

```
/*
EJERCICIO 1.*/

/*
LOGIN:
*/
/*1. Crea un Login inicial que se llame 'cfgsmz' a partir de autenticación de SQL
Server.
Investiga cómo crearlo para que al iniciar sesión en él te pida que cambies la
contraseña introducida inicialmente.
*/

CREATE LOGIN cfgsmz
WITH PASSWORD = 'abcd'
MUST_CHANGE, /*NOTA: La opción MUST_CHANGE no se puede utilizar cuando se
ha desactivado CHECK_EXPIRATION.*/
CHECK_EXPIRATION = ON;
GO

/*2. Renombra el Login creado anteriormente a 'gbdasix'*/

ALTER LOGIN cfgsmz
WITH NAME = gbdasix;

/*3. Deshabilita el Login creado.*/

ALTER LOGIN cfgsmz DISABLE;

/*4. Vuelve a habilitarlo.*/

ALTER LOGIN cfgsmz ENABLE;

/*5. Borra el Login e inmediatamente vuelve a crearlo con el nombre 'gbdasix'.*/

DROP LOGIN cfgsmz;

CREATE LOGIN cfgsmz
WITH PASSWORD = 'abcd'

/*USERS:*/
/*6. Eres el administrador de la Base de Datos, y te piden varios
departamentos usuarios para acceder a la BD.
Realiza la siguiente creación de usuarios:*/
/*7. Crea un usuario de tipo administrador, es decir, que tenga permisos para todo y
sea el que vayas a usar tú.*/

CREATE USER mpf_asix FOR LOGIN cfgsmz;

USE clinica;
GRANT ALL ON DATABASE::clinica TO mpf_asix;
USE videoclub;
GRANT ALL ON DATABASE::videoclub TO mpf_asix;
```

```
51 USE proyectos;
52 GRANT ALL ON DATABASE::proyectos TO mpf_asix;
53 USE Servicio_tecnico;
54 GRANT ALL ON DATABASE::Servicio_tecnico TO mpf_asix;
55
56
57 /*8. Crea un usuario genérico para el departamento de Marketing,
58 que sólo tenga privilegios para consultar datos en las BD ClinicaDB, Proyectos y
59 Concesionarios.*/
60 CREATE LOGIN marketing
61     WITH PASSWORD = 'Usermarketing1';
62 CREATE USER user_consulta FOR LOGIN marketing;
63
64 GRANT SELECT ON DATABASE::clinica TO user_consulta;
65 GRANT SELECT ON DATABASE::proyectos TO user_consulta;
66 GRANT SELECT ON DATABASE::concesionario TO user_consulta;
67 GO
68
69 /*9. Crea un usuario genérico para el departamento de Ventas, que tenga privilegios
70 para consultar, borrar e insertar datos para la BD Proyectos.*/
71
72 CREATE LOGIN ventas
73     WITH PASSWORD = 'Userventas1';
74 CREATE USER user_consulta FOR LOGIN ventas;
75
76 GRANT SELECT, DELETE, INSERT ON DATABASE::proyectos TO user_consulta;
77
78 GO
79
80 /*10. Crea tres usuarios para el departamento de TI que tengan todos los permisos para
81 cualquiera de las tablas de la BD.*/
82 CREATE LOGIN TI
83     WITH PASSWORD = 'UserTI1';
84
85 CREATE USER user_ti1 FOR LOGIN TI
86 GRANT ALL ON DATABASE::concesionario TO user_ti1;
87
88 CREATE USER user_ti2 FOR LOGIN TI
89 GRANT ALL ON DATABASE::concesionario TO user_ti2;
90
91 CREATE USER user_ti3 FOR LOGIN TI
92 GRANT ALL ON DATABASE::concesionario TO user_ti3;
93
94 /*11. A uno de estos tres últimos usuarios, que será para el equipo de desarrollo,
95 quítale los permisos de REFERENCES y ALTER.*/
96
97 USE concesionario;
98 REVOKE REFERENCES, ALTER ON DATABASE::concesionario FROM user_ti3;
99 GO
100
101
102 /*12. Crea otro usuario que permita solamente acceder a la información de los
pacientes
```

```
103 y ejecutar algún procedimiento de los creados en la Unidad 7 sobre la BD ClinicaBD.*/
104
105 CREATE LOGIN clinica
106     WITH PASSWORD = 'User1'
107 CREATE USER user_consulta_pacientes FOR LOGIN clinica
108
109 GRANT SELECT ON DATABASE:: clinicaBD TO user_consulta_pacientes;
110 USE clinicaBD;
111 GRANT EXECUTE ON OBJECT::dbo.visitaspacientes TO user_consulta_pacientes;
112 GO
113
114
115 /*ROLES:*/
116 /*13. Investiga para qué sirven los roles en SQL Server, y que roles por defecto
    existen ya.*/
117
118
119     /*Los roles son los conjuntos de permisos. Estos conjuntos existen a tres
    niveles distintos: servidor, base de datos y aplicación.
120     Los roles permiten agrupar los derechos y gestionar más fácilmente los
    diferentes usuarios y las conexiones.
121     Siempre es preferible asignar los derechos a los roles y posteriormente
    asignar los roles a los usuarios.
122     Con una estructura como esta, la adición y la modificación de permisos o de
    usuarios son más sencillas.
123
124     Es posible definir un rol como un conjunto identificado de permisos.
125     Para facilitar la gestión de los permisos, SQL Server ofrece los roles
    predefinidos, también llamados fijos,
126     ya que no es posible añadir o eliminar privilegios en estos roles.
127
128     Estos roles fijos se definen en dos niveles:
129
130         1-.Servidor.
131         2-.Base de datos.
132
133     Además de estos roles fijos, es posible gestionar otros roles.
134     Es conveniente establecer un nombre único para definir un rol y posteriormente
    asignar uno o varios permisos
135     respetando un procedimiento en todo punto similar al utilizado para asignar
    los permisos a los usuarios.
136     Estos roles se pueden definir en tres niveles:
137
138         Servidor.
139         Base de datos.
140         Aplicación.
141
142     */
143
144
145 /*14. ¿Qué es el rol 'public' y para qué sirve?*/
146
147     /*Es un rol fijo de la base de datos, y tiene una particularidad que es el unico
    el cual se pueden modificar los permisos a diferencia de los demas roles fijos*/
148
149
```

```
150 /*15. ¿Qué es la cuenta de usuario dbo y para qué sirve?*/
151 /*
152     dbo (database owner) es un usuario con permisos implícitos para realizar todas las
    actividades en la base de datos
153     Función que se asigna a un usuario que es propietario de una base de datos.
154 */
155
156 /*16. ¿Qué es la cuenta de usuario guest y para qué sirve?*/
157 /*
158     La cuenta Guest permite iniciar una sesión sin una cuenta de usuario que de
    acceso a la base de datos.
159     Se pueden añadir y eliminar cuentas de usuario invitado en cualquier base de
    datos,
160     excepto en las bases Master y Tempdb.
161 */
162
163
164 /*17. Crea un rol que se llame 'rolexecutegebdb' y asigne los privilegios de
    ejecutar cualquier objeto de la BD y consultar cualquier tabla de la BD.*/
165
166
167     CREATE ROLE rolexecutegebdb
168     GRANT SELECT ON DATABASE ::clinicaBD
169
170
171 /*18. Posteriormente crea dos nuevos usuarios y asígnales este rol.*/
172
173     CREATE LOGIN asix2
174         WITH PASSWORD = 'User1'
175
176     CREATE USER frank FOR LOGIN asix2
177     EXEC sp_addrolemember 'clinicaBD', 'frank'
178
179     CREATE USER antonio FOR LOGIN asix2
180     EXEC sp_addrolemember 'clinicaBD', 'frank'
181
182
183 /*
184 EJERCICIO 2. Crea vistas para las siguientes BDs
185 A) CLINICA BD
186 1.Realiza una vista que permita mostrar las visitas del día actual.
187 Además, también deberá mostrar la información del paciente de la visita.
188 Esta vista ayudará a que la gente de administración sepa las visitas del día en el que
    se encuentran.*/
189
190     CREATE VIEW consultasdiarias
191     AS
192         SELECT * FROM visitas v
193         INNER JOIN pacientes p
194         ON v.CODPACIENTE=p.codigo
195         WHERE v.FECHA = CONVERT(varchar(10), getdate(), 120)
196
197
198 /*2. Modifica esta vista para que, además, sólo muestre la de los pacientes que no
    estén allí por una gripe.*/
199
```

```
200 ALTER VIEW consultasdiarias
201 AS
202     SELECT * FROM visitas v
203     INNER JOIN pacientes p
204     ON v.codpaciente=p.codigo
205     WHERE v.fecha = CONVERT(varchar(10), getdate(), 120)
206     AND v.enfermedad <> 'Gripe';
207
208 /*3. Borra la vista creada anteriormente.*/
209
210 DROP VIEW consultasdiarias
211
212 /*B) VIDEOCLUB
213 4. Crea una vista que muestre todos los datos de los socios, exceptuando el DNI y la
    fecha de nacimiento.*/
214
215 CREATE VIEW datos_socios
216 AS
217     SELECT codigo, nombre, apellido, telefono, clientevip
218     FROM socios
219
220 /*5. Modifica los datos de la tabla para que el cliente con código 2 (Federico), sea
    CLIENTEVIP también (1).
221 Consulta ahora la vista. ¿Qué sucede? ¿Ha cambiado también el valor en la vista?*/
222
223 USE clinica
224 UPDATE socios
225 SET CLIENTEVIP = 1
226 WHERE codigo = 2
227
228 /*
229 Ha cambiado el valor del registro CLIENTEVIP a 1, y por tanto en la vista veremos
    el
230 valor actualizado a CLIENTEVIP = 1
231 */
232
233 /*6. Devuelve a su estado anterior (0) el valor modificado.*/
234
235 USE clinica
236 UPDATE socios
237 SET CLIENTEVIP = 0
238 WHERE codigo = 2
239
240
241 /*7. Modifica ahora el mismo campo para Federico, pero esta vez en la vista. ¿Qué
    sucede?
242 ¿Permite realizar la modificación de los datos reales a partir de la vista?*/
243
244 UPDATE datos_socios
245 SET CLIENTEVIP = 1
246 WHERE codigo = 2
247
248 /*
249 Se podra modificar los registros si el usuario posee permisos UPDATE, INSERT o
    DELETE
```

```
250     en la tabla de destino, en función de la acción que realizará
251 */
252
253 /*8. Bien, como ya hemos visto, a partir de esta vista hemos podido modificar los
    valores reales,
254 ya que la vista no es nada físico, sino simplemente una representación de los valores.
255 Por tanto, tendréis ahora que crear un usuario, que SÓLO tendrá privilegios para
    consultar
256 la información de esta vista,y de esta manera evitamos que pudiera modificar.*/
257
258 CREATE LOGIN asix2
259     WITH PASSWORD = 'User1'
260     CREATE USER frank FOR LOGIN asix2
261     GRANT SELECT ON OBJECT:: datos_socios TO frank;
262
263
264 /*9. Modifica ahora la vista para que permita ver los datos de socios (Exceptuando su
    DNI y fecha de nacimiento)
265 que hayan alquilado alguna vez alguna película que dure más de 180 minutos).*/
266
267 ALTER VIEW datos_socios
268 AS
269     SELECT s.codigo, s.nombre, s.apellido, s.telefono, s.clientevip
270     FROM socios s
271     INNER JOIN alquiler a ON s.codigo = a.CODIGO_SOCIO
272     INNER JOIN stock stk ON a.CODIGO_STOCK = stk.CODIGO
273     INNER JOIN peliculas p ON stk.CODIGO_PELICULA = p.CODIGO
274     WHERE p.duracion >= 180
275
276
277 /*10. Trata ahora de realizar una modificación de la misma forma sobre CLIENTEVIP. ¿Te
    ha permitido?*/
278
279 /*
280     No se puede efectuar la modificación ya que el usuario solo tiene privilegios
    de consulta
281     sobre la vista
282 */
283
284 /*C) PROYECTOS
285 11. Crea una vista que permita visualizar la información de solamente los empleados
286 que trabajan en el departamento de DIR.*/
287
288 CREATE VIEW empleados_deptDIR
289 AS
290     SELECT e.intCodempleado, e.strNombreem, e.strApellidoem, e.mnySueldo,
291     e.strNombredp, e.strCiudadp, e.strNumpro
292     FROM tblempleados e
293     INNER JOIN tbldepartamentos d ON e.strNombredp = d.strNombredp
294     WHERE e.strNombredp LIKE 'DIR'
295
296 GO
297
298 /*12. Tendrás que crear un usuario genérico para que este departamento consulte la
    información en esta vista.
```

```
299 Además, tendrán permiso también para insertar y modificar datos sobre la propia
    vista.*/
300
301     CREATE LOGIN proyectos
302         WITH PASSWORD = 'User1proyectos'
303     CREATE USER us01proyectos FOR LOGIN proyectos
304     GRANT SELECT, INSERT, UPDATE ON empleados_deptDIR TO us01proyectos;
305
306 /*13. Borra la vista creada anteriormente.*/
307
308     DROP VIEW empleados_deptDIR
309
310
311
```

