

```

1  /*EJERCICIO 1. Realiza un script con un cursor que muestre todo el dinero ganado en
   un
2  mes para cada una de las pizzas. El mes que se mostrará será el que más pizzas se
   haya
3  comprado de ese tipo concreto. Puedes utilizar internamente funciones y
4  procedimientos.*/
5
6  IF OBJECT_ID ('dbo.beneficioPizzafavorita') IS NOT NULL
7      DROP PROCEDURE dbo.beneficioPizzafavorita;
8  GO
9
10 CREATE PROCEDURE beneficioPizzafavorita
11     @Npizza VARCHAR(20),
12     @beneficioTotal DECIMAL(4,2) OUTPUT
13
14 AS
15     -- body of the stored procedure
16 DECLARE @mes int----,@beneficioTotal INT
17
18 SELECT TOP 1 @beneficioTotal = SUM(p.price),@mes = datepart(MONTH from s.date2)
19 FROM pizzas p
20 INNER JOIN sales s on p.Npizza = s.Npizza
21 WHERE s.Npizza LIKE @Npizza
22 GROUP BY datepart(MONTH from s.date2)
23 ORDER BY SUM(p.price) DESC
24
25 --DECLARE @Npizza nvarchar(20), @beneficioTotal int
26 --EXECUTE dbo.beneficioPizzafavorita @Npizza, @beneficioTotal OUTPUT
27 --PRINT CAST(@beneficioTotal as nvarchar(10))
28 GO
29 --CURSOR
30 DECLARE @Npizza VARCHAR(20),
31         @mes INT,
32         @importetotal INT
33
34 DECLARE BeneficiosMesPizzas CURSOR
35 FOR
36     SELECT TOP 1 (s.Npizza) ,
37     DATEPART(MONTH FROM s.date2), --as mes,
38     SUM(p.price) --as[Importe Total]
39     FROM sales s
40     inner join pizzas p on s.Npizza =p.Npizza
41     GROUP BY s.Npizza,DATEPART(MONTH FROM s.date2)
42     ORDER BY SUM(p.price)
43
44 OPEN BeneficiosMesPizzas
45 FETCH BeneficiosMesPizzas INTO @Npizza, @mes, @importetotal
46 WHILE(@@FETCH_STATUS = 0)
47     BEGIN
48         --declare @mesletras nvarchar(10)
49         --set @mesletras = (SELECT DATENAME(datepart, @mes));
50         DECLARE @beneficioTotal decimal(4,2)
51         EXECUTE dbo.beneficioPizzafavorita @Npizza, @beneficioTotal OUTPUT
52         PRINT 'MES:'+cast(@mes as nvarchar(20))
53         PRINT 'PIZZA:'+@Npizza
54         PRINT 'BENEFICIO:'+cast(@beneficioTotal as nvarchar(20))
55         PRINT space(1)
56         FETCH BeneficiosMesPizzas INTO @Npizza, @mes, @importetotal
57     END
58 CLOSE BeneficiosMesPizzas
59 DEALLOCATE BeneficiosMesPizzas

```

```
60
61
62
63 /*EJERCICIO 2. Realiza un script con un cursor en el que se muestren todas las
64 películas
65 que han obtenido al menos un 10 de calificación por alguno de los socios. Además,
66 deberás mostrar el socio que la calificó con esta nota.*/
67 --CURSOR
68 GO
69 USE videoclub
70 GO
71 DECLARE @titulo nvarchar(40),
72          @codigo int,
73          @codigo_socio int,
74          @nombre nvarchar(40)
75
76 DECLARE PeliculasCalificacion10 CURSOR
77 FOR
78     select p.titulo, a.codigo_socio, s.codigo, s.nombre
79     from peliculas p
80     inner join STOCK stk on stk.CODIGO_PELICULA = p.codigo
81     inner join alquiler a on a.codigo_stock = stk.CODIGO
82     inner join socios s on a.CODIGO_SOCIO = s.CODIGO
83     where CALIFICACION = 10
84
85 OPEN PeliculasCalificacion10
86 FETCH PeliculasCalificacion10 INTO @titulo, @codigo_socio, @codigo, @nombre
87 PRINT '**-----Películas con Calificación de 10-----**'
88 PRINT '-----'
89 WHILE(@@FETCH_STATUS = 0)
90     BEGIN
91         PRINT 'Titulo:' + @titulo
92         PRINT 'Socio nº:' + cast(@codigo as nvarchar(20))+space(3)+@nombre
93         print space(1)
94         FETCH PeliculasCalificacion10 INTO @titulo, @codigo_socio, @codigo, @nombre
95     END
96 CLOSE PeliculasCalificacion10
97 DEALLOCATE PeliculasCalificacion10
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
```

```

120 /*EJERCICIO 3. Muestra los socios que hayan alquilado al menos una película y
121 posteriormente muestra las categorías de las que han obtenido películas.*/
122
123 --SOLUCION
124 /*Vamos a realizar un cursor que recorra las tabla socio-alquiver para mostrar
125 socios
126 con + 1 alquiler.
127 Las categorias las obtendremos de otro cursor
128 */
129 --CURSOR
130 DECLARE @codigo INT,
131          @nombre NVARCHAR(22),
132          @apellidos NVARCHAR(22),
133          @Nalquileres INT
134
135 DECLARE sociosalquiler CURSOR
136 FOR
137     SELECT s.codigo,s.nombre,s.apellido, count(*) --as Nalquileres
138     from socios s
139     inner join alquiler a on CODIGO_SOCIO = codigo
140     group by codigo,nombre,apellido
141     having count(*)>1
142
143 OPEN sociosalquiler
144 FETCH sociosalquiler INTO @codigo, @nombre, @apellido, @Nalquileres
145 WHILE(@@FETCH_STATUS = 0)
146     BEGIN
147         PRINT 'Socio nº:' + cast(@codigo as nvarchar(20))+space(3)
148         PRINT 'Nombre:' + @nombre
149         PRINT 'Apellidos:' + @apellido
150         print space(1)
151         --2º cursor
152         DECLARE @codigosocio INT, @nombrecategoria NVARCHAR(25)
153         DECLARE CategoriasSocio CURSOR
154         FOR
155             SELECT a.CODIGO_SOCIO, c.nombre
156             FROM alquiler a
157             INNER JOIN stock stk ON stk.codigo= a.codigo_socio
158             inner join peliculas p on p.codigo = stk.CODIGO_PELICULA
159             inner join categoria c on c.codigo = p.CODIGO_CATEGORIA
160             where CODIGO_SOCIO = @codigo
161         OPEN CategoriasSocio
162         FETCH CategoriasSocio INTO @codigosocio, @nombrecategoria
163         WHILE(@@FETCH_STATUS = 0)
164             BEGIN
165                 PRINT 'CATEGORIA:' + @nombrecategoria
166
167                 FETCH CategoriasSocio INTO @codigosocio, @nombrecategoria
168             END
169         CLOSE CategoriasSocio
170         DEALLOCATE CategoriasSocio
171     FETCH sociosalquiler INTO @codigo, @nombre, @apellido, @Nalquileres
172     END
173 CLOSE sociosalquiler
174 DEALLOCATE sociosalquiler
175
176
177
178
179

```

```
180
181 /*EJERCICIO 4. Vamos a realizar un pequeño programa que analice si nos sale rentable
182 tener a ciertos pacientes en nuestro seguro médico. Para ello, tendremos que añadir
183 un
184 campo a la tabla PACIENTES que se llamará 'TARIFA' y será de tipo DECIMAL (8, 2).
185 A este campo le indicaremos como valor por defecto 50€.
186
187 A partir de aquí, crear un cursor que recorra la siguiente consulta:
188 NOMBRE PACIENTE, CIUDAD, NUM_VISITAS
189 Para cada línea del cursor se debe realizar:
190     □ Si el número de visitas de pacientes es > 4 debe:
191         1. Imprimir el mensaje de: PACIENTE PROBLEMÁTICO
192         2. Aumentar la tasa mínima a cobrar por el paciente un 20%.
193
194     □ Ejecutar un Procedimiento Almacenado, donde le pasaremos la Ciudad y nos
195     devuelva la media del Importe de las visitas correspondientes a pacientes de esa
196     Ciudad.
197
198     □ Si la media devuelta por el procedimiento almacenado es mayor que 250, debe
199     realizar lo siguiente:
200         1. Insertar en la tabla PACIENTES_PROBLEMATICOS (que tendremos que
201         crear), el Nombre del paciente y su código, la media del importe de
202         visitas, la tarifa y la fecha en la que se realiza la inserción.
203
204     □ Mostrar por pantalla:
205     PACIENTE: XXXXXXXXXXXX(Nombre)
206     TARIFA: XXXXX
207     MEDIA DEL IMPORTE: XXXX (Media devuelta por el procedimiento)*/
208
209 ALTER TABLE pacientes ADD tarifa decimal(8,2)
210
211 CREATE TABLE  PACIENTES_PROBLEMATICOS
212 (
213     codigo INT PRIMARY KEY,
214     nombre nvarchar(25),
215     mediaimporte decimal(5,2),
216     tarifa decimal (5,2),
217     fecharegistro date
218 )
219
220 --STORED PROCEDURE(SP)
221 -- Create a new stored procedure called 'Mediaimporte'
222 -- Drop the stored procedure if it already exists
223 USE clinica;
224 GO
225 IF OBJECT_ID ('dbo.Mediaimporte') IS NOT NULL
226     DROP PROCEDURE dbo.Mediaimporte;
227 GO
228 CREATE PROCEDURE Mediaimporte
229     @ciudad nvarchar(25),
230     @mediavisitas decimal(8,2) OUTPUT
231 AS
232     -- body of the stored procedure
233     SELECT @mediavisitas = AVG(importe)
234     FROM visitas
235     INNER JOIN pacientes on codpaciente = codigo
236     WHERE ciudad = @ciudad
237 GO
238 -- example to execute the stored procedure we just created
239 --DECLARE @mediavisitas int
```

```
240 --EXECUTE dbo.Mediamporte 'ALICANTE', @mediavisitas OUTPUT
241 --PRINT 'MEDIA IMPORTE VISITAS:'+cast(@mediavisitas as nvarchar(20))
242
243
244 --CURSOR
245 DECLARE @nombre NVARCHAR(25),
246         @ciudad NVARCHAR(25),
247         @Nvisitas int,
248         @tarifa decimal(8,2),
249         @codigo int,
250         @tarifaup decimal(8,2)
251
252 DECLARE RentabilidadPacientes CURSOR
253 FOR
254     SELECT codigo,nombre, ciudad, COUNT(*),tarifa
255     FROM pacientes
256     INNER JOIN visitas ON codpaciente = codigo
257     group by codigo,nombre,ciudad,tarifa
258
259 OPEN RentabilidadPacientes
260 FETCH RentabilidadPacientes INTO @codigo, @nombre, @ciudad, @Nvisitas, @tarifa
261 WHILE(@@FETCH_STATUS = 0)
262     BEGIN
263         PRINT 'PACIENTE:' + @nombre
264         IF (@Nvisitas > 4)
265             BEGIN
266                 PRINT 'PACIENTE PROBLEMÁTICO'
267                 SET @tarifaup = @tarifa * 1.20
268                 PRINT 'TARIFA:' + CAST(@tarifa as nvarchar(10))
269                 --print space(1)
270             END
271         ELSE
272             BEGIN
273                 PRINT 'TARIFA:' + cast(@tarifa as nvarchar(25))
274             END
275         DECLARE @mediavisitas INT
276         EXECUTE dbo.Mediamporte @ciudad, @mediavisitas OUTPUT
277         IF (@mediavisitas > 250)
278             BEGIN
279                 INSERT INTO PACIENTES_PROBLEMATICOS (codigo, nombre,
280 mediaimpor, tarifa, fecharegistro)
281                 VALUES (@codigo, @nombre, @mediavisitas, @tarifa, GETDATE())
282             END
283         PRINT 'MEDIA IMPORTE VISITAS:' + CAST(@mediavisitas as NVARCHAR(20))
284         PRINT '*****'
285         print space(1)
286
287     FETCH RentabilidadPacientes INTO @codigo, @nombre, @ciudad, @Nvisitas, @tarifa
288     END
289 CLOSE RentabilidadPacientes
290 DEALLOCATE RentabilidadPacientes
```