

Ejercicio de evaluación P101

Población aleatoria de una base de datos

Ejercicio de evaluación P101	1
Población aleatoria de una base de datos	1
La base de datos	2
Procedimientos almacenados	3
Los requisitos	3
Generar usuarios aleatorios	3
Nombre y apellidos	4
Cómo seleccionar un nombre y unos apellidos	4
Obtener una fecha aleatoria en un periodo de tiempo	5
Construir el usuario de correo electrónico	5
Esqueleto de la creación de usuarios	6

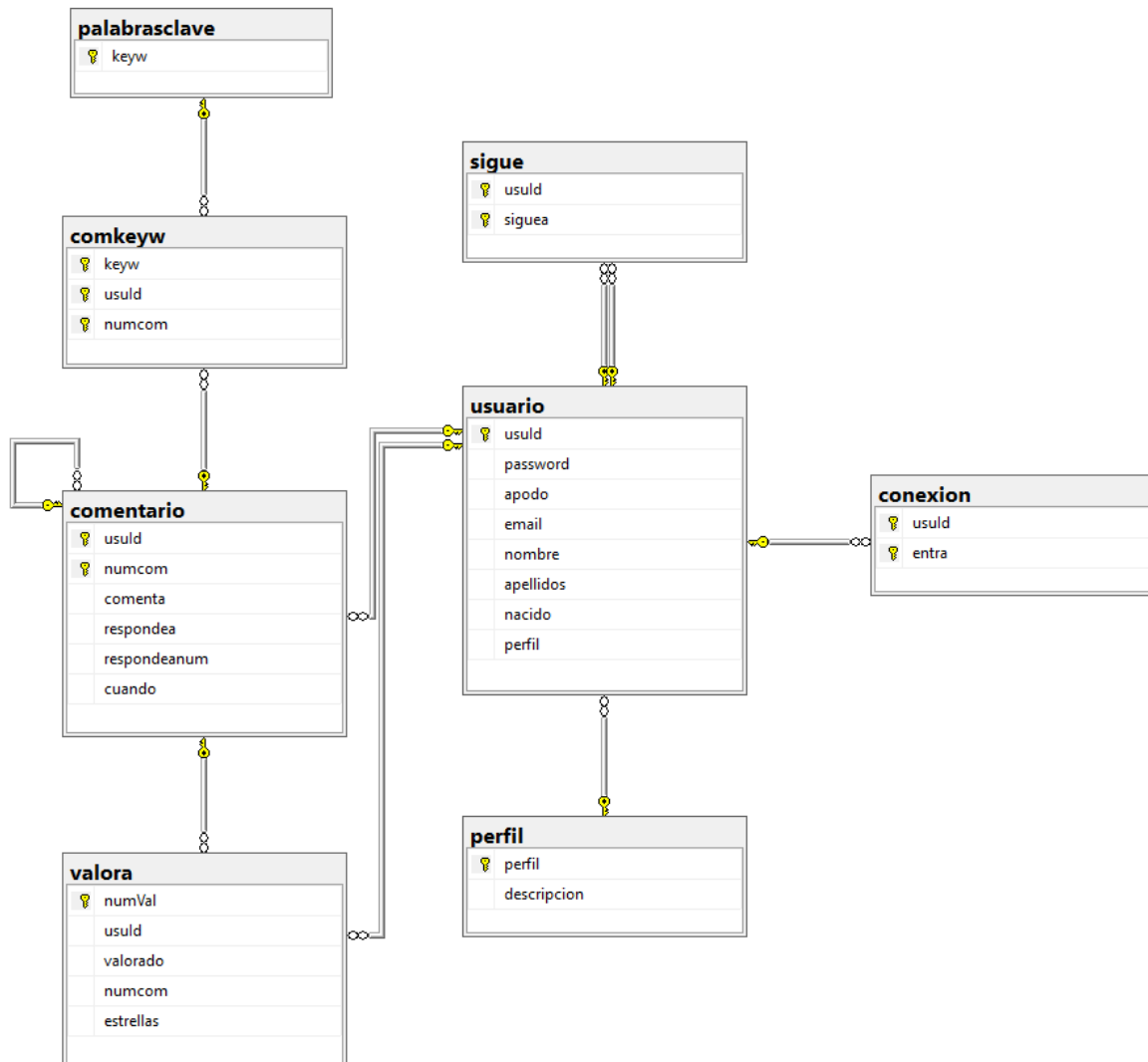
Se trata de rellenar unas tablas con datos aleatorio, algunos extraídos de otras tablas. Debes encapsularlo en procedimientos almacenados, con uno de ellos como el principal que llama a todos los demás. Aquí se describe cómo ha de ser la base de datos. Para los datos aleatorios puedes utilizar SQL dinámico.



Puedes necesitar conocer:

1. [TOP \(select\)](#)
2. [Función random](#)
3. [Función replicate](#)
4. [Funciones de cadena](#)
5. [Función newid](#)
6. [Ordenación aleatoria de una consulta](#); [también aquí](#); [en tablas grandes aquí](#)
7. [SQL dinámico: orden EXECUTE](#); [también aquí](#)

La base de datos



La base de datos representa, más o menos, una red social, sin necesitar ser fiel a la realidad. Salvo las tablas PERFIL y PALABRASCLAVE, las demás estarán vacías y tendréis que rellenarlas de datos.

La creación de las tablas y los datos iniciales los encontraréis en este [fichero de creación de la base de datos](#).

Procedimientos almacenados

Los requisitos

- Lo que codifiques irá a continuación de lo ya codificado en el script anterior.
- El procedimiento principal se llamará **generar_usuasios**
- Admitirá un parámetro que será la cantidad de usuarios a insertar.
 - Por ejemplo: `exec generar_usuarios 10;`
- Los datos de nombre, apellidos (2 apellidos separados por espacio) se obtendrán aleatoriamente de las tablas temporales creadas en el script anterior.
- El dato de correo electrónico se compondrá de `nombre.primerapellido@servidor`, siendo `servidor` un valor extraído aleatoriamente de la tabla temporal creada en el script anterior.

Desarrollarás el código a continuación de la creación de la base de datos, en un único fichero. Eso permitirá reejecutar como si fuera la primera vez cuando se necesite hacer un cambio y comprobar su éxito. Más tarde se puede encapsular ese lote en el procedimiento almacenado.

Generar usuarios aleatorios¹

La tabla USUARIOS es central para la generación de los demás datos. La definición de esta es:

```
create table usuario (  
    usuId int IDENTITY(1,1) constraint PK_usuario primary key,  
    password varbinary(64),  
    apodo varchar(15),  
    email varchar(254) not null,  
    nombre varchar(50) not null,  
    apellidos varchar(75) not null,  
    nacido date not null,  
    perfil char(3) default '0',  
    constraint FK_usuario_perfil  
        foreign key (perfil) references perfil(perfil)  
);
```

El PA necesita un parámetro, la cantidad de usuarios a generar —que llamaremos *cantusu*, pero que tú puedes llamar como quieras—. Ahora vamos a pensar cómo generar un único usuario, después se montará un bucle para generarlos todos.

¹ Algunas de las tablas de trabajo que mencionaremos son candidatas a trabajarlas directamente en memoria RAM. En SQL Server se pueden utilizar estas tablas. Por ejemplo:

```
create table nomtabla (restricciones...)  
WITH (MEMORY_OPTIMIZED=ON, DURABILITY = SCHEMA_ONLY);
```

Esto [crea una tabla en memoria](#) cuyo contenido, las filas, se destruye al finalizar sesión. Hay otras opciones de persistencia, cada una con sus requisitos. Puede necesitarse una modificación de la configuración del servidor.

Nombre y apellidos

Necesitamos hacer unas operaciones previas. Dispones de dos tablas temporales para generar los nombres de los usuarios: #apellido y #nombre. Para el nombre, solo hay que hacer una selección aleatorio. Para los apellidos utilizaremos el producto cartesiano.

El primer paso es generar otra tabla temporal con todas las posibles combinaciones de estos. Esa tabla tendrá 3 columnas: orden (entero), nombre (varchar de 25) y apellidos (varchar 75). Por ejemplo:

Results		Messages	
	orden	nombre	apellidos
1	1	Adrian	Andreo Andreo
2	2	Adrian	Adáñez Andreo
3	3	Adrian	Alejándrez Andreo

La columna *orden* la utilizaremos para seleccionar *cantusu* filas aleatoriamente que insertaremos, finalmente, en la tabla USUARIO —junto con los otros datos requeridos—. La generación de estos valores se consigue con la función *row_number()*.

Prueba esto:

```
select row_number() over (order by nombre) orden, nombre from #nombre;
```

La combinación de nombre y apellidos se consigue con el producto cartesiano:

```
select * from #nombre n, #apellido a1, #apellido a2
```

La concatenación de texto se consigue con el operador +:

```
select a1.apellido + ' ' + a2.apellido from #apellido a1, #apellido a2
```

Ya tienes todo lo que necesitas para crear esa tabla.

Cómo seleccionar un nombre y unos apellidos

La tabla temporal que estamos manejando tiene una columna numérica *orden*. Nos basaremos en ella para, aleatoriamente, seleccionar algunas de sus filas.

Prueba esto varias veces:

```
declare @n float;
declare @cuantos integer = 10;

declare @contador integer = 1;
while @cuantos >= @contador
begin
```

```
set @n=rand();  
select @n, @n*3, floor(@n*3)+1;  
set @contador=@contador+1;  
end;
```

Ya tienes un método para conseguir un número aleatorio entre 1 y 3.

Obtener una fecha aleatoria en un periodo de tiempo

Aquí vamos a utilizar una función desarrollada por terceros que está declarada en el [fichero de creación de la base de datos](#). Solo tienes que usarla:

```
select dbo.getRandomDate('1960-01-01', '2019-10-11')
```

Con ella conseguimos una fecha aleatoria entre el 1/1/1960 y el 11/10/2019.

Construir el usuario de correo electrónico

Recuerda que la tabla temporal #correo contiene varios servidores. Se trata de elegir uno para cada usuario.

```
declare @serv varchar(50);  
select top 1 @serv='@'+correo from #correo order by newid();  
select @serv;
```

Si lo ejecutas varias veces verás que nos construye un servidor de correo aleatorio basándose en los datos almacenados en #correo.

Para construir el nombre de usuario, suponiendo que nos los hemos guardado en alguna variable, solo tenemos que extraer el primero de ellos.

```
declare @losapellidos varchar(75) = 'Pérez Gómez'  
declare @elnombre varchar(25) = 'María';  
declare @usuario varchar (75);  
  
select charindex(' ',@losapellidos) posición;  
  
select substring(@losapellidos,1,charindex(' ',@losapellidos)-1)  
primer_apellido;  
  
select @elnombre+'.'+substring(@losapellidos,1,charindex(' ',@losapellidos)-1) usuario;  
  
select @elnombre+'.'+substring(@losapellidos,1,charindex(' ',@losapellidos)-1) + cast(floor(rand()*999) as varchar(3))  
usuario_rand;
```

La última parte añade un número aleatorio entre 0 y 999 para evitar —pobrememente, no es el mejor método— que se produzcan duplicados de correo.

Esqueleto de la creación de usuarios

Más o menos, ya tenemos solucionados todos los problemas que nos vamos a encontrar. El procedimiento a definir tendrá esta estructura —los nombres de variable y tabla, los que quieras—:

```
crear PA (parámetro @cantusu)
declaración de variables
generar tabla de nombres y apellidos (TNA)
mientras @cuenta <= @cuantos {
  seleccionar 1 (nombre,apellidos) aleatoriamente desde TNA: (@nom y @apes)
  seleccionar 1 servidor de correo aleatorio: (@serv)
  construir el usuario de correo electrónico: (@email)
  insertar en USUARIO (email,nombre,apellidos,nacido) =
    (@email,@nom,@apes,fecha_aleatoria)
```

Recuerda que la tabla usuario tiene definida una clave primaria autoincremental (IDENTITY). Esto hace que no sea necesario especificarla en la inserción que crearéis en el PA, se generará automáticamente.