



TAREA EVALUABLE 2

Bases de Datos
CFGs DAW



Pau Miñana Climent
2020/2021

PARTE I (8 puntos)

Para la siguiente actividad evaluable vamos a recuperar la base de datos diseñada en la anterior actividad. Debido a la naturaleza del diseño de la misma es más sencillo usar Mysql, pues Oracle usa el mismo campo fecha para almacenar tanto fechas como horas, así que el diseño no acaba de ser adecuado para la misma y según está requiere de algún trabajo extra para realizar/mostrar correctamente las consultas, de forma similar a como se comenta en la consulta 12. De todos modos quien quiera puede realizar la actividad con Oracle, se puede hacer y resulta además un estudio interesante que recomiendo, aunque eso sí, yo os lo recomiendo ya fuera de la actividad evaluable una vez realizada, pero es vuestra decisión.

Tenéis disponible un script para crear la base de datos que se puede usar tanto en Oracle como en mysql, pero falta quitar los comentarios en las partes adecuadas. El primer trabajo que tenéis es analizar el script y dejarlo adecuado para poder ejecutarlo con el SGBD escogido. Podéis ver el esquema de la BD en la imagen correspondiente

Los datos para rellenar las tablas se proporcionan en una hoja de cálculo. Conviene crear los archivos .csv correspondientes a cada tabla encriptados en utf8 antes de importarlos.

Para la importación depende del SGBD elegido:

- En Oracle se recomienda importar cada csv a su tabla correspondiente desde SQLDeveloper y activar posteriormente la restricción.
- En MySQL se pueden usar las últimas líneas del script para realizar la importación, pero hay que tener en cuenta que no se permite cualquier ruta para los archivos. La ruta disponible para importar archivos la podéis encontrar ejecutando

```
SHOW VARIABLES LIKE "secure_file_priv";
```

Si está en blanco entonces podéis añadir la línea:

```
secure_file_priv="ruta_que_usarás"
```

Al archivo my.config y reiniciar la base de datos. Posteriormente debéis especificar esa ruta en el script.

Puesto que esta parte es necesaria para realizar la actividad queda fuera de la parte evaluable y podéis pedir ayuda al profesor si la necesitáis.

Una vez realizado el proceso podéis comenzar con las siguientes consultas, no es necesario incluir capturas de las mismas, sólo el texto necesario para realizar cada una de las mismas. Valen 0,5 puntos cada una independientemente de su complejidad.

1. Listado de profesores con nombre, apellido1 y el departamento al que pertenecen.
(13 rows)

2. Listado de reservas de aulas (con el **nombre** del profesor que hace la reserva) ordenados por aula. (31 rows)
3. Listado de profesores y los módulos que imparten, ambos con los nombres. (29 rows)
4. Nombre de los módulos que imparte el profesor con dni 67538464E. (3 rows)
5. Nombre completo en un único campo (se puede usar la función concat(campo1,campo2..) para unir campos varchar, también acepta texto directamente entre comillas) y dni de ALUMNOS que empiezan con la letra S. (2 rows)
6. Id, fecha y alumno de las tutorías pedidas en el mes de diciembre. (80 rows).
7. Lista de Id de los horarios de tutoría de los que se ha reservado alguna sesión. (84 rows)
8. Id, día hora y dni del profesor de los horarios de tutoría que no se han pedido todavía en ninguna fecha. (3 rows)
9. Nombre de los módulos con el total de alumnos que tienen matriculados. (24 rows)
10. Número de alumnos del módulo con más alumnos.
11. Nombre y número de alumnos del módulo con menos alumnos.
12. Listado de reserva de aulas que coincide con una tutoría de ese profesor. Es decir datos de la reserva de aula que coincida en fecha y hora con una tutoría reservada para ese mismo profesor. Para poder comparar fechas y horas con la fecha+hora completa de reserva_aula se pueden usar funciones como DATE(fecha) o TIME(fecha) que extraen sólo esa parte de la información. (1 row)
13. Nombre y apellidos del alumno que ha pedido más tutorías y cuántas son.
14. Puesto que no se han implementado las restricciones para que los profesores no puedan matricularse en módulos impartidos por alguien de su departamento, crear una vista que muestre el dni del profesor y el código del módulo que incumplen esta restricción (1 coincidencia, pero es probable que salgan 2 rows, una por cada profesor del módulo).
15. Consulta para borrar la fila referida en la consulta anterior. Los datos del profesor y la columna deben introducirse directamente en la propia consulta no puede crearse una subconsulta de la vista anterior, puesto que las consultas DELETE no permiten incluir a la propia tabla en las condiciones del where. En caso de no haber resuelto el punto anterior simplemente elige un profesor+alumno y un módulo de los que está matriculado y borra esa fila.
16. Actualizar la capacidad máxima de todas las aulas bajándola en 2, excepto el laboratorio y el taller. (A15 y B15)

PARTE II (2 puntos)

Como habréis observado, la parte de reserva de libros ha quedado fuera de esta actividad. Debéis añadir al script las instrucciones necesarias para implementar esa parte según el siguiente modelo:

Libros (ISBN, Nombre, Autor, Editorial)

CP: {ISBN}

Ejemplares (ISBN, N_Copia, Estado)

CP: {ISBN, N_Copia}

CAj: {ISBN} → Libros

Lineas_Reserva (ID_Reserva, N_Linea, Fecha_Dev, ISBN, N_Copia)

CP: {ID_Reserva, N_Linea}

CAj: {ID_Reserva} → Reserva_Libros

CAj: {ISBN, N_Copia} → Ejemplares

Reserva_Libros (ID_Reserva, Fecha_Reserva, Hora, Fecha_Recogida, DNI)

CP: {ID_Reserva}

CAj: {DNI} → Usuarios

UNIQUE {Fecha_Reserva, Hora,DNI}

Además el script debe añadir datos correspondientes a 3 libros, 10 ejemplares y 2 reservas, una de una línea y otra de 3 líneas. Los datos los podéis inventar libremente siempre que cumplan las restricciones y la coherencia de la base de datos.

Las modificaciones para adecuar el script a las condiciones exigidas y su explicación en caso de ser necesaria se valora con 2 puntos.