

PRACTICA

BASES DE DATOS

Curso 2018-19

Grado en Matemáticas e Informática

Creación de una bases de datos desde un programa Java y realización de consultas

Objetivos Generales

- Crear una base de datos, tablas y restricciones siguiendo un modelo relacional proporcionado.
- Crear los procesos de carga de datos mediante un programa desarrollado en Java.
- Implementación del acceso a la base de datos creada para la realización de consultas SQL mediante un programa Java, mediante el uso del conector JDBC.

El alumno debe conectarse a un servidor de bases de datos MySQL por lo que se recomienda tener instalado el sistema gestor de bases de datos MySQL (parte servidora) en su ordenador.

Se recomienda al alumno la utilización de un cliente SQL como por ejemplo MySQL Workbench, HeidiSQL o SequelPro, para poder probar de una manera más cómoda las consultas SQL que se deben crear.

El código java a desarrollar deberá funcionar con el conector JDBC de MySQL que se puede encontrar en Moodle:

(<https://moodle.upm.es/titulaciones/oficiales/mod/resource/view.php?id=962881>).

Objetivos específicos y descripción del trabajo a realizar

El alumno debe ser capaz de:

- Realizar conexiones a un SGBD desde un programa Java.
- Realizar consultas simples y complejas en SQL, manejando las clases adecuadas del conector JDBC.

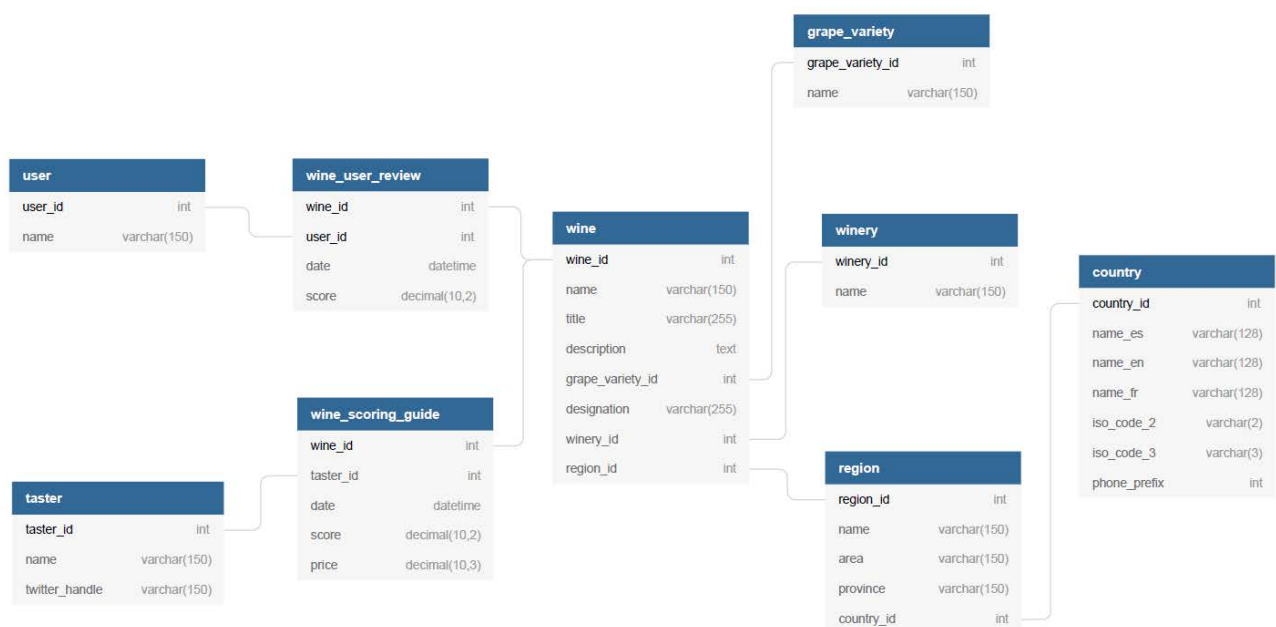
- Manejar correctamente los resultados de las consultas SQL dentro del código Java.
- Definir la gestión de las posibles excepciones que puedan resultar de la gestión de consultas.
- Implementar un sistema basado en el concepto de transacciones usando código Java.

Tareas concretas:

Se desea realizar un programa que administre la información sobre valoraciones de la cata de distintos vinos. Las funcionalidades que se requieren implementar en el programa son:

- 1) Creación de la base de datos basada en el modelo relacional mostrado a continuación.

Se debe entregar un fichero SQL conteniendo todas las consultas (script) de creación de las tablas, así como la creación de las claves primarias y de las restricciones de clave foránea. **[0,5 puntos]**



- 2) Carga de los datos suministrados en formato de sentencias SQL INSERT de algunas tablas del modelo. **[0,5 puntos]**

Se debe cargar su contenido usando el cliente de SQL que el alumno considere oportuno. Se debe entregar el fichero SQL conteniendo las consultas (script) de inserción de datos en las tablas en el orden usado por el alumno.

- 3) Creación de una clase Java que maneje las conexiones de con la base de datos de manera centralizada para todos los accesos a la misma desde la práctica **[0,5 puntos]**.

Se valorará la modularidad, eficiencia, limpieza del código y gestión de posibles errores, además de la buena utilización de los recursos del ordenador.

4) Capacidad cargar datos desde código Java a través de JDBC **[1 punto en total]**:

- i. Carga de los datos suministrados en un fichero con formato CSV, que contiene datos de la tabla *wine_scoring_guide* [0,5 puntos].
- ii. Carga de los datos suministrados en un fichero con formato CSV, que contiene datos de la tabla *wine_user_review* [0,5 puntos].

Para ello el alumno deberá implementar un método que realice la carga de los datos de cada una de las tablas desde el fichero CSV, generando sentencias SQL INSERT a través del conector JDBC de Java.

Se valorará la modularidad, eficiencia, limpieza del código y gestión de posibles errores, además de la buena utilización de los recursos del ordenador.

5) Realización de consultas a través de JDBC **[2,5 punto en total]**:

- i. Consulta que devuelva el **número de vinos existentes en la base de datos** [0,5 puntos]
- ii. Consulta que permita **buscar un vino por nombre o descripción** dada una cadena de texto [0,75 puntos]
- iii. **Listado de valoraciones de usuarios de un para un vino dado su identificador** (numérico) [0,75 puntos]
- iv. **Número de valoraciones, y valoración media** de las de usuarios; además de la **valoración profesional para un vino dado su identificador** (numérico) [0,5 puntos]

Cada una de las consultas pedidas deberá estar implementada en un método distinto, que reciba los parámetros necesarios, y devuelva como resultados las Un string formateado, conteniendo la tabla resultado de la consulta (incluyendo cabeceras).

Se valorará la modularidad, eficiencia, limpieza del código y gestión de posibles errores, además de la buena utilización de los recursos del ordenador.

6) Realización de consultas a través de JDBC, usando un pseudo-ORM **[1,5 punto en total]**:

- i. Obtener vino por id (numérico), devolverá un objeto tipo vino (clase Wine), o null en caso de no existir el vino en la base de datos. [0,5 puntos]
- ii. Obtener vinos por nombre o descripción (String), devolverá una lista de objetos tipo vino (clase Wine) [0,5 puntos]
- iii. Obtener la lista de usuarios (clase User) que hayan realizado valoraciones de un vino, el método recibirá como parámetro el identificador numérico del vino. [0,5 puntos]

La creación de un ORM (Object-Relational-Model mapping) implica la creación de las clases necesarias en Java que representen las entidades presentes en la base de datos. Cada una de estas clases Java deberá incluir, aparte del constructor, los métodos getters y setters de sus atributos, y el toString [0,5 puntos]

Cada una de las operaciones a ser implementadas en este apartado deberá estar implementada en un método distinto, y deberá recibir los parámetros necesarios, y devolver como resultados las entidades del ORM que se necesiten.

Se valorará la modularidad, eficiencia, limpieza del código y gestión de posibles errores, además de la buena utilización de los recursos del ordenador.

7) Inserción de datos desde Java **[1 puntos]**. En este punto el alumno deberá implementar el código necesario para lograr introducir usando **en una sola transacción por cada instancia de la tabla vino**, los datos suministrados en el fichero *nuevas_catas.csv*, de manera que la información sea insertada en las tablas correspondientes de forma correcta.

El fichero *nuevas_catas.csv*, contiene información de nuevas catas de vino realizadas por un experto que deben ser insertadas (o actualizadas) en el sistema. Por cada línea que aparece en el fichero aparece la información de una cata y valoración de un vino, conteniendo los siguientes datos:

title; variety; winery; country; description; designation; points; price; province; region; area; taster_name; taster_twitter_handle

Se deberán comprobar los distintos campos obteniendo los identificadores de los datos asociados a la valoración (de las tablas *winery*, *region*, *country*, *taster*, ...). Posteriormente se deberán construir las sentencias SQL para la inserción de datos que sean necesarias para poder introducir los datos que aparecen en el fichero en tablas *wine* y *wine_scoring_guide*. Para ello se debe comprobar usando el campo *title* si el vino y/o su valoración existen ya en la base de dato. En caso de que los datos del vino o la valoración ya existan se deberán actualizar los datos y puntuación del vino (usando el identificador que posea). En el caso de no existir se deberá crear una entrada nueva para el vino y la puntuación dejando que la base de datos le asigne un nuevo identificador (*wine_id*).

- 8) Obtención de consultas estadísticas de la base de datos y de su contenido. **[2 puntos]**:
- Obtener la **lista de usuarios**, incluyendo su nombre, **que no han realizado ninguna valoración** aún en el sistema. [0,5 puntos]
 - Obtener el usuario o **listado de los usuarios que más valoraciones han realizado durante el año 2018**. [0,5 puntos].
 - Obtener el **listado de marcas de vino (winery)** con su nombre, de aquellas marcas **que comercializan vino de más de un país**. Incluir en el resultado la lista de regiones (separadas por coma) a las que pertenecen los vinos, y el número de países distintos en los que trabaja la marca [0,5 puntos].
 - Obtener el **listado de los usuarios cuyas valoraciones son las más aproximadas a las de los catadores oficiales** (desviación media menor), incluyendo el nombre de usuario, la desviación media en sus valoraciones con respecto a las profesionales, y el número de vinos de los que ha realizado valoraciones. [0,5 puntos].

Cada una de las consultas pedidas deberá estar implementada en un método distinto, que reciba los parámetros necesarios, y devuelva como resultados las Un string formateado, conteniendo la tabla resultado de la consulta (incluyendo cabeceras).

Se valorará la modularidad, eficiencia, limpieza del código y gestión de posibles errores, además de la buena utilización de los recursos del ordenador.

9) Se quiere desarrollar un programa principal ejecutable (*main*), que ofrezca por medio de un menú textual de manera interactiva al usuario la posibilidad de ejecutar todas las funcionalidades implementadas en la práctica (enunciadas en los puntos 5, 6, 7 y 8). **[0,5 puntos]**

Entrega de la Práctica

La práctica se puede realizar en grupos de dos alumnos o de manera individual. Cada grupo de prácticas deberá entregar mediante la plataforma Moodle, en una tarea habilitada para ello, un único fichero comprimido (*.zip, *.rar) cuyo nombre sea:

(Grupo_N_JDBC.rar)

Siendo N el número asignado al grupo de prácticas. Dicho fichero deberá contener tanto los scripts SQL de creación y carga de la base de datos pedidos en los puntos 1 y 2, así como los ficheros de código fuente implementados en Java que implementan el resto de los apartados de la práctica. Del proyecto Java desarrollado se debe entregar un fichero comprimido conteniendo la carpeta src del proyecto de eclipse.

La entrega debe ser realizada por SOLO uno de los miembros del grupo.

La fecha límite para la entrega de esta práctica es el viernes día 12 de Abril de 2019 antes de las 15:00.

Sistema anti-copias

Se utilizará un sistema de detección de copias sobre el código y las consultas SQL generadas.