

Memoria de la parte 1

- Esquema E/R global, las decisiones tomadas más importantes, y las restricciones necesarias.
- Descripción breve de las relaciones que se han creado y las decisiones tomadas durante dicho proceso, la normalización de las mismas, y el esquema relacional final, incluyendo el fichero SQL correspondiente. Comentar cualquier otra tarea realizada al crear la BD en SQL para Oracle.
- Datos sobre reuniones, división del trabajo, problemas, etc. relativos a la coordinación del grupo.

¹ Nuevas versiones sobre más o menos la misma historia y personajes, ej. “Batman” (1989) y “Batman Begins” (2005). No son secuelas ni precuelas, sino una nueva manera de contar la misma historia.

² Películas desarrolladas en el mismo contexto que otras previas pero en un periodo posterior. Ej. las secuelas de “La Guerra de las Galaxias” (1977) son “El Imperio Contraataca” (1980), y “El Retorno del Jedi” (1983) (ambas continúan la historia de la primera); la tercera también es secuela de la segunda.

³ Películas desarrolladas en el mismo contexto que otras previas pero en un periodo anterior. Ej. Las precuelas de “La Guerra de las Galaxias” (1977) son “La Amenaza Fantasma” (1999), “El Ataque de los Clones” (2002), y “La venganza de los Sith” (2005) (realizadas posteriormente pero se desarrollan en un tiempo anterior), que a su vez son secuelas cada una de la siguiente. “Precuela” no es un término reconocido por la RAE.

Parte 2: Optimización: Introducción de datos y ejecución de preguntas sobre una base de datos

Se cargarán en la BD Oracle los datos necesarios para poder obtener respuestas más o menos reales al ejecutar las preguntas indicadas. También se practicará el borrado y actualización de tuplas. Los datos a introducir en la base de datos diseñada se pueden extraer de una BD MySQL que puede ser accedida desde vuestra cuenta en Hendrix:

```
$ mysql -h hendrix-mysql -u <usuario> -p miniIMDB
```

donde usuario es el mismo usuario que utilizáis con Oracle. Os pedirá un password que podéis encontrar en el fichero `README.mysql` en Hendrix. Podéis listar las tablas de la base de datos `miniIMDB` con el comando `"show tables;"`. El comando `"describe <tabla>;"` muestra la estructura de la tabla indicada.

NOTA: no es necesario cargar *todos* los datos suministrados sino sólo aquellos necesarios para que la base de datos cumpla con lo se pide en el enunciado de la parte 1 y puedan responderse la consultas SQL que se describen más adelante. Si hicieran falta datos que no aparecen en dicho fichero, comenta el problema con los profesores de prácticas. El formato en que están organizados los datos tampoco debería influenciar en vuestra base de datos, extraedlos y adaptadlos a vuestra base de datos, no al revés.

Cada grupo es libre de elegir la mejor estrategia para introducir la información que necesite para poblar su base de datos. En moodle se proporciona un ejemplo que muestra cómo se introducirían los datos de forma automatizada desde MySQL a Oracle mediante un programa Java que incluye el driver JDBC necesario para utilizar MySQL.

Se pide:

1) **Realizar las siguientes consultas SQL:**

- i. Dado el nombre de cualquier población del mundo, busca las películas relacionadas con esa población, ya sea en su título, por el lugar de nacimiento de alguna de las personas que participan en esa película o que se comente algo de esa ciudad en los keywords de la película. La respuesta incluirá el título de la película, el año, y la causa por la que se incluye en esa lista. Deben estar ordenadas de más modernas a más antiguas.
- ii. Muestra las 3 personas con más cargos en una película o serie
- iii. Directores que empezaron como actores/actrices.

2) **Realizar otras tres consultas (no triviales)** que tengan sentido práctico para al menos algún tipo de usuario de la base de datos. Verificad con los profesores de prácticas que las consultas que habéis pensando son de la complejidad adecuada.

Memoria de la parte 2

- Resumir los pasos seguidos para poblar la BD con datos, destacando los principales problemas encontrados y las decisiones tomadas. Mostrar datos estadísticos sobre cada una de las tablas (número de tuplas, tamaño *aproximado* en Kbytes) y listar de forma tabular 5 tuplas cualesquiera de las cargadas en cada tabla.
- Para cada una de las seis consultas, explicar cómo se ha construido en SQL, incluyendo un árbol sintáctico que la describa en álgebra relacional, posibles alternativas, y la respuesta obtenida. Incluir un fichero con cada consulta SQL.
- Datos sobre reuniones, división del trabajo, problemas, coordinación del grupo, etc..
-

Parte 3: Optimización, Diseño Físico, y Triggers

En esta práctica se van a revisar aspectos diversos como la optimización de preguntas, diseño físico, y creación de *triggers* en la BD Oracle creada anteriormente.

Se pide:

- 1) **Diseño físico y mejora de eficiencia.** Considerando las tres consultas obligatorias planteadas en la Parte 2, y con el objetivo de mejorar su rendimiento, realizar el diseño físico necesario a la BD (creación de índices, desnormalización, JOINS pre calculados, etc.). Oracle permite obtener informes y estadísticas sobre la ejecución de sentencias SQL, lo cual te puede ayudar a comprobar cómo has mejorado la eficiencia de las mismas con tus decisiones a nivel de diseño físico. Para ello hay que crear una tabla que Oracle utilizará para crear sus informes. Descarga el fichero `utlxplan.sql` que se encuentra localizado en moodle y sigue los siguientes pasos:
 - i. Crear la tabla `PLAN_TABLE` (ejecuta *Utlxplan.sql* en *sqlplus*)
 - ii. Ejecutar en *sqlplus* “set autotrace on”
 - iii. Ejecutar la pregunta SQL que desees y analizar el plan de ejecución y estadísticas mostradas después de la respuesta.
 - iv. Repetir el paso iii. para las preguntas que desees.
- 2) **Crear tres *triggers* que:**
 - i. Cumplan tareas de mantenimiento de la consistencia (al menos uno de ellos deberá ser de este tipo)
 - ii. Faciliten tareas no triviales.
 - iii. Tengan sentido práctico para los usuarios de la base de datos desarrollada.

Memoria de la parte 3

- Para cada una de las tres consultas SQL obligatorias desarrolladas en la Parte 2:
 - Presentar y explicar con tus propias palabras, los resultados de los informes obtenidos con *autotrace* para dichas consultas, antes del diseño físico.
 - Explicar y justificar las suposiciones o hipótesis en las que se haya basado la toma de decisiones del diseño físico, y las diferentes acciones tomadas en dicha fase para mejorar el rendimiento. Incluir el código SQL correspondiente a esas acciones.
 - Una vez acabado el diseño físico, presentar y justificar el nuevo informe obtenido con *autotrace*, destacando la mejora obtenida con tus decisiones de diseño físico.
- Para cada *trigger* desarrollado, justificar su utilidad, explicar su código y funcionalidad, principales problemas encontrados, y comentar las alternativas si es que existen. Incluir el código SQL de cada *trigger* y una traza de su funcionamiento desde SQL Plus.
- Datos sobre reuniones, división del trabajo, problemas, etc. relativos a la coordinación del grupo.