

## Adquisición y Tratamiento de Datos. (30/01/2001). TEORIA.

1) Enumera los:

- a) Dos tipos de organización física de un archivo. (0.25 pts.)
- b) Tres tipos de organización lógica de un archivo. (0.25 pts.)
- c) Tres tipos de bases de datos existentes. (0.25 pts.)

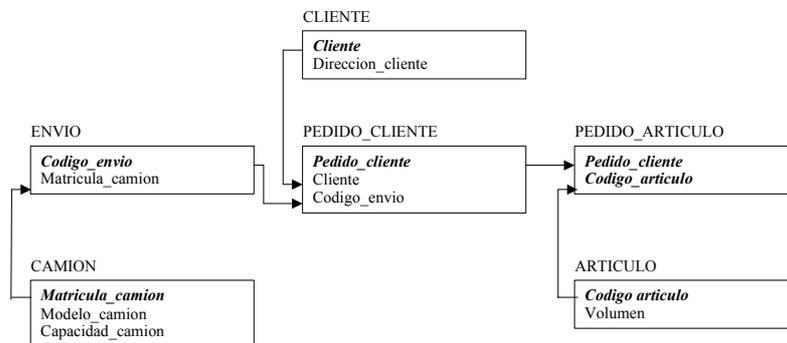
2) Dada la siguiente relación:

PROFESOR				
CODIGO	NOMBRE	APELLIDOS	CATEGORÍA	SALARIO
H0001	Antonio	García García	Catedrático	350.000
H0002	Amparo	Pérez Pérez	Ayudante	150.000
H0003	Isabel	Fernández Fernández	Titular	240.000
H0004	José	Hernández Hernández	Ayudante	160.000
H0005	Carlos	Martínez Martínez	Titular	260.000

Escribir la relación resultante de aplicar los operadores siguientes:

- a)  $\sigma_{\text{categoria}=\text{ayudante}}(\text{PROFESOR})$  (0.25 pts.)
- b)  $\pi_{\text{categoria}}(\text{PROFESOR})$  (0.25 pts.)
- c) PROFESOR AGRUPACION\_POR CATEGORIA, MEDIA(SALARIO) (0.25 pts.)

3) Dada el esquema de la siguiente base de datos:



Indicar que preguntas contestan las siguientes sentencias SELECT de SQL:

a) (0.5 pts.).

```
SELECT cliente.cliente, COUNT(*) FROM cliente, pedido_cliente WHERE
cliente.cliente=pedido_cliente.cliente GROUP BY cliente.cliente ORDER BY 2 DESC
```

b) (0.5 pts.).

```
SELECT AVG(volumen) FROM articulo, pedido_articulo, pedido_cliente WHERE
pedido_cliente.pedido_cliente=pedido_articulo.pedido_cliente AND
pedido_articulo.codigo_articulo=articulo.codigo_articulo AND cliente=XXX GROUP BY
pedido_cliente.pedido_cliente
```

c) (0.5 pts.).

```
SELECT * FROM camion WHERE capacidad_camion>=ALL(SELECT capacidad_camion
FROM camion)
```

4) Indica que realizan las sentencias de SQL COMMIT WORK y ROLLBACK WORK (1 pts.)

5) ¿Cuál es la diferencia existente entre una clave primaria y una clave foránea de una tabla en una base de datos relacional? (1 pto.)

6) Supongamos una base de datos donde el administrador acaba de crear una nueva tabla, llamada tablaX. A continuación, el administrador ejecuta el siguiente comando:

```
GRANT ALL PRIVILEGES ON tablaX TO usuarioA.
```

Una vez ejecutada la anterior sentencia, el usuarioA ejecuta la sentencia siguiente:

```
GRANT SELECT ON tablaX TO usuarioB.
```

¿Puede el usuarioB ejecutar una sentencia SELECT?. Explica la respuesta. (1 pto.)

7) Se desea diseñar una base de datos para la realización de un congreso. Al congreso se presentan una serie de artículos que han de ser evaluados por una serie de revisores en función del tema del artículo. El revisor emite un informe evaluando la calidad del artículo. Se desea que la base de datos contenga la siguiente información:

- Sobre los artículos se desea conocer el título, la fecha de recepción y si ha sido aceptado o no.
- Sobre los autores se desea conocer el nombre, departamento, organización, dirección, código postal, país, correo electrónico y teléfono.
- Sobre los revisores se desea conocer su nombre y apellidos, departamento, organización, dirección, país, correo electrónico y teléfono.
- La evaluación del artículo se hace en términos de su originalidad, importancia y claridad y se asigna una puntuación que varía entre 1 y 5.

Se asume que:

- Un artículo puede tener varios autores.
- Un autor puede presentar varios artículos.
- Un revisor puede revisar varios artículos.
- Un artículo puede ser revisado por varios revisores.
- Un revisor tiene asociado varios tópicos.

Diseñar una base de datos de acuerdo a las especificaciones anteriores y normalizarla hasta 3FN. (3 ptos.).

8) Dado un directorio de LINUX cuya respuesta a la salida del comando **ls -l** es:

```
-rw-rw-rw- 1 usuario grupo    23214 Jan 30 9:00 copia_enero.gz
-rw-rw-rw- 1 usuario grupo   345678 Jan 30 9:00 copia_enero2.gz
-rw-rw-rw- 1 usuario grupo    2456 Jan 30 9:00 datos_febrero.gz
-rw-rw-rw- 1 usuario grupo  123984 Jan 30 9:00 datos_febrero2.gz
-rw-rw-rw- 1 usuario grupo    4567 Jan 30 9:00 copia_datos.gz
-rw-rw-rw- 1 usuario grupo  234112 Jan 30 9:00 copia_datos2.gz
-rw-rw-rw- 1 usuario grupo    110 Jan 30 9:00 copia_.gz
-rw-rw-rw- 1 usuario grupo   45600 Jan 30 9:00 copia2_.gz
```

Escribir el comando (un solo comando y un solo atributo en el comando) que borre los seis primeros ficheros del directorio sin borrar los dos últimos. Como ayuda se os recuerda la existencia de caracteres comodín (1 pto.).

## Adquisición y Tratamiento de Datos. (30/01/2001). PRÁCTICAS.

*El alumno deberá contestar tres preguntas, eligiendo para ello 2 de las tres primeras preguntas (preguntas 1, 2 y 3), y la pregunta 4. Si se contestan más de tres preguntas el examen será suspendido automáticamente.*

1) En la práctica 1, teníamos un fichero en formato dos ficheros BINARIOS con la siguiente estructura de datos:

```
struct PRODUCTO{ /* Estructura del fichero producto.bin */
  int codigo; /*Codigo del producto */
  float precio; /* Precio del producto */
};
struct EXISTENCIAS{ /* Estructura del fichero existencias.bin */
  int codigo; /*Codigo del producto */
  int numero; /* Existencias del producto */
};
```

Dado el siguiente trozo de programa:

```
FILE *fp[2];
struct PRODUCTO producto;
struct EXISTENCIAS existencias;
float total=0.0,valor;
int valido;

while(fread(&producto,sizeof(struct PRODUCTO),1,fp[0])==1)
{
  fseek(fp[1],0L,SEEK_SET);
  while ((valido=(fread(&existencias,sizeof(struct EXISTENCIAS),1,fp[1])==1))==1)
    if (existencias.codigo==producto.codigo)
      break;
  if (valido)
  {
    valor=producto.precio*existencias.numero;
    total+=valor;
    printf("El valor del inventario del producto %d es %f\n", producto.codigo,valor);
  }
  else
    printf("Producto %d no encontrado en existencias\n",producto.codigo);
}
```

a) Explica lo que hace. ¿Cuál es el motivo del uso de fseek con fp[1]? ¿porqué no se usa también para el fichero fp[0]? (1.5 pts.).

b) Suponiendo que existe la misma cantidad de elementos en el fichero de productos que en el fichero de existencias, y que para cada registro de producto existe un registro de existencia y además escritos en el mismo orden, modifica el trozo de programa para que quede lo mas simplificado posible. (2 pts.).

2) En la práctica 2, disponíamos de unos ficheros que almacenaban cabecera y datos con el siguiente formato:

- 1 byte que contiene el código A3h y que indica que el fichero es de nuestro formato.
- 1 byte que en formato binario contienen el número de campos de cabecera existentes. Su valor puede ir de 1 a 255.

Para cada uno de los campos existentes:

- 1 byte con el tipo de datos, que puede ser E entero (int), F coma flotante (float), C carácter y S cadena de caracteres (string).
- 1 byte con la longitud en binario del tipo de datos, solo usado para indicar la longitud, incluyendo el carácter terminador ('\0') de la cadena de caracteres (string). Su valor puede ir de 2 a 255
- 16 bytes con el nombre de los datos en formato de cadena de caracteres. Los 16 bytes incluyen el carácter terminador de string ('\0').

A continuación de los campos de cabecera se encuentran los datos de cada uno de los registros del fichero.

```

struct CABECERA{
    unsigned char id; /* Identificador del fichero */
    unsigned char ncampos; /* Número de campos de la cabecera */
};
struct CAMPO{
    char tipo; /* Tipo de datos (E, F, C o S)*/
    unsigned char tamaño; /* Tamaño de los datos, solo usado si tipo es S */
    char nombre[16]; /* Nombre del tipo de datos */
};

int main(int argc,char *argv[])
{
    struct CABECERA cabecera;
    struct CAMPO campo[255];
    FILE *fp;
    register int i;
    int sumai=0;
    float sumaf=0;

    if ((fp=fopen(argv[1],"rb"))==NULL)
        return 0;

    if (LeerCabecera(fp,&cabecera)!=0)
        Error("Error leyendo la cabecera del fichero ",argv[1]);

    for(i=0;i<cabecera.ncampos;i++)
        if (LeerCampo(fp,&campo[i])!=0)
            Error("Error leyendo los campos del fichero ",argv[1]);
        else
            MostrarCampo(&campo[i]);

    while (LeerDatos(fp,cabecera.ncampos,campo,&sumai,&sumaf)==0);

    fclose(fp);

    return 0;
}

```

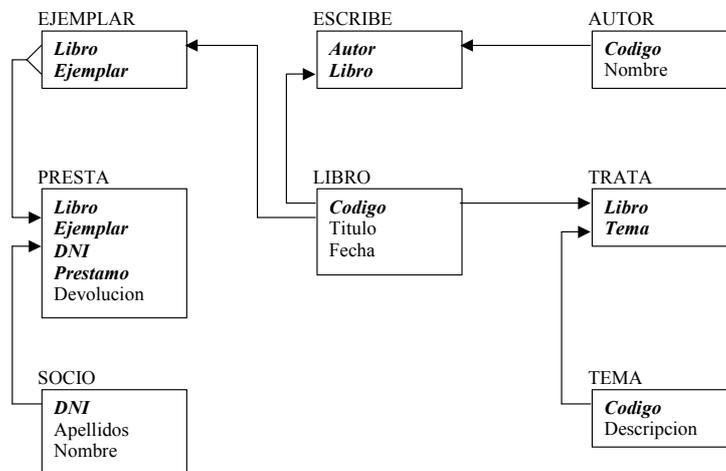
Utilizando la estructuras de datos dadas y el programa principal dado, hacer la función LeerDatos, que además de lo que se pedía en la práctica (mostrar todos los datos de los registros), deberá almacenar en “sumai” la suma de todos los datos enteros y en “sumaf” la suma de todos los datos reales. (3.5 pts.).

3) Escribir un programa que lea una serie de números enteros en código de un fichero “datos.dat” (en formato ASCII), y devuelva por pantalla la media ( $\sum_i^N \frac{X_i}{N}$ ), el valor máximo, y la desviación típica ( $\sqrt{\sum_i^N (x_i - \bar{x})^2 / N}$ ). La parte principal del programa constará de tres llamadas secuenciales a las siguientes funciones a desarrollar por el alumno:

- lee\_numeros(.....) -> lee los números enteros del fichero
- calcula\_estadistica(.....) -> calcula la media, valor máx. y desviación típica.
- visualiza\_resultados(.....) -> muestra los cálculos por pantalla.

Queda a cargo de los alumnos, los parámetros a utilizar en cada función, y los valores de retorno de cada una de ellas, teniendo en cuenta que no se podrá utilizar ninguna variable global y tan solo existirá una constante en el programa: MAX=100 (número máximo de enteros de que consta el fichero “datos.dat”). (3.5 pts.).

4) Dado el siguiente diseño de base de datos:



Escribir las sentencias SQL que contestan a las siguientes preguntas:

- Título y número de ejemplares existentes de cada libro ordenados de mayor a menor número de ejemplares. (0.75 pts.).
- Dni, nombre y apellidos de los socios que no han recibido en préstamo más de tres libros. (0.75 pts.).
- Título de los libros y nombre de los autores de aquellos libros que han sido escritos por cinco o más autores. (0.75 pts.).
- Código y título de los libros (sin tener en cuenta su ejemplar) que más veces han sido prestados. (0.75 pts.).