

Práctica: modelo E-R y SQL para para gestionar un animalario

Una empresa que se dedica a la cría de ratones transgénicos para su posterior venta a laboratorios necesita una aplicación de escritorio que les ayude a recoger información relativa a distintas poblaciones de ratones que cría en sus instalaciones. La aplicación debe permitir llevar un registro de los ratones y las características de los ratones que forman parte de una población, población que finalmente será vendida como un todo a un laboratorio.

Para cada población de ratones, inicialmente el científico indicará: un nombre para la población, el nombre de la persona de la compañía responsable de dicha población y un número de días durante los cuales la población estará en las instalaciones de la empresa procreando. Ese número de días siempre será inferior a 270 días. Para cada ratón que forma parte de la población deberá poder indicarse un código de referencia (un número entero que no será introducido por el usuario sino asignado por el programa), su fecha de nacimiento (una fecha), su peso (en gramos, un número entero), su sexo (sólo podrá tomar valores, es decir, un Enum "Macho" y "Hembra"), su temperatura corporal en grados centígrados (un número real), y un campo de texto describiendo el ratón que puede contener más de una línea de texto. Cada ratón contiene además el par 23 de cromosomas (dos cromosomas) que varían en función del sexo. En el caso de los machos contienen un cromosoma XY, y en el caso de las hembras contienen un par de cromosomas XX. Además, para cada ratón habrá que describir posibles mutaciones que afectan a sus cromosomas X e Y. El cromosoma X podrá ser normal, o contener una mutación que hace que el portador sea estéril (en el caso de los machos, si su cromosoma X tiene la mutación el macho será estéril; en el caso de las hembras ambos cromosomas X deberán contener la mutación). El cromosoma Y podrá ser normal, o podrá contener una mutación que hace al ratón macho propenso a la poligamia (emparejarse con múltiples hembras. Todos los datos que se pidan al usuario deberán ser validados (excepciones), y contener conjuntos de tests en el main de cada módulo. Cada población de ratones se almacenará en un RDBMS de MySQL.

Existe una aplicación que permite manejar las siguientes opciones a través de un menú principal:

1. Abrir un archivo que contenga una población de ratones (por defecto o pidiendo el nombre al usuario). La estructura del fichero se indica mediante un fichero de ejemplo.
2. Crear una población virtual de ratones a partir del tamaño de la población, el porcentaje de machos, hembras y mutaciones.
3. Crear una nueva población de ratones.
4. Listar los códigos de referencia de todos los ratones de una población.
5. Añadir un nuevo ratón a una población ya existente, indicando todos sus datos.
6. Añadir un nuevo ratón a una población ya existente indicando su nombre y asignando mediante funciones aleatorias el sexo, el peso entre 50 y 100 gramos,

la temperatura entre 36 y 38 grados, y las mutaciones en sus cromosomas con un porcentaje del % de posibilidades de contener mutaciones en cualquier de sus cromosomas definidas por un parámetro de la función con valor por defecto de 20%.

7. Eliminar un ratón de una población indicando su código de referencia. Al seleccionar esta opción se listan todos los ratones por pantalla.
8. Modificar los datos de un ratón, indicando su código de referencia.
9. Ver información detallada de un ratón, habiendo especificado previamente su código de referencia.
10. Los ratones pueden formar familias. Los ratones sin mutación en su gen Y sólo pueden ser parte de familias monogámicas. Para formar familias
 - a. Se selecciona un ratón macho de la población
 - i. Si no es un ratón con gen de poligamia, se selecciona un ratón hembra de la población y se crea una familia “normal” de ratones. Ambos ratones son retirados de la población inicial.
 - b. Si es un ratón con el gen de poligamia, se selecciona un ratón hembra de la población y se genera un número aleatorio entre 0 y 9.
 - i. Si el número es < 5 , se termina la formación de la familia y se crea una familia "Poligamica" con el ratón macho y la ratona/las ratonas hembras, aunque esto permita la creación de familias “Poligámicas” de una única pareja macho-hembra.
 - ii. Si el número es ≥ 5 , seleccionamos un ratón hembra de la población y la añadimos a la actual familia, y volvemos al punto 2.b.

El proceso de formación de familias termina cuando se agoten los ratones macho o los ratones hembra. Los ratones macho o hembra no emparejados no forman parte de los ratones que forman familias.

11. Una vez formadas las familias de ratones se comienza la simulación de la reproducción de estas familias. En la naturaleza, la fecundidad de los ratones depende de la disponibilidad de comida. En nuestro caso, los ratones siempre tendrán disponible toda la comida que quieran, y su fertilidad será elevada y constante a lo largo de todo el período de procreación. La reproducción se realiza del siguiente modo. Las familias pueden procrear y crear nuevos ratones con fecha de nacimiento al día donde procrean y con valores aleatorios para su nombre y asignando mediante funciones aleatorias el sexo, el peso entre 50 y 100 gramos, la temperatura entre 36 y 38 grados, y las mutaciones en sus cromosomas con un porcentaje del % de posibilidades de contener mutaciones en cualquier de sus cromosomas definidas por un parámetro de la función con valor por defecto de 20%. Para cada familia de ratones
 - a. Si la familia está compuesta por un único ratón macho y un único ratón hembra y ninguno es estéril.
 - i. Se genera un numero aleatorio N entre 0 y 99; el número de crías de la camada será:

Número aleatorio	Número de crías
$N < 5$	2
$5 \leq N < 15$	3
$15 \leq N < 30$	4
$30 \leq N < 50$	5

50 <= N <70	6
70 <= N <85	7
85 <= N <95	8
95 <= N <=99	9

- b. Si la familia está compuesta por un ratón macho no estéril y 2 o más hembras, si bien la fertilidad de las hembras no se ve alterada, la fertilidad del macho relativa al número de hembras decrementa. Por ello, para cada una de las hembras.

- i. Se genera un numero aleatorio N entre 0 y 99; el número de crías de la camada de dicha hembra será:

Número aleatorio	Número de crías
N <10	2
10 <= N <25	3
25 <= N <45	4
45 <= N <60	5
60 <= N <75	6
75 <= N <95	7
95 <= N <=99	8

- c. Si el macho de la familia es estéril, hay una cierta probabilidad de que la hembra quede embarazada por otro ratón macho y salga adelante la camada. Por ello, para simular la posible descendencia:

- i. Se genera un numero aleatorio N entre 0 y 99; el número de crías de la camada será:

Número aleatorio	Número de crías
N <15	2
15 <= N <35	3
35 <= N <70	4
70 <= N <90	5
90 <= N <=99	6

- d. Si la hembra de la familia es estéril, la familia producirá 0 crías.

12. Guardar (se supone que para usar esta opción previamente hemos abierto un archivo).
13. Guardar como (es decir, crear una copia diferente de la población de ratones).

Cuando el usuario selecciona la opción 7, 8 o 9, se le pide al usuario el código de referencia del ratón tras mostrarse todos los códigos de referencia de los ratones por pantalla.

Sin embargo, los responsables de venta de la aplicación han detectado varios problemas a la hora de trabajar con ella. Si existe algún problema en la aplicación o el equipo dónde se está ejecutando, la información desde la última vez que se ejecutó la opción de guardar no se puede recuperar.

Además, los ficheros donde se guarda la información se pueden corromper en caso de que el fichero estuviese abierto por la aplicación.

No se pueden compartir datos del mismo experimento desde diferentes aplicaciones. Esto es un problema puesto que multitud de clientes tienen terminales personales y no pueden acceder a los experimentos compartidos.

Por suerte, se han informado de que existe una tecnología que les permite administrar la información de manera consistente, que está disponible desde un modelo distribuido (se puede acceder desde diferentes aplicaciones al mismo tiempo) y evita errores de consistencia cuando dos trabajadores realizan acciones sobre una misma población al mismo tiempo.

I. Entrega I

Debido a ello, se requiere que:

- 1- Se cree un modelo E-R que represente la información almacenada en los ficheros en tercera forma normal para evitar duplicidades.
- 2- Se poble dicho modelo E-R con las poblaciones que aparecen en los ficheros Excel que la empresa ya tenía. Se deben guardar los scripts SQL para que este proceso sea reproducible.
- 3- Se creen las sentencias SQL que realicen las diferentes acciones de la práctica. Dichas sentencias no tienen por qué estar integradas en el código para la entrega 1, pero sí en la entrega final. En el caso de necesitar diferentes sentencias, se deben listar en el orden adecuado, si aplica.

En este punto se debe tener en cuenta que algunas acciones como la creación de valores aleatorios es totalmente transparente para las sentencias SQL. Los valores se crean en la aplicación que consume el modelo de datos y posteriormente se realizan las acciones SQL.

Para las opciones 10 y 11, se deben indicar las sentencias que se van a ejecutar en función de la casuística que ocurra. Por ejemplo, el punto 10 va a realizar diferentes acciones (sentencias SQL) según el tipo de familia que se cree.

- 4- Se deben crear los índices necesarios para que las consultas a partir del nombre de los ratones sean accesibles de forma eficiente ($O(\log(n))$).

II. Entrega II

Los responsables del animalario desean no tener que replicar las acciones que realizan en su aplicación en una BBDD externa. Esto resta mucho tiempo de sus empleados y consideran que pueden aumentar su productividad si la aplicación realiza de forma automática todas las acciones de BBDD. Por ello, ya no quieren saber nada más de ficheros que no pueden ser utilizados por varias personas a la vez y que no garantizan la consistencia de los datos. Así, se debe persistir la información de su aplicación en una base de datos que permita realizar todas las acciones descritas en la aplicación.

En este caso, el guardado de los datos no se hará al realizar las acciones 11 y 12, sino que todas las acciones se guardarán tan pronto como se hagan. Se debe tener en cuenta que la acción 1 ahora debe cargar todos los ratones de una población concreta, así como las familias que se hayan creado previamente en dicha población. Esta casuística que pudiera parecer muy complicada en ficheros, en base de datos consiste en una relación

extra a cargar. Si nos fijamos bien en los requisitos de la aplicación, los ratones que formen parte de una familia como padre o madre (poligámica o normal, es indiferente) deben no pertenecer a la población, sino que la relación con la población se hace a través de la familia.

Se debe tener en cuenta el uso de transacciones para operaciones que deban ser atómicas (o se realizan todas, o ninguna).

La aplicación debe abrir una única conexión a la base de datos en el fichero lab.py, y se aconseja que se realicen métodos extra en un fichero .py que se encargue de gestionar la entrada/salida por base de datos. Así, algunos métodos no requerirán la conexión a BBDD porque la información ya está en memoria. Se recomienda consensuar con el profesor las acciones a realizar tras diseñarlas con antelación a su implementación.

Ítems a entregar: Implementación íntegra de los requisitos establecidos en el enunciado de la práctica junto a la memoria descriptiva del trabajo realizado. Se evaluará de forma más positiva la descripción de requisitos no implementados y conocidos descritos en la memoria que la falta de funcionalidad desconocida y no descrita.

Como resultado de esta entrega final se deberá enviar un fichero zip con la carpeta completa que contenga el proyecto software de Python correspondiente a la práctica (asegúrese de que incluye todo el código fuente, es la carpeta donde están todos los paquetes, si se utilizan paquetes). Si este proyecto requiere de algún requisito previo para la ejecución por parte del usuario (el profesor), dicha información debe aparecer en la memoria de la práctica. La entrega de la práctica forma parte de la evaluación de la práctica.

La memoria incluirá una portada con el nombre del alumno y se ajustará a la siguiente estructura:

- Análisis y descripción de la aplicación. Este análisis y descripción dará respuesta a las siguientes preguntas:
 - Cómo se han organizado y estructurado las entidades y las relaciones.
 - Qué decisiones de diseño se han tomado. ¿Por qué se accede o no se accede a la BBDD?
 - Diagramas E-R
- **Listado de fallos conocidos y funcionalidades definidas en el enunciado que no se han implementado** en el código entregado.
- Listado de todo el código fuente de la aplicación organizado por paquetes (si aplica) y módulos.
- Conclusiones (que incluirán, obligatoriamente, una valoración del tiempo dedicado a la práctica).

Consejos

Habla con el profesor para confirmar el esquema E-R que se va a utilizar para guardar la información en una Sistema Gestor de Base de Datos (RDBMS). Las tablas deben estar normalizadas.

Sobre la entrega de la práctica

La práctica debe ser entregada en formato electrónico antes del 20 de mayo(inclusive). El retraso en la entrega de la práctica se considerará como un 0 en su evaluación.

Junto con la práctica, el alumno deberá entregar un listado de los fallos que existen en la práctica, y/o la funcionalidad que no le ha dado tiempo de implementar. A la hora de calificar la práctica, se penalizarán menos la existencia de un fallo/carencia de funcionalidad conocido por el alumno (y que por tanto forma parte del listado de fallos), pero que no le ha dado tiempo a resolver, que un fallo/carencia de funcionalidad desconocido por el alumno (y que por tanto no forma parte del listado de fallos).

El alumno tendrá que realizar una defensa de la práctica; si el alumno es incapaz de defender la práctica y explicar su funcionamiento, su nota en la práctica será un 0.