TP5: Índices - Búsqueda y recuperación de información

Alumno: Juan Cruz Mateos
Mat. 15134

1) Busque 3 ejemplos de entidades con 2 relaciones para cada una de ellas.

Entidades: empleado, guitarra, libro.

Empleado:
Un empleado pertenece a un departamento.
Un empleado reporta a un jefe.

Guitarra:
Una guitarra posee cuerdas.
Una guitarra es tocada por un músico.

Libro:
Un libro posee hojas.
Un libro es escrito por un autor.

2) A partir de la siguiente información de vehículos, implemente algún tipo de organización indexada. Justifique la elección. Defina ventajas y desventajas.

Patente (campo clave)	Marca	Modelo	Color	Puertas
AA098JD	Ford	Fiesta	Rojo	3
AC563FF	Peugeot	207	Negro	3
AD835CE	Ford	Ka	Blanco	2
AA523KS	Fiat	Mobi	Gris	3
AB964AA	Ford	Ka	Gris	4
AB002MM	Fiat	500	Azul	5
AC934US	Peugeot	5008	Rojo	5
AA051AR	Ford	Ecosport	Azul	5
AD467VD	Peugeot	207	Blanco	3

Utilizaría indices invertidos, que es una organización a partir de indices secundarios basados en la clave primaria. En este tipo de organización la información del indice no tiene por que estar en el dato, ya que es redundante. Son útiles cuando las búsquedas se realizan sobre una combinación de atributos ya que permite un acceso eficiente para diferentes criterios (distintos campos). Ademas, permiten responder a preguntas complejas sin necesidad de acceder a los datos. Sin embargo, un inconveniente de esta organización es que si se necesita obtener los datos completos de un auto en particular habría que hacer una búsqueda sobre la totalidad de los datos.

Tomando como área principal de datos la tabla dada, las listas invertidas son:

Marcas:

Ford	AA098JD, AD835CE, AB964AA, AA051AR
Peugeot	AC563FF, AC934US, AD467VD
Fiat	AA523KS, AB002MM

Modelos:

Fiesta	AA098JD
207	AC563FF, AD467VD
Ka	AD835CE, AB964AA
Mobi	AA523KS
500	AB002MM
5008	AC934US
Ecosport	AA051AR

Colores:

Rojo	AA098JD, AC934US
Negro	AC563FF
Blanco	AD835CE, AD467VD
Gris	AA523KS, AB964AA
Azul	AB002MM, AA051AR

Puertas:

2	AD835CE
3	AA098JD, AC563FF, AA523KS, AD467VD
4	AB964AA
5	AB002MM, AC934US, AA051AR

3) Realice los 4 pasos para los siguientes textos:

1) Parseo
2) Ordenamiento
3) Agrupamiento.
4) Generación del Diccionario y Posting.

“Ya no me necesitas es lo mejor
Eras alguien a quien yo solía conocer
Fue muy simple despegar
Solo un corto tiempo
Y te buscaste un nuevo corazón
Ahora tienes tu propio show
Como un rey vengador, vengador
No te alcanza con improvisar
El descaro baby es parte de la diversión”

“¿Que importan ya tus ideales?
¿Que importa tu canción?
La grasa de las capitales
cubre tu corazón.
¿Por qué tenes que llorar?

Es que hay otro en tu lugar que dice:
"Vamos, vamos, la fama."
Tu oportunidad está ahí,
lo mismo me pasó a mí.
Lo tienes todo, todo
no hay nada."

Parseo:

Termino	D#	
ya		1
no		1
me		1
necesitas		1
es		1
lo		1
mejor		1
eras		1
alguien		1
a		1
quien		1
yo		1
solía		1
conocer		1
fue		1
muy		1
simple		1
despegar		1
solo		1
un		1
corto		1
tiempo		1
y		1
te		1
buscaste		1
un		1
nuevo		1
corazón		1
ahora		1
tienes		1
tu		1
propio		1
show		1
como		1
un		1
rey		1
vengador		1
vengador		1
no		1
te		1
alcanza		1

con	1
improvisar	1
el	1
descaro	1
baby	1
es	1
parte	1
de	1
la	1
diversión	1
que	2
importan	2
ya	2
tus	2
ideales	2
que	2
importa	2
tu	2
canción	2
la	2
grasa	2
de	2
las	2
capitales	2
cubre	2
tu	2
corazón	2
por	2
qué	2
tenes	2
que	2
llorar	2
es	2
que	2
hay	2
otro	2
en	2
tu	2
lugar	2
que	2
dice	2
vamos	2
vamos	2
la	2
fama	2
tu	2
oportunidad	2
está	2
ahí	2
lo	2
mismo	2
me	2

pasó	2
a	2
mí	2
lo	2
tienes	2
todo	2
todo	2
no	2
hay	2
nada	2

Ordenamiento:

Termino	D#
a	1
a	2
ahora	1
ahí	2
alcanza	1
alguien	1
baby	1
buscaste	1
canción	2
capitales	2
como	1
con	1
conocer	1
corazón	1
corazón	2
corto	1
cubre	2
de	1
de	2
descaro	1
despegar	1
dice	2
diversión	1
el	1
en	2
eras	1
es	1
es	1
es	2
está	2
fama	2
fue	1
grasa	2
hay	2
hay	2
ideales	2
importa	2
importan	2

improvisar	1
la	1
la	2
la	2
las	2
llorar	2
lo	1
lo	2
lo	2
lugar	2
me	1
me	2
mejor	1
mismo	2
muy	1
mí	2
nada	2
necesitas	1
no	1
no	1
no	2
nuevo	1
oportunidad	2
otro	2
parte	1
pasó	2
por	2
propio	1
que	2
que	2
que	2
que	2
que	2
quien	1
qué	2
rey	1
show	1
simple	1
solo	1
solía	1
te	1
te	1
tenes	2
tiempo	1
tienes	1
tienes	2
todo	2
todo	2
tu	1
tu	2
tu	2
tu	2

tu	2
tus	2
un	1
un	1
un	1
vamos	2
vamos	2
vengador	1
vengador	1
y	1
ya	1
ya	2
yo	1

Agrupamiento:

Termino	D#	F#
a	1	1
a	2	1
ahora	1	1
ahí	2	1
alcanza	1	1
alguien	1	1
baby	1	1
buscaste	1	1
canción	2	1
capitales	2	1
como	1	1
con	1	1
conocer	1	1
corazón	1	1
corazón	2	1
corto	1	1
cubre	2	1
de	1	1
de	2	1
descaro	1	1
despegar	1	1
dice	2	1
diversión	1	1
el	1	1
en	2	1
eras	1	1
es	1	2
es	2	1
está	2	1
fama	2	1
fue	1	1
grasa	2	1
hay	2	2
ideales	2	1
importa	2	1

importan	2	1
improvisar	1	1
la	1	1
la	2	2
las	2	1
llorar	2	1
lo	1	1
lo	2	2
lugar	2	1
me	1	1
me	2	1
mejor	1	1
mismo	2	1
muy	1	1
mí	2	1
nada	2	1
necesitas	1	1
no	1	2
no	2	1
nuevo	1	1
oportunidad	2	1
otro	2	1
parte	1	1
pasó	2	1
por	2	1
propio	1	1
que	2	5
quien	1	1
qué	2	1
rey	1	1
show	1	1
simple	1	1
solo	1	1
solía	1	1
te	1	2
tenes	2	1
tiempo	1	1
tienes	1	1
tienes	2	1
todo	2	2
tu	1	1
tu	2	4
tus	2	1
un	1	3
vamos	2	2
vengador	1	2
y	1	1
ya	1	1
ya	2	1
yo	1	1

Diccionario:

Termino	TD#	FT#	#RR
a	2	2	0
ahora	1	1	2
ahí	2	1	3
alcanza	1	1	4
alguien	1	1	5
baby	1	1	6
buscaste	1	1	7
canción	2	1	8
capitales	2	1	9
como	1	1	10
con	1	1	11
conocer	1	1	12
corazón	2	2	13
corto	1	1	15
cubre	2	1	16
de	2	2	17
descaro	1	1	19
despegar	1	1	20
dice	2	1	21
diversión	1	1	22
el	1	1	23
en	2	1	24
eras	1	1	25
es	2	3	26
está	2	1	28
fama	2	1	29
fue	1	1	30
grasa	2	1	31
hay	2	2	32
ideales	2	1	33
importa	2	1	34
importan	2	1	35
improvisar	1	1	36
la	2	3	37
las	2	1	39
llorar	2	1	40
lo	2	3	41
lugar	2	1	43
me	2	2	44
mejor	1	1	46
mismo	2	1	47
muy	1	1	48
mí	2	1	49
nada	2	1	50
necesitas	1	1	51
no	2	3	52
nuevo	1	1	54
oportunidad	2	1	55
otro	2	1	56

parte	1	1	57
pasó	2	1	58
por	2	1	59
propio	1	1	60
que	2	5	61
quien	1	1	62
qué	2	1	63
rey	1	1	64
show	1	1	65
simple	1	1	66
solo	1	1	67
solía	1	1	68
te	1	2	69
tenes	2	1	70
tiempo	1	1	71
tienes	2	2	72
todo	2	2	74
tu	2	5	75
tus	2	1	77
un	1	3	78
vamos	2	2	79
vengador	1	2	80
y	1	1	81
ya	2	2	82
yo	1	1	84

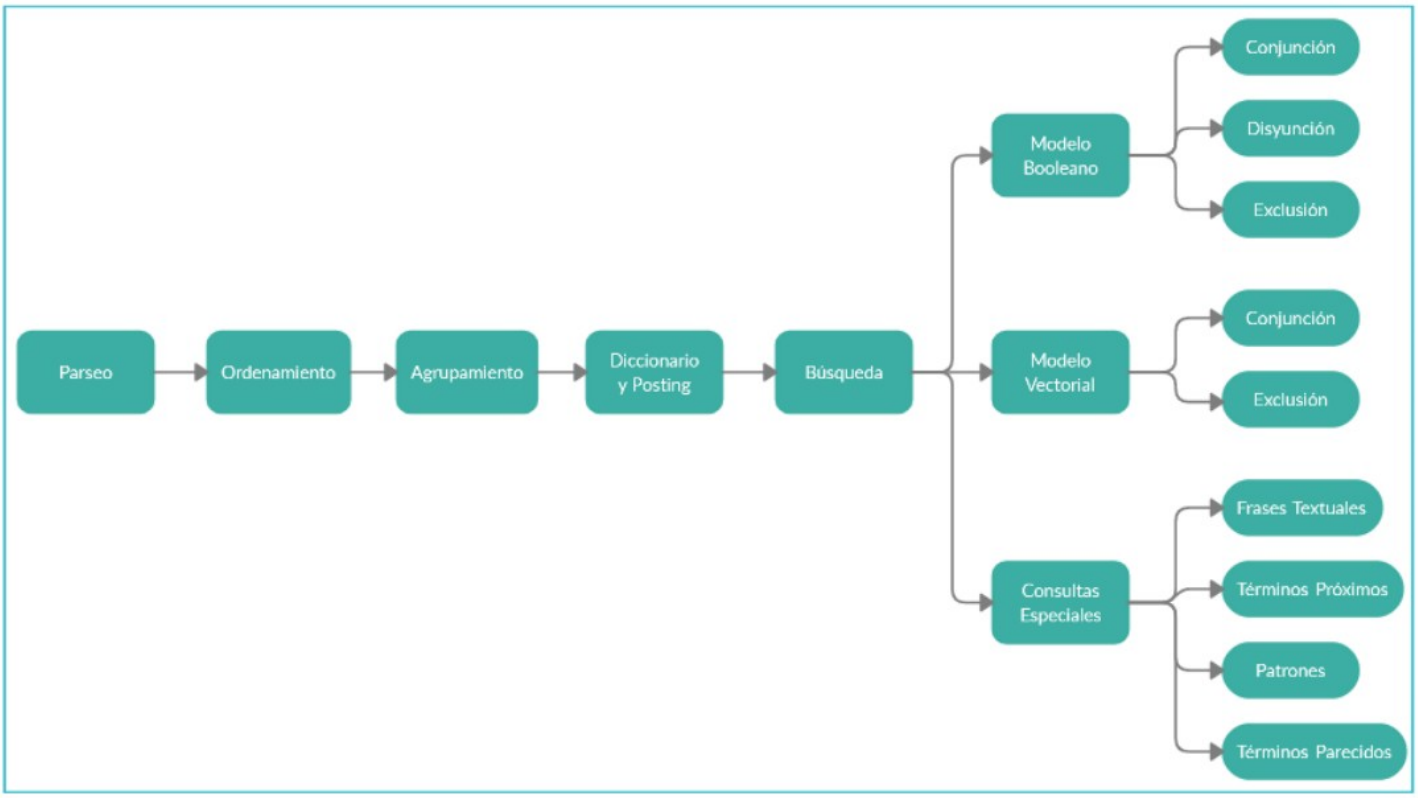
Posting:

#RR	D#	F#	#PR
0	1	1	1
1	2	1	-1
2	1	1	-1
3	2	1	-1
4	1	1	-1
5	1	1	-1
6	1	1	-1
7	1	1	-1
8	2	1	-1
9	2	1	-1
10	1	1	-1
11	1	1	-1
12	1	1	-1
13	1	1	14
14	2	1	-1
15	1	1	-1
16	2	1	-1
17	1	1	18
18	2	1	-1
19	1	1	-1
20	1	1	-1
21	2	1	-1

22	1	1	-1
23	1	1	-1
24	2	1	-1
25	1	1	-1
26	1	2	27
27	2	1	-1
28	2	1	-1
29	2	1	-1
30	1	1	-1
31	2	1	-1
32	2	2	-1
33	2	1	-1
34	2	1	-1
35	2	1	-1
36	1	1	-1
37	1	1	38
38	2	2	-1
39	2	1	-1
40	2	1	-1
41	1	1	42
42	2	2	-1
43	2	1	-1
44	1	1	45
45	2	1	-1
46	1	1	-1
47	2	1	-1
48	1	1	-1
49	2	1	-1
50	2	1	-1
51	1	1	-1
52	1	2	53
53	2	1	-1
54	1	1	-1
55	2	1	-1
56	2	1	-1
57	1	1	-1
58	2	1	-1
59	2	1	-1
60	1	1	-1
61	2	5	-1
62	1	1	-1
63	2	1	-1
64	1	1	-1
65	1	1	-1
66	1	1	-1
67	1	1	-1
68	1	1	-1
69	1	2	-1
70	2	1	-1
71	1	1	-1
72	1	1	73
73	2	1	-1

74	2	2	-1
75	1	1	76
76	2	4	-1
77	2	1	-1
78	1	3	-1
79	2	2	-1
80	1	2	-1
81	1	1	-1
82	1	1	83
83	2	1	-1
84	1	1	-1

4) Grafique el proceso de un motor de búsqueda. En cada paso qué estructura de datos usaría.



- ***Parseo***: utilizaría una lista para almacenar los términos junto con el id del documento en el cual se encuentran.
- ***Ordenamiento***: mantendría la misma estructura que en el parseo, solo ordenándola por termino, manteniendo los duplicados.
- ***Agrupamiento***: utilizaría una lista para almacenar los términos, junto con el id del archivo donde se encuentran y su frecuencia en dicho archivo.
- ***Diccionario***: utilizaría un archivo inverso con indice no denso. El archivo del diccionario direccionaría al posting.
- ***Posting***: utilizaría un archivo con lista inversa.

5) Ejercicio de programación - Opcional

Implemente una aplicación con la siguiente funcionalidad general:

Debe procesar un directorio (tomado como parámetro) y del mismo indexar todos los archivos de texto que encuentre en él, permitiendo posteriormente realizar búsquedas por palabras y obtener la lista de archivos donde aparecen estas palabras. Para la indexación debe generar las estructuras necesarias (pueden ser solo en memoria), utilizando índices inversos. Como mínimo permitirá una búsqueda por una sola palabra. Se realizará el control mínimo de errores (path incorrecto o mal formateado, ausencia de archivos, etc).

Ejemplo de ejecución:

usuario@hal9000:/home/sources/indx$./indx ./pruebas
3 archivo(s) encontrado(s)
Parseando … 100% listo
Ordenando … 100% listo
Agrupando … 100% listo
Generando diccionario y posteando … 100% listo
Ingrese la palabra a buscar : GATO
Encontrada en el archivo : texto1.txt , 3 veces
Encontrada en el archivo : texto4txt , 1 veces
Buscamos otra palabra? NO
usuario@hal9000:/home/sources/indx$

*Resuelto en **motor.py***

Ejecución (sobre Linux): python3 ./motor.py [path]

Obs.: si se omite el path por defecto se toma ./