

Gestión de Datos

Problema 1

I) De las 3 tablas anexas, escribe una consulta en SQL (Cualquier intérprete) que te dé los nombres de aquellos estudiantes cuyos amigos perciben un ingreso mayor que ellos. Los nombres deben de estar ordenados por el salario que ganan sus amigos.

Estudiantes

ID	Integer
Nombre	String

Amigos

ID	Integer
Amigos_ID	Integer

Salario

ID	Integer
Salario	Float

Estudiantes

ID	Nombre
1	Manuel
2	Tania
3	Pedro
4	Ana
5	Luis
6	Marta
7	Jorge
8	Clara
9	Pablo
10	Sara
11	David
12	Laura
13	Mario
14	Silvia
15	Carlos
16	Isabel
17	Antonio
18	Elena
19	Francisco
20	Sofia

Amigos

ID	Amigo_ID
1	1
2	20
3	3
4	12
5	4
6	5
7	1
8	13
9	6
10	12
11	17
12	18
13	19
14	8
15	9
16	15
17	10
18	7
19	10
20	11

Salario

ID	Salario
2	15200.1
2	10060.2
3	11500.5
4	12120
5	13200.75
6	9500.4
7	14500
8	10300.3
9	16000
10	8500.5
11	9000
12	17000.75
13	11000.2
14	12500.4
15	13500.1
16	9200.6
17	10100.8
18	11200.9
19	14000.25
20	14500.5

II) De las 3 tablas anexas, escribe una consulta que te dé el nombre del estudiante y el promedio del salario de sus amigos, pero solo para aquellos estudiantes cuyo promedio de salario de amigos sea mayor que el promedio de salario de todos los estudiantes. Los resultados deben estar ordenados por el promedio del salario de sus amigos en orden descendente.

Problema 2

Desarrolla un Datamart anexando los scripts que emplearías y responde las siguientes preguntas:

I) ¿Qué es un Datamart explicado en tus palabras? Describe detalladamente cómo un Datamart difiere de un Data Warehouse y da ejemplos de casos de uso específicos para un Datamart considerando como ejemplo, un banco.

II) Desarrolla las tablas del Datamart considerando las siguientes Dimensiones/Catálogos y construyéndose en base a la información que crees estas pudieran contener: Clientes, Productos que oferta el Banco, Tipo de cambio, Historial de transacciones. Además, debes incluir:

a) Un diagrama **ERD** (*Entity-Relationship Diagram*) mostrando las relaciones entre todas las tablas.

III) Describe las relaciones entre las tablas especificando el tipo de relación (una a muchas, muchas a muchas, etc.) y justificando por qué ese tipo de relación es adecuado para cada par de tablas. Incluye también:

a) Explicación de cómo estas relaciones ayudan a satisfacer necesidades específicas del área de un banco.

b) De ser posible ¿Plantearías una relación que facilitara la conexión entre tablas? Es decir, una tabla intermedia que pueda manejar los datos de manera más óptima.

IV) Identifica las llaves primarias y foráneas de cada tabla explicando lo siguiente:

a) El razonamiento detrás de la selección de éstas llaves.

b) Cómo manejarías las situaciones de datos faltantes o inconsistentes en estas llaves.

V) Establece controles para alimentar las tablas asegurando información correcta, íntegra, precisa y confiable. Para cada tabla, sigue las instrucciones siguientes::

a) Describe las validaciones específicas que le harías a los datos

b) Describe cómo regularías que éstas validaciones se lleven a cabo.

c) ¿Qué estrategias propones para evitar la pérdida de datos?

VI) Desarrolla un análisis para determinar los riesgos posibles utilizando un lenguaje de programación.

VII) Describe las características que debe tener un archivo de texto para ser importado en la base de datos y cómo automatizarías éste proceso..

VIII) Realiza validaciones en un lenguaje de programación que cumplan con los criterios que estableciste.

Problema 3

De forma complementaria, una vez que se encuentre listo el Datamart, el área tiene la necesidad de explotar la información y desea realizarlo vía alguna herramienta de visualización de datos interactiva.

I) Desarrolla un ejemplo de visualización que muestre los resultados de tu análisis apoyándote en los datos del problema 1.

II) Considera que estas a cargo de 3 personas. Un practicante, un analista y un subgerente en el área de datos. Realiza un diagrama de Gantt para organizar el cumplimiento de las tareas propuestas para el datamart aprovechando las habilidades de cada puesto mencionado.

Problema 4

Los problemas siguientes se evaluarán de la siguiente manera:

- 1) Lógica implementada
- 2) Buenas prácticas aplicadas
- 3) Optimización de la función o clase desarrollada.

I) Dado un diccionario de productos cualesquiera que contiene el nombre de un producto arbitrario como clave y una lista de tuplas como valor, donde cada tupla representa un pedido, con la cantidad y el precio por unidad, de la siguiente manera:

```
productos = {  
    "P_1": [x1=(10, 1.5), x2=(3, 1.4), x3=(5, 1.6)],  
    "P_2": [x1=(7, 2.1), x2=(2, 2.0), x3=(1, 2.2), x4=(NA, NA)],  
    "P_3": [x1=(12, 1.7), x2=(8, 1.6), x3=(3, 1.8)]  
}
```

Escribe una función que retorne un diccionario con el nombre de cada producto P_i y el precio total de dicho producto.

II) Dado un diccionario anidado como el siguiente:

```
Elementos = {  
    "e_1": {v1: 8, "v_2": 7, v3: 6},  
    "e_2": {"v_1": 9, v2: 6, v3: 7},  
}
```

" e_3 ": $\{v_1: 7, v_2: 8, v_3: 8\}$
}

Escribe una función que determine que e_i tiene el promedio más alto y además pueda ordenar los diccionarios anteriores.

III) Usando comprensión de listas y diccionarios, reescribe la función anterior para que el código sea más eficiente y compacto. El usuario debe ser capaz de escoger si sólo quiere ordenar los valores de los diccionarios, mostrar el promedio o ambos.

Problema 5

I) Diseña una función que, dado un arreglo de enteros (puede incluir números negativos), encuentre el subarreglo contiguo con la suma más alta. Usa el siguiente algoritmo para optimizar su función:

$$\max_{t(i)} = \max(A[i], \max_{t(i-1)} + A[i])$$

En dónde $\max_{t(i)}$ es la suma máxima del subarreglo que termina en el índice i en el tiempo t y $A[i]$ es el valor actual del arreglo.

Ejemplo:

$$A = [-2, 1, -3, 4, -1, 2, 1, -5, 4]$$

Inicializando tenemos que $\max_{t(i)} = -2$ por tanto el máximo actual es 2.

Luego, hacemos

$$\max_{t(i)} = \max(1, -2 + 1) = \max(1, -1) = 1$$

máximo hasta esta iteración: 1

Ahora, para el índice 2 cuyo valor es -3:

$$\max_{t(i)} = \max(-3, 1 + (-3)) = \max(-3, -2) = -2$$

máximo hasta esta iteración: 1

Y así sucesivamente para cada índice. Al final, el resultado es la suma máxima de un subarreglo contiguo.

II) Crea una función o clase que reciba una lista de montos de créditos solicitados por los clientes y un capital disponible total del banco. La función o clase debe encontrar la mayor cantidad de créditos que se pueden aprobar sin exceder el capital disponible.

Problema 6

I) Diseña una clase que modele la estructura de un banco con diferentes tipos de cuentas bancarias como se muestra a continuación:

Clase Banco: Contiene una lista de clientes, donde cada cliente tiene una cuenta bancaria.

Clase Cliente: Cada cliente tiene un nombre y una cuenta.

Clase Cuenta (clase base o "padre"): Tiene atributos como número de cuenta y saldo.

Tiene métodos como depositar(monto) y retirar(monto).

Clase Cuenta Corriente (hereda de Cuenta):

Esta clase tiene una característica adicional: permite sobregiros. Es decir, el saldo puede ser negativo hasta cierto límite.

Clase Cuenta Ahorros (hereda de Cuenta):

Esta clase gana intereses sobre el saldo, y no permite sobregiros.

Características a implementar:

Banco: Debe poder agregar clientes (y sus cuentas) a la lista de clientes.

Cliente: Debe contener atributos como nombre y su cuenta bancaria asociada (puede ser de cualquiera de los dos tipos: Cuenta Corriente o Cuenta Ahorros).

Cuenta (Clase base): Debe tener los métodos básicos como depositar(monto) retirar(monto) (en Cuenta Ahorros no puede retirar si no hay saldo suficiente, en Cuenta Corriente puede si no excede el límite de sobregiro).

Cuenta: Debe contener el saldo y métodos para depositar y retirar dinero.

Cuenta Corriente: debe permitir retirar dinero, incluso si el saldo es menor que el monto solicitado (permitiendo un saldo negativo), pero con una penalización del 7% sobre el monto del sobregiro.

Cuenta Ahorros: No permite sobregiros, pero acumula intereses mensuales a una tasa fija.

II) Añade un método en la clase Banco que permita transferir dinero entre dos cuentas de clientes diferentes. Si la cuenta origen es una Cuenta Corriente y se produce un sobregiro, se debe aplicar la penalización automáticamente.

III) Implementa un sistema de auditoría que registre en un archivo CSV con todas las transacciones realizadas en el banco, indicando la fecha, el cliente, el tipo de cuenta y la cantidad transferida o retirada.

El planteamiento de los problemas pretende evaluar la organización, gestión del tiempo, creatividad, conocimiento básico del sector bancario, conocimientos técnicos y solución de problemas por lo que es importante detallar lo mayor posible, describir consideraciones o supuestos, para tener visibilidad de los aspectos a evaluar y pasar a la siguiente etapa. Es deseable el envío de código en caso de que aplique.

¡Éxito!